

PENGOPTIMUMAN IDENTIFIKASI SAMPAH DAN
NAVIGASI ROBOT BERAUTONOMI DI
PERKARANGAN RUMAH

EMIL NAF'AN

UNIVERSITI KEBANGSAAN MALAYSIA

PENGOPTIMUMAN IDENTIFIKASI SAMPAH DAN NAVIGASI ROBOT
BERAUTONOMI DI PERKARANGAN RUMAH

EMIL NAF'AN

TESIS YANG DIKEMUKAKAN UNTUK MEMPEROLEH
IJAZAH DOKTOR FALSAFAH

INSTITUT INFORMATIK VISUAL (IVI)
UNIVERSITI KEBANGSAAN MALAYSIA
BANGI

2024



UNIVERSITI KEBANGSAAN MALAYSIA
The National University of Malaysia

**PERAKUAN TESIS SARJANA / DOKTOR FALSAFAH
(DECLARATION OF MASTER / DOCTOR OF PHILOSOPHY THESIS)**

A. MAKLUMAT PELAJAR/STUDENT DETAILS		
Nama <i>Name</i>	Emil Nafan	
No. Pendaftaran <i>Registration No.</i>	P88912	
Fakulti/Institut <i>Faculty/Institute</i>	Institut Informatik Visual (IVI)	
Program Pengajian <i>Program of Study</i>	Sarjana / Doktor Falsafah <i>Masters / Doctor of Philosophy</i>	
Tajuk Tesis <i>Thesis Title</i>	Pengoptimuman Identifikasi Sampah Dan Navigasi Robot Berautonomi Di Perkarangan Rumah	
Mel-e/Email: p88912@siswa.ukm.edu.my		No. Telefon/Tel.No: +60137988017
B. PERAKUAN / DECLARATION		
Merujuk kepada Dasar Harta Intelek UKM 2018, tesis adalah hak milik UKM seperti keterangan dibawah: 4.2.1 Harta Intelek yang direka cipta oleh pelajar termasuk tesis dan semua hasil penyelidikan ditulis oleh Pelajar UKM dalam tempoh pengajiannya di UKM direka cipta, dibangunkan atau dihasilkan menggunakan kemudahan, bahan, dana, atau sumber lain adalah hak milik UKM melainkan dinyatakan sebaliknya dalam Dasar ini atau dipersetujui sebaliknya oleh UKM dan Pelajar/Penaja. <i>Referring to the UKM Intellectual Property Policy 2018, the thesis is the property of UKM as described below:</i> <i>4.2.1 Intellectual property created by the student, including the thesis and all research output produced by the UKM Student during their studies at UKM, including designed, developed or produced output using facilities, materials, funds, or other resources are the property of UKM unless otherwise stated in this Policy or otherwise agreed by UKM and the Student/Sponsor.</i>		
	RAHSIA CONFIDENTIAL	Mengandungi maklumat rahsia di bawah AKTA RAHSIA RASMI 1972 <i>Consisting of classified information under the OFFICIAL SECRETS ACT 1972</i>
	TERHAD RESTRICTED	Mengandungi maklumat TERHAD yang telah ditentukan oleh organisasi / badan di mana penyelidikan dijalankan <i>Consisting of RESTRICTED information which has been determined by the organisation/body where the research was conducted</i>
✓	AKSES TERBUKA/ TIDAK TERHAD OPEN ACCESS/ NON-RESTRICTED	Saya membenarkan tesis ini diterbitkan secara akses terbuka, teks penuh atau dibuat salinan untuk tujuan pengajian, pembelajaran & penyelidikan sahaja <i>I give permission for this thesis to be published through open access, full text or copied only for study, learning, and research purposes.</i>

Bagi kategori Akses Terbuka/Tidak Terhad, saya membenarkan tesis (Sarjana/Doktor Falsafah) ini di simpan di Perpustakaan Universiti Kebangsaan Malaysia (UKM)* dengan syarat-syarat kegunaan seperti berikut:

For the Open Access/Non-Restricted category, I permit this (Master's/Doctoral) Thesis to be kept in the Universiti Kebangsaan Malaysia (UKM) Library with the following conditions:

1. Perpustakaan UKM mempunyai hak untuk membuat salinan untuk tujuan pengajian, pembelajaran, penyelidikan sahaja.

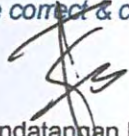
UKM Library has the right to reproduce the thesis only for study, learning and research purposes.

2. Perpustakaan Universiti Kebangsaan Malaysia dibenarkan membuat satu (1) salinan tesis ini untuk tujuan pertukaran antara institusi pengajian tinggi dan mana-mana badan/ agensi kerajaan, tertakluk kepada terma dan syarat.

UKM Library is allowed to make one (1) copy of this thesis for exchange purposes among higher education institutions and any government body/agency, subject to the terms and conditions.

Saya mengakui maklumat & tesis yang diberikan adalah benar & terkini.

I acknowledge the information & the thesis provided are correct & current.



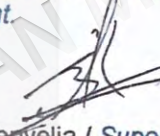
Tandatangan Pelajar:

Student's Signature

Tarikh / Date: 29 April 2024

Saya memperakukan maklumat & tesis yang diberikan adalah benar & terkini.

I verify that the information & the thesis provided are correct & current.



Tandatangan Penyelia / Supervisor's Signature:

Nama / Name:

PROF. IR. DR. RIZA SULAIMAN
Felo Kanan

Cop Rasmi / Official Stamp:

Institut Informatik Visual (IVI)
Universiti Kebangsaan Malaysia

Tarikh / Date: 29 April 2024

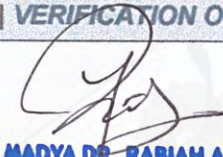
C. PENGESAHAN FAKULTI/INSTITUT | VERIFICATION OF FACULTY/INSTITUTE

Tandatangan/Signature:

Nama/Name:

Tarikh/Date: 23/5/24

Cop Rasmi/Official Stamp:


PROF. MADYA DR. RABIAH ABDUL KADIR
PENGARAH
Institut Informatik Visual (IVI)
Universiti Kebangsaan Malaysia

**Kuatkuasa: 15 Julai 2021

PENGAKUAN

Saya akui karya ini adalah hasil kerja saya sendiri kecuali nukilan dan ringkasan yang tiap-tiap satunya telah saya jelaskan sumbernya.

29 April 2024

EMIL NAF'AN
P88912

PENGHARGAAN

Alhamdulillah segala puji dan syukur dipanjatkan kehadrat Allah Subhanahu wata'ala atas limpahan rahmat dan kurnia serta hidayahNya, maka tesis ini telah dapat diselesaikan. Semoga hasil penyelidikan ini dapat memberikan manfaat terutamanya dalam bidang penglihatan dan navigasi robot.

Pertama sekali saya ingin mengucapkan terima kasih kepada Prof Dr Riza Sulaiman dan Prof Madya Nazlena Mohamad Ali yang telah membimbing saya dalam menyempurnakan tesis ini.

Terima kasih khas ditujukan kepada bapak H.Herman Nawas dan Ibu Dr.Hj.Zerni Melmusi SE,Ak.CA serta sivitas akademika UPI “YPTK” Padang, atas semua sokongan kewangan dan do'a hingga memudahkan saya menyempurnakan tesis ini.

Akhir sekali, ucapan terima kasih tidak terhingga kepada ibunda Siti Aisyah, isteri saya Rini Nurman, anak saya Abdel Haqqi dan Abdel Warits Rabbani serta keluarga besar yang tercinta dan rakan seperjuangan IVI dan B309 atas semua sokongan dan doa yang tiada putusnya di mana akhirnya memudahkan saya menyempurnakan tesis ini.

ABSTRAK

Perkarangan rumah yang bersih adalah impian semua orang. Sampah di perkarangan rumah mungkin terdiri daripada daun pokok, botol plastik, bungkusan makanan ringan dan objek lain. Pada masa ini terdapat banyak robot yang digunakan untuk membantu manusia. Selaras dengan perkembangan visi komputer dan teknologi robotik, timbul idea untuk menggunakan robot berautonomi yang mampu mencari sampah di perkarangan rumah. Atas sebab ini, robot mesti dapat mengidentifikasi sampah dan menavigasi dengan betul. Terdapat beberapa kajian mengenai proses identifikasi sampah menggunakan robot, namun belum optimum. Kebanyakan kaedah identifikasi objek tidak dapat membezakan sama ada objek itu adalah imej sebenar (3D) atau imej fotografi di atas kertas (2D). Ia akan menjadi masalah jika objek yang dikesan dialihkan dari satu tempat ke tempat lain. Jika objek adalah 2D, pencengkam robot akan cuba mencengkam, tetapi tidak mencengkam sebarang objek, disebabkan oleh geometri pencengkam robotik. Dalam tesis ini, untuk mengoptimumkan identifikasi sampah, kaedah *Sequential Camera LiDAR* (SCL) dicadangkan. Kaedah ini adalah gabungan Rangkaian Saraf Konvolusi (CNN) dan penderia *Light Detection and Ranging* (LiDAR). CNN memproses output kamera, diikuti dengan proses pengimbasan pantas oleh penderia LiDAR. Empat CNN pra-latihan digunakan, iaitu AlexNet, VGG16, GoogleNet dan ResNet18. Penderia LiDAR yang digunakan ialah jenis TF40 yang mempunyai ketepatan $\pm 2\text{mm}$. Untuk mengoptimumkan proses navigasi robot berautonomi, kaedah *Modified Freeman Chain Code Path Planning* (MFCC-PP) dicadangkan. Kaedah ini mengubah suai bilangan kod rantai Freeman yang pada asalnya hanya maksimum 8 kod, diubah suai kepada 16 dan 32 kod. Selepas kaedah SCL dilaksanakan terhadap 11 jenis sampah. Empat CNN pra-latihan digunakan, iaitu AlexNet, VGG16, GoogleNet dan ResNet18 mampu mengidentifikasi sampah dengan betul, dengan keputusan ketepatan ialah AlexNet (80.5%), VGG16 (95.6%), GoogleNet (98.3%), dan ResNet18 (97.5%). Keputusan ini sangat sesuai untuk identifikasi sampah. Selepas sampah diidentifikasi dengan betul, urutan kaedah SCL seterusnya adalah untuk menentukan sama ada sampah dalam bentuk 2D atau 3D menggunakan penderia LiDAR. Objek yang diidentifikasi sebagai sampah akan diabaikan jika proses pengimbasan pantas dengan penderia LiDAR mengesan sampah bentuk 2D. Jika bentuk 3D, sampah akan diimbas terperinci untuk menentukan kedudukan pencengkam dalam mengutip sampah. Kecekapan masa yang dijana oleh pengimbasan pantas adalah antara 13.33% dan 59.26% bergantung pada saiz objek. Lebih besar objek, lebih besar kecekapan masa yang diperolehi. Selepas kaedah MFCC-PP dilaksanakan dalam mencari sampah di perkarangan rumah, kaedah ini kemudiannya dibandingkan dengan kaedah navigasi algoritma A*, Dijkstra, dan *Freeman Direction Chain Code Path Planning* (FDCC-PP). Pengujian dilakukan dengan menyediakan 2 jenis peta yang berbeza. Berdasarkan dua jenis peta yang berbeza, panjang laluan yang dirancang oleh algoritma MFCC-PP ialah 5.69% dan 18.32% lebih panjang daripada laluan terpendek yang dirancang oleh algoritma Dijkstra. Pengubahsuaian kod rantai menghasilkan pusingan yang lebih lancar berbanding dengan algoritma lain. Secara keseluruhannya, kaedah SCL dan algoritma MFCC-PP yang dicadangkan dalam identifikasi sampah dan navigasi robot di perkarangan rumah mempunyai peningkatan yang ketara berbanding algoritma A*, Dijkstra, dan FDCC-PP.

OPTIMIZATION OF TRASH IDENTIFICATION AND AUTONOMOUS ROBOT NAVIGATION ON THE HOUSE COMPOUND

ABSTRACT

A clean house compound is a dream for everyone. The trash in the house compound may consist of tree leaves, plastic bottles, snack packages, and other objects. Currently, many robots are used to help humans. In line with the development of computer vision and robotics technology, the idea arose to use an autonomous robot to find trash in the house compound. For this reason, the robot must be able to identify trash and navigate correctly. There are several studies on the trash identification process using robots, but it is not yet optimal. Because most object identification methods cannot distinguish whether the object is a real image (3D) or a photographic image on paper (2D). It will be a problem if the detected object is moved from one place to another. If the object is 2D, the robot gripper will try to grip but will not grip any object due to the geometry of the robotic gripper. In this thesis, the Sequential Camera_LiDAR (SCL) method is proposed to optimize trash identification. This method combines Convolutional Neural Networks (CNN) and Light Detection and Ranging (LiDAR) sensors. The CNN processes the camera output, followed by a fast scanning process by the LiDAR sensor. Here, four pre-trained CNNs are used: AlexNet, VGG16, GoogleNet, and ResNet18. The LiDAR sensor is a TF40 type with an accuracy of $\pm 2\text{mm}$. The Modified Freeman Chain Code Path Planning (MFCC-PP) method is proposed to optimize the autonomous robot navigation process. This method modifies the number of Freeman chain codes, initially only a maximum of 8, modified to 16 and 32. After the SCL method was implemented against 11 types of trash, four pre-trained CNNs were used, namely AlexNet, VGG16, GoogleNet, and ResNet18 were able to correctly identify trash, with the accuracy results being AlexNet (80.5%), VGG16 (95.6%), GoogleNet (98.3%), and ResNet18 (97.5%). This result is very suitable for trash identification. After the trash is correctly identified, the following sequence of the SCL method is to determine whether the trash is in 2D or 3D form using the LiDAR sensor. Objects identified as the trash will be ignored if the fast scanning process with the LiDAR sensor detects 2D trash. If it is 3D, the trash will be scanned in detail to determine the position of the robot gripper in collecting the trash. The time efficiency generated by fast scanning is between 13.33% and 59.26%, depending on the object's size. After the MFCC-PP method is implemented in finding trash in the house compound, this method is compared with the navigation method of the A*, Dijkstra's, and the Freeman Direction Chain Code Path Planning (FDCC-PP) algorithm. Testing is done by preparing two different types of maps. Based on two different types of maps, the length of the path planned by the MFCC-PP algorithm is 5.69% and 18.32% longer than the shortest path planned by the Dijkstra algorithm. However, modifying the chain code results in smoother rounds than other algorithms. Overall, the SCL method and MFCC-PP algorithm proposed in trash identification and robot navigation in house compound significantly improve than A*, Dijkstra, and FDCC-PP algorithms.

KANDUNGAN

	Halaman
PENGAKUAN	ii
PENGHARGAAN	iii
ABSTRAK	iv
ABSTRACT	v
KANDUNGAN	vi
SENARAI JADUAL	x
SENARAI ILUSTRASI	xii
SENARAI SINGKATAN	xviii
BAB I	PENDAHULUAN
1.1	Pengenalan 1
1.2	Permasalahan Kajian 3
1.3	Persoalan Kajian 7
1.4	Objektif Kajian 7
1.5	Skop Kajian 8
1.6	Sumbangan Kajian 8
1.7	Organisasi Bab 8
BAB II	KAJIAN LITERATUR
2.1	Pengenalan 10
	2.1.1 Definisi Sampah 10
	2.1.2 Jenis-jenis Sampah 11
	2.1.3 Pengurusan Sampah 12
2.2	Pembelajaran Dalam (<i>Deep Learning</i>) 12
	2.2.1 Pemprosesan Imej 15
	2.2.2 Pengimejan Perubatan 17
	2.2.3 Biometrik 18
	2.2.4 Menyelesaikan Masalah Graf 19
	2.2.5 Pemprosesan Ucapan 19
2.3	Rangkaian Saraf Konvolusi (CNN) 19
	2.3.1 Reka Bentuk Rangkaian Saraf Konvolusi 21
	2.3.2 Lapisan Rangkaian Saraf Konvolusi 23
	2.3.3 Lapisan Konvolusi 24

2.3.4	Fungsi Pengaktifan (ReLU)	25
2.3.5	Lapisan Pengumpulan (<i>Pooling Layer</i>)	25
2.3.6	Lapisan yang disambungkan sepenuhnya (<i>Fully Connected Layer</i> / MLP)	27
2.4	Beberapa Seni Bina Rangkaian Saraf Konvolusi (CNN)	27
2.4.1	Seni Bina LeNet	27
2.4.2	Seni Bina AlexNet	28
2.4.3	Seni Bina VGG	30
2.4.4	Seni Bina GoogleNet	32
2.4.5	Seni Bina ResNet	33
2.4.6	Perbandingan pada model yang berbeza	35
2.5	Penyelidikan Rangkaian Saraf Konvolusi (CNN) Terdahulu Dalam Pengurusan Sampah	36
2.6	Ketepatan Mengukur Model	38
2.6.1	Persimpangan di atas Kesatuan (IoU)	38
2.6.2	Matriks Kekeliruan (<i>Confusion Matrix</i>)	39
2.6.3	Ketepatan	40
2.6.4	Kepersisan	41
2.6.5	Ingat Kembali (<i>Recall</i>)	42
2.6.6	Ukuran F / Skor F1	43
2.7	Penderia LiDAR (<i>Light Detection and Ranging</i>)	44
2.8	Kerja Sebelumnya Menggunakan LiDAR Dalam Pengesanan Objek	45
2.9	Kod Rantai Freeman (<i>Freeman Chain Code</i>)	47
2.10	Robot Berautonomi	48
2.11	Navigasi Robot	51
2.11.1	Algoritma A*	54
2.11.2	Algoritma Dijkstra	58
2.11.3	Algoritma <i>Freeman Direction Chain Code-Path Planning</i> (FDCC-PP)	62
2.12	Kerja Sebelumnya Menggunakan Penderia Ultrasonik dan Pengekod Dalam Navigasi Robot	64
2.13	Kesimpulan	67
BAB III	METODOLOGI PENYELIDIKAN	
3.1	Pengenalan	69
3.2	Reka Bentuk Robot Autonomi	71
3.2.1	Gambar rajah blok identifikasi sampah	74
3.2.2	Gambar rajah navigasi robot	75
3.2.3	Gambar rajah blok lengan robot	77
3.2.4	Pembinaan modul program	79

3.3	Identifikasi Sampah di Perkarangan Rumah	90
3.3.1	Penentuan permasalahan	90
3.3.2	Pengumpulan data dan pra-pemrosesan identifikasi sampah	90
3.3.3	Pembangunan algoritma identifikasi sampah	95
3.3.4	Imbas imej pantas menggunakan LiDAR	100
3.3.5	Imbas imej terperinci menggunakan LiDAR	105
3.4	Navigasi Robot Autonomi di Perkarangan Rumah	108
3.5	Kesimpulan	114

BAB IV HASIL DAN PERBINCANGAN

4.1	Pengenalan	116
4.2	Hasil Reka Bentuk Robot Autonomi	117
4.2.1	Hasil reka bentuk bahagian identifikasi sampah	117
4.2.2	Hasil reka bentuk bahagian navigasi robot	119
4.2.3	Hasil reka bentuk bahagian lengan robot	123
4.2.4	Hasil reka bentuk keseluruhan robot berautonomi	124
4.3	Hasil Identifikasi Sampah Menggunakan Rangkaian Saraf Konvolusi (CNN)	125
4.3.1	Identifikasi sampah menggunakan AlexNet	126
4.3.2	Identifikasi sampah menggunakan VGG16	129
4.3.3	Identifikasi sampah menggunakan GoogleNet	133
4.3.4	Identifikasi sampah menggunakan ResNet18	137
4.4	Hasil Imbasan Penderia LiDAR	141
4.4.1	Hasil imbasan imej pantas	142
4.4.2	Hasil imbasan imej terperinci	142
4.5	Hasil NAVIGASI ROBOT BERAUTONOMI	144
4.5.1	Hasil Pengujian Menggunakan Peta Pertama	144
4.5.2	Hasil Pengujian Menggunakan Peta Kedua	157
4.6	Kesimpulan	169

BAB V KESIMPULAN DAN KAJIAN MASA DEPAN

5.1	Pengenalan	171
5.2	Kesimpulan	172
5.3	Sumbangan Kajian	173
5.4	Cadangan Kajian Lanjutan	173

RUJUKAN	175
----------------	------------

LAMPIRAN

Lampiran A	Senarai Penerbitan	188
Lampiran B	Keputusan Ujian Seni Bina AlexNet	189
Lampiran C	Keputusan Ujian Seni Bina VGG16	207
Lampiran D	Keputusan Ujian Seni Bina GoogleNet	224
Lampiran E	Keputusan Ujian Seni Bina ResNet18	241



SENARAI JADUAL

No. Jadual		Halaman
Jadual 2.1	Ralat-5% teratas dengan parameter pengiraan dan Mac untuk model CNN yang berbeza	36
Jadual 2.2	Aplikasi Rangkaian Saraf Konvolusi (CNN), 2005–2022	36
Jadual 2.3	Parameter utama TF40	44
Jadual 2.4	Kerja Sebelumnya Menggunakan LiDAR Dalam Pengesanan Objek	45
Jadual 2.5	Kerja sebelumnya menggunakan penderia ultrasonik dan pengekod	65
Jadual 3.1	Objektif penyelidikan dan tahapan penyelidikan	70
Jadual 3.2	Fungsi dan Spesifikasi Motor Servo Pada Lengan Robot	78
Jadual 3.3	Parameter Latihan Digunakan untuk Identifikasi Sampah	97
Jadual 4.1	Selisih nilai kitaran tugas (<i>duty cycle</i>) antara teori dan praktikal	118
Jadual 4.2	Hasil Uji Kinerja Motor Servo	119
Jadual 4.3	Hasil Uji Kinerja Motor DC	120
Jadual 4.4	Hasil Uji Kinerja Penderia Pengekod	121
Jadual 4.5	Hasil Uji Kinerja Penderia Kompas	122
Jadual 4.6	Hasil Uji Kinerja Penderia Ultrasonik	123
Jadual 4.7	Hasil ketepatan yang dicapai oleh seni bina CNN pra-latihan AlexNet	129
Jadual 4.8	Hasil ketepatan yang dicapai oleh seni bina CNN pra-latihan VGG16	133
Jadual 4.9	Hasil ketepatan yang dicapai oleh seni bina CNN pra-latihan GoogleNet	136
Jadual 4.10	Hasil ketepatan yang dicapai oleh seni bina CNN pra-latihan ResNet18	140
Jadual 4.11	Keputusan kemajuan latihan	140

Jadual 4.12	Purata Ketepatan CNN Pra-Latihan dalam identifikasi Sampah Masa Nyata	140
Jadual 4.13	Perbandingan antara algoritma MFCC-PP dan tiga algoritma navigasi lain berdasarkan peta pertama	156
Jadual 4.14	Perbandingan antara algoritma MFCC-PP dan tiga algoritma navigasi lain berdasarkan peta kedua	167
Jadual 4.15	Pencapaian Matlamat Navigasi untuk Setiap Kaedah	168



SENARAI ILUSTRASI

No. Rajah		Halaman
Rajah 1.1	Perkarangan Rumah	1
Rajah 1.2	Perkarangan rumah dengan jenis sampah daun	2
Rajah 1.3	Perkarangan rumah dengan sampah kertas (<i>tissue</i>) dan pembungkusan makanan	2
Rajah 1.4	Contoh pengesanan objek (tetikus) dalam 2D dan 3D	4
Rajah 1.5	Sampel pengesanan objek (tetikus). (a) Foto tetikus di atas kertas (2D); (b) Foto tetikus sebenar (3D)	5
Rajah 2.1	Pembelajaran Dalam (<i>Deep Learning</i>) dalam konteks Kecerdasan Buatan	13
Rajah 2.2	Konsep rangkaian saraf (<i>neural</i>)	14
Rajah 2.3	Lapisan rangkaian saraf konvolusi (CNN)	22
Rajah 2.4	Ilustrasi seni bina CNN	23
Rajah 2.5	Contoh Pengumpulan Maksimum (<i>Max Pooling</i>)	26
Rajah 2.6	Seni Bina LeNet	28
Rajah 2.7	Seni bina AlexNet: Konvolusi, pengumpulan maksimum, LRN dan lapisan bersambung sepenuhnya (<i>Fully Connected Layer</i>)	29
Rajah 2.8	Blok binaan asas rangkaian VGG: Konvolusi (Conv) dan FC untuk lapisan bersambung sepenuhnya	31
Rajah 2.9	Konsep Lapisan permulaan (<i>Inception Layer</i>)	32
Rajah 2.10	Lapisan permulaan dengan pengurangan dimensi	33
Rajah 2.11	Gambar rajah asas blok Baki (<i>Residual Block</i>)	34
Rajah 2.12	Gambar rajah blok asas untuk unit Permulaan-Sisa (<i>Inception-Residual</i>)	35
Rajah 2.13	Persimpangan di atas Kesatuan	39
Rajah 2.14	Matriks kekeliruan	40
Rajah 2.15	Metrik ketepatan	41
Rajah 2.16	Metrik kepersisan	42

Rajah 2.17	Metrik perolehan kembali (<i>recall</i>)	43
Rajah 2.18	Bentuk fizikal penerima TF40 LiDAR dan dimensinya	45
Rajah 2.19	Nombor arah untuk (a) kod rantai 4 arah, dan (b) kod rantai 8 arah	47
Rajah 2.20	Proses pensampelan semula sempadan halangan	48
Rajah 2.21	Proses pengiraan bucu graf dengan kod rantai	48
Rajah 2.22	(A) Persekitaran dengan halangan dan konfigurasi permulaan dan matlamat, dan (B) satu daripada banyak laluan yang mungkin dari permulaan kepada konfigurasi matlamat	52
Rajah 2.23	Nod permulaan dan matlamat dilabelkan	55
Rajah 2.24	Perbandingan prestasi algoritma Dijkstra (kiri) dan A* (kanan)	57
Rajah 2.25	Demonstrasi algoritma carian mendalam-pertama pendalaman berulang	58
Rajah 2.26	Algoritma Dijkstra untuk mencari laluan terpendek dari nod permulaan A ke semua nod lain	60
Rajah 2.27	Perbandingan antara robot tunggal dan berbilang robot apabila merancang laluan oleh algoritma FDCC-PP	63
Rajah 2.28	Algoritma FDCC-PP	64
Rajah 3.1	Proses penyelidikan	71
Rajah 3.2	Reka bentuk fizikal robot berautonomi	71
Rajah 3.3	Gambar rajah blok robot berautonomi	73
Rajah 3.4	Susun atur reka bentuk sistem pengimbasan menggunakan penerima LiDAR	74
Rajah 3.5	Skim litar sistem pengecaman sampah	75
Rajah 3.6	Susunan reka bentuk penerima ultrasonik	76
Rajah 3.7	Litar sistem navigasi robot	77
Rajah 3.8	Susunan reka bentuk motor servo pada lengan robot 6 <i>dof</i>	78
Rajah 3.9	Skema litar sistem lengan robot	79
Rajah 3.10	Paparan skrin telefon pintar	80
Rajah 3.11	Paparan aplikasi Kamera IP	80

Rajah 3.12	Paparan alamat bagi kamera IP	81
Rajah 3.13	Paparan perisian Matlab untuk ujian kamera IP	81
Rajah 3.14	Paparan program ujian kamera IP yang sedang berjalan	82
Rajah 3.15	Paparan modul program ujian penderia LiDAR pada Arduino IDE	82
Rajah 3.16	Paparan modul program ujian motor servo 1 dan motor servo 2	83
Rajah 3.17	Paparan modul program ujian motor DC 1 dan motor DC 2	84
Rajah 3.18	Paparan modul program ujian penderia pengekod	85
Rajah 3.19	Paparan modul program ujian penderia kompas	86
Rajah 3.20	Paparan modul program ujian penderia ultrasonik	87
Rajah 3.21	Paparan modul program ujian motor servo 1 hingga motor servo 6	88
Rajah 3.22	Paparan modul program ujian motor servo 7	89
Rajah 3.23	Metodologi yang dicadangkan untuk identifikasi sampah	90
Rajah 3.24	Aliran dalam pengumpulan data dan pra-pemrosesan	91
Rajah 3.25	Lapan arah pengambilan gambar	92
Rajah 3.26	Lapan hasil imej yang dipangkas	93
Rajah 3.27	Hasil imej tambahan (<i>augmented image</i>)	94
Rajah 3.28	Hasil ubah saiz imej	95
Rajah 3.29	(a) Contoh imej set data sampah sendiri. (b) Imej contoh set data TrashNet. (c) Imej contoh set data LeafSnap	96
Rajah 3.30	Pembelajaran pemindahan mudah menggunakan seni bina CNN (AlexNet, VGGG16, GooogleNet dan ResNet18)	97
Rajah 3.31	Mengubah suai seni bina GoogleNet menggunakan Deep Network Designer	98
Rajah 3.32	Carta alir kajian yang dicadangkan, SCL (<i>Sequential Camera LiDAR</i>)	99
Rajah 3.33	Menukar kotak sempadan kepada koordinat x, y, dan z	100
Rajah 3.34	Laluan proses imbasan imej pantas	100

Rajah 3.35	Kedudukan penderia LiDAR semasa proses imbasan	102
Rajah 3.36	Contoh bacaan penderia LiDAR	103
Rajah 3.37	Carta alir imej pengimbasan pantas	104
Rajah 3.38	Laluan proses imbasan imej terperinci	106
Rajah 3.39	Carta alir mengimbas imej terperinci	107
Rajah 3.40	Laluan biasa pergerakan robot	109
Rajah 3.41	Pengesanan X_1 dan X_2 menggunakan penderia ultrasonik	109
Rajah 3.42	Laluan pergerakan robot jika nilai $X_1 = X_2$	110
Rajah 3.43	Laluan pergerakan robot jika nilai $X_1 < X_2$	110
Rajah 3.44	Laluan pergerakan robot jika nilai $X_1 > X_2$	111
Rajah 3.45	Peta untuk menguji algoritma navigasi robot dengan satu halangan	111
Rajah 3.46	Peta untuk menguji navigasi robot dengan dua halangan	112
Rajah 3.47	Bentuk Fizikal Peta Grid Navigasi Robot	113
Rajah 3.48	Carta alir MFCC-PP	114
Rajah 4.1	Gambar sebenar reka bentuk sistem pengimbasan menggunakan penderia LiDAR	117
Rajah 4.2	Gambar sebenar reka bentuk sistem navigasi robot	119
Rajah 4.3	Gambar sebenar reka bentuk sistem lengan robot	124
Rajah 4.4	Gambar sebenar reka bentuk sistem robot berautonomi	125
Rajah 4.5	Hasil Latihan Kemajuan Menggunakan Seni Bina Pra-latih AlexNet	126
Rajah 4.6	Hasil matriks kekeliruan dengan pra-latih AlexNet	127
Rajah 4.7	Hasil identifikasi sampah menggunakan kamera IP. (a) Kadbod; (b)Pembungkus makanan	127
Rajah 4.8	Hasil identifikasi sampah menggunakan kamera IP. (a) Logam; (b) Kertas	128
Rajah 4.9	Hasil identifikasi sampah menggunakan kamera IP. (a) Kaca; (b) Plastik	128

Rajah 4.10	Hasil identifikasi sampah yang salah	128
Rajah 4.11	Hasil Latihan Kemajuan Menggunakan Seni Bina Pra-latih VGG16	130
Rajah 4.12	Hasil matriks kekeliruan dengan pra-latih VGG16	130
Rajah 4.13	Hasil identifikasi sampah menggunakan kamera IP. (a) Kadbod; (b)Pembungkus makanan	131
Rajah 4.14	Hasil identifikasi sampah menggunakan kamera IP. (a) Logam; (b) Kertas	131
Rajah 4.15	Hasil identifikasi sampah menggunakan kamera IP. (a) Kaca; (b) Plastik	132
Rajah 4.16	Hasil identifikasi sampah yang salah	132
Rajah 4.17	Hasil Latihan Kemajuan Menggunakan Seni Bina Pra-latih GoogleNet	133
Rajah 4.18	Hasil matriks kekeliruan dengan pra-latih GoogleNet	134
Rajah 4.19	Hasil identifikasi sampah menggunakan kamera IP. (a) Kadbod; (b)Pembungkus makanan	134
Rajah 4.20	Hasil identifikasi sampah menggunakan kamera IP. (a) Logam; (b) Kertas	135
Rajah 4.21	Hasil identifikasi sampah menggunakan kamera IP. (a) Kaca; (b) Plastik	135
Rajah 4.22	Hasil identifikasi sampah yang salah	136
Rajah 4.23	Hasil Latihan Kemajuan Menggunakan Seni Bina Pra-latih ResNet18	137
Rajah 4.24	Hasil matriks kekeliruan dengan pra-latih ResNet18	137
Rajah 4.25	Hasil identifikasi sampah menggunakan kamera IP. (a) Kadbod; (b)Pembungkus makanan	138
Rajah 4.26	Hasil identifikasi sampah menggunakan kamera IP. (a) Logam; (b) Kertas	138
Rajah 4.27	Hasil identifikasi sampah menggunakan kamera IP. (a) Kaca; (b) Plastik	139
Rajah 4.28	Hasil identifikasi sampah yang salah	139
Rajah 4.29	Sampel hasil pengimbasan imej objek sampah pantas	142

Rajah 4.30	Sampel hasil pengimbasan imej objek sampah terperinci	143
Rajah 4.31	Perbandingan masa antara pengimbasan pantas dan terperinci	144
Rajah 4.32	Hasil simulasi algoritma Dijkstra dengan Matlab	145
Rajah 4.33	Hasil pengujian algoritma Dijkstra dalam masa nyata	147
Rajah 4.34	Hasil simulasi algoritma A* dengan Matlab	148
Rajah 4.35	Hasil pengujian algoritma A* dalam masa nyata	150
Rajah 4.36	Hasil pergerakan robot dalam bentuk garis	151
Rajah 4.37	Hasil pengujian algoritma FDCC-PP dalam masa nyata	152
Rajah 4.38	Hasil pergerakan robot dalam bentuk garis	153
Rajah 4.39	Hasil pengujian algoritma MFCC-PP dalam masa nyata	155
Rajah 4.40	Hasil simulasi algoritma Dijkstra dengan Matlab	157
Rajah 4.41	Hasil pengujian algoritma Dijkstra dalam masa nyata	160
Rajah 4.42	Hasil simulasi algoritma Dijkstra dengan Matlab	161
Rajah 4.43	Hasil pengujian algoritma A* dalam masa nyata	163
Rajah 4.44	Hasil pengujian algoritma FDCC-PP dalam masa nyata	165
Rajah 4.45	Hasil pengujian algoritma MFCC-PP dalam masa nyata	167

SENARAI SINGKATAN

AI	Artificial Intelligence
CNN	Convolutional Neural Network
CCW	Counter Clock Wise
CW	Clock Wise
DL	Deep Learning
DOF	Degree Of Freedom
FDCC-PP	Freeman Direction Chain Code Path Planning
FN	False Negative
FP	False Positive
GPU	Graphic Processing Unit
IP	Internet Protocol
LiDAR	Light Detection and Ranging
MFCC-PP	Modified Freeman Chain Code Path Planning
OSR	Outdoor Service Robot
RGB	Red Green Blue
SCL	Sequential Camera LiDAR
UKM	Universiti Kebangsaan Malaysia

BAB I

PENDAHULUAN

1.1 PENGENALAN

Sebuah kompleks rumah yang bersih adalah impian bagi semua orang. Menjaga kebersihan adalah nasihat Nabi Muhammad Shallallahu 'alaihi wa sallam, kebersihan itu sebahagian daripada iman (Hadis Riwayat At-Tirmidzi). Hal ini diperkuat dengan sabda Nabi Shallallahu 'alaihi wa sallam, Allah itu indah dan menyukai keindahan (Hadis Riwayat Muslim, Ahmad, dan At-Tirmidzi).



Rajah 1.1 Perkarangan Rumah

Kawasan perkarangan rumah yang bersih tentu sahaja bermakna tidak ada sampah di sekitar kawasan tersebut. Sampah boleh disebabkan oleh daun yang gugur dari pokok, sampah plastik seperti botol plastik, paket makanan ringan dan sampah lainnya. Biasanya sampah di perkarangan rumah dibersihkan oleh pemilik rumah atau orang yang ditugaskan untuk melakukannya. Sekiranya pemilik rumah sibuk dan tidak sempat membersihkan, maka akan kelihatan tidak bersih kerana sampah bertaburan di perkarangan rumah.



Rajah 1.2 Perkarangan rumah dengan jenis sampah daun



Rajah 1.3 Perkarangan rumah dengan sampah kertas (*tissue*) dan pembungkusan makanan

Selaras dengan perkembangan teknologi komputer dan robotik, timbul idea untuk mengatasi masalah tersebut. Salah satunya dilakukan oleh Y.Fuchikawa pada tahun 2005 menggunakan robot *Outdoor Service Robot* (OSR) untuk mengutip sampah (Fuchikawa et al. 2005). Penyelidikan ini hanya untuk mengumpul sampah plastik dalam bentuk botol PET. Satu lagi penyelidikan telah dijalankan oleh kumpulan penyelidik Itali yang diketuai oleh Barbara Mazzolai (Salvini et al. 2011). Nama robotnya iaitu *Dust Cart Robot*. Robot ini mengumpul jenis sampah berdasarkan input pengguna. Selepas pengguna memasukkan jenis sampah, robot membuka stor sampah mengikut jenis input sampah. Penggunaan kamera pada robot adalah untuk navigasi dalam mengelak halangan, bukan untuk mengklasifikasikan jenis sampah.

Dalam penyelidikan ini, dilakukan penyepaduan teknologi sedia ada dalam robotik dan penglihatan komputer untuk identifikasi sampah dan menavigasi pada robot berautonomi. Pengesanan objek sampah dijalankan menggunakan kamera. Jumlah sampah yang boleh dikesan dijangka lebih daripada penyelidikan sebelum ini. Robot

juga mesti boleh mengemudi secara kemudi dengan baik dan mengenali persekitarannya.

Bagi membolehkan perkara di atas dilakukan, adalah penting untuk sistem robot tersebut memiliki keadah pengesanan sampah yang optimum. Penyelidikan ini mencadangkan aplikasi teknik penglihatan komputer yang menggunakan gabungan Rangkaian Saraf Konvolusi (CNN) dan penderia *Light Detection and Ranging* (LiDAR). Bagi sistem navigasi robot pula, kaedah pengoptimuman laluan yang dicadangkan adalah melalui penggunaan teknik Perancangan Laluan (*path planning*) yang digabungkan dengan penderia ultrasonik, penderia pengekod, dan penderia kompas.

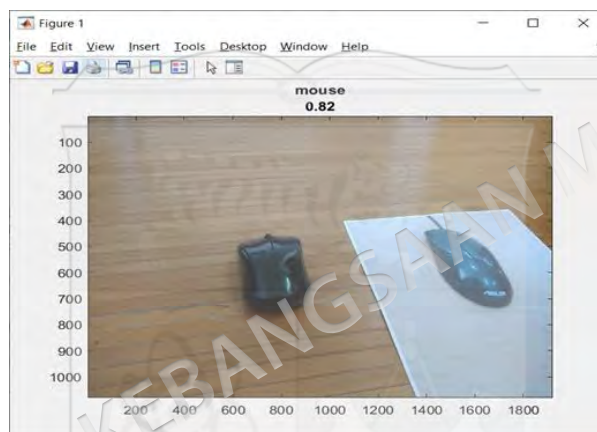
1.2 PERMASALAHAN KAJIAN

Pada masa ini, perkhidmatan perbandaran yang penting, seperti air, sanitasi, dan pengurusan sisa pepejal, dianggap sebagai tanggungjawab kerajaan tempatan atau negara (Baud et al. 2004). Penyelidikan dalam identifikasi dan pengelasan sampah telah dilakukan oleh beberapa penyelidik sebelum ini. Walau bagaimanapun, jumlah sampah yang dikesan adalah kurang optimum dan kurang tepat. Pengoptimuman boleh memainkan peranan penting dalam penglihatan komputer kerana banyak algoritma penglihatan komputer menggunakan langkah pengoptimuman di dalam beberapa fasa yang penting, seperti pra-pemprosesan serta pengecaman objek (Treiber 2013).

Identifikasi sampah adalah langkah penting sebelum pengasingan, dan ia boleh dilakukan dengan cekap dengan bantuan pembelajaran mesin yang berbeza dan algoritma pemprosesan imej. Penambahbaikan dalam identifikasi sampah telah dijalankan di Universiti Stanford di mana penyelidikan mengenai jenis sampah telah dikelaskan menggunakan kaedah CNN (M. Yang & Thung 2016). Walau bagaimanapun, jumlah sampah yang boleh diidentifikasi hanya sehingga 6 item sampah sahaja (Mao et al. 2021) (M. W. Rahman et al. 2022) (Adedeji & Wang 2019), manakala di perkarangan rumah, sampah melebihi enam item.

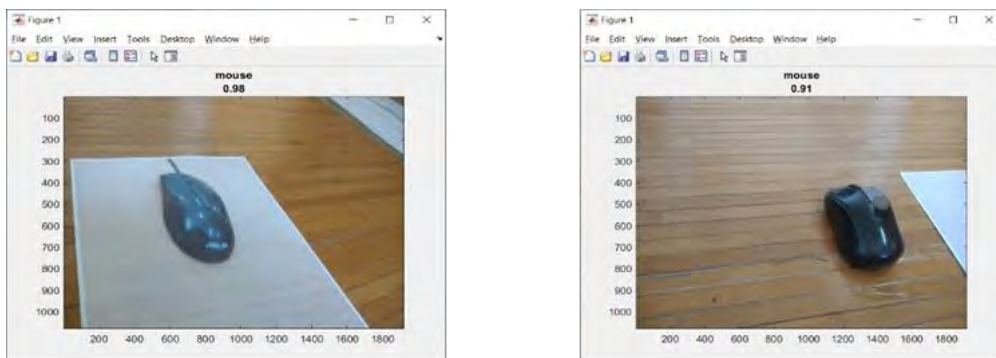
Memajukan peningkatan dalam kebolehpercayaan dan kelajuan pemprosesan sistem penglihatan dan eksperimen dengan sampah mesti dibuat (Fuchikawa et al.

2005). CNN paling disukai untuk pengkelasan imej (Hulyalkar S., Deshpande R., Makode K. 2018). Walau bagaimanapun, kebanyakan kaedah identifikasi objek tidak dapat membezakan sama ada objek itu adalah imej dalam bentuk 3D atau imej dalam bentuk gambar pada kertas (2D) (Mao et al. 2021) (Z.-Q. Zhao et al. 2019) (Funch et al. 2021) (Ren et al. 2021)(C. Liu et al. 2022) (Wu et al. 2020). Rajah 1.4 menunjukkan objek ujian (tetikus) dalam 2D dan 3D. Objek dikesan menggunakan CNN. Keputusan ujian pengesanan objek dalam bentuk 2D dan 3D dapat dikesan dengan baik, dengan ramalan 82%. Dalam Rajah 1.4, objek 2D terletak di sebelah kanan Rajah 1.4. Pada masa yang sama, objek 3D terletak di sebelah kiri Rajah 1.4.



Rajah 1.4 Contoh pengesanan objek (tetikus) dalam 2D dan 3D

Kemudian, setiap objek dikesan iaitu objek 2D dan objek 3D menggunakan CNN. Perkakasan menggunakan komputer riba Intel Core i7 (generasi ke-8), RAM 16GByte, kad grafik NVIDIA GeForce RTX2060 dengan perisian perisian MATLAB 2020a. Untuk objek 2D, sampel digunakan sebagai foto objek di atas kertas. Keputusan ujian untuk objek 2D boleh dilihat dalam Rajah 1.5 (a). Keputusan ujian diperolehi dengan ramalan 98%.



Rajah 1.5 Sampel pengesanan objek (tetikus). (a) Foto tetikus di atas kertas (2D); (b) Foto tetikus sebenar (3D)

Tambahan pula, ujian untuk objek dalam bentuk 3D. Objek ini diletakkan di sebelah objek 2D. Keputusan ujian boleh dilihat dalam Rajah 1.5 (b). Keputusan ujian diperolehi dengan ramalan 91%. Daripada ketiga-tiga keputusan ujian ini, boleh dikatakan CNN dapat mengesan objek dengan baik. Bermakna bahawa CNN boleh melabelkan nama objek, tetapi tidak dapat membezakan antara objek sebenar (3D) dan yang hanya dalam bentuk foto (2D). Ini akan menjadi masalah jika objek dialihkan dari satu tempat ke tempat lain menggunakan pencengkam robot. Bagaimana jika robot ingin mengalihkan sampah (objek) sedangkan sampah (objek) hanyalah gambar sampah di atas kertas? Oleh itu, robot mesti mampu mengidentifikasi sampah (objek) dalam bentuk 3D.

Dalam sebarang aplikasi, boleh digunakan pengelas tertentu untuk cuba mengoptimumkan prestasinya. Pendekatan yang biasa dilakukan adalah untuk mencuba beberapa pengelas yang berbeza dan memilih yang berprestasi terbaik pada set pengesanan (*validation*) yang berasingan (Dougherty 2013). Dalam kes ini, kami mencadangkan CNN digabungkan dengan penerima LiDAR untuk identifikasi sama ada objek itu dalam bentuk 3D atau tidak (2D). CNN dipilih kerana, pada masa ini, CNN adalah kaedah pembelajaran mesin terbaik dalam pengecaman objek. Oleh itu, kajian ini mencadangkan pengoptimuman pengecaman sampah di kawasan perkarangan rumah menggunakan CNN.

Navigasi robot sangat penting dalam sistem robotik. Matlamat utama navigasi adalah untuk robot bergerak dari kedudukan awal ke kedudukan matlamat dengan mengikut laluan yang selamat dan menjana panjang laluan yang optimum. Terdapat

beberapa peraturan dalam navigasi robot, antara lain; panjang trek mestilah minimum, trek yang ideal ialah laluan yang boleh dilalui oleh robot dalam masa yang sesingkat mungkin, trek mestilah sejauh mungkin daripada halangan, trek mestilah licin dan bebas daripada selekoh tajam, trek mesti mematuhi sekatan perjalanan.

Terdapat cadangan baharu dalam navigasi global, iaitu Kod Rantaian Arah Freeman–Perancangan Laluan (*Freeman Direction Chain Code – Path Planning / FDCC-PP*). Algoritma FDCC-PP ini telah dicadangkan oleh M. Zhao dan Lu pada 2019 (M. Zhao & Lu 2019). Penyelidikan ini lebih tertumpu kepada penambahbaikan algoritma FDCC-PP yang dicadangkan oleh mereka.

Algoritma FDCC-PP telah dibandingkan dengan algoritma lain. Hasilnya ialah algoritma ini boleh mengurangkan lebih carian dalam ruang kosong dan memendekkan masa perancangan laluan tanpa menjejaskan keselamatan robot. Algoritma ini membolehkan perancangan laluan pantas dalam persekitaran yang kompleks. Berbeza daripada algoritma lain, maklumat koordinat dan maklumat tentang arah sempadan halangan diperoleh daripada pemprosesan peta dalam algoritma FDCC-PP. Oleh itu, apabila robot berada berhampiran halangan, ia hanya perlu bertanya kod rantai arah halangan yang sepadan dan memilih lencongan terpendek, bukannya mengelilingi halangan untuk mencari laluan lencongan terbaik sebelum bergerak ke arah gol semula. Walau bagaimanapun, algoritma ini masih menggunakan kod rantai 4 dan 8 Freeman.

Dalam kod rantai 4, robot berpusing pada sudut 90 darjah. Terdapat beberapa keburukan sekiranya putaran 90 darjah ini dijalankan iaitu; 1. Robot mesti mengurangkan kelajuan Bergeraknya dan kemudian berhenti. Satu teknik yang boleh digunakan ialah Modulasi Lebar Denyut (*Pulse Width Modulation / PWM*). Selepas itu, robot akan berpusing pada sudut 90 darjah. Keadaan ini akan melambatkan masa robot untuk mencapai matlamat. 2. Jika robot berhenti serta-merta tanpa mengurangkan kelajuan, kemudian berpusing dengan memutar satu arah roda motor DC robot, maka ini boleh merosakkan pemandu motor DC dan motor DC dalam jangka masa panjang. Faktor merosakkan ini dikenali sebagai tekanan haba.

Manakala dalam kod rantai 8, terdapat dua pilihan untuk membelok iaitu pada sudut 90 darjah dan 45 darjah. Ini dianggap tidak lancar, jadi perlu meningkatkan sudut pusingan robot. Selain itu, robot tersebut boleh mengalami kesukaran melepasi halangan. Untuk mengatasinya, robot mesti boleh bergerak dengan lebih lancar. Dalam penyelidikan ini, kod rantai 16 dan 32 dicadangkan. Kod rantai 16 akan menghasilkan sudut pusingan maksimum 22.5 darjah. Kod rantai 32 menghasilkan sudut pusingan maksimum 11.25 darjah. Dengan cara ini, diharapkan robot akan lebih lancar membelok dan dapat mengelak halangan yang memerlukan robot bergerak pada sudut kurang daripada 45 darjah.

1.3 PERSOALAN KAJIAN

Berdasarkan pernyataan masalah yang dikemukakan, maka pernyataan permasalahan dalam bentuk Persoalan Kajian (*Research Question*, RQ) adalah sebagai berikut :

1. Bagaimana membina protaip sistem robot berautonomi yang mampu mengoptimasi identifikasi sampah dan navigasi robot di perkarangan rumah ?
2. Bagaimana membangunkan algoritma optimasi identifikasi sampah di perkarangan rumah ?
3. Bagaimana membangunkan algoritma optimasi navigasi robot berautonomi di perkarangan rumah ?
4. Bagaimana menguji keberkesanan dan prestasi keseluruhan sistem yang dibangunkan ?

1.4 OBJEKTIF KAJIAN

Kajian ini berfokus pada pengembangan teknik baru untuk :

1. Membina protaip sistem robot berautonomi yang mampu mengoptimasi identifikasi sampah dan navigasi robot di perkarangan rumah.
2. Membangunkan algoritma optimasi identifikasi sampah di perkarangan rumah.

3. Membangunkan algoritma optimasi navigasi robot berautonomi di perkarangan rumah.
4. Menguji keberkesanan dan prestasi keseluruhan sistem yang dibangunkan.

1.5 SKOP KAJIAN

Kajian ini akan memfokuskan kepada identifikasi sampah di perkarangan rumah di atas satah rata. Sampah yang dikenal pasti adalah sampah dalam bentuk objek 3D. Sekiranya terdapat sampah kertas dalam bentuk 2D, maka ini akan diabaikan kerana pencengkam robot tidak dapat mengambil sampah tersebut. Navigasi robot berautonomi dilakukan di dalam kawasan rumah yang telah dimodelkan. Manakala proses mengutip sampah tidak termasuk dalam skop penyelidikan. Proses identifikasi sampah dan menavigasi robot hanya dilakukan dalam cuaca cerah, bukan hujan atau gelap.

1.6 SUMBANGAN KAJIAN

Kepentingan kajian ini ialah:

1. Menyumbang dalam mengoptimumkan proses identifikasi sampah sama ada dari segi kuantiti maupun pengesanan ketepatan.
2. Menyumbang dalam mengoptimumkan proses navigasi robot berautonomi untuk mengenali persekitaran sekitar kawasan rumah.
3. Memberi sumbangan dalam menjaga kebersihan persekitaran bebas sampah terutama sampah di perkarangan rumah.
4. Mungkin berimpak kepada meningkatkan kualiti hidup.

1.7 ORGANISASI BAB

Tesis ini mengandungi lima (5) bab. Berikut adalah ringkasan penerangan bagi setiap bab:

Bab I membicarakan tentang pengenalan umum kajian yang menyediakan ulasan latar belakang kajian, pernyataan masalah, persoalan kajian dan objektif kajian. Selain daripada itu bab ini juga menerangkan skop kajian dan sumbangan yang dihasilkan daripada kajian.

Bab II menyediakan ulasan literatur yang lengkap mengenai kerja penyelidikan yang berkaitan dengan penyeldidikan ini. Topik yang dikaji termasuk pengenalan tentang sampah dan jenis-jenis sampah, Pembelajaran Dalam, CNN, navigasi robot berautonomi, kajian terdahulu yang berkaitan dengan CNN dalam identifikasi sampah, dan kajian terdahulu yang berkaitan dengan menavigasi robot berautonomi.

Bab III menerangkan aliran kajian yang dijalankan berdasarkan metodologi penyelidikan. Bab ini juga memberi penerangan yang terperinci tentang reka bentuk robot berautonomi, pembuatan robot berautonomi, pengumpulan dan ciri-ciri set data, pra-pemrosesan data, pembuatan algoritma optimasi identifikasi sampah, pembuatan algoritma optimasi navigasi robot berautonomi. Penerangan tentang pembinaan algoritma untuk identifikasi sampah dan menavigasi robot berautonomi menggunakan MATLAB mengakhiri Bab III.

Bab IV membicarakan pengujian keberkesanan dan prestasi keseluruhan sistem yang dibangunkan, dengan menganalisis hasil eksperimen kajian pada setiap peringkat implementasi pembinaan model untuk identifikasi sampah dan menavigasi robot berautonomi. Pengoptimuman dalam identifikasi sampah dan navigasi robot berautonomi yang dicadangkan juga diterangkan dan dibincangkan dengan lebih terperinci.

Bab V akan merumuskan penemuan dan sumbangan utama kajian ini. Bab ini juga membincangkan peluang penyelidikan masa hadapan yang memanjangkan penemuan dan batasan kajian ini.

BAB II

KAJIAN LITERATUR

2.1 PENGENALAN

Tujuan utama kajian ada 2 iaitu ; 1. Melaksanakan penyelidikan yang berkaitan dengan pengoptimuman mengenal pasti sampah di perkarangan rumah. Tujuan berikutnya iaitu pengoptimuman robot dalam bernavigasi di perkarangan rumah. Bab ini akan mengupas penyelidikan, terminologi, kaedah dan pendekatan yang berkaitan dengan kajian ini. Dalam kajian ini sampah yang diidentifikasi di perkarangan dikelompokkan ke dalam 11 jenis sampah. Metode yang digunakan dalam mengenal pasti sampah adalah Rangkaian Saraf Konvolusi (CNN). Rangkaian Saraf Konvolusi (CNN) ialah salah satu daripada algoritma Pembelajaran Dalam (*Deep Learning*) yang paling berkuasa. CNN mempunyai banyak aplikasi dalam pengelasan imej, pembahagian dan pengesanan (C. Zhang et al. 2018), (Krizhevsky et al. 2017), (Sermanet et al. 2013), (Zeiler & Fergus 2014). Oleh itu, dalam penyelidikan ini CNN dicadangkan untuk mengenal pasti sampah. Arsitektur yang digunakan dalam penyelidikan ini diantaranya adalah ; AlexNet (Krizhevsky et al. 2017), VGG (Simonyan & Zisserman 2014), GoogleNet (Szegedy et al. 2016), dan ResNet (He et al. 2016).

2.1.1 Definisi Sampah

Menurut Undang-undang No. 18 (2008) Republik Indonesia tentang Pengelolaan Sampah, sampah bermaksud sisa pepejal dari kegiatan manusia sehari-hari dan/atau proses alam (Ambina 2019). Bahan buangan ini dihasilkan setiap kali manusia melakukan aktiviti harian. Pengurusan sampah adalah isu penting bagi semua negara, baik yang maju mahupun membangun. Pemilahan dan pengurusan sampah adalah bahagian penting dalam melindungi alam sekitar dan memaksimumkan kecekapan sumber. Selain mengurangkan pembaziran dan pembaziran, komoditi boleh guna

semula diunjurkan membawa faedah yang ketara kerana ia mengurangkan pergantungan kita kepada bahan mentah. Penggunaan kompos sampah boleh mengembang melebihi baja dan baja selepas mengitar semula logam, bahan kimia dan kaca (Kshirsagar et al. 2022).

Untuk meminimumkan masalah sampah, rawatan sampah hendaklah bermula di punca. Pengurusan sampah adalah satu aktiviti yang sistematik, menyeluruh dan berterusan melibatkan pengurangan dan pengurusan sampah. Pengurusan sampah perlu dilaksanakan dan disepadukan sepenuhnya dari hulu ke hilir. Ini demi memberi manfaat kepada ekonomi, sihat untuk masyarakat, selamat untuk alam sekitar dan mengubah tingkah laku manusia.

2.1.2 Jenis-jenis Sampah

Jenis sampah-sampah boleh dibahagikan kepada dua jenis, organik dan bukan organik, mengikut sifatnya (Priti & Mandal 2022).

1. Sampah organik (terurai); Definisi sampah organik ialah sampah yang boleh diuraikan dan di kompos. Contohnya, sampah mentah, daun mati, sayur-sayuran.
2. Sampah tak organik (tidak terurai); takrifan sampah tak organik ialah sampah yang berterusan dan tidak boleh terurai. Walau bagaimanapun, sampah bukan organik boleh dikitar semula menjadi perkara baru yang berguna. Contoh: Botol PET, kertas terpakai, kad bodi, tin terpakai.

Berdasarkan bentuknya, sampah boleh dibahagikan kepada dua kumpulan: pepejal dan cecair.

1. Pepejal:

Pepejal ialah bahan yang dibuang oleh manusia (selain najis manusia). Jenis sampah ini termasuk plastik terpakai, kaca pecah, tin terpakai, sisa dapur.

2. Cecair:

Cecair ialah bahan cecair yang tidak diperlukan dan dibuang ke dalam tong sampah. Contohnya, daripada sisa cecair dari tandas kepada cecair dari dapur dan dobi.

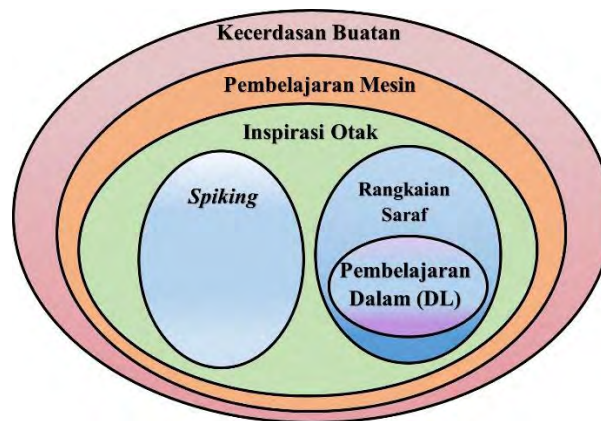
2.1.3 Pengurusan Sampah

Daripada semua jenis sampah, yang paling penting untuk perhatian dunia ialah sampah plastik (Abdel-Shafy & Mansour 2018) (Subramanian 2000). Ia bukan sahaja merosakkan daratan, malah sampah plastik turut dibawa ke laut sehingga mengancam ekosistem marin. Kecenderungan orang menggunakan plastik, jika dilihat dari sudut sosiologi, adalah fenomena di mana orang ingin lekas dan praktikal. Daripada menggunakan pembungkus daun dan sebagainya, plastik ini agak lebih pantas, praktikal, murah dan mudah didapati di mana-mana. Oleh itu, perlu ada pendidikan tentang kepentingan pengetahuan tentang bahaya sampah plastik, kesedaran tentang cara bertindak balas terhadap plastik.

Selaras dengan perkembangan teknologi robotik, dalam penyelidikan ini sampah di perkarangan rumah diuruskan oleh robot berautonomi. Tugas robot adalah untuk mengidentifikasi sampah dan mencari sampah di perkarangan rumah. Agar tugas berfungsi dengan baik, proses identifikasi dan navigasi robot berautonomi mesti dioptimumkan sebaik mungkin.

2.2 PEMBELAJARAN DALAM (*DEEP LEARNING*)

Sejak tahun 1950-an, subset kecil Kecerdasan Buatan (*Artificial Intelligence*), sering dipanggil Pembelajaran Mesin (*Machine Learning*), telah merevolusikan beberapa bidang dalam beberapa dekad yang lalu. Rangkaian Saraf (*Neural Network*) ialah subbidang *Machine Learning*, dan subbidang inilah yang melahirkan Pembelajaran Dalam (*Deep Learning*). Sejak penubuhannya, Pembelajaran Dalam telah menunjukkan kejayaan cemerlang dalam hampir setiap domain aplikasi. Rajah 2.1 menunjukkan taksonomi Kecerdasan Buatan (*Artificial Intelligence* / AI) (Sze et al. 2017).



Rajah 2.1 Pembelajaran Dalam (*Deep Learning*) dalam konteks Kecerdasan Buatan

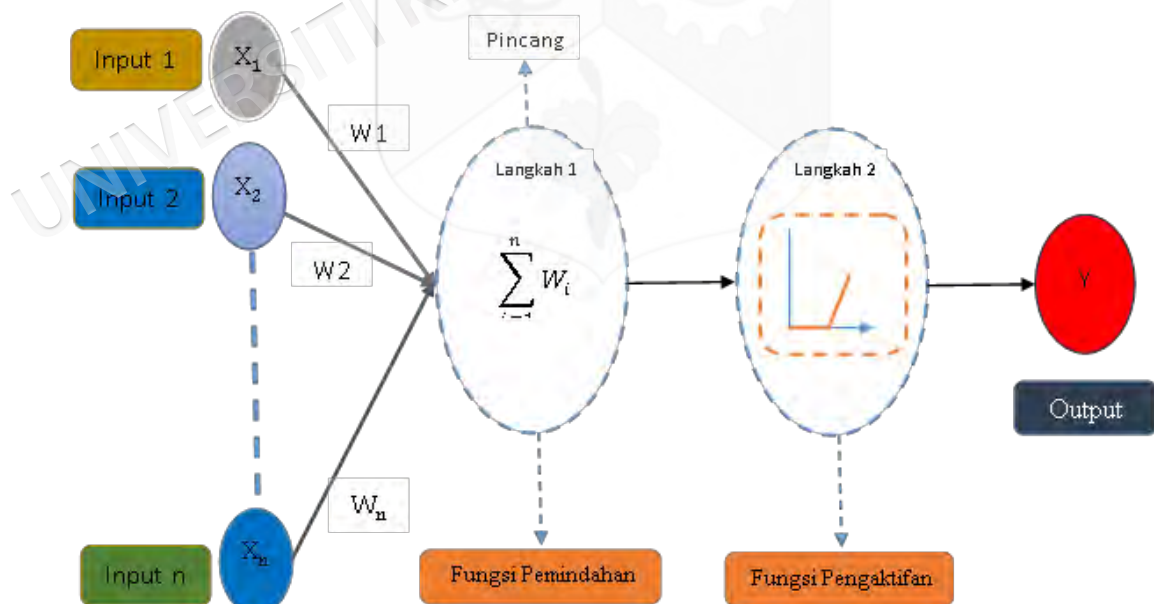
Sumber: Sze et al. 2017

Awal populariti istilah Pembelajaran Dalam (*Deep Learning*) bermula apabila Geoffrey Hinton memperkenalkan varian rangkaian saraf tiruan yang dipanggil jaring kepercayaan yang dalam (*deep belief nets*) dalam kertas kerjanya pada tahun 2006 (Hinton et al. 2006). Ideanya adalah untuk melatih salah satu daripada jenis model rangkaian saraf ini dengan melatih dua lapisan kemudian menambah satu lapisan di atasnya, kemudian melatih hanya lapisan atas dan seterusnya. Dengan cara ini, adalah mungkin untuk melatih model rangkaian saraf dengan lebih banyak lapisan (berbilang lapisan) daripada model sebelumnya. Model ini boleh mempunyai berjuta-juta parameter dan oleh itu memerlukan sejumlah besar data untuk mencari kesesuaian optimum. Walau bagaimanapun, walaupun terdapat keperluan tinggi pada set data dan perkakasan pengkomputeran, Pembelajaran Dalam ialah teknik yang sangat berkuasa yang digunakan dalam banyak sistem lanjutan dalam Penglihatan Komputer (*Computer Vision*) (M. Tan & Le 2019), dan Pemprosesan Bahasa Semula Jadi (*Natural Speech Processing*) (Coltman 2002), dan banyak lagi.

Selepas istilah Pembelajaran Dalam menjadi popular, Pembelajaran Dalam tidak menjadi sangat menarik kepada penyelidik. Ini kerana rangkaian saraf tiruan dengan banyak lapisan mempunyai algoritma yang sangat kompleks untuk memproses data, memerlukan komputer yang sangat ditentukan, yang tidak cekap secara pengiraan pada masa itu. Kemudian, pada tahun 2009, Andrew Ng dan rakan-rakannya memperkenalkan penggunaan Unit Pemprosesan Grafik (GPU) untuk Pembelajaran Dalam pada artikel bertajuk "Pembelajaran Tanpa Pengawasan Dalam Berskala Besar

menggunakan Pemproses Grafik". Menggunakan GPU untuk rangkaian saraf tiruan boleh mempercepatkan proses pengiraan berbanding menggunakan CPU. Dengan perkakasan yang mencukupi tersedia untuk pembangunan Pembelajaran Dalam, teknologi Pembelajaran Dalam itu sendiri telah melahirkan produk seperti pengecaman muka, kereta pandu sendiri dan Pemrosesan Ucapan (*Speech Processing*) pengecaman pertuturan (Alzubaidi et al. 2021).

Pembelajaran Dalam (DL) ialah teknologi yang mampu berfungsi menggunakan algoritma tertentu. Malah, tiada algoritma Pembelajaran Dalam yang dianggap sempurna. Kerana, setiap jenis mempunyai keupayaan yang berbeza. Oleh itu, pembangun aplikasi mesti memilih jenis algoritma yang paling sesuai dengan keperluan mereka, untuk memilih algoritma yang betul. Pembelajaran Dalam menggunakan rangkaian saraf tiruan untuk melakukan pengiraan yang canggih pada sejumlah besar data yang berfungsi pada struktur dan fungsi otak manusia. Rajah 2.2 adalah rangkaian neural berstruktur seperti otak manusia dan terdiri daripada neuron buatan, juga dikenali sebagai nod. Nod ini disusun bersebelahan antara satu sama lain dalam tiga lapisan iaitu lapisan input, lapisan tersembunyi, lapisan keluaran.



Rajah 2.2 Konsep rangkaian saraf (*neural*)

Sumber: Krizhevsky et al. 2017

Data menyediakan setiap nod dengan maklumat dalam bentuk input. Nod mendarabkan input dengan berat rawak, mengiranya, dan menambah bias. Akhir sekali, fungsi tak linear, juga dikenali sebagai fungsi pengaktifan (*activation function*), digunakan untuk menentukan neuron mana yang akan diaktifkan.

Algoritma Pembelajaran Dalam (*Deep Learning*) berfungsi dengan hampir semua jenis data dan memerlukan sejumlah besar kuasa pengkomputeran dan maklumat untuk menyelesaikan masalah yang kompleks. Oleh itu, Pembelajaran Dalam boleh digunakan dalam pelbagai aplikasi kehidupan sebenar. Sebagai contoh, dalam Pemrosesan Imej Digit (*Digital Image Processing*); pewarnaan imej skala kelabu daripada gambar yang pernah dilakukan secara manual oleh pengguna yang terpaksa memilih setiap warna berdasarkan pertimbangan mereka sendiri. Dengan mengaplikasikan algoritma Pembelajaran Dalam, pewarnaan boleh dilakukan secara automatik oleh komputer (Najafabadi et al. 2015). Sebagai contoh lain, bunyi sebagai input ke dalam drum video yang dimatikan (*mute drumming video*) menggunakan Rangkaian Saraf Berulang (RNN) adalah sebahagian daripada kaedah Pembelajaran Dalam. (Marra et al. 2018).

Pembelajaran Dalam boleh difahami sebagai kaedah untuk meningkatkan hasil dan mengoptimumkan masa pemprosesan dalam beberapa proses pengkomputeran. Dalam bidang pemprosesan bahasa semula jadi, kaedah Pembelajaran Dalam telah digunakan untuk Penjanaan Kapsyen Imej (Mohamed, Abdel-rahman and Dahl, George and Hinton 2009) dan Penjanaan Tulisan Tangan (Deng 2014). Aplikasi berikut dikategorikan dalam pemprosesan imej digital tulen, perubatan dan biometrik.

2.2.1 Pemprosesan Imej

Sebelum Pembelajaran Dalam secara rasmi muncul sebagai pendekatan penyelidikan baharu, beberapa aplikasi telah dijalankan dalam konsep pengecaman corak melalui pemprosesan lapisan. Pada tahun 2003, satu contoh menarik telah dibangunkan dengan menggunakan Penapisan Zarah (*Particle Filtering*) dan Penyebaran Kepercayaan Bayesian (*Bayesian-Belief Propagation*). Konsep utama aplikasi ini mencadangkan bahawa manusia boleh mengenali wajah seseorang dengan hanya menonton separuh - gambar muka yang dipangkas, komputer boleh membina semula imej wajah daripada

yang dipotong (Mosavi & Vaezipour 2012). Kemudian pada tahun 2006, algoritma tamak (*greedy algorithm*) dan hierarki (*hierarchy*) telah digabungkan menjadi aplikasi yang mampu memproses digit tulisan tangan (Khurana et al. 2023). Penyelidikan terkini telah menggunakan Pembelajaran Dalam sebagai alat utama untuk pemrosesan imej digital. Sebagai contoh, menggunakan Rangkaian Saraf Konvolusi (CNN) untuk Pengecaman Iris boleh menjadi lebih berkesan daripada menggunakan penderia iris konvensional. Keberkesanan CNN boleh mencapai sehingga 99.79% ketepatan (M. M. Rahman et al. 2019). Pengecaman lokasi mudah alih pada masa kini membolehkan pengguna mengetahui alamat yang ditentukan berdasarkan gambar. Algoritma Semantik Terkawal–Memelihara Pencincangan Dalam (*Supervised Semantics – Preserving Deep Hashing / SSPDH*) telah membuktikan peningkatan yang ketara berbanding dengan Bit Cincang Visual (*Visual Hash Bit/VHB*) dan Pemilihan Cap Jari Space-Saliency (*Space-Saliency Fingerprint Selection/SSFS*). Ketepatan SSPDH adalah 70% lebih cekap (J.-G. Lee et al. 2017).

Terdapat juga tinjauan yang baik mengenai pendekatan Pembelajaran Dalam untuk pemrosesan imej dan tugas berkaitan Penglihatan Komputer (*Computer Vision*) (Chai et al. 2021). Resolusi super imej tunggal menggunakan kaedah CNN (S. Kim et al. 2021). Penghapusan imej menggunakan CNN padanan blok (Arvanitis et al. 2020). Penilaian estetik foto menggunakan *A-Lamp (Adaptive Layout-Aware Multi-Patch Deep CNN)* (Ma et al. 2017). DCNN untuk pengimejan spektrum hiper untuk segmentasi menggunakan Markov Random Field (MRF) (Cao et al. 2018). Pendaftaran imej menggunakan CNN (B. Kim et al. 2019). *The Hierarchical Deep CNN* untuk Pemindahan Gaya Artistik Pantas (Jing et al. 2020). Pembahagian latar belakang menggunakan DCNN (Hirner & Fraundorfer 2021). Pengecaman aksara tulisan tangan menggunakan pendekatan CNN (Kusetogullari et al. 2021). Klasifikasi imej optik menggunakan pendekatan Pembelajaran Dalam (Ara et al. 2022). Pengecaman objek menggunakan rangkaian berulang serentak selular dan Rangkaian Saraf Konvolusi (CNN) (Alom et al. 2017).

Akhir sekali, satu lagi aplikasi yang luar biasa dalam pemrosesan imej digital, menggunakan kaedah Pembelajaran Dalam, ialah pengecaman muka. Google, Facebook dan Microsoft mempunyai model pengecaman muka Pembelajaran Dalam

yang unik (Domingues & Rosário 2019). Akhir-akhir ini, pengenalan berdasarkan gambar muka telah berubah kepada pengecaman automatik dengan menetapkan umur dan jantina sebagai parameter awal. Sighthound Inc. misalnya menguji algoritma Rangkaian Saraf Konvolusi (CNN) yang mendalam yang mampu mengenali bukan sahaja umur dan jantina, malah emosi (Agarwal et al. 2021). Tambahan pula, sistem yang teguh telah dibangunkan untuk menentukan umur dan jantina seseorang dengan tepat daripada satu imej dengan penerapan seni bina pembelajaran pelbagai tugas yang mendalam (Liao et al. 2022).

2.2.2 Pengimejan Perubatan

Dalam Pengimejan Perubatan untuk tugas klasifikasi, pengesanan, dan segmentasi imej terdapat tinjauan yang baik menggunakan Pembelajaran Dalam (*Deep Learning/DL*) (Litjens et al. 2017). Beberapa kertas kerja yang diterbitkan selepas kajian ini termasuk: MD Net, yang dibangunkan untuk diagnosis perubatan dengan imej dan penerangan teks yang sepadan (Z. Zhang et al. 2017). Kajian lain iaitu ; 1. Segmentasi Jantung menggunakan MRI Paksi pendek (*short-axis MRI*) (Tran 2016), 2. Segmentasi cakera optik dan vaskular retina menggunakan Rangkaian Saraf Konvolusi (CNN) (J. H. Tan et al. 2017), 3. Pembahagian tumor otak menggunakan hutan rawak dengan ciri yang dipelajari dengan Rangkaian Saraf Konvolusi Sepenuhnya (FCNN) (Moeskops et al. 2016).

Pemprosesan imej tidak dapat dinafikan merupakan bahagian penting dalam bidang penyelidikan di mana kaedah Pembelajaran Dalam boleh digunakan. Oleh itu, aplikasi klinikal telah diuji baru-baru ini. Sebagai contoh, perbandingan antara pembelajaran cetek dan Pembelajaran Dalam pada rangkaian saraf membawa kepada prestasi yang lebih baik pada ramalan penyakit. Imej yang diambil daripada Pengimejan Resonans Magnetik (MRI) (Chauhan et al. 2019) daripada otak manusia telah diproses untuk meramalkan kemungkinan penyakit Alzheimer (Patil et al. 2022). Walaupun kejayaan awal prosedur ini, beberapa isu perlu dipertimbangkan untuk permohonan masa hadapan. Latihan dan pergantungan pada kualiti tinggi adalah beberapa batasan. Jumlah, kualiti dan kerumitan data adalah aspek yang mencabar, namun penyepaduan jenis data heterogen adalah aspek potensi seni bina Pembelajaran Dalam (Sarker 2021). *Optical Coherence Tomography* (OCT) ialah satu lagi contoh di mana kaedah

Pembelajaran Dalam menunjukkan hasil yang berharga. Secara konvensional, imej dirawat dengan pembangunan manual matriks konvolusi (Halupka et al. 2018). Malangnya, kekurangan set latihan, menghadkan kaedah Pembelajaran Dalam. Namun begitu dalam beberapa tahun, pengenalan set latihan yang lebih baik akan meramalkan patologi retina secara berkesan dan mengurangkan kos teknologi OCT (Pekala et al. 2019).

2.2.3 Biometrik

Pada tahun 2009, aplikasi pengecaman pertuturan automatik telah dijalankan untuk mengurangkan Kadar Ralat Telefon (*Phone Error Rate* / PER) dengan menggunakan dua seni bina rangkaian kepercayaan mendalam yang berbeza (Dargan et al. 2020). Sejak tahun 2012, kaedah CNN (Bhatia et al. 2022) telah digunakan dalam rangka Rangkaian Neural Hibrid - Model Markov Tersembunyi (NN – HMM). Hasilnya, PER sebanyak 20.07 % telah dicapai. PER yang diperolehi adalah lebih baik berbanding dengan kaedah garis dasar rangkaian saraf 3 lapisan yang digunakan sebelum ini (Xing et al. 2017). Telefon pintar dan resolusi kamera mereka telah diuji pada pengecaman iris. Dengan menggunakan telefon bimbit yang dibangunkan oleh syarikat yang berbeza, ketepatan pengecaman iris boleh mencapai sehingga 87% keberkesanan (D. Kim et al. 2016) (Malgheet et al. 2021).

Dari segi keselamatan, terutamanya kawalan akses; Pembelajaran Dalam (DL) digunakan bersama dengan ciri biometrik (Păvăloaia & Husac 2022). DL telah digunakan untuk mempercepatkan pembangunan dan pengoptimuman peranti pengenalan wajah FaceSentinel. Menurut pengilang ini, peranti mereka boleh mengembangkan proses pengenalan daripada satu-ke-satu kepada satu-ke-banyak dalam sembilan bulan (Mosavi & Rabczuk 2017). Kemajuan enjin ini boleh mengambil masa 10 tahun tanpa pengenalan DL. Ia mempercepatkan pengeluaran dan pelancaran peralatan. Peranti ini digunakan di lapangan terbang Heathrow di London dan mempunyai potensi untuk digunakan untuk masa dan kehadiran dan dalam sektor perbankan (Guo et al. 2020).

2.2.4 Menyelesaikan Masalah Graf

Mempelajari struktur data graf ialah masalah biasa dengan pelbagai aplikasi berbeza dalam perlombongan data dan tugas pembelajaran mesin. Teknik DL telah menghubungkan antara pembelajaran mesin dan kumpulan perlombongan data. Rangkaian Saraf Konvolusi (CNN) yang cekap untuk pemprosesan graf arbitrari telah dicadangkan pada tahun 2016. (Fu et al. 2021).

2.2.5 Pemprosesan Ucapan

Pemrosesan ucapan ialah kaedah menterjemah isyarat pertuturan kepada satu siri perkataan menggunakan program komputer dan algoritamanya. Matlamat utamanya adalah untuk membolehkan mesin mengecam bunyi dan bertindak mengikutnya. Keupayaan komputer untuk mengenal pasti "menerima dan mentafsir" pertuturan dan menterjemahkannya ke dalam bentuk atau teks yang boleh dibaca dikenali sebagai pengecaman pertuturan automatik (Fadlilah & Djamal 2019). Kaedah CNN juga digunakan untuk pemprosesan ucapan bagi meningkatkan prestasi pengecaman ucapan menggunakan CNN secara multimodal (Ahmed et al. 2021), dan penandaan audio menggunakan Rangkaian Berulang Berpagar Konvolusi (*Convolutional Gated Recurrent Network / CGRN*) (Y. Xu et al. 2017).

2.3 RANGKAIAN SARAF KONVOLUSI (CNN)

Pada tahun 1950-an dan 1960-an, David Hubel dari Amerika dan Torsten Wiesel dari Sweden mula menyelidik sistem visual kucing dan monyet di Sekolah Perubatan Johns Hopkins. Mereka menerbitkan satu siri kertas kerja yang membentangkan teori bahawa neuron dalam korteks visual masing-masing terhad kepada bahagian tertentu medan visual. Mereka selanjutnya menyatakan bahawa pemprosesan visual berlangsung dalam lata, daripada *neuron* khusus kepada bentuk mudah ke arah neuron yang mengambil corak yang lebih kompleks. Penemuan mereka memenangi Hadiah Nobel dalam Fisiologi atau Perubatan 1981. Pada tahun 1980, saintis komputer Jepun Kuniyiko Fukushima mencipta *neocognitron* (Sejnowski 2018), sejenis rangkaian saraf yang terdiri daripada lapisan konvolusi dan lapisan pensampelan bawah, mengambil inspirasi daripada penemuan Hubel dan Wiesel. Neocognitron boleh melakukan beberapa tugas pemprosesan imej asas seperti pengecaman aksara (Chaudhuri et al. 2017).

Dari tahun 1987 hingga 1990, ramai penyelidik termasuk Alex Waibel dan Kouichi Yamaguchi mengadaptasikan lagi neocognitron, memperkenalkan inovasi seperti perambatan belakang yang menjadikannya lebih mudah untuk dilatih. Kemudian pada tahun 1998, Yann LeCun membangunkan LeNet, Rangkaian Saraf Konvolusi (CNN) dengan lima lapisan konvolusi yang mampu mengenali digit poskod tulisan tangan dengan ketepatan yang tinggi (Long & Zeng 2022). Satu lagi peristiwa penting ialah AlexNet Rangkaian Saraf Konvolusi (CNN) Alex Krizhevsky pelajar PhD Ukraine-Kanada, diterbitkan pada 2012 (Srinivas et al. 2017). AlexNet menewaskan semua calon lain dalam pertandingan pengecaman imej ImageNet dengan lebih daripada 10 mata peratusan, menandakan permulaan era baru dalam penglihatan komputer.

Sebelum tahun 2015, sistem pengenalan wajah biasanya dilakukan melalui kod program yang susah payah. Sistem ini dengan mengambil ciri-ciri wajah seperti kening dan hidung. Banyak pengiktirafan aksara optik (OCR) atau aplikasi pengenalan wajah tidak menggunakan pembelajaran mesin sama sekali. Kemajuan pesat dalam kuasa pengkomputeran, dan ketersediaan set data besar, Unit Pemprosesan Grafik (GPU), dan perisian Pembelajaran Dalam menjadikan sistem pengenalan wajah lebih mudah. Rangkaian Saraf Konvolusi (CNN) mampu memberikan ketepatan yang lebih baik daripada kaedah tradisional. Rangkaian Saraf Konvolusi (CNN) tiba-tiba menjadi standard untuk hampir semua tugas berkaitan penglihatan komputer di akademik dan industri. Kini pengguna telefon pintar rata-rata mungkin mempunyai satu atau dua aplikasi yang menjalankan Rangkaian Saraf Konvolusi (CNN) di dalam poket mereka, konsep yang tidak dapat difikirkan pada tahun 2010.

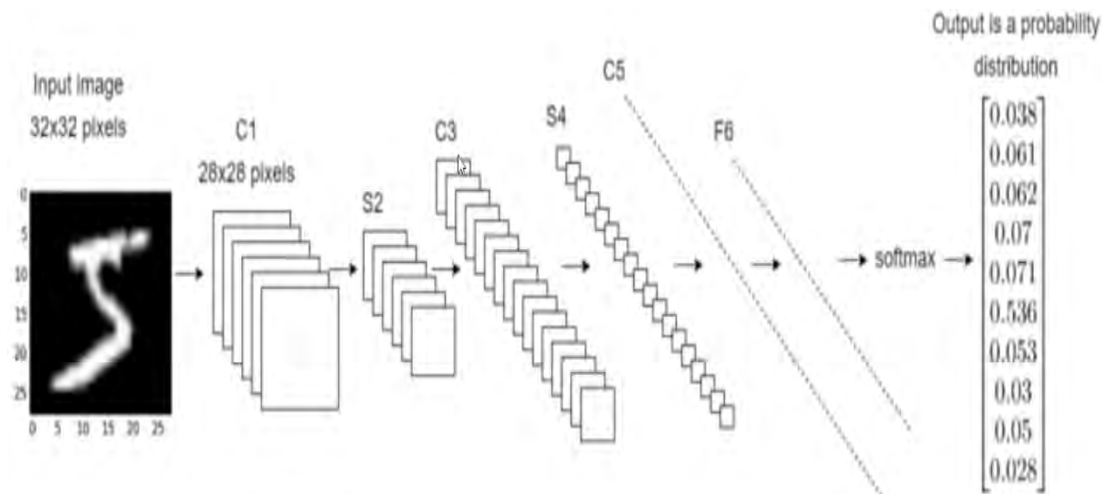
Rangkaian Saraf Konvolusi atau *Convolutional Neural Network* (CNN) yang dikenali sebagai ConvNets, terdiri daripada berbilang lapisan yang digunakan untuk pemprosesan imej dan pengesanan objek. Untuk mengenali aksara seperti kod pos dan nombor, Yann LeCun membangunkan CNN pertama pada tahun 1988 yang dikenali sebagai LeNet (LeCun et al. 1989). CNN ialah rangkaian neural Pembelajaran Dalam yang direka untuk memproses tatasusunan data berstruktur seperti imej. CNN digunakan secara meluas dalam penglihatan komputer dan telah menjadi keadaan seni untuk banyak aplikasi visual seperti klasifikasi imej, dan juga telah menemui kejayaan

dalam pemrosesan bahasa semula jadi untuk pengelasan teks. CNN sangat baik dalam mengambil corak dalam imej input, seperti garisan, kecerunan, bulatan, malah mata dan muka. Sifat inilah yang menjadikan Rangkaian Saraf Konvolusi (CNN) begitu berkuasa untuk penglihatan komputer. Tidak seperti algoritma penglihatan komputer terdahulu, CNN boleh beroperasi secara langsung pada imej mentah dan tidak memerlukan sebarang prapemprosesan.

CNN ialah rangkaian neural suapan ke hadapan, selalunya dengan sehingga 20 atau 30 lapisan. Kuasa Rangkaian Saraf Konvolusi (CNN) berasal dari jenis lapisan khas yang dipanggil lapisan convolutional. CNN mengandungi banyak lapisan konvolusi yang disusun di atas satu sama lain, setiap satu mampu mengenali bentuk yang lebih canggih. Dengan tiga atau empat lapisan konvolusi adalah mungkin untuk mengenali digit tulisan tangan dan dengan 25 lapisan adalah mungkin untuk membezakan wajah manusia. Penggunaan lapisan konvolusi dalam CNN mencerminkan struktur korteks visual manusia, di mana satu siri lapisan memproses imej masuk dan mengenal pasti ciri yang semakin kompleks.

2.3.1 Reka Bentuk Rangkaian Saraf Konvolusi

Seni bina Rangkaian Saraf Konvolusi (CNN) ialah rangkaian neural suapan ke hadapan berbilang lapisan, dibuat dengan menyusun banyak lapisan tersembunyi di atas satu sama lain mengikut urutan. Reka bentuk berjujukan inilah yang membolehkan CNN mempelajari ciri hierarki. Lapisan tersembunyi biasanya lapisan konvolusi diikuti oleh lapisan pengaktifan, sebahagian daripadanya diikuti oleh lapisan pengumpulan. CNN mudah yang membantu pemahaman prinsip reka bentuk teras ialah CNN awal LeNet-5, diterbitkan oleh Yann LeCun pada tahun 1998. LeNet mampu mengenali aksara tulisan tangan.



Rajah 2.3 Lapisan rangkaian saraf konvolusi (CNN)

Sumber: LeCun et al. 1989

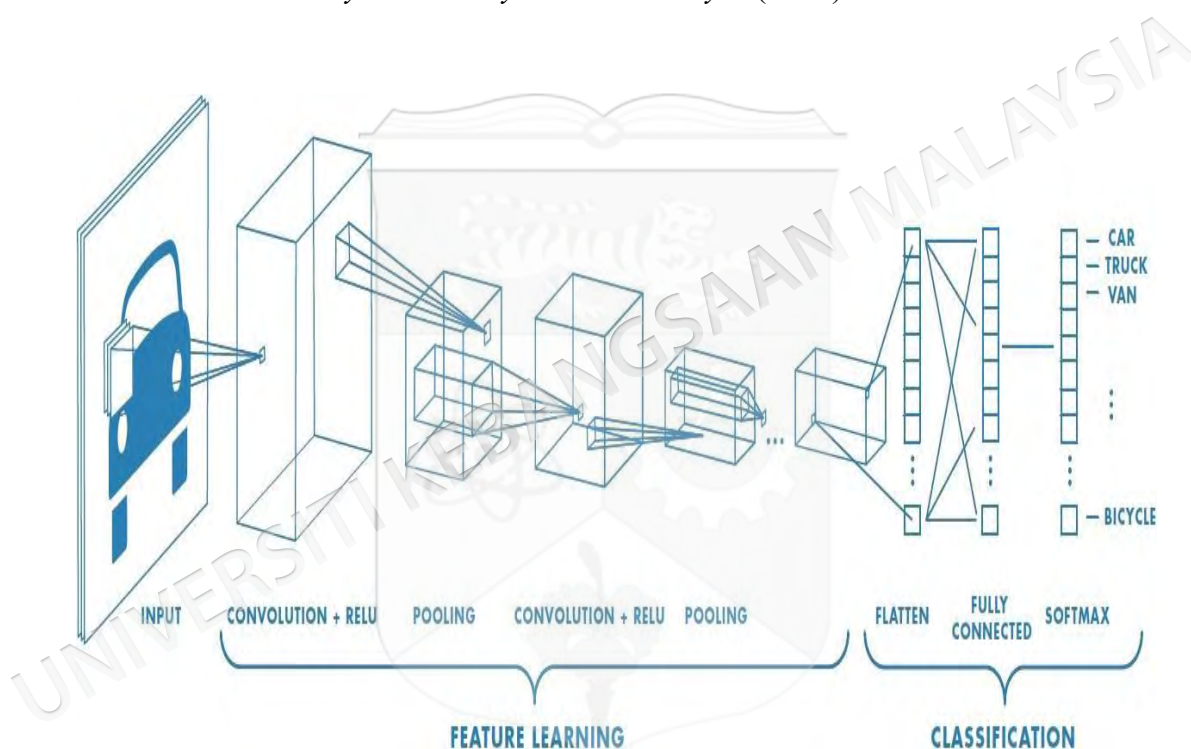
Rajah 2.3 iaitu contoh lapisan CNN, dimana LeNet mengambil imej input digit tulisan tangan bersaiz 32x32 piksel dan menghantarnya melalui timbunan lapisan berikut. Setiap lapisan kecuali yang terakhir diikuti dengan fungsi pengaktifan *tanh*.

- a. C1. Lapisan konvolusi pertama. Ini terdiri daripada enam biji konvolusi bersaiz 5x5, yang 'berjalan di atas' imej input. C1 mengeluarkan enam imej bersaiz 28x28. Lapisan pertama CNN biasanya mengenal pasti ciri asas seperti tepi lurus dan sudut.
- b. S2. Lapisan subsampling, juga dikenali sebagai lapisan pengumpulan purata. Setiap segi empat sama empat piksel dalam output C1 dipuratakan kepada satu piksel. S2 mengecilkan enam imej 28x28 dengan faktor 2, menghasilkan enam imej output bersaiz 14x14.
- c. C3. Lapisan konvolusi kedua. Ini terdiri daripada 16 kernel konvolusi, setiap satu bersaiz 5x5, yang mengambil enam imej 14x14 dan berjalan di atasnya semula, menghasilkan 16 imej bersaiz 10x10.
- d. S4. Lapisan pengumpulan purata kedua. S4 mengecilkan enam belas imej 10x10 kepada enam belas imej 5x5.
- e. C5. Lapisan konvolusi bersambung sepenuhnya dengan 120 output. Setiap satu daripada 120 nod keluaran disambungkan kepada semua 400 nod ($5 \times 5 \times 16$) yang datang dari S4. Pada ketika ini output bukan lagi imej, tetapi tatasusunan 1D dengan panjang 120.

- f. F6. Lapisan bersambung sepenuhnya memetakan tatasusunan 120 kepada tatasusunan baru dengan panjang 10. Setiap elemen tatasusunan ini sepadan dengan digit tulisan tangan 0-9.
- g. Lapisan output. Fungsi softmax yang mengubah output F6 menjadi taburan kebarangkalian 10 nilai yang berjumlah 1.

2.3.2 Lapisan Rangkaian Saraf Konvolusi

Seni bina daripada Rangkaian Saraf Konvolusi (CNN) dibagi menjadi 2 bagian besar, *Feature Extraction Layer* dan *Fully-Connected Layer* (MLP).



Rajah 2.4 Ilustrasi seni bina CNN

Sumber: The Mathworks 2018

Lapisan *Feature Extraction*, proses yang berlaku dalam bahagian ini ialah melakukan “encoding” sesuatu imej kepada ciri dalam bentuk nombor yang mewakili imej tersebut (*Feature Extraction*). Lapisan pengekstrakan ciri terdiri daripada dua bahagian iaitu Lapisan konvolusi dan lapisan penyatuan. Tetapi kadang-kadang terdapat beberapa kajian/kertas kerja yang tidak menggunakan pengumpulan. Lapisan *Fully Connected*, peta ciri yang dijana daripada pengekstrakan ciri masih dalam bentuk tatasusunan berbilang dimensi, jadi ia mesti "meratakan" atau membentuk semula peta

ciri menjadi vektor supaya ia boleh digunakan sebagai input daripada lapisan yang disambungkan sepenuhnya.

Lapisan *fully-connected* ialah lapisan di mana semua neuron aktiviti dari lapisan sebelumnya semuanya disambungkan ke neuron dalam lapisan seterusnya sama seperti yang boleh dilakukan oleh rangkaian saraf tiruan. Setiap aktiviti daripada lapisan sebelumnya perlu ditukar kepada data satu dimensi sebelum ia boleh disambungkan kepada semua neuron dalam lapisan *fully connected*. Lapisan *fully-connected* biasanya digunakan dalam kaedah *multi-layer perceptron* dan bertujuan untuk memproses data supaya ia boleh diklasifikasikan. Perbezaan antara lapisan *fully-connected* dan lapisan lilitan biasa ialah neuron dalam lapisan lilitan hanya disambungkan ke kawasan input tertentu. Manakala lapisan *fully-connected* mempunyai neuron yang bersambung sepenuhnya. Walau bagaimanapun, kedua-dua lapisan masih mengendalikan *dot product*, jadi fungsinya tidak begitu berbeza.

2.3.3 Lapisan Konvolusi

Lapisan konvolusi terdiri daripada neuron yang disusun sedemikian rupa sehingga membentuk penapis dengan panjang dan tinggi (piksel). Sebagai contoh, lapisan pertama dalam lapisan pengekstrakan ciri ialah lapisan konvolusi dengan saiz 5x5x3. Panjang 5 piksel, tinggi 5 piksel, dan tebal 3 keping/ jumlah mengikut saluran imej.

Stride ialah parameter yang menentukan berapa banyak peralihan penapis. Jika nilai langkah ialah 1, maka tapis konvolusi akan beralih sebanyak 1 piksel secara mendatar dan kemudian secara menegak. Semakin kecil langkah, lebih banyak maklumat terperinci yang kita dapat daripada input, tetapi ia memerlukan lebih banyak pengiraan jika dibandingkan dengan langkah yang besar. Walau bagaimanapun, perlu diingat bahawa dengan menggunakan langkah kecil tidak akan sentiasa mendapat prestasi yang baik.

Padding atau *padding* sifar ialah parameter yang menyatakan bilangan piksel (mengandungi nilai 0) untuk ditambahkan pada setiap sisi input. Ia digunakan untuk tujuan memanipulasi dimensi output penukaran lapisan (peta ciri / *feature map*). Dengan menggunakan padding, akan dapat diukur dimensi output untuk

memastikannya sama dengan dimensi input. Jadi kita boleh menggunakan lapisan konvolusi yang lebih dalam supaya lebih banyak ciri boleh diekstrak. Prestasi model yang dipertingkatkan disebabkan oleh penukaran lapisan akan memberi tumpuan kepada maklumat sebenar yang berada di antara padding sifar. Untuk mengira dimensi peta ciri, formula ...(2.1) berikut digunakan:

$$Output = \frac{W+N+2P}{s} + 1 \quad \dots(2.1)$$

Huraian:

W = Panjang/Tinggi input

N = Panjang/Penuras tinggi

P = *Padding* sifar

S = *Stride*/langkah

2.3.4 Fungsi Pengaktifan (ReLU)

Fungsi pengaktifan adalah pada peringkat sebelum melakukan lapisan pengumpulan dan selepas melakukan proses konvolusi. Pada peringkat ini, nilai berbelit-belit tertakluk kepada fungsi pengaktifan. Terdapat beberapa fungsi pengaktifan yang sering digunakan dalam rangkaian konvolusi, termasuk tanh() atau ReLU. Pengaktifan ReLU adalah pilihan bagi sesetengah penyelidik kerana sifatnya yang lebih berfungsi dengan baik.

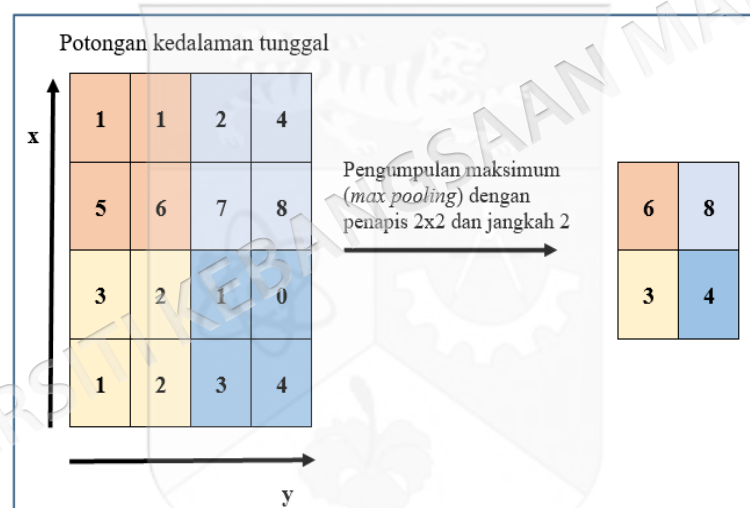
Fungsi yang digunakan untuk pengaktifan dalam ReLU, fungsi ReLU adalah nilai output neuron boleh dinyatakan sebagai 0 jika input negatif. Jika nilai input fungsi pengaktifan adalah positif, maka output neuron adalah nilai input pengaktifan itu sendiri.

2.3.5 Lapisan Pengumpulan (*Pooling Layer*)

Lapisan pengumpulan biasanya datang selepas lapisan konvolusi. Pada dasarnya, lapisan pengumpulan terdiri daripada penapis dengan saiz dan langkah tertentu yang beralih ke seluruh kawasan peta ciri. Pengumpulan yang biasa digunakan ialah Pengumpulan Maksimum (*Max Pooling*) dan Pengumpulan Purata (*Average Pooling*). Tujuan menggunakan lapisan pengumpulan adalah untuk mengurangkan dimensi peta

ciri (pensampelan rendah), dengan itu mempercepatkan pengiraan kerana lebih sedikit parameter perlu dikemas kini dan tidak *overfitting*.

Perkara yang paling penting dalam membina model CNN ialah memilih pelbagai jenis lapisan pengumpulan. Ini boleh memanfaatkan prestasi model (C. Y. Lee et al. 2018). Lapisan pengumpulan berfungsi pada setiap tindanan peta ciri dan mengurangkan saiznya. Bentuk lapisan pengumpulan yang paling biasa ialah menggunakan penapis 2x2 yang digunakan dalam langkah 2 dan kemudian beroperasi pada setiap keping input. Bentuk seperti ini akan mengurangkan peta ciri kepada 75% daripada saiz asalnya. Pengumpulan Maksimum (*Max Pooling*) ditunjukkan dalam Rajah 2.5.



Rajah 2.5 Contoh Pengumpulan Maksimum (*Max Pooling*)

Sumber: Krizhevsky et al. 2017

Lapisan pengumpulan akan beroperasi pada setiap keping kedalaman volumn input secara bergilir-gilir. Dalam imej di atas, lapisan pengumpulan menggunakan salah satu operasi maksimum yang merupakan operasi yang paling biasa. Rajah 2.5. menunjukkan operasi dengan langkah dua dan saiz penapis 2x2. Daripada saiz input 4x4, pada setiap satu daripada empat nombor dalam input operasi mengambil nilai maksimumnya dan membuat saiz output baru 2x2.

2.3.6 Lapisan yang disambungkan sepenuhnya (*Fully Connected Layer* / MLP)

Peta ciri yang dijana daripada pengekstrakan ciri masih dalam bentuk tatasusunan berbilang dimensi, jadi ia mesti "meratakan" atau membentuk semula peta ciri menjadi vektor supaya ia boleh digunakan sebagai input daripada lapisan yang disambungkan sepenuhnya.

Lapisan *fully connected* ialah lapisan di mana semua neuron aktiviti dari lapisan sebelumnya semuanya disambungkan ke neuron dalam lapisan seterusnya sama seperti yang boleh dilakukan oleh rangkaian saraf tiruan. Setiap aktiviti daripada lapisan sebelumnya perlu ditukar kepada data satu dimensi sebelum ia boleh disambungkan kepada semua neuron dalam lapisan yang disambungkan sepenuhnya. Lapisan ini biasanya digunakan dalam kaedah *Multi-layer Perceptron*. Tujuannya adalah untuk memproses data supaya ia boleh dikelaskan. Perbezaan antara lapisan tersambung sepenuhnya dan lapisan lilitan biasa ialah neuron dalam lapisan lilitan hanya disambungkan ke kawasan input tertentu. Manakala lapisan yang disambungkan sepenuhnya mempunyai neuron yang bersambung sepenuhnya. Walau bagaimanapun, kedua-dua lapisan masih mengendalikan *dot product*, jadi fungsinya tidak begitu berbeza.

2.4 BEBERAPA SENI BINA RANGKAIAN SARAF KONVOLUSI (CNN)

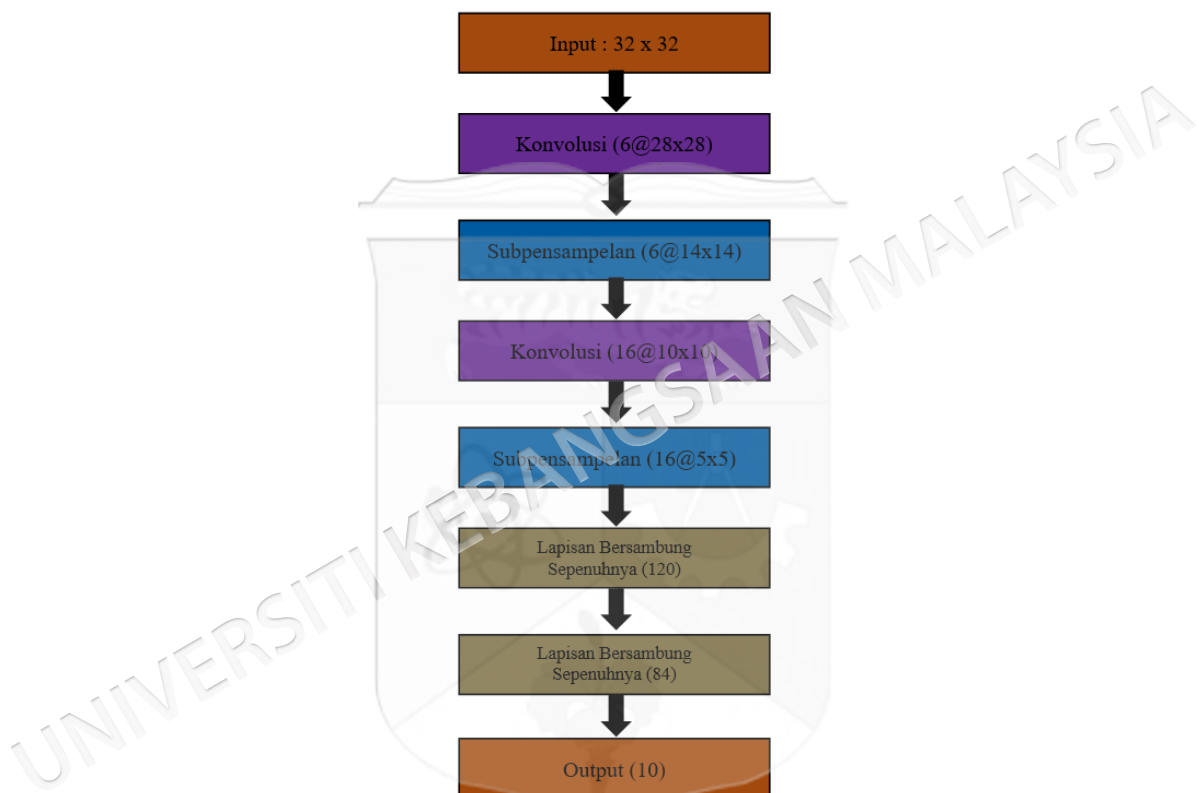
Terdapat banyak seni bina yang ada pada Rangkaian Saraf Konvolusi (CNN), antara lain ; Le-Net, AlexNet, VGG16, GoogleNet, ResNet18, tetapi tidak semuanya dibincangkan disini.

2.4.1 Seni Bina LeNet

Walaupun LeNet dicadangkan pada tahun 1990-an, keupayaan pengiraan terhad dan kapasiti memori menjadikan algoritma sukar untuk dilaksanakan sehingga kira-kira 2010 (Lecun et al. 1998). LeCun, bagaimanapun, mencadangkan CNN dengan algoritma perambatan belakang dan bereksperimen pada set data digit tulisan tangan untuk mencapai ketepatan terkini. Seni bina beliau terkenal sebagai LeNet-5 (Lecun et al. 1998). Konfigurasi asas LeNet-5 ialah (lihat Rajah 2.6): 2 lapisan lilitan (penukaran),

2 lapisan sub-persampelan, 2 lapisan bersambung sepenuhnya dan lapisan keluaran dengan sambungan Gaussian. Jumlah bilangan pemberat dan Darab dan Terkumpul (MAC) masing-masing ialah 431k dan 2.3M.

Apabila perkakasan pengiraan mula bertambah baik dalam keupayaan, CNN menyatakan menjadi popular sebagai pendekatan pembelajaran yang cekap dalam visi komputer dan komuniti pembelajaran mesin.



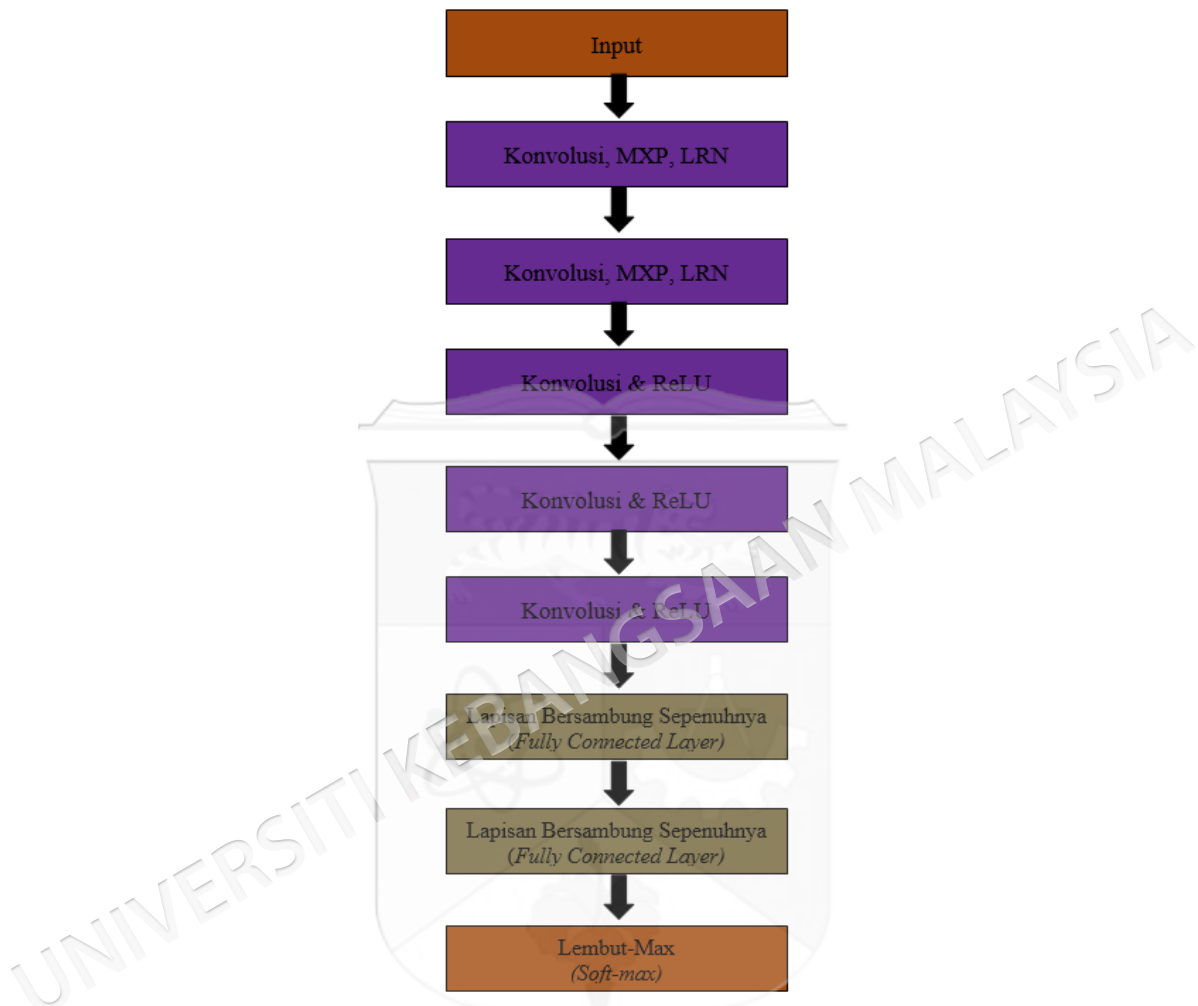
Rajah 2.6 Seni Bina LeNet

Sumber: Lecun et al. 1998

2.4.2 Seni Bina AlexNet

Pada tahun 2012, Alex Krizhevsky dan yang lain mencadangkan model CNN yang lebih mendalam dan lebih luas berbanding LeNet dan memenangi cabaran ImageNet yang paling sukar untuk pengecaman objek visual yang dipanggil *ImageNet Large Scale Visual Recognition Challenge* (ILSVRC) pada tahun 2012 (Krizhevsky et al. 2017). AlexNet mencapai ketepatan pengecaman terkini terhadap semua pembelajaran mesin tradisional dan pendekatan penglihatan komputer. Ia merupakan satu kejayaan

besar dalam bidang pembelajaran mesin dan penglihatan komputer untuk tugas pengecaman dan pengelasan visual dan merupakan titik dalam sejarah di mana minat terhadap Pembelajaran Dalam meningkat dengan pesat.



Rajah 2.7 Seni bina AlexNet: Konvolusi, pengumpulan maksimum, LRN dan lapisan bersambung sepenuhnya (*Fully Connected Layer*)

Sumber: Krizhevsky et al. 2017

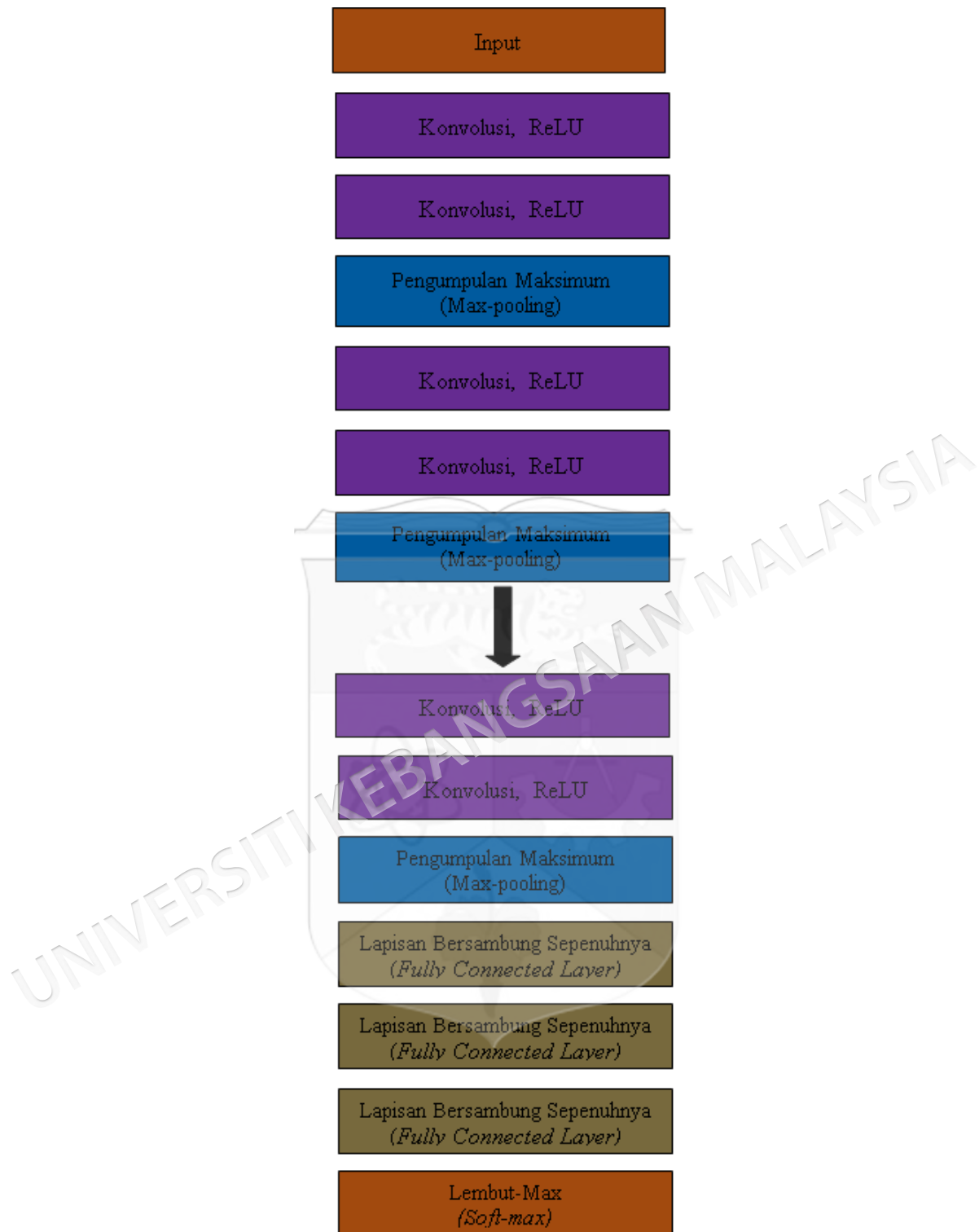
Seni bina AlexNet ditunjukkan dalam Rajah 2.7. Lapisan konvolusi pertama melakukan konvolusi dan pengumpulan maksimum dengan Normalisasi Respons Setempat (LRN) di mana 96 penapis penerimaan berbeza digunakan yang bersaiz 11×11 . Operasi pengumpulan maksimum dilakukan dengan penapis 3×3 dengan saiz langkah 2. Operasi yang sama dilakukan pada lapisan kedua dengan penapis 5×5 . Penapis 3×3 digunakan dalam lapisan konvolusi ketiga, keempat dan kelima dengan peta ciri 384, 384 dan 296 masing-masing. Dua lapisan bersambung sepenuhnya (FC) digunakan dengan kecaciran diikuti dengan lapisan Softmax pada penghujungnya. Dua

rangkaian dengan struktur yang serupa dan bilangan peta ciri yang sama dilatih secara selari untuk model ini. Dua konsep baharu, *Local Response Normalization* (LRN) dan dropout, diperkenalkan dalam rangkaian ini. LRN boleh digunakan dalam dua cara berbeza: mula-mula digunakan pada saluran tunggal atau peta ciri, dengan tampalan $N \times N$ dipilih daripada peta ciri yang sama dan dinormalkan berdasarkan satu nilai kejiranan. Kedua, LRN boleh digunakan merentasi saluran atau peta ciri (kejiranan sepanjang dimensi ketiga tetapi satu piksel atau lokasi).

AlexNet mempunyai 3 lapisan lilitan dan 2 lapisan bersambung sepenuhnya. Apabila memproses set data ImageNet, jumlah bilangan parameter untuk AlexNet boleh dikira seperti berikut untuk lapisan pertama: sampel input ialah $224 \times 224 \times 3$, penapis (kernel atau topeng) atau medan penerimaan yang mempunyai saiz 11, langkah ialah 4, dan keluaran lapisan lilitan pertama ialah $55 \times 55 \times 96$. Mengikut persamaan dalam bahagian 3.1.4, kita boleh mengira bahawa lapisan pertama ini mempunyai 290400 ($55 \times 55 \times 96$) neuron dan 364 ($11 \times 11 \times 3 = 363 + 1$ berat sebelah). Parameter untuk lapisan lilitan pertama ialah $290400 \times 364 = 105,705,600$. Jadual 2.1 menunjukkan bilangan parameter bagi setiap lapisan dalam berjuta-juta. Jumlah bilangan pemberat dan MAC untuk keseluruhan rangkaian ialah 61M dan 724M masing-masing.

2.4.3 Seni Bina VGG

Kumpulan Geometri Visual (VGG), adalah naib juara ILSVRC 2014 (Simonyan & Zisserman 2014). Sumbangan utama kerja ini ialah ia menunjukkan bahawa kedalaman rangkaian adalah komponen kritikal untuk mencapai pengiktirafan atau ketepatan pengelasan yang lebih baik dalam CNN. Seni bina VGG terdiri daripada dua lapisan konvolusi yang kedua-duanya menggunakan fungsi pengaktifan ReLU. Mengikuti fungsi pengaktifan ialah lapisan pengumpulan maksimum tunggal dan beberapa lapisan bersambung sepenuhnya juga menggunakan fungsi pengaktifan ReLU. Lapisan akhir model ialah lapisan Softmax untuk pengelasan. Dalam VGG-E (Simonyan & Zisserman 2014) saiz penapis lilitan ditukar kepada penapis 3×3 dengan langkah 2. Tiga model VGG-E (Simonyan & Zisserman 2014), VGG-11, VGG-16 dan VGG-19; telah dicadangkan model mempunyai 11, 16, dan 19 lapisan masing-masing.



Rajah 2.8 Blok binaan asas rangkaian VGG: Konvolusi (Conv) dan FC untuk lapisan bersambung sepenuhnya

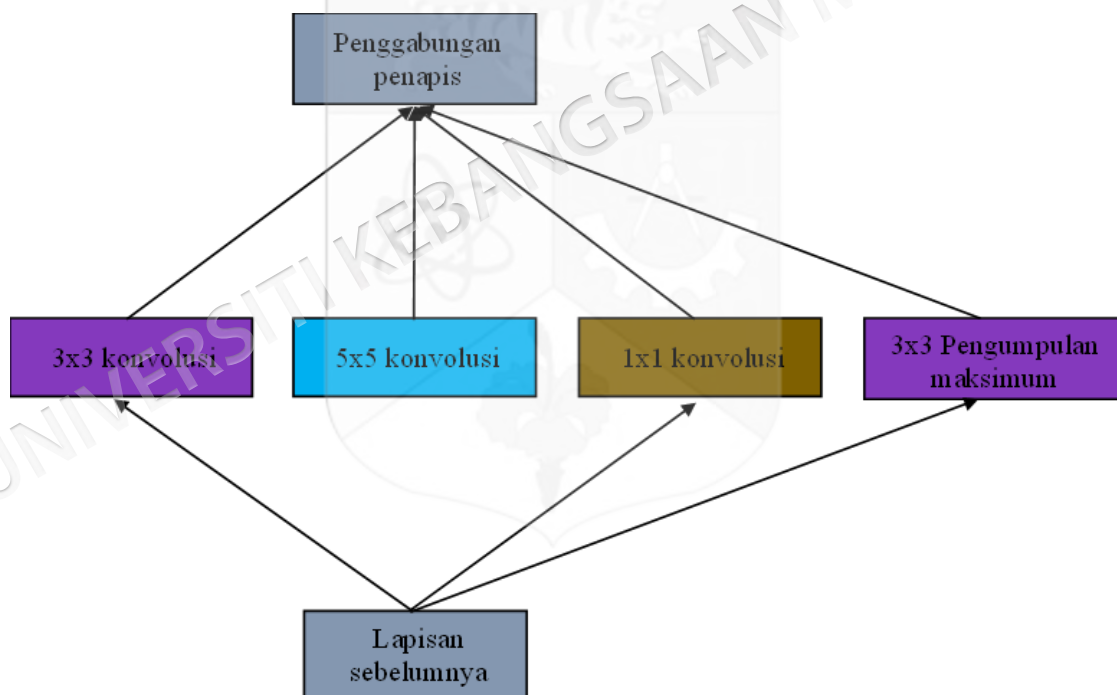
Sumber: Simonyan & Zisserman 2014

Semua versi model VGG-E berakhir sama dengan tiga lapisan bersambung sepenuhnya. Walau bagaimanapun, bilangan lapisan lilitan berbeza-beza VGG-11

mengandungi 8 lapisan lilitan, VGG-16 mempunyai 13 lapisan lilitan, dan VGG-19 mempunyai 16 lapisan lilitan. VGG-19, model paling mahal pengiraan, mengandungi 138M *weights* dan mempunyai 15.5M MACs.

2.4.4 Seni Bina GoogleNet

GoogLeNet, pemenang ILSVRC 2014 (Szegedy et al. 2015), ialah model yang dicadangkan oleh Christian Szegedy dari Google dengan objektif untuk mengurangkan kerumitan pengiraan berbanding CNN tradisional. Kaedah yang dicadangkan adalah untuk menggabungkan "Lapisan Permulaan" yang mempunyai medan penerimaan berubah-ubah, yang dicipta oleh saiz kernel yang berbeza. Medan penerimaan ini mencipta operasi yang menangkap corak korelasi yang jarang dalam timbunan peta ciri baharu.

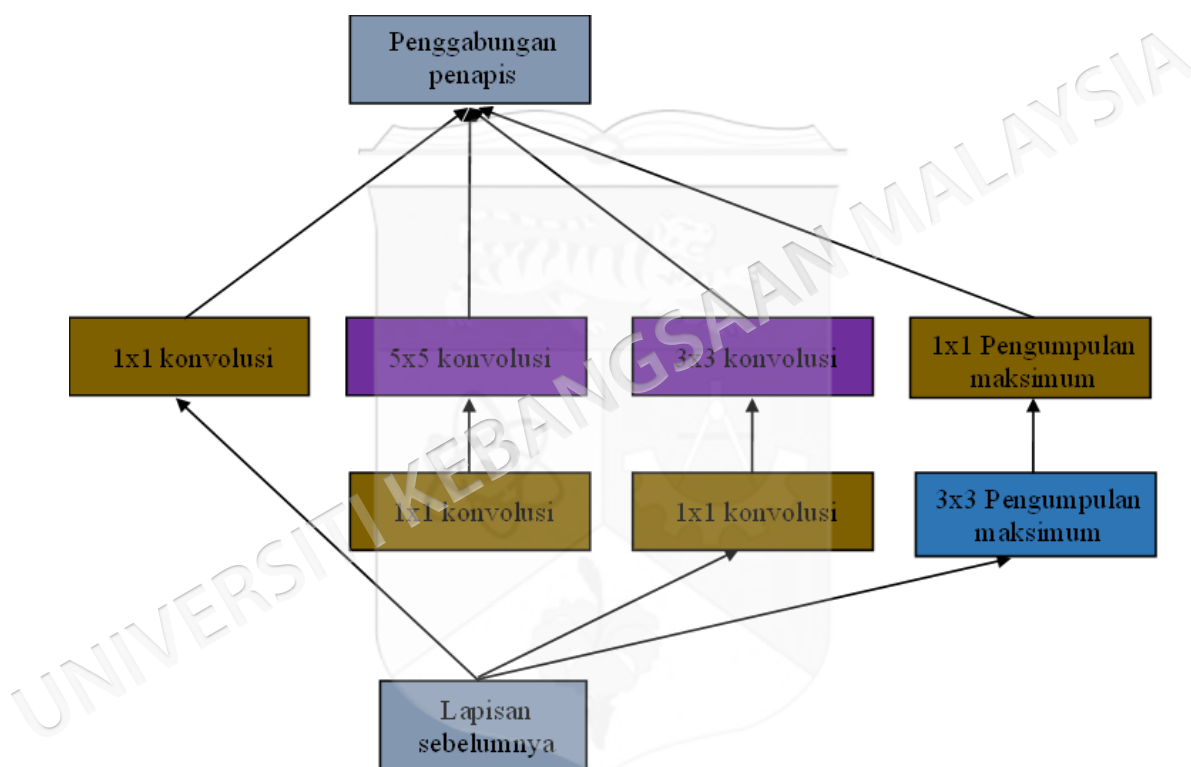


Rajah 2.9 Konsep Lapisan permulaan (*Inception Layer*)

Sumber: Szegedy et al. 2015

Konsep awal Lapisan Permulaan (*Inception Layer*) boleh dilihat dalam Rajah 2.9. GoogLeNet menambah baik keadaan ketepatan pengecaman seni menggunakan timbunan lapisan Inception yang dilihat dalam Rajah 2.10. Perbezaan antara lapisan

permulaan naif dan Lapisan Inception akhir ialah penambahan daripada isirong lilitan 1×1 . Ini ini dibenarkan untuk pengurangan dimensi sebelum lapisan yang mahal secara pengiraan. GoogLeNet terdiri daripada 22 lapisan secara keseluruhan, yang jauh lebih besar daripada mana-mana rangkaian sebelum ini. Walau bagaimanapun, bilangan parameter rangkaian yang digunakan GoogLeNet adalah jauh lebih rendah daripada AlexNet atau VGG sebelumnya. GoogLeNet mempunyai parameter rangkaian 7M apabila AlexNet mempunyai 60M dan VGG-19 138M. Pengiraan untuk GoogLeNet juga adalah 1.53G MAC jauh lebih rendah daripada AlexNet atau VGG.



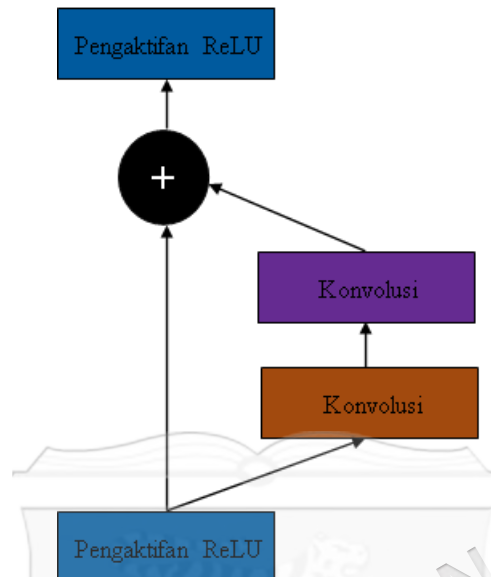
Rajah 2.10 Lapisan permulaan dengan pengurangan dimensi

Sumber: Szegedy et al. 2015

2.4.5 Seni Bina ResNet

Pemenang ILSVRC 2015 ialah seni bina Rangkaian Sisa, ResNet (He et al. 2016). Resnet telah dibangunkan oleh Kaiming He dengan tujuan mereka bentuk rangkaian ultra-dalam yang tidak mengalami masalah kecerunan yang hilang seperti yang dialami oleh pendahulu. ResNet dibangunkan dengan pelbagai bilangan lapisan; 34, 50, 101, 152 dan juga 1202. ResNet50 yang popular mengandungi 49 lapisan lilitan dan 1 lapisan

bersambung sepenuhnya di hujung rangkaian. Jumlah bilangan pemberat dan MAC untuk keseluruhan rangkaian ialah 25.5M dan 3.9M masing-masing.



Rajah 2.11 Gambar rajah asas blok Baki (*Residual Block*)

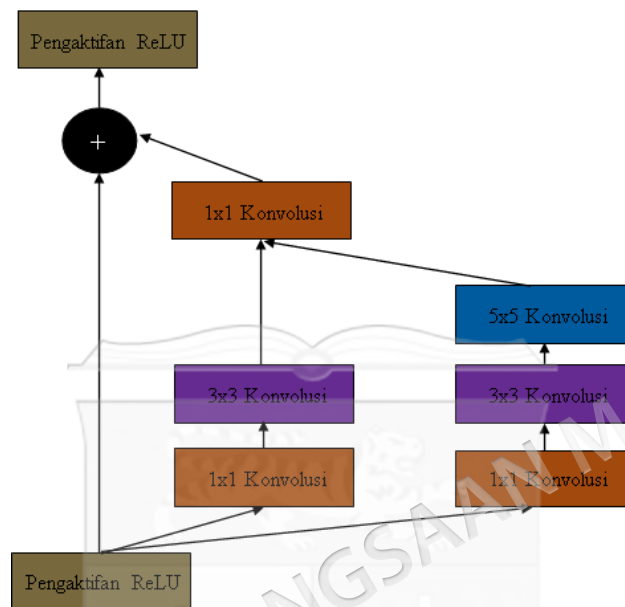
Sumber: He et al. 2016

Gambar rajah blok asas seni bina ResNet ditunjukkan dalam Rajah 2.11. ResNet ialah rangkaian ke hadapan suapan tradisional dengan sambungan baki. Keluaran lapisan baki boleh ditakrifkan berdasarkan keluaran $(l-1)^{th}$ yang datang daripada lapisan sebelumnya yang ditakrifkan sebagai x_{l-1} . $\mathcal{F}(x_{l-1})$ ialah output selepas melakukan pelbagai operasi (cth. lilitan dengan saiz penapis yang berbeza, Penormalan Kelompok (BN) diikuti dengan fungsi pengaktifan seperti ReLU pada x_{l-1}). Keluaran akhir unit baki ialah x_l yang boleh ditakrifkan dengan persamaan ... (2.2).

$$x_l = \mathcal{F}(x_{l-1}) + x_{l-1} \quad \dots(2.2)$$

Rangkaian sisa (*Residual Network*) terdiri daripada beberapa blok baki asas. Walau bagaimanapun, operasi dalam blok sisa boleh diubah bergantung pada seni bina rangkaian sisa yang berbeza (He et al. 2016). Versi rangkaian sisa yang lebih luas telah dicadangkan oleh Zagoruyko et al. pada tahun 2016 (Zagoruyko & Komodakis 2016). Satu lagi pendekatan rangkaian sisa yang lebih baik yang dikenali sebagai transformasi sisa agregat telah dicadangkan pada 2016 (Xie et al. 2017). Baru-baru ini, beberapa varian lain model sisa telah dicadangkan berdasarkan seni bina Rangkaian Residual

(Veit, Andreas and Wilber, Michael J and Belongie 2016) (Abdi & Nahavandi 2016) (X. Zhang et al. 2016). Tambahan pula, terdapat beberapa seni bina termaju yang telah dicadangkan dengan gabungan unit Inception dan Residual. Gambar rajah konsep asas unit Permulaan-Sisa (*Inception-Residual*) ditunjukkan dalam Rajah 2.12 berikut.



Rajah 2.12 Gambar rajah blok asas untuk unit Permulaan-Sisa (*Inception-Residual*)

Sumber: Zagoruyko & Komodakis 2016

Secara matematik, konsep ini boleh diwakili sebagai:

$$x_l = \mathcal{F}(x_{l-1}^{3 \times 3} \odot x_{l-1}^{5 \times 5}) + x_{l-1} \quad \dots(2.3)$$

Pada persamaan ... (2.3), simbol $\odot$ merujuk operasi kepekatan antara dua output daripada penapis 3×3 dan 5×5 . Selepas itu operasi lilitan dilakukan dengan penapis 1×1 . Akhir sekali, output ditambah dengan input blok x_{l-1} ini. Konsep blok Inception dengan sambungan baki diperkenalkan dalam seni bina Inception-v4 (Szegedy et al. 2016). Versi rangkaian Inception-Residual yang lebih baik yang dikenali sebagai PolyNet telah dicadangkan baru-baru ini (Szegedy et al. 2016) (Alom et al. 2020).

2.4.6 Perbandingan pada model yang berbeza

Perbandingan model yang dicadangkan baru-baru ini berdasarkan ralat, parameter rangkaian, dan bilangan sambungan maksimum diberikan dalam Jadual 2.1.

Jadual 2.1 Ralat-5% teratas dengan parameter pengiraan dan Mac untuk model CNN yang berbeza

Kaedah	LeNet-5 (Fukushima 1988)	AlexNet (Krizhevsk y et al. 2017)	OverFeat (fast) (Krizhevsky et al. 2017)	VGG-16 (Simonyan & Zisserman 2014)	GoogLeNet (Szegedy et al. 2015)	ResNet- 50(v1) (He et al. 2016)
Ralat 5 teratas	n/a	16.4	14.2	7.4	6.7	5.3
Saiz input	28x28	227x227	231x231	224x224	224x224	224x224
Bilangan lapisan konvolusi	2	5	5	16	21	50
Saiz penapis	5	3,5,11	3,7	3	1,3,5,7	1,3,7
Bilangan peta ciri	1,6	3-256	3-1024	3-512	3-1024	3-1024
Langkah	1	1,4	1,4	1	1,2	1,2
Bilangan berat	26k	2.3M	16M	14.7M	6.0M	23.5M
Bilangan MAC	1.9M	666M	2.67G	15.3G	1.43G	3.86G
Bilangan lapisan FC	2	3	3	3	1	1
Bilangan berat	406k	58.6M	130M	124M	1M	1M
Bilangan MAC	405k	58.6M	130M	124M	1M	1M
Jumlah berat	431k	61M	146M	138M	7M	25.5M
Jumlah MAC	2.3M	724M	2.8G	15.5G	1.43G	3.9G

2.5 PENYELIDIKAN RANGKAIAN SARAF KONVOLUSI (CNN) TERDAHULU DALAM PENGURUSAN SAMPAH

Jadual 2.2 meringkaskan beberapa aplikasi yang dijalankan sepanjang tahun-tahun sebelumnya mengenai penyelidikan Rangkaian Saraf Konvolusi (CNN) dalam pengurusan sampah.

Jadual 2.2 Aplikasi Rangkaian Saraf Konvolusi (CNN), 2005–2022

No.	Pengarang	Sumbangan	Deskripsi
1	(Fuchikawa et al. 2005)	Mengutip sampah menggunakan <i>Outdoor Service Robot</i> .	Robot hanya mengumpul sampah plastik dalam bentuk botol PET
2	(Salvini et al. 2011)	Mengumpulkan sampah menggunakan <i>Dust Cart</i> robot.	Robot ini mengumpulkan jenis sampah berdasarkan input pengguna. Selepas pengguna memasukkan jenis sampah, robot membuka stor sampah mengikut jenis input sampah

bersambung...

...sambungan

3	(M. Yang & Thung 2016)	Pengelasan sampah menggunakan CNN	Jenis sampah yang boleh diidentifikasi hanya sehingga 6 item sampah sahaja.
4	(Hulyalkar S., Deshpande R., Makode K. 2018)	SmartBin menggunakan CNN.	Jenis sampah yang boleh diidentifikasi hanya sehingga 4 item sampah sahaja, iaitu logam, kaca, kertas, dan plastik.
5	(Salimi et al. 2019)	Pengelasan sampah menggunakan CNN pada <i>Trash Bin Robot</i> .	Selepas robot itu melihat sampah, ia mengeluarkan bunyi dan mengajak orang ramai datang, mengutip sampah yang ditemui oleh robot, dan membuangnya ke dalam tong sampah yang melekat pada robot.
6	(Adedeji & Wang 2019)	Pengelasan sampah menggunakan CNN.	Jenis sampah yang boleh diidentifikasi hanya sehingga 6 item sampah sahaja.
7	(Raza et al. 2021)	Pengelasan sampah menggunakan CNN.	Membangunkan pengesanan sampah masa nyata menggunakan kamera CCTV. Jenis sampah yang boleh diidentifikasi hanya sehingga 8 item sampah sahaja.
8	(Funch et al. 2021)	Pengelasan sampah logam dan kaca menggunakan CNN.	Membangunkan prototaip untuk mengklasifikasikan kehadiran kaca dan logam dalam beg sampah pengguna.
9	(Longo et al. 2021)	Pengelasan sampah menggunakan CNN pada <i>Smart Trash Bin</i> .	Membangunkan prototaip tong sampah pintar, dapat mengelaskan sampah dengan algoritma pengelasan penderia/ imej hibrid, serta mengasingkan bahan buangan yang berbeza secara automatik.
10	(Mao et al. 2021)	Pengelasan sampah menggunakan CNN.	Jenis sampah yang boleh diidentifikasi hanya sehingga 6 item sampah sahaja.
11	(Ren et al. 2021)	Pengelasan sampah pantai menggunakan CNN.	Jenis sampah yang boleh diidentifikasi hanya sehingga 6 item sampah sahaja, iaitu: plastik, kaca, kertas, <i>butt</i> , logam, dan kayu.
12	(Yuan & Liu 2022)	Pengelasan sampah menggunakan CNN.	Jenis sampah yang boleh diidentifikasi hanya sehingga 6 item sampah sahaja iaitu: logam, kertas, plastik, kadbod, kaca, dan sampah.
13	(Faisal et al. 2022)	Pengelasan sampah plastik menggunakan Faster R-CNN	Mengurangkan sampah plastik dan botol di lautan dengan kes untuk pemeliharaan Penyu. bersambung...

...sambungan

14	(Kshirsagar et al. 2022)	Pengelasan sampah menggunakan CNN dan teknik robotik dalam pengasingan bahan buangan.	Membangunkan prototaip Lengan Robot yang boleh menggenggam sampah yang boleh digunakan semula.
15	(M. W. Rahman et al. 2022)	Pengelasan sampah menggunakan CNN	Membangunkan prototaip <i>TrashBin</i> untuk pengelasan sampah. Jenis sampah yang boleh diidentifikasi hanya sehingga 6 item sampah sahaja iaitu: kadbod, kaca, logam, kertas, plastik, sampah.

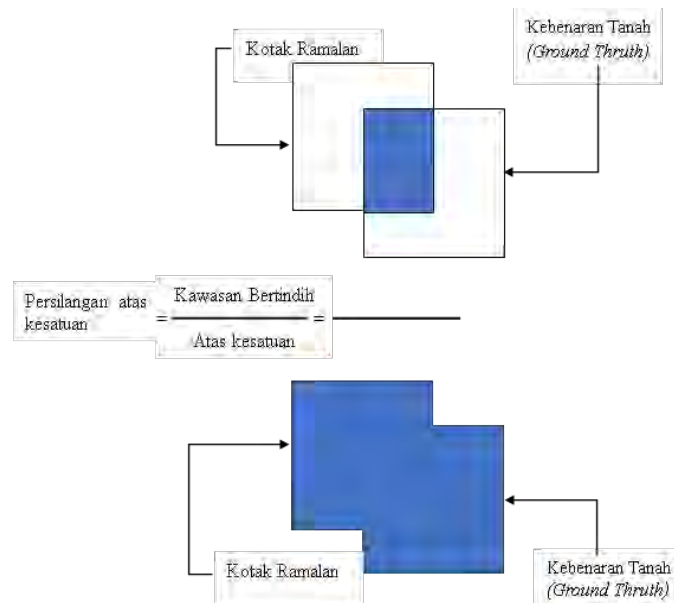
Jadual 2.2 menunjukkan bahawa CNN biasanya digunakan dalam kajian pengelasan sampah. Ini disebabkan oleh kadar kejayaan pengelasan sampah yang agak tinggi. Walau bagaimanapun, terdapat beberapa penyelidikan yang menggabungkan CNN dan penderia untuk mengesan sampah ini. Dalam Jadual 2.2 sesetengah menggunakan penderia ultrasonik dan penderia julat dalam kombinasi dengan CNN, tetapi penggunaannya hanya untuk mengesan kehadiran objek dalam sistem yang direka bentuk. Jadual 2.2 juga mempunyai prototaip lengan robot untuk mengutip sampah. Walau bagaimanapun, robot belum mempunyai penderia LiDAR yang menghasilkan bentuk 2D dan 3D. Jadi jika objek berbentuk 2D, pencengkam robot tidak akan dapat memegang sampah.

2.6 KETEPATAN MENGUKUR MODEL

Mean Average Precision (MAP) ialah metrik untuk menilai model pengesanan objek. MAP mempunyai nilai antara 0 dan 100. Lebih tinggi nilai, lebih baik. MAP boleh dikira dengan cara ketepatan purata (AP) secara berasingan untuk setiap kelas. Nilai AP setiap kelas kemudiannya dipuratakan (Gollapudi 2019).

2.6.1 Persimpangan di atas Kesatuan (IoU)

Intersection over Union (IoU) atau Persimpangan di atas Kesatuan ialah nisbah kawasan bertindih antara *bounding box* yang diramalkan dengan *ground truth bounding box* (Szeliski 2021). Ambang yang biasa digunakan ialah 0.5. Ilustrasi IoU pada Rajah 2.13.



Rajah 2.13 Persimpangan di atas Kesatuan

Sumber: Szeliski 2021

2.6.2 Matriks Kekeliruan (*Confusion Matrix*)

Matriks kekeliruan atau *Confusion Matrix* ialah pengukuran prestasi untuk masalah klasifikasi pembelajaran mesin di mana output boleh menjadi dua atau lebih kelas. Matriks kekeliruan ialah jadual dengan empat kombinasi berbeza nilai ramalan dan nilai sebenar. Terdapat empat istilah yang mewakili hasil proses pengelasan dalam matriks kekeliruan iaitu *True Positive*, *True Negative*, *False Positive*, dan *False Negative*.

True Positive (TP): apabila nilai sebenar adalah Positif dan diramalkan juga Positif.

True Negative (TN): apabila nilai sebenar adalah Negatif dan ramalan juga Negatif.

False Positive (FP): Apabila yang sebenar adalah negatif tetapi ramalan adalah Positif. Juga dikenali sebagai ralat Jenis 1.

False Negative (FN): Apabila yang sebenar adalah Positif tetapi ramalannya Negatif. Juga dikenali sebagai ralat Jenis 2.

		Kelas Ramalan	
		Negatif 0	Positif 1
Kelas Sebenar	Negatif 0	TN	FP
	Positif 1	FN	TP

Rajah 2.14 Matriks kekeliruan

Sumber: J. Xu et al. 2020

Dalam Rajah 2.14 yang ditunjukkan, nilai positif sebenar ditakrifkan sebagai tuple positif yang dikelaskan dengan betul oleh model. Negatif sebenar ialah tuple negatif yang dikelaskan dengan betul oleh model. Sementara itu, positif palsu ialah tuple negatif yang diklasifikasikan sebagai positif oleh model. Negatif palsu ialah tuple positif yang diklasifikasikan sebagai kelas negatif oleh model. Berdasarkan matriks kekeliruan dalam rajah Rajah 2.14, prestasi model pengelasan boleh dikira.

2.6.3 Ketepatan

Ketepatan atau *accuracy* hanya mengukur kekerapan pengelasan membuat ramalan yang betul. Ia adalah nisbah antara bilangan ramalan yang betul dan jumlah bilangan ramalan. Metrik ketepatan tidak sesuai untuk kelas yang tidak seimbang. Ketepatan mempunyai kelemahannya sendiri, untuk data yang tidak seimbang, apabila model meramalkan bahawa setiap titik tergolong dalam label kelas majoriti, ketepatannya akan menjadi tinggi. Tetapi, model itu tidak tepat. Ia adalah ukuran ketepatan yang dicapai dalam ramalan sebenar. Dalam perkataan mudah, ia memberitahu kita berapa banyak ramalan yang sebenarnya positif daripada semua jumlah positif yang diramalkan. Metrik ketepatan ditunjukkan dalam Rajah 2.15.

		NILAI RAMALAN	
		Positive (1)	Negative (0)
NILAI SEBENAR	Positive (1)	6 TRUE POSITIVE	1 FALSE NEGATIVE
	Negative (0)	2 FALSE POSITIVE	11 TRUE NEGATIVE

Rajah 2.15 Metrik ketepatan

Sumber: Anderson 2018

Ketepatan ialah pilihan penilaian yang sah untuk masalah klasifikasi yang seimbang dan tidak condong atau tiada ketidakseimbangan kelas. Ketepatan ditakrifkan sebagai peratusan data ujian yang dikelaskan kepada kelas yang betul. Ketepatan boleh dinyatakan dalam persamaan...(2.4).

$$\text{Ketepatan} = \frac{TP+TN}{TP+TN+FP+FN} = \frac{6+11}{6+11+2+1} = 85\% \quad \dots(2.4)$$

2.6.4 Kebersihan

Kebersihan atau *precision* adalah ukuran kebersihan yang dicapai dalam ramalan sebenar. Dalam perkataan mudah, ia memberitahu kita berapa banyak ramalan yang sebenarnya positif daripada semua jumlah positif yang diramalkan. Kebersihan ialah metrik yang berguna dalam kes di mana *False Positive* (FP) adalah kebimbangan yang lebih tinggi daripada *False Negative* (FN). Metrik kebersihan ditunjukkan dalam Rajah 2.16.

		NILAI RAMALAN	
		Positive (1)	Negative (0)
NILAI SEBENAR	Positive (1)	6 TRUE POSITIVE	1 FALSE NEGATIVE
	Negative (0)	2 FALSE POSITIVE	11 TRUE NEGATIVE

Rajah 2.16 Metrik kepersisan

Sumber: Anderson 2018

Kepersisan ditakrifkan sebagai nisbah jumlah bilangan kelas positif yang dikelaskan dengan betul dibahagikan dengan jumlah bilangan kelas positif yang diramalkan. Atau, daripada semua kelas positif ramalan, jumlah yang diramalkan dengan betul. Ketepatan harus tinggi (sebaik-baiknya 1). Kepersisan boleh dinyatakan dalam persamaan...(2.5).

$$\text{Kepersisan} = \frac{TP}{TP + FP} = \frac{6}{6 + 2} = 0.75 \quad \dots(2.5)$$

2.6.5 Ingat Kembali (*Recall*)

Ingat kembali atau *recall* adalah ukuran pemerhatian sebenar yang diramalkan dengan betul, iaitu berapa banyak pemerhatian kelas positif sebenarnya diramalkan sebagai positif. Ia juga dikenali sebagai sensitiviti. *Recall* ialah pilihan metrik penilaian yang sah apabila kita ingin menangkap sebanyak mungkin positif. Metrik Ingat Kembali (*Recall*) ditunjukkan dalam Rajah 2.17.

		NILAI RAMALAN	
		Positive (1)	Negative (0)
NILAI SEBENAR	Positive (1)	6 TRUE POSITIVE	1 FALSE NEGATIVE
	Negative (0)	2 FALSE POSITIVE	11 TRUE NEGATIVE

Rajah 2.17 Metrik perolehan kembali (*recall*)

Sumber: Anderson 2018

Recall ditakrifkan sebagai nisbah jumlah bilangan kelas positif yang dikelaskan dengan betul dibahagikan dengan jumlah bilangan kelas positif. Atau, daripada semua kelas positif, berapa banyak yang telah diramalkan dengan betul. *Recall* harus tinggi (sebaik-baiknya 1). *Recall* ialah metrik yang berguna dalam kes di mana FN mengatasi FP. Perolehan kembali boleh dinyatakan dalam persamaan...(2.6).

$$\text{Perolehan kembali} = \frac{TP}{TP + FN} = \frac{6}{6 + 1} = 0.85 \quad \dots(2.6)$$

2.6.6 Ukuran F / Skor F1

Skor F1 ialah nombor antara 0 dan 1 dan merupakan min harmonik bagi *precision* dan *recall*. Kajian ini menggunakan min harmonik kerana ia tidak sensitif kepada nilai yang sangat besar, tidak seperti purata mudah. Skor F1 semacam mengekalkan keseimbangan antara *precision* dan *recall* untuk pengelas anda. Jika *precision* rendah, F1 adalah rendah dan jika *recall* rendah sekali lagi markah F1 anda rendah. Akan ada kes di mana tiada perbezaan yang jelas antara sama ada *precision* lebih penting atau *recall*. Dalam amalan, apabila dicuba meningkatkan ketepatan model, penarikan balik akan menurun dan begitu juga sebaliknya. Skor F1 menangkap kedua-dua arah aliran dalam satu nilai (Anderson 2018).

$$Skor F1 = 2 * \frac{(Recall * Precision)}{(Recall + Precision)} = 2 * \frac{(0.85 * 0.75)}{(0.82 + 0.75)} = 0.79 \quad \dots(2.7)$$

Skor F1 ialah min harmonik bagi *precision* dan *recall*. Berbanding dengan Min Aritmetik, Min Harmonik mempunyai nilai ekstrem. Skor-F hendaklah tinggi (sebaik-baiknya 1).

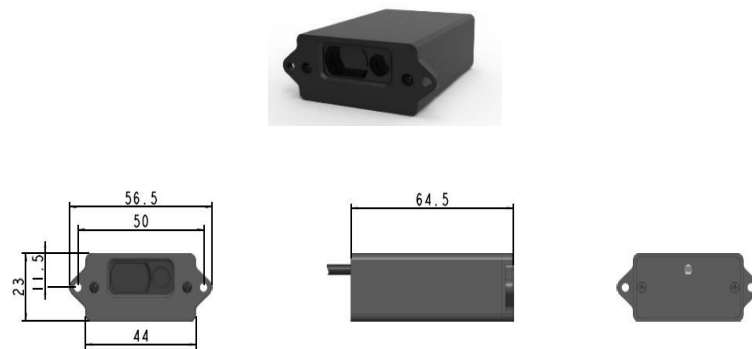
2.7 PENDERIA LIDAR (*LIGHT DETECTION AND RANGING*)

Penderia LiDAR ialah penderia yang digunakan secara meluas dalam reka bentuk unit perkhidmatan (Rosique et al. 2019). TF40 ialah penderia LiDAR yang digunakan dalam penyelidikan ini. TF40 mempunyai ketepatan peringkat milimeter dengan julat sehingga 40 m (Benewake 2022) (Iordan, Daniela and Popescu 2015) (W. Yang et al. 2022).

Ketepatan penderia LiDAR dalam mm adalah penting kerana objek sampah agak kecil, 15 cm × 20 cm. Saiz dilaraskan kepada lebar pencengkam robot. TF40 mempunyai ciri-ciri berikut: ketepatan tinggi, kecil, titik cahaya kecil, laser boleh dilihat dan sasaran yang lebih mudah. Jadual 2.3 ialah parameter utama TF40, dan Rajah 2.18 menunjukkan bentuk fizikal penderia TF40 LiDAR dan dimensinya.

Jadual 2.3 Parameter utama TF40

No.	Nama Parameter	Versi standard
1	Prestasi produk	Julat
		0.04 – 40 m 90% daya pemantulan, 0.04 – 20 m 10% daya pemantulan
		Ketepatan
		± 2 mm
2	Parameter optik	Resolusi jarak
		1 mm
		Kadar bingkai
		5 Hz
		Sumber cahaya
		LD
3	Parameter elektrik	Panjang gelombang
		635 nm
		Kelas laser
		KELAS 2 (EN 60825)
		Sudut pengesanan
		< 1 mrad
3	Parameter elektrik	Voltan bekalan
		3.3 V
		Arus purata
		≤ 180 mA
		Penggunaan kuasa
		≤ 0.6 W
		Tahap voltan Komunikasi
		LVTTL (3.3 V)



Rajah 2.18 Bentuk fizikal penderia TF40 LiDAR dan dimensinya

Sumber: Benewake 2022

2.8 KERJA SEBELUMNYA MENGGUNAKAN LIDAR DALAM PENGESANAN OBJEK

Penyelidikan yang dicadangkan mempunyai persamaan dengan gabungan penderia. Dalam gabungan penderia, berbilang penderia digabungkan untuk melakukan pengukuran. Matlamatnya ialah data yang terhasil adalah lebih tepat daripada hanya menggunakan satu penderia. Dalam kajian ini, kamera dan penderia LiDAR telah digunakan. Banyak kajian menggunakan gabungan penderia dan kamera LiDAR. Jadual 2.4 menunjukkan perbandingan penggunaannya (Khan et al. 2022).

Jadual 2.4 Kerja Sebelumnya Menggunakan LiDAR Dalam Pengesanan Objek

Rujukan	Penderia	Kajian yang Dicadangkan	Kaedah
(Qiu et al. 2019)	Kamera, RGB, LiDAR	Gabungan Awal	
(Cheng et al. 2019)	Kamera, RGB, LiDAR		
(Van Gansbeke et al. 2019)	Kamera, RGB, LiDAR		Data daripada imej kamera RGB dan penderia LiDAR adalah input secara langsung untuk Pembelajaran Dalam dan diproses bersama. Hasilnya adalah kedalaman yang lengkap.
(Hu et al. 2021)	Kamera, RGB, LiDAR		
(Yan et al. 2022)	Kamera, RGB, LiDAR		
(Nazir et al. 2022)	Kamera, RGB, LiDAR		
(L. Liu et al. 2020)	Kamera, RGB, LiDAR		

bersambung....

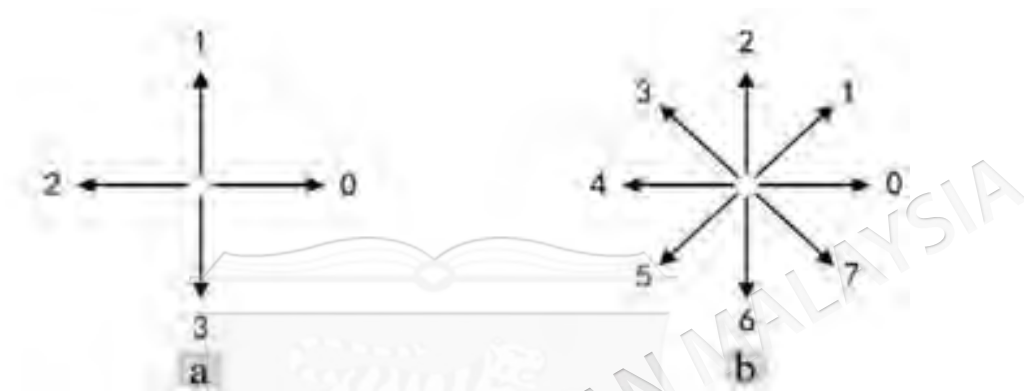
sambungan...

(Hu et al. 2021)	Kamera, RGB, LiDAR	Gabungan Berjujukan	Data daripada imej kamera RGB mula-mula diproses oleh Pembelajaran Dalam. Hasilnya ialah kedalaman RGB. Kemudian, keputusan ini dan data penderia LiDAR secara langsung menjadi input untuk Pembelajaran Dalam seterusnya. Kedua-duanya diproses bersama, dan hasilnya adalah kedalaman yang lengkap
(Nazir et al. 2022)	Kamera, RGB, LiDAR		
(Eldesokey et al. 2020)	Kamera, RGB, LiDAR		
(Hu et al. 2021)	Kamera, RGB, LiDAR	Gabungan Lewat	Data daripada imej kamera RGB mula-mula diproses oleh Pembelajaran Dalam. Hasilnya ialah kedalaman RGB. Selain itu, data penderia LiDAR juga diproses melalui Pembelajaran Dalam. Hasilnya ialah kedalaman LiDAR. Tambahan pula, output masing-masing diproses bersama, menghasilkan kedalaman yang lengkap
(Nazir et al. 2022)	Kamera, RGB, LiDAR		
(Jaritz et al. 2018)	Kamera, RGB, LiDAR		

Dalam jadual 2.4, dapat dilihat bahawa semua kajian menggunakan penderia gabungan yang sama, iaitu : kamera, RGB dan LiDAR. Perbezaannya terletak pada kajian yang dicadangkan. Terdapat 3 kaedah, iaitu Gabungan Awal, Gabungan Berjujukan, dan Gabungan Lewat. Semua kajian di atas memperoleh output yang sama iaitu pengesanan kedalaman yang baik dan lengkap. Walau bagaimanapun, kerana masa penderiaan menggunakan LiDAR adalah lebih lama, maka kajian ini mencadangkan penggunaan 2 pendekatan penderiaan, iaitu Pengimbasan Pantas dan Pengimbasan Terperinci. Sebelum pengimbasan menggunakan penderia LiDAR dilakukan, data imej dari kamera RGB diproses menggunakan Pembelajaran Dalam untuk mengesan kehadiran objek sampah. Jika objek itu merupakan sampah, maka Pengimbasan Pantas dilakukan untuk memastikan bahawa objek dalam bentuk 3D. Jika objek adalah 3D, maka Pengimbasan Terperinci dilakukan untuk menentukan kedudukan yang betul dalam mengangkat objek sampah menggunakan penggenggam robot. Dengan menggunakan kaedah ini, maka diharapkan proses identifikasi sampah lebih cepat dilakukan dan robot memang mengangkat objek dalam bentuk 3D.

2.9 KOD RANTAI FREEMAN (*FREEMAN CHAIN CODE*)

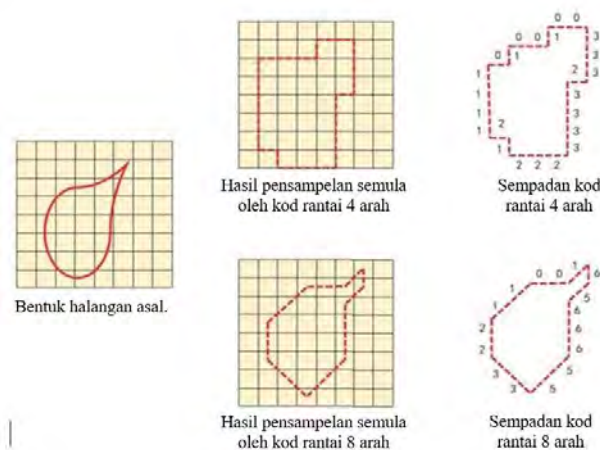
Kod rantai arah Freeman juga dikenali sebagai kod rantai. Kod ini digunakan untuk mewakili sempadan dengan jujukan segmen garis lurus yang disambungkan dengan panjang dan arah tertentu. Biasanya, perwakilan ini adalah berdasarkan 4- atau 8-kesambungan segmen, seperti yang ditunjukkan Rajah 2.19.



Rajah 2.19 Nombor arah untuk (a) kod rantai 4 arah, dan (b) kod rantai 8 arah

Sumber: M. Zhao & Lu 2019

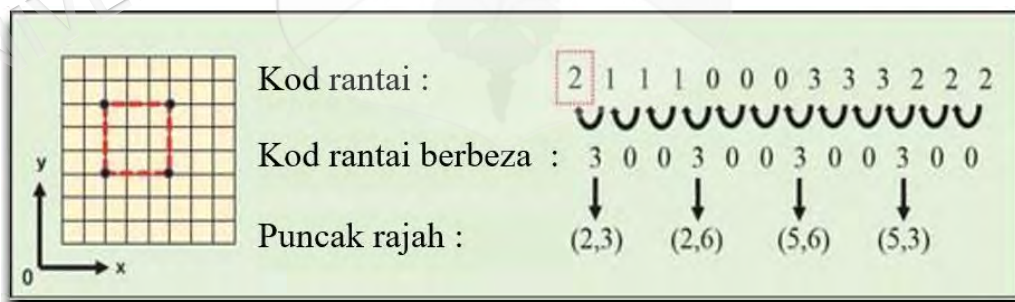
Oleh kerana imej digital biasanya diperolehi dan diproses dalam format grid dengan jarak yang sama, kod rantai arah Freeman boleh dijana dengan mudah dengan mengikut sempadan mengikut arah jam dan memberikan arah kepada segmen yang menghubungkan setiap pasangan piksel (Onishi & Ono 2011). Kod rantai arah Freeman digunakan secara meluas dalam pemprosesan imej digital. Untuk mengelakkan kod rantai terlalu panjang dan mengambil kira ciri piksel imej, kaedah biasa digunakan dalam pengesanan sempadan imej ialah pensampelan semula sempadan dengan memilih jarak grid yang lebih besar, seperti ditunjukkan dalam Rajah 2.20.



Rajah 2.20 Proses pensampelan semula sempadan halangan

Sumber: M. Zhao & Lu 2019

Di samping itu, disebabkan oleh keberkataan kod rantai, puncak grafik boleh diperoleh dengan memperkenalkan kod rantaian pembezaan. Kod ini membantu robot menentukan titik matlamat sementara, jika robot hendak mengelakkan halangan. Kod rantaian pembezaan diperoleh dengan mengira bilangan perubahan arah dalam arah lawan jam dalam Rajah 2.20, yang memisahkan dua elemen bersebelahan kod. Sebagai contoh, kod rantaian pembezaan kod rantai 4 arah 111000333222 ialah 300300300300, titik koordinat permulaan yang sepadan dengan nilai bukan sifar bagi jujukan kod ini ialah bucu graf, dan proses khusus adalah seperti Rajah 2.21 berikut.



Rajah 2.21 Proses pengiraan bucu graf dengan kod rantai

Sumber: M. Zhao & Lu 2019

2.10 ROBOT BERAUTONOMI

Robot berautonomi merupakan robot yang dalam pengoperasiannya tidak ada campur tangan manusia (Hui & Pratihari 2010). Robot mudah alih boleh ditakrifkan sebagai sistem yang boleh bergerak dalam ruang tertentu tanpa terikat dengan persekitaran.

Mereka boleh dibahagikan kepada beberapa kumpulan utama, berkaitan dengan persekitaran tempat mereka bergerak (Klančar et al. 2017).

1. Robot mudah alih darat

Terdapat pelbagai jenis platform mudah alih di sini, antaranya : kenderaan mudah alih dengan roda atau ulat, robot berkaki (humanoid atau haiwan mimetik), atau robot yang meniru pergerakan tertentu haiwan lain seperti ular. Robot mudah alih darat beroda yang tidak membawa pengendali sering dirujuk sebagai kenderaan darat tanpa pemandu (Mo et al. 2020). Robot ini biasanya digunakan untuk kawasan rata. Untuk kawasan yang tidak rata atau kawasan yang kasar, robot beroda tidak sesuai digunakan. (Nagatani et al. 2016) Dengan menggunakan mekanik rantai tangki boleh membantu robot yang khusus dalam bergerak di lapangan. Untuk sistem kemudi, kemudi selip (*skid steering*) digunakan, yang merupakan sistem kemudi kenderaan rantai. Sistem kemudi ini mempunyai prinsip yang sama dengan Kemudi Pembezaan (*Differential Steering*). Kelajuan putaran rantai yang berbeza antara kedua-dua sisi rantai digunakan untuk menyesuaikan kemudi.

2. Robot mudah alih udara

Kumpulan ini menerbangkan udara tertentu di angkasa (kapal terbang, helikopter, dron, roket, sistem penerbangan yang meniru haiwan; apabila digunakan tanpa juruterbang, mereka dipanggil kenderaan udara tanpa pemandu) atau apa-apa yang mengelilingi Bumi atau badan angkasa. Ia terdiri daripada sistem mudah alih satelit (González-Jorge et al. 2017).

3. Robot mudah alih air dan bawah air

Kumpulan ini termasuk pelbagai jenis kapal, bot, kapal selam dan kenderaan bawah air berautonomi. (Zeng et al. 2015)

Pada hakikatnya ini bermakna sistem mudah alih menerima arahan daripada pengendali manusia berdasarkan tahap autonomi yang mampu dilakukan oleh sistem. Sistem bertindak balas dengan cuba melaksanakan tugas yang dipesan dan subtugas

yang sesuai pada tahap yang lebih rendah. Tugasannya biasanya dilaksanakan pada selang masa tetap tanpa sebarang kontingensi. (Rubio et al. 2019).

Berdasarkan tahap autonomi robot, arahan am berikut boleh diperolehi daripada operator (Klančar et al. 2017) :

- Kelajuan roda yang dikehendaki
Robot menerima arahan tentang kelajuan roda dan algoritma kawalan yang sangat mudah yang menggunakan penderia yang sesuai (biasanya pengekod berputar) untuk memutar roda seperti yang diarahkan.
- Kelajuan robot membujur dan sudut yang dikehendaki
Program komputer yang berjalan pada robot mengetahui model kinematik robot dan boleh mengira kelajuan roda yang sesuai untuk mencapai kelajuan robot yang dikehendaki.
- Laluan robot atau trajektori robot yang dikehendaki
Robot boleh menentukan dan mengawal posturnya dalam persekitarannya. Pose biasanya ditakrifkan sebagai maklumat yang dikongsi tentang kedudukan dan orientasi berkenaan dengan sistem koordinat. Dalam tahap ini, anda akan menemui penyetempatan robot yang memperoleh anggaran terbaik pose robot menggunakan pelbagai penderia yang dipasang pada robot atau persekitaran. Kawalan juga terjejas disebabkan oleh ciri bukan linear sistem, maklumat penderia yang salah, gelinciran roda, model buruk, kelewatan, dan lain-lain sebagainya.
- Operasi yang dikehendaki di dalam persekitaran yang diketahui dengan potensi halangan
Robot mesti melakukan operasi dalam persekitaran yang diketahui, tetapi dengan beberapa halangan (statik atau mudah alih). Pada tahap ini, robot boleh merancang semula laluan jika halangan menghalangnya daripada melengkapkan manuver.

- Operasi yang dikehendaki di dalam persekitaran yang tidak diketahui
Robot itu tidak mengetahui persekitarannya. Robot perlu menentukan kedudukannya dan memetakan persekitarannya pada masa yang sama. Pendekatan ini dikenali sebagai *Simultaneous Localization and Mapping* (SLAM).
- Misi yang dikendaki
Robot akan memulakan misi yang mesti diselesaikan. Ia berfungsi dalam persekitaran di mana ia boleh bekerjasama dengan robot dan ejen lain. Robot mesti mempunyai sedikit pemahaman tentang misinya. Untuk membolehkan robot membatalkan misi tertentu atau melaksanakan misi keutamaan yang lebih tinggi, robot mesti mempunyai keutamaan khusus untuk misi tersebut. Banyak keputusan perlu dibina ke dalam robot. Contoh mudah ialah apabila robot perlu memeriksa tahap tenaga baterinya dan mengecas semula bateri jika perlu.

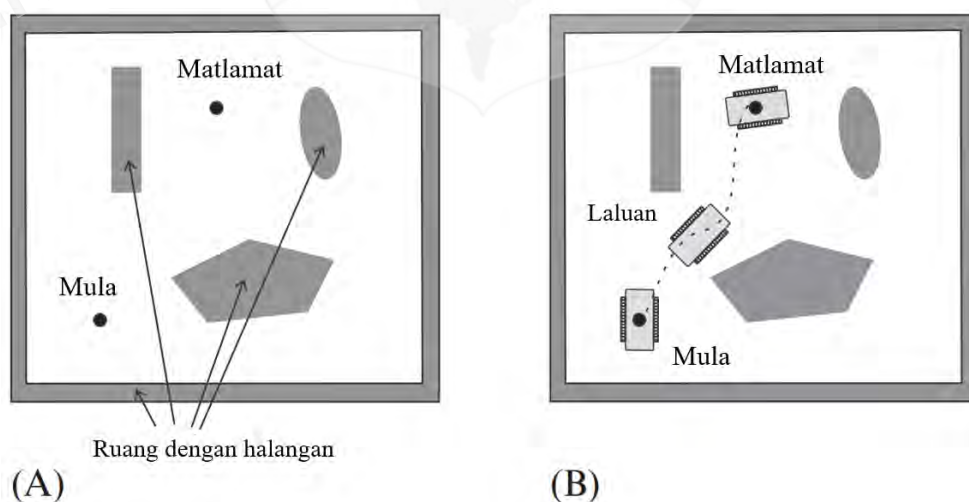
2.11 NAVIGASI ROBOT

Navigasi robot sangat penting dalam sistem robotik. Matlamat utama navigasi adalah untuk robot bergerak dari kedudukan permulaan ke kedudukan matlamat dengan mengikut laluan yang selamat dan menjana panjang laluan yang optimum. Perancangan laluan (*path planning*) ialah salah satu bahagian terpenting dalam navigasi robot mudah alih berautonomi. Sepanjang dua dekad yang lalu, penyelidik telah menangani masalah perancangan laluan, dan beberapa kaedah telah dibangunkan untuk berbuat demikian. Perancangan laluan melibatkan penentuan laluan bebas perlanggaran dari satu titik ke titik lain sambil meminimumkan kos keseluruhan laluan yang berkaitan. Bergantung pada jenis persekitaran, perancangan laluan boleh dibahagikan kepada persekitaran statik dan dinamik. Jika halangan berubah kedudukan dari semasa ke semasa, ini dipanggil perancangan laluan statik, dan jika kedudukan dan orientasi halangan berubah, ini dipanggil perancangan laluan dinamik. (Gul et al. 2019).

Navigasi boleh dibahagikan kepada dua jenis: Navigasi *Global* dan *Local* (Alajlan et al. 2013) (Sedighi et al. n.d.). Untuk jenis navigasi global, pengetahuan awal tentang persekitaran harus diketahui, yang juga dipanggil mod Luar Talian untuk perancangan laluan (*path planning*), Contohnya: Algoritma Dijkstra (Soltani et al. 2002), algoritma A* (Xiang Liu & Daoxiong Gong 2011), Graf kebolehlihatan

(Masehian & Amin-Naseri 2004), Kaedah medan potensi buatan (Gomez et al. 2013), dan kaedah penguraian sel (Park et al. 2001). Untuk navigasi *Local*, juga dikenali sebagai mod Dalam Talian untuk perancangan laluan di mana robot menentukan kedudukan dan orientasinya serta boleh mengawal pergerakannya menggunakan penderia yang dilengkapi luaran contohnya: penderia inframerah, penderia ultrasonik, penderia laser dan penderia penglihatan (kamera) boleh digunakan untuk membetulkan secara automatik orientasi robot melalui perisian, teknik lain melibatkan NN (Engedy & Horvath 2010), Logik Fuzzy (Montaner & Ramirez-Serrano 1998), Neuro Fuzzy (Zhu & Yang 2007), PSO (Ahmadzadeh, Sedigheh and Ghanavati 2012), GA (Ghorbani et al. 2009), dan ACO (Garcia et al. 2009).

Perancangan laluan digunakan untuk menyelesaikan masalah dalam banyak bidang, daripada perancangan laluan spatial yang mudah kepada memilih tindakan yang betul yang diperlukan untuk mencapai matlamat tertentu. Oleh kerana persekitaran tidak selalu diketahui lebih awal, jenis perancangan ini selalunya terhad kepada persekitaran yang telah direka bentuk atau persekitaran yang boleh diterangkan dengan ketepatan yang mencukupi sebelum proses perancangan. Perancangan laluan boleh digunakan dalam persekitaran yang diketahui sepenuhnya atau sebahagiannya diketahui serta persekitaran yang tidak diketahui sepenuhnya di mana maklumat deria mentakrifkan pergerakan robot yang dikehendaki (Klančar et al. 2017).



Rajah 2.22 (A) Persekitaran dengan halangan dan konfigurasi permulaan dan matlamat, dan (B) satu daripada banyak laluan yang mungkin dari permulaan kepada konfigurasi matlamat

Sumber: Klančar et al. 2017

Perancangan laluan ialah tugas mencari laluan berterusan yang memacu robot dari awal hingga akhir konfigurasi. Keseluruhan laluan mestilah dalam ruang kosong seperti dilihat dalam Rajah 2.22. Untuk perancangan laluan, sistem menggunakan peta persekitaran yang diketahui disimpan dalam ingatan robot. Keadaan atau konfigurasi menentukan kemungkinan pose robot dalam persekitaran dan boleh diwakili sebagai titik dalam ruang konfigurasi yang mengandungi semua kemungkinan keadaan robot. Robot boleh bergerak dari satu keadaan ke keadaan lain dengan melakukan tindakan yang berbeza. Oleh itu, laluan yang baik diterangkan sebagai satu siri tindakan yang membawa robot daripada konfigurasi awal melalui beberapa konfigurasi perantara kepada konfigurasi matlamat. Tindakan mana yang dipilih dalam keadaan semasa dan keadaan mana yang dipilih seterusnya bergantung pada algoritma perancangan laluan dan kriteria yang digunakan. Algoritma memilih keadaan terbaik seterusnya daripada set semua keadaan yang mungkin boleh diakses daripada keadaan semasa. Keputusan ini biasanya dibuat mengikut fungsi kriteria yang ditakrifkan oleh salah satu ukuran jarak seperti jarak Euclidean terpendek ke matlamat.

Antara keadaan permulaan yang diberikan dan matlamat, mungkin terdapat satu atau lebih laluan yang menghubungkan keadaan, atau tiada laluan langsung. Biasanya terdapat beberapa laluan yang mungkin iaitu laluan yang tidak bertembung dengan halangan. Untuk mengehadkan pilihan yang mungkin, keperluan atau kriteria tambahan diperkenalkan yang menentukan optimum yang dikehendaki (Klančar et al. 2017).

- Panjang laluan mestilah minimum.
- Laluan yang ideal ialah laluan yang boleh dilalui oleh robot dalam masa yang paling singkat.
- Laluan hendaklah sejauh mungkin daripada halangan.
- Laluan hendaklah licin dan bebas daripada selekoh tajam.
- Laluan mesti mematuhi sekatan perjalanan.

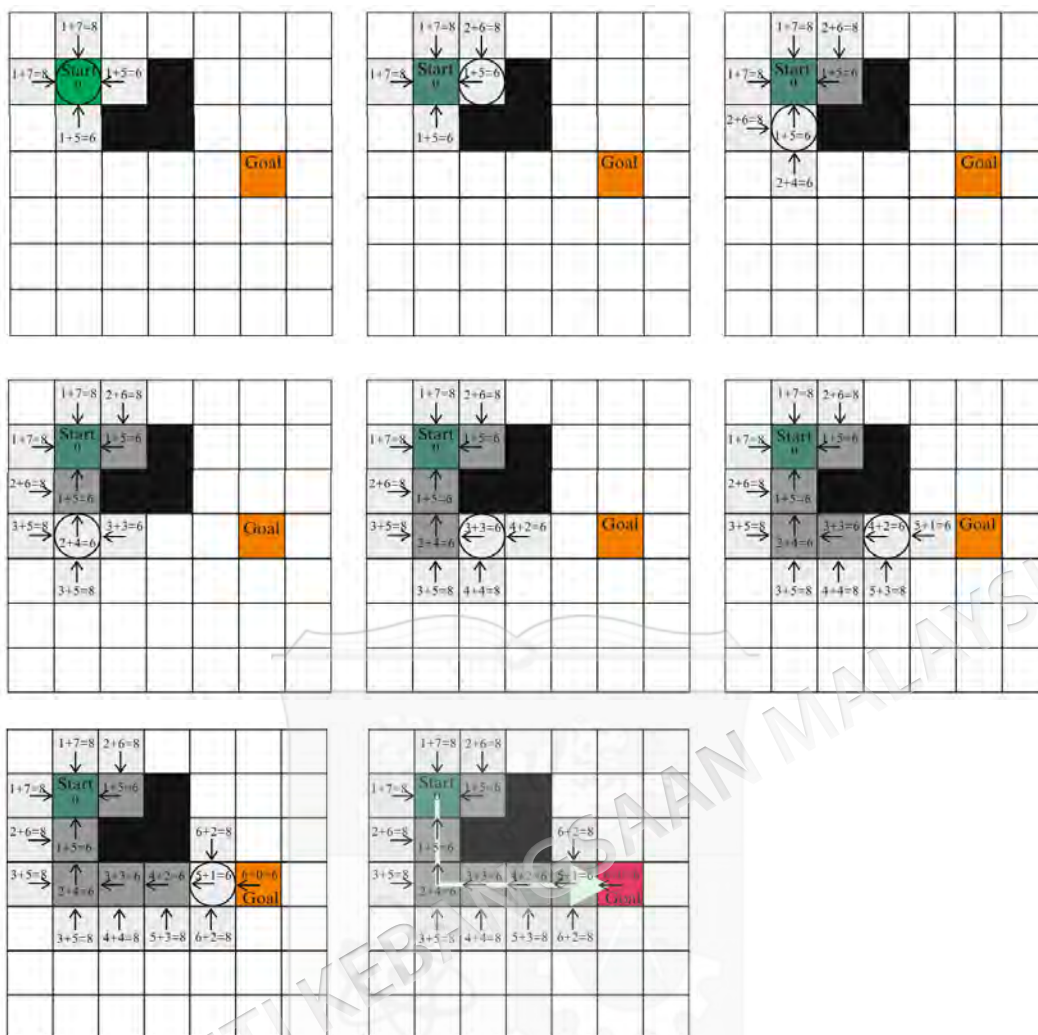
Dalam kajian ini, dibincangkan mengenai perancangan laluan menggunakan algoritma A*, Dijkstra, Perancangan Laluan Kod rantai langsung Freman (*Freman Direct Chain Code-Path Planning* / FDCC-PP).

2.11.1 Algoritma A*

Algoritma A* ialah sebahagian daripada algoritma navigasi klasik (Duchoň et al. 2014). A* ialah algoritma carian yang juga boleh digunakan untuk mencari laluan. Algoritma A* telah digunakan secara meluas sejak ia dicadangkan (Hart et al. 1968). Algoritma secara berterusan mencari tempat yang belum diterokai dalam graf. Semua lokasi dalam grafik dicari dan algoritma berhenti apabila lokasi yang dikehendaki dicapai. Jika matlamat tidak tercapai, robot mencari laluan terpendek antara semua robot jiran (poorva et al. 2018).

A* ialah algoritma bermaklumat kerana ia mengandungi maklumat tambahan atau heuristik. Heuristik adalah untuk menganggarkan kos laluan dari nod semasa ke destinasi memandangkan bahagian pokok graf yang masih belum diterokai. Ini membolehkan algoritma membezakan antara nod yang menjanjikan dan tidak menjanjikan, dengan itu mencari penyelesaian dengan lebih cekap. Untuk setiap nod, algoritma mengira fungsi heuristik yang mewakili anggaran kos perjalanan dari nod itu ke destinasi. Ini dipanggil kos ke destinasi. Fungsi heuristik ini boleh menjadi jarak Euclidean atau jarak Manhattan iaitu jumlah pergerakan menegak dan mendatar dari nod semasa ke nod matlamat. Heuristik juga boleh dikira oleh fungsi lain yang sesuai. Semasa pelaksanaan algoritma, jumlah kos laluandikira untuk setiap nod. Ini terdiri daripada kos sehingga kini dan kos ke matlamat. Senarai nod terbuka dan senarai nod tertutup juga digunakan semasa mencari laluan.

Algoritma A* dilaksanakan seperti berikut: pada mulanya, senarai terbuka hanya mengandungi nod permulaan. Di sini kosnya adalah sifar dan tiada pautan ke nod sebelumnya. Kemudian langkah-langkah berikut juga ditunjukkan dalam Rajah 2.23 Nod permulaan dan matlamat dilabelkan diulang.



Rajah 2.23 Nod permulaan dan matlamat dilabelkan

Sumber: Klančar et al. 2017

1. Dapatkan nod pertama daripada senarai terbuka. Ini dipanggil nod semasa. Senarai terbuka diisi mengikut tertib menaik jumlah kos laluan, dengan nod pertama ialah nod dengan jumlah kos laluan terendah.
2. Untuk semua nod yang boleh dicapai daripada nod semasa, hitung:
 - Kos ke matlamat,
 - Kos ke sini sebagai jumlah kos ke sini nod semasa dan kos peralihan, dan
 - Kos-seluruh-laluan sebagai jumlah kos-ke-sini dan kos-ke-matlamat.
3. Bagi setiap nod yang belum lagi menyimpan kos ke sini, nilai sambungan daripada nod semasa, kos ke matlamat dan kos laluan keseluruhan disimpan.

4. Jika dalam langkah sebelumnya beberapa nod telah menyimpan nilai kos daripada beberapa lelaran sebelumnya, maka kedua-dua kos ke sini iaitu kos semasa dan sebelumnya dibandingkan dan nod dengan nilai yang lebih rendah disimpan, bersama dengan sambungan yang sepadan dan kos laluan keseluruhan.

Nod semasa ditandakan dengan bulatan, nod dalam senarai terbuka adalah kelabu muda, dan nod dalam senarai tertutup adalah kelabu gelap, manakala halangan berwarna hitam. Peralihan boleh dilakukan dalam empat arah (kiri, kanan, atas dan bawah). Dalam setiap sel yang dilawati (nod) arah ke nod induk dilambangkan dengan anak panah. Setiap sel yang dilawati mengandungi kos laluan, iaitu jumlah kos ke sini dan kos ke matlamat. Untuk fungsi kos jarak Manhattan digunakan. Laluan yang ditemui boleh dijejaki dengan mengikuti sambungan yang ditandakan dengan anak panah.

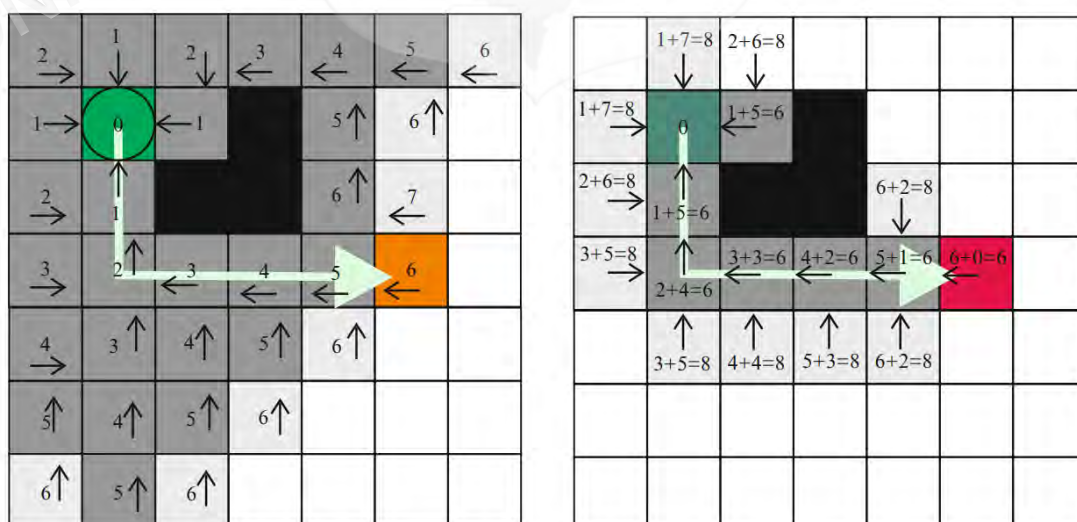
1. Nod yang nilai kosnya telah dikira buat kali pertama ditambahkan pada senarai terbuka. Nod yang telah berada dalam senarai terbuka dan dikemas kini disimpan dalam senarai terbuka. Nod yang telah dikemas kini dan berada dalam senarai tertutup dialihkan ke senarai terbuka. Senarai terbuka diisi mengikut kos-seluruh-laluan dalam susunan yang semakin meningkat. Nod semasa dipindahkan ke senarai tertutup.

Dalam subfigura pertama (dalam Rajah 2.23) nod semasa ialah nod permulaan dan nod penggantinya (diakses dalam empat arah: kiri, kanan, atas dan bawah daripada nod semasa) dipilih. Untuk semua nod pengganti, kos-ke-sini adalah satu kerana ia hanya satu langkah dari nod bintang dan ko-ke-matlamat ialah jarak Manhattan (heuristik) dari nod pengganti ke nod matlamat, yang juga boleh diukur melalui halangan. Jumlah kedua-dua kos ialah kos keseluruhan laluan. Nod pengganti juga mempunyai sambungan yang ditandakan dengan anak panah (arah) ke nod induknya iaitu nod semasa yang ditandakan dengan bulatan. Senarai terbuka kini mengandungi empat nod pengganti dan senarai tertutup mengandungi nod permulaan.

Dalam subfigura kedua dalam Rajah 2.23, nod dengan kos keseluruhan laluan terendah dengan kos 6 dipilih daripada senarai terbuka sebagai nod semasa. Nod semasa hanya mempunyai satu pengganti. Sel lain disekat oleh halangan atau dalam senarai tertutup. Kos-ke-sini untuk nod pengganti ini ialah 2 kerana ia berada dua langkah jauhnya (jarak Manhattan) dari nod permulaan dan matlamat kos ialah 8. Nod semasa diletakkan pada senarai tertutup dan nod pengganti ke senarai terbuka. Dalam subfigura ketiga dalam Rajah 2.23, nod semasa menjadi nod daripada senarai terbuka dengan kos keseluruhan laluan terendah, iaitu 6. Kemudian algoritma diteruskan sama seperti dalam dua langkah pertama.

Algoritma A* dijamin untuk mencari laluan optimum dalam graf jika heuristik untuk pengiraan kos ke matlamat boleh diterima atau optimistik, yang bermaksud anggaran kos ke matlamat adalah lebih kecil atau sama dengan kos sebenar kepada-matlamat. Algoritma tamat apabila nod matlamat ditambahkan pada senarai tertutup.

Algoritma A* ialah algoritma yang lengkap kerana ia mencari laluan jika ia wujud, dan seperti yang telah disebutkan, ia adalah optimum jika heuristik boleh diterima (optimis). Kelemahannya ialah penggunaan memori yang besar. Jika semua kos ke matlamat ditetapkan kepada sifar operasi A* bersamaan dengan algoritma Dijkstra.



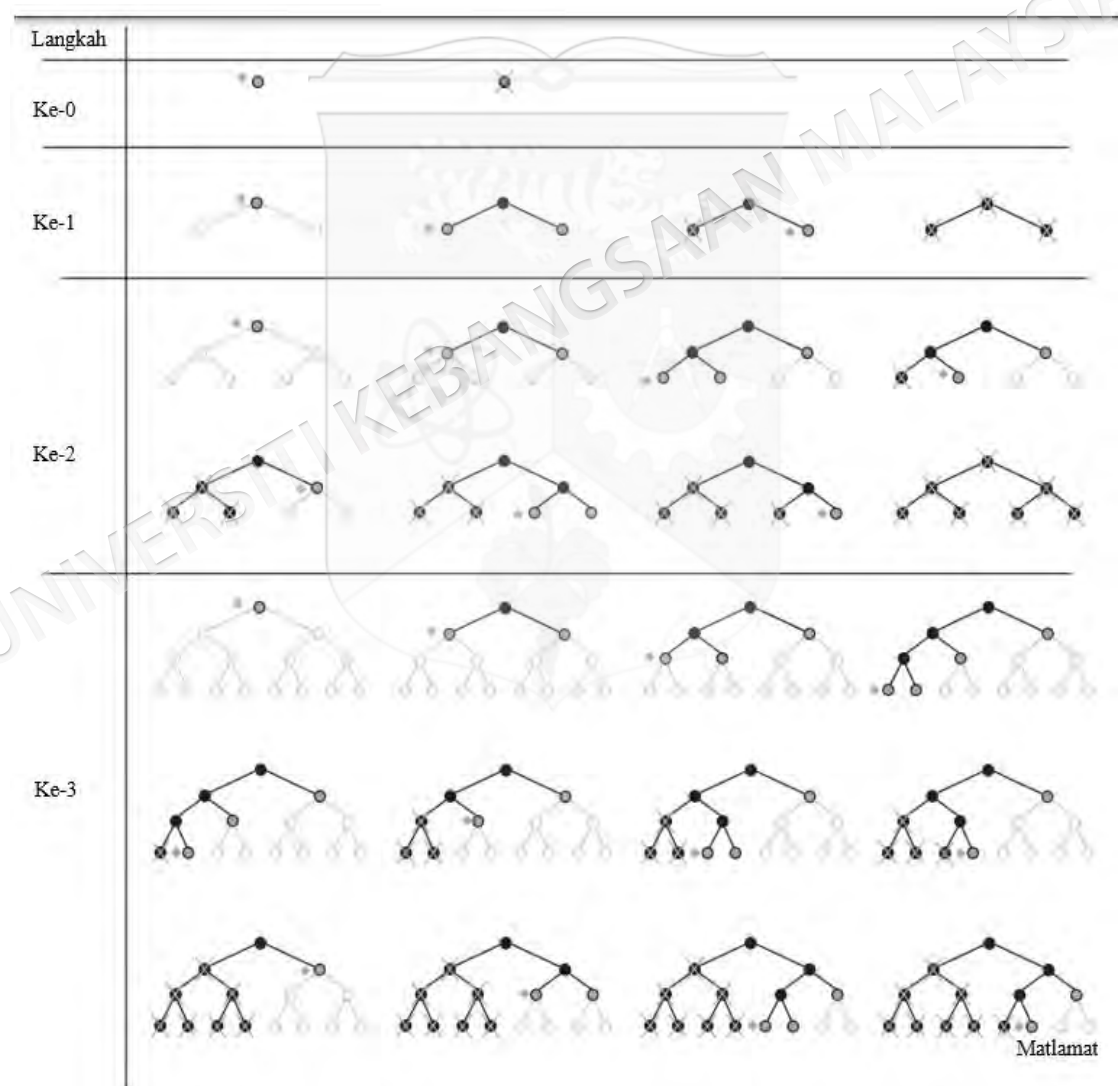
Rajah 2.24 Perbandingan prestasi algoritma Dijkstra (kiri) dan A* (kanan)

Sumber: Klančar et al. 2017

Kedua-duanya adalah algoritma mencari laluan terpendek ke matlamat; bagaimanapun, algoritma A* memerlukan kurang lelaran (*iteration*) disebabkan oleh heuristik yang digunakan semasa mencari graf.

2.11.2 Algoritma Dijkstra

Algoritma Dijkstra ialah algoritma tidak bermaklumat yang mencari laluan terpendek dari satu nod (nod sumber) ke semua nod lain dalam graf (Mehlhorn, Kurt and Sanders, Peter and Sanders 2008).



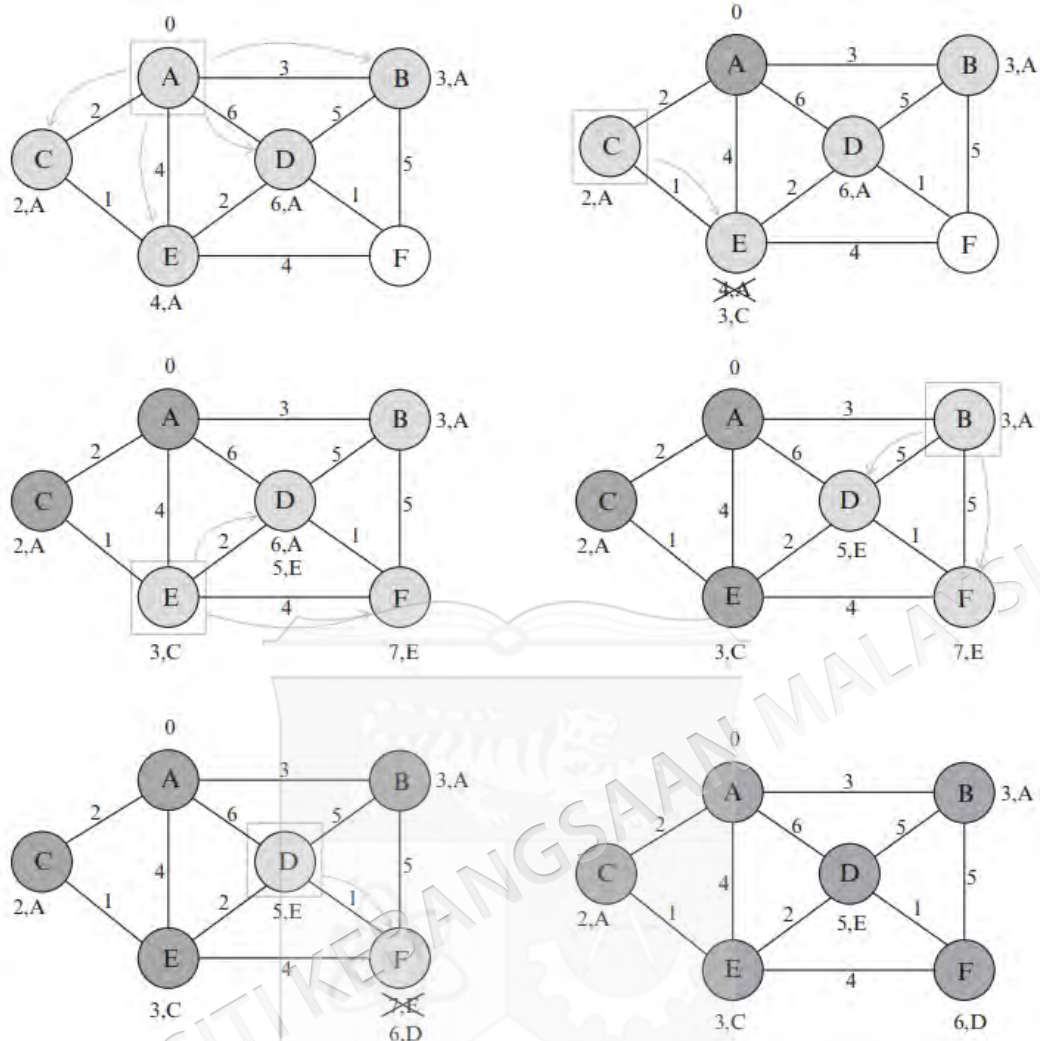
Rajah 2.25 Demonstrasi algoritma carian mendalam-pertama pendalaman berulang

Sumber: Mehlhorn, Kurt and Sanders, Peter and Sanders 2008

Pada Rajah Rajah 2.25 boleh dilihat demonstrasi algoritma carian mendalam-pertama pendalaman berulang, di mana carian diulang hingga kedalaman langkah pokok. Nod yang dikeluarkan daripada ingatan ditandakan dengan salib. Algoritma ditamatkan apabila nod matlamat dalam langkah ketiga ditemui. Hasilnya ialah pokok laluan terpendek dari nod sumber ke mana-mana nod lain. Pada asalnya ia dicadangkan oleh Dijkstra (Dijkstra 1959) dan kemudiannya diperluaskan dengan banyak pengubahsuaian. Dalam masalah di mana laluan terpendek antara satu nod permulaan dan satu nod matlamat mesti ditemui algoritma Dijkstra mungkin kelihatan tidak cekap kerana pengiraan laluan optimum kepada semua nod lain. Dalam kes ini, algoritma boleh ditamatkan sebaik sahaja laluan terpendek yang dikehendaki dikira.

Algoritma mencari laluan terpendek antara nod permulaan dan nod matlamat kerana ia mengira fungsi kos untuk laluan antara nod permulaan ke nod semasa, yang kami namakan kos ke sini. Kos laluan ke nod semasa (kos ke sini) dikira sebagai jumlah kos laluan ke nod sebelumnya (dari mana kita datang ke nod semasa) dan kos peralihan antara nod sebelumnya dan nod semasa. Dalam kes beberapa laluan terpendek (laluan dengan kos yang sama) algoritma mengembalikan hanya satu, dan tidak penting yang mana satu.

Untuk menjalankan algoritma, peralihan antara nod dan kosnya perlu ditakrifkan. Untuk setiap nod yang dilawati, algoritma menyimpan kos laluan terpendek pada masa ini (kos ke sini) dan sambungan dari tempat kami tiba ke nod semasa. Semasa carian, senarai nod terbuka dan nod tertutup disesuaikan. Operasi algoritma adalah seperti berikut. Pada mulanya, senarai terbuka hanya mengandungi nod permulaan, yang mempunyai sifar kos ke sini dan ia tanpa sambungan ke nod sebelumnya. Senarai nod tertutup kosong. Kemudian langkah-langkah berikut juga ditunjukkan dalam Rajah 2.26.



Rajah 2.26 Algoritma Dijkstra untuk mencari laluan terpendek dari nod permulaan A ke semua nod lain

Sumber: Klančar et al. 2017

Nod semasa dikelilingi dengan segi empat sama kelabu dan penggantinya ditandakan dengan anak panah. Kos untuk peralihan antara nod ditandakan pada sambungan. Di sepanjang nod kos ke sini dan sambungan ke nod sebelumnya diberikan. Nod dalam senarai terbuka diwarnakan dengan kelabu muda manakala nod dalam senarai tertutup diwarnakan dengan kelabu gelap. Contoh: Jika kita perlu mendapatkan laluan terpendek antara nod A dan F kosnya ialah kos $F-D-E-C-A = 6$, manakala laluan melalui nod $A \rightarrow C \rightarrow E \rightarrow D \rightarrow F$.

1. Dari senarai terbuka ambil nod pertama; memanggilnya nod semasa. Senarai terbuka diisih mengikut kos ke sini dalam susunan yang semakin meningkat, di mana nod pertama ialah nod dengan kos ke sini terkecil.
2. Untuk semua nod yang boleh dicapai dari nod semasa hitung kos ke sini sebagai jumlah kos ke sini dan kos peralihan nod semasa.
3. Kira dan simpan kos ke sini dan sambungan yang sesuai ke nod semasa untuk semua nod yang belum mempunyai maklumat ini disimpan.
4. Jika dalam langkah sebelumnya beberapa nod sudah mempunyai sambungan kos-ke-sini dan bersesuaian daripada beberapa lelaran sebelumnya, kemudian bandingkan kedua-dua kos-ke-sini dan simpan dalam storan kos-ke-sini yang lebih rendah dan sambungannya.
5. Nod ditambahkan pada senarai terbuka dan diisih mengikut tertib kos-ke-sini yang semakin meningkat. Senarai sedemikian dipanggil baris gilir keutamaan dan membolehkan nod dengan harga terendah ke sini ditemui lebih cepat daripada dalam senarai yang tidak diisih. Nod semasa dialihkan ke senarai tertutup.

Pada asalnya, algoritma Dijkstra ditamatkan apabila senarai nod terbuka kosong dan hasilnya adalah laluan terpendek dari nod permulaan ke semua nod lain. Jika hanya laluan terpendek dari nod permulaan dan beberapa nod matlamat diperlukan maka algoritma ditamatkan apabila nod matlamat ditambah ke senarai tertutup.

Laluan yang terhasil boleh diperolehi dengan menjejak kembali sambungan yang membawa kita ke nod matlamat. Pada nod matlamat sambungan ke nod sebelumnya diperolehi kemudian sambungan ke nod sebelumnya dibaca dan seterusnya sehingga nod permulaan dicapai. Akhirnya, senarai sambungan yang diperolehi diterbalikkan. Algoritma Dijkstra adalah algoritma lengkap dan optimum jika semua sambungan mempunyai kos lebih tinggi daripada sifar.

2.11.3 Algoritma *Freeman Direction Chain Code-Path Planning* (FDCC-PP)

Algoritma FDCC-PP ialah algoritma yang menggunakan Kod Rantai Freeman dalam menavigasi robot. Algoritma ini masih agak baharu, yang dicadangkan oleh M. Zhao dan Lu pada tahun 2019 (M. Zhao & Lu 2019).

Algoritma FDCC-PP terdiri daripada dua proses. Proses pertama ialah pemprosesan peta, di mana kod rantai arah Freeman diguna pakai untuk mengekstrak maklumat tentang kedudukan dan arah halangan pada peta. Proses kedua algoritma FDCC-PP ialah perancangan laluan, yang termasuk bergerak ke arah titik matlamat dalam ruang kosong dan membuat lencongan pantas mengikut kod pautan rantai Freeman berhampiran halangan. Algoritma FDCC-PP telah pun membuat perbandingan dengan algoritma lain. Hasilnya ialah algoritma ini boleh mengurangkan lebih carian dalam ruang kosong dan memendekkan masa perancangan laluan tanpa menjejaskan keselamatan robot. Algoritma ini membolehkan perancangan laluan pantas dalam persekitaran yang kompleks. Walau bagaimanapun, algoritma ini masih menggunakan 4 dan 8 kod rantai daripada Freeman, supaya robot maksimum hanya boleh berpusing pada 45 darjah. Ini dianggap belum lancar, jadi ia perlu diubah suai. Penyelidikan yang dijalankan adalah untuk mengubahsuai algoritma FDCC-PP kepada 16 dan 32 kod.

Berbeza daripada algoritma lain, maklumat koordinat dan maklumat arah sempadan halangan telah pun diperolehi dalam pemprosesan peta dalam algoritma FDCC-PP. Oleh itu, apabila robot berada berhampiran halangan, ia hanya perlu menanyakan kod rantai arah halangan yang sepadan dan memilih lencongan terpendek, bukannya bergerak mengelilingi halangan untuk mencari laluan lencongan terbaik sebelum bergerak ke arah destinasi semula. Memandangkan terdapat beberapa perbezaan dalam perancangan laluan untuk robot tunggal dan berbilang robot, dalam bahagian ini, kami akan menerangkan perancangan laluan untuk robot tunggal dan berbilang robot masing-masing, dan membandingkannya dalam Rajah 2.27.

Perancangan Laluan	Untuk Robot Tunggal
Di Ruang Kosong	 - Bergerak di sepanjang garisan antara titik mula dan titik akhir.
Berhampiran Halangan	 - Bergerak di sepanjang garisan antara titik mula dan titik akhir. - Cari kod rantai berhampiran halangan dan pilih lencongan pendek.

Rajah 2.27 Perbandingan antara robot tunggal dan berbilang robot apabila merancang laluan oleh algoritma FDCC-PP

Sumber: M. Zhao & Lu 2019

Untuk meminimumkan laluan robot, kami mengamalkan prinsip bergerak ke arah titik matlamat, bergerak di sepanjang garis antara titik mula dan titik matlamat dalam ruang bebas. Jika kedudukan seterusnya yang robot akan bergerak adalah halangan, di mana robot mungkin berlanggar. Kami akan mengambil tiga langkah untuk memintas halangan itu. Pertama sekali, kita perlu mencari indeks halangan, yang boleh diperolehi dengan menanyakan maklumat sempadan halangan mengikut koordinat kedudukan seterusnya. Kemudian, bucu halangan boleh diperolehi dengan mengira kod rantaian pembezaan halangan. Kemudian robot boleh mendapatkan destinasi semasa mengelilingi halangan dengan membandingkan jarak Euclidean dari setiap bucu halangan ke titik matlamat. Untuk memintas halangan secepat mungkin dan bergerak ke titik matlamat semula, puncak halangan yang paling hampir dengan titik matlamat dipilih sebagai destinasi semasa robot dalam algoritma FDCC-PP. Akhirnya, laluan dengan langkah bergerak yang lebih sedikit diperolehi dengan membandingkan bilangan

langkah yang diperlukan untuk mengikut arah jam dan lawan jam memintas halangan, dan kod rantai yang sepadan direkodkan sebagai petunjuk arah untuk pergerakan robot.

Kod pseudo algoritma FDCC-PP yang digunakan untuk perancangan laluan robot tunggal dapat dilihat pada Rajah 2.28. Di mana *cp*, *np*, *S* dan *E* ialah kedudukan semasa, kedudukan seterusnya, titik mula dan titik matlamat robot dalam perancangan laluan, masing-masing.

```

1: Import peta bengkel sebagai gambar
2: Mulakan S, E, cp, np, laluan = []
3: Pemprosesan peta
4: while cp  $\neq$  E //perancangan laluan
5:   kira np
6:   if peta (np) = 1
7:     cari nombor halangan di mana np berada
8:     kira destinasi semasa mengelilingi halangan
9:     bandingkan panjang laluan dua lencongan
10:    simpan lencongan terpendek dalam laluan
11:    cp = destinasi semasa
12:  else
13:    simpan np dalam laluan
14:    cp = np
15:  end if
16: end while
17: Output laluan sebagai laluan yang dirancang untuk robot

```

Rajah 2.28 Algoritma FDCC-PP

Sumber: M. Zhao & Lu 2019

Algoritma FDCC-PP menggunakan kod rantai arah Freeman untuk mengekstrak maklumat tentang kedudukan dan arah halangan semasa pemprosesan peta. Untuk merancang laluan sesingkat mungkin, robot dibenarkan untuk bergerak ke titik matlamat di ruang bebas dan memilih lencongan terpendek dengan bantuan maklumat halangan dalam jarak yang hampir dengan halangan. Selain itu, kerana algoritma FDCC-PP menggunakan sepenuhnya maklumat halangan yang diperoleh daripada peta, hubungan antara nod dalam peta adalah tetap.

2.12 KERJA SEBELUMNYA MENGGUNAKAN PENDERIA ULTRASONIK DAN PENGEKOD DALAM NAVIGASI ROBOT

Pada penyelidikan ini, untuk mengetahui jarak penderia dengan halangan digunakan penderia ultrasonik. Gelombang ultrasonik ini bergerak melalui udara pada kelajuan

lebih kurang 344 meter sesaat. Untuk mengetahui jarak antara penderia dan halangan, ia dikira berdasarkan tempoh masa perjalanan gelombang ultrasonik (t_{IN}) didarab dengan kelajuan gelombang dibahagikan dengan 2. Jadi jarak ke objek yang diukur ialah $[(t_{IN}) \cdot s \times 344 \text{ m/s}] \div 2$ meter.

Penderia ultrasonik sering digunakan dalam navigasi robot. Penderia ultrasonik boleh berdiri sendiri atau digabungkan dengan penderia pengekod. Penggunaan penderia pengekod di sini adalah untuk menentukan bilangan putaran motor DC yang memacu roda robot. Biasanya penderia pengekod ini dipasang selari dengan aci motor DC. Dengan bantuan penderia pengekod dan penderia ultrasonik, navigasi robot boleh menjadi lebih baik. Penggunaan penderia ini dalam bidang navigasi robot dapat dilihat daripada beberapa kajian yang boleh dilihat pada Jadual 2.5.

Jadual 2.5 Kerja sebelumnya menggunakan penderia ultrasonik dan pengekod

No.	Pengarang	Penderia	Method	Deskripsi
1	(Lye et al. 2014)	Penderia ultrasonik, pengekod.	<i>A* Algorithm</i>	Robot mudah alih dilengkapi penderia ultrasonik, pengekod dalam menavigasi persekitaran dalaman.
2	(Omrane et al. 2016)	Penderia julat inframerah (IR), pengekod.	<i>Fuzzy logic</i>	Robot mudah alih dilengkapi sembilan penderia julat inframerah (IR) untuk mengelakkan halangan.
3	(Oh et al. 2018)	Penderia inframerah, penderia ultrasonik, dan pengekod.	2D SLAM	Membangunkan robot mudah alih autonomi dalam persekitaran dalaman.
4	(C.-H. Chen et al. 2021)	Penderia ultrasonik, pengekod.	<i>Knowledge-based Neural Fuzzy Controller (KNFC)</i>	Robot mudah alih menavigasi menggunakan KNFC berdasarkan input daripada penderia ultrasonik dan pengekod.

bersambung...

sambungan...

5	(Asad et al. 2022)	Penderia ultrasonik, pengekod.	<i>Fuzzy Logic</i>	Robot mudah alih Qbot2 dilengkapi dengan tiga penderia ultrasonik (SRF-04) yang dipasang pada bampar hadapan dan motor DC dengan pengekod. Pengawal mengambil input daripada tiga penderia ultrasonik dan menjana arahan kelajuan untuk mengelakkan sebarang halangan di laluannya.
6	(Hoang et al. 2022)	Penderia inersia (IMU), pengekod.	<i>Kalman Filter</i>	Robot mudah alih beroperasi dalam persekitaran tertentu yang dicadangkan. Sudut orientasi dan kedudukan robot dianggarkan oleh data yang dikumpul daripada penderia inersia. Penderia pengekod pada paksi motor dikira oleh algoritma penapis Kalman.
7	(Koodtalang et al. 2022)	Penderia ultrasonik, pengekod, GPS, kompas.	-	Setiap destinasi robot mudah alih boleh dicapai dengan kaedah navigasi yang dicadangkan, digabungkan dengan GPS, pengekod dan penderia kompas digital. Selain itu, tiga penderia ultrasonik diletakkan pada jarak 45 darjah, yang telah digunakan untuk mengelakkan halangan secara automatik apabila robot bergerak ke arah lokasi matlamat.
8	(Phueakthong et al. 2022)	Penderia ultrasonik, pengekod, pengimbas laser 2D.	<i>Deep reinforcement learning</i>	Robot mudah alih menggunakan pembelajaran pengukuhan men-dalam untuk navigasi robot dan pengelakan halangan di kawasan yang tidak diketahui sebelum ini atau tanpa peta pra-dibuat.

bersambung...

sambungan...

9	(Divya Vani et al. 2022)	Penderia ultrasonik, pengekod.	Dijkstra	Kajian ini mencadangkan tentang navigasi autonomi <i>multi-robot</i> untuk pengangkutan dalam persekitaran menggunakan algoritma Dijkstra untuk mengelakkan halangan.
---	--------------------------	--------------------------------	----------	---

Dalam Jadual 2.5 dapat dilihat bahawa dalam navigasi robot sentiasa menggunakan penderia ultrasonik atau penderia pengekod. Pelbagai kaedah digunakan, contohnya A*, Dijkstra, *Fuzzy logic*, *Kalman Filter*, *Neural Fuzzy Logic*. Kaedah ini dikenali sebagai kaedah klasik. Kaedah ini biasanya digunakan dalam navigasi menggunakan peta. Namun begitu, sekiranya robot itu sedang menavigasi di kawasan yang belum dipetakan, maka kaedah yang sering digunakan adalah, contohnya: Penyetempatan dan Pemetaan Serentak (*Simultaneous Localization and Mapping/SLAM*) dan Pembelajaran Peneguhan Dalam (*Deep Reinforcement Learning*). Namun begitu, kerana navigasi dalam kajian ini menggunakan peta, navigasi menggunakan kaedah SLAM dan *Deep Reinforcement Learning* tidak perlu digunakan. Penggunaan kaedah navigasi klasik adalah lebih sesuai kerana kawasan itu telah dipetakan dan dimodelkan terlebih dahulu.

2.13 KESIMPULAN

Optimasi identifikasi sampah dan navigasi robot merupakan perkara yang sangat penting bagi memastikan robot berautonomi mendapatkan sampah yang sebenar (bentuk 3D) dan juga memudahkan robot berautonomi bergerak di perkarangan rumah dalam mendapatkan sampah tersebut. Terdapat beberapa seni bina Rangkaian Neural Convolution yang boleh digunakan untuk pengecaman sampah, iaitu AlexNet, VGG16, GoogleNet dan ResNet18. Namun, penggunaan seni bina ini tidak dapat memastikan objek yang dikesan adalah sampah sebenar. Beberapa kajian terdahulu telah menunjukkan bahawa penderia LiDAR boleh digunakan untuk mengesan objek dalam 3D. Penderia LiDAR yang digunakan mempunyai ketepatan $\pm 2\text{mm}$. Pemilihan penderia ini mengambil kira beberapa perkara antaranya; pertama, resolusi bentuk 3D hasil imbasan adalah lebih baik daripada penderia lain, menjadikannya lebih mudah untuk

menentukan kedudukan kutipan sampah oleh pencengkam robot. Kedua, berkemungkinan peratusan robot dalam mengesan objek sebenar (3D) mempunyai ketepatan yang lebih tinggi berbanding menggunakan penderia lain. Purata penderia LiDAR dalam kerja terdahulu mempunyai ketepatan dalam sentimeter (cm), manakala penderia LiDAR yang digunakan mempunyai ketepatan dalam milimeter (mm). Dalam algoritma navigasi robot, terdapat beberapa kajian lepas yang telah dibincangkan iaitu; Dijkstra, A* dan Kod Rantaian *Freeman*–Perancangan Laluan (FDCC-PP). Antara ketiga-tiga algoritma ini, algoritma FDCC-PP adalah lebih baik. Bagaimanapun, algoritma FDCC-PP ini masih dianggap kurang lancar dalam pergerakan robot, jadi algoritma ini masih perlu dibuat lebih lancar supaya robot lebih mahir dalam mengemudi.

Tesis ini akan fokus pada penyelesaian cabaran-cabaran ini. Pertama, kajian terhadap pembangunan reka bentuk robot berautonomi. Kedua, kajian bagi menyelesaikan masalah optimasi identifikasi sampah di perkarangan rumah. Ketiga, kajian bagi menyelesaikan masalah optimasi dalam navigasi robot berautonomi di perkarangan rumah. Keempat, kajian untuk menguji keberkesanan dan prestasi keseluruhan sistem yang dibangunkan.

BAB III

METODOLOGI PENYELIDIKAN

3.1 PENGENALAN

Dalam mengatasi atau menyelesaikan permasalahan dan mencapai objektif penyelidikan maka penggunaan metodologi penyelidikan yang tepat adalah amat penting. Bab 3 ini akan menjelaskan secara terperinci mengenai kaedah yang digunakan untuk mengoptimumkan identifikasi sampah dan navigasi robot berautonomi. Penyelidikan dilaksanakan di Makmal Permodelan dan Simulasi Institut Informatik Visual (IVI) Universiti Kebangsaan Malaysia (UKM). Bagi memudahkan pemahaman berkaitan dengan kaedah yang digunakan dalam penyelidikan ini, maka bab ini akan dibincangkan berasaskan objektif penyelidikan. Sehingga tergambarkan antara objektif yang hendak dicapai dengan kaedah yang digunakan untuk mencapai objektif tersebut.

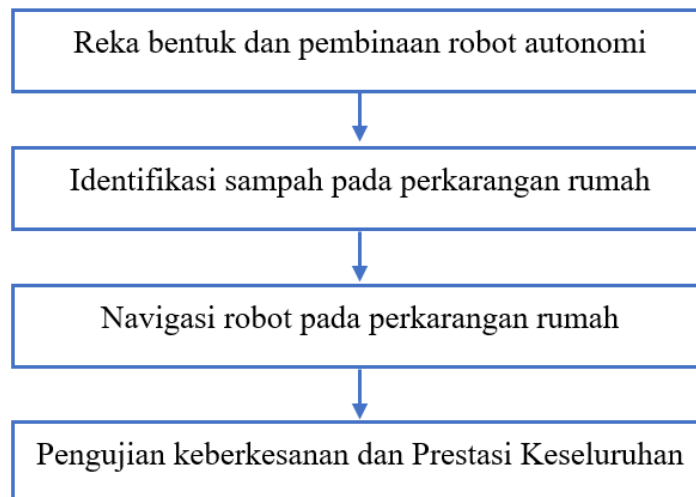
Dalam penyelidikan ini, dicadangkan prototaip robot berautonomi untuk mengidentifikasi sampah dan menavigasi pada perkarangan rumah, algoritma optimasi identifikasi sampah, algoritma optimasi navigasi robot dan evaluasi optimasi identifikasi sampah dan navigasi robot berautonomi. Metodologi dan tahap penyelidikan secara terperinci berasaskan objektif penyelidikan ditunjukkan pada Jadual 3.1.

Penyelidikan dimulakan dengan pembinaan robot berautonomi untuk mengidentifikasi sampah dan menavigasi di perkarangan rumah. Seterusnya proses membangunkan algoritma optimasi identifikasi sampah. Dan yang terakhir adalah membangunkan algoritma optimasi navigasi robot berautonomi di perkarangan rumah. Secara terperinci akan kaedah-kaedah ini akan dibincangkan pada subtopik 3.2, 3.3 dan 3.4. Proses penyelidikan boleh dilihat dalam Rajah 3.1.

Jadual 3.1 Objektif penyelidikan dan tahapan penyelidikan

NO.	OBJEKTIF PENYELIDIKAN	METODOLOGI DAN TAHAPAN KAJIAN
1	Membina protaip sistem robot berautonomi yang mampu mengoptimasi identifikasi sampah dan navigasi robot pada perkarangan rumah	- Reka bentuk robot berautonomi - Pembuatan robot berautonomi - Pengujian prestasi robot berautonomi
2	Membangunkan algoritma optimasi identifikasi sampah pada perkarangan rumah	- Pembuatan algoritma optimasi identifikasi sampah pada perkarangan rumah. - Membina perisian menggunakan MATLAB 2020 berasaskan algoritma yang dibuat. - Melakukan pengesahan algoritma yang dibuat dengan sampah berbentuk 2D dan 3D. - Membandingkan antara prestasi algoritma yang dikembangkan dengan algoritma yang sedia ada.
3	Membangunkan algoritma optimasi navigasi robot berautonomi pada perkarangan rumah	- Pembuatan algoritma optimasi navigasi robot berautonomi. - Membina perisian menggunakan MATLAB berasaskan algoritma yang dibuat. - Membandingkan antara prestasi algoritma yang dikembangkan dengan algoritma yang sedia ada.

Penyelidikan dimulakan dengan pembinaan robot berautonomi untuk mengidentifikasi sampah dan menavigasi di perkarangan rumah. Seterusnya proses mengembangkan algoritma optimasi identifikasi sampah. Dan yang terakhir adalah mengembangkan algoritma optimasi navigasi robot berautonomi pada perkarangan rumah. Secara terprinci akan kaedah-kaedah ini akan dibincangkan pada subtopik 3.2, 3.3 dan 3.4. Rajah 3.1 menunjukkan proses yang dilakukan dalam penyelidikan.



Rajah 3.1 Proses penyelidikan

3.2 REKA BENTUK ROBOT AUTONOMI

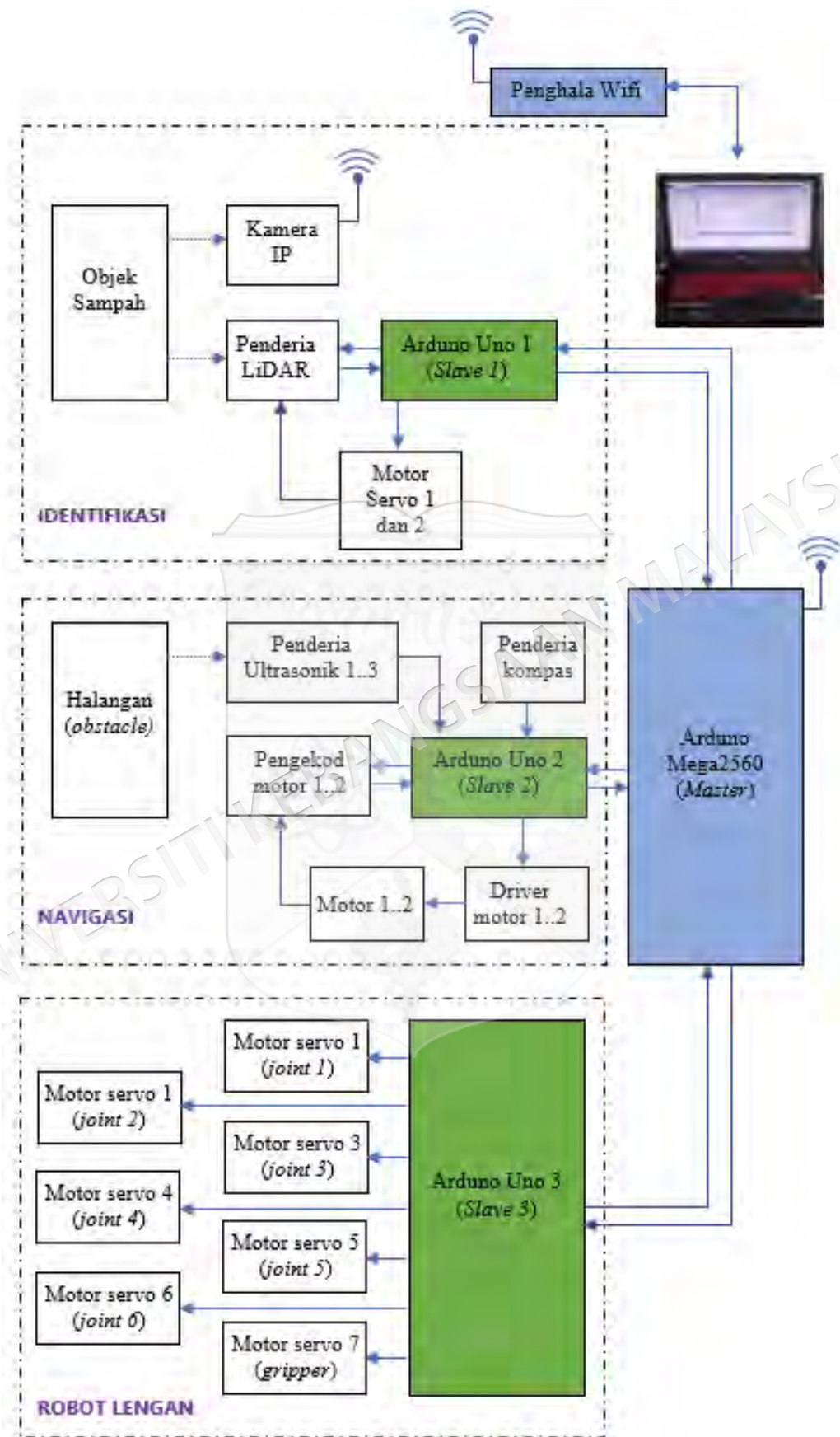
Reka bentuk robot dibuat sebelum prototaip robot berautonomi dibina. Untuk pergerakan robot, trek berterusan digunakan. Ini menganggap terdapat empat laluan yang perlu dilalui oleh robot iaitu: asfalt, simen, tanah dan rumput. Reka bentuk fizikal robot boleh dilihat dalam Rajah 3.2.



Rajah 3.2 Reka bentuk fizikal robot berautonomi

Dalam Rajah 3.2 dapat dilihat bahawa robot berbentuk tangki itu digerakkan oleh dua motor DC 12Volt. Motor DC dilengkapi dengan penderia pengekod, untuk menentukan pergerakan robot. Di bahagian hadapan robot terdapat tiga penderia ultrasonik. Penderia pertama diletakkan di hadapan. Penderia kedua dan ketiga diletakkan di sebelah kiri dan kanan robot dengan sudut 30 darjah. Penderia ini berfungsi untuk menentukan jarak dan halangan di hadapan robot. Di bahagian atas robot terdapat bahagian kawalan yang diletakkan di dalam kotak. Kotak ini berfungsi untuk mengelakkan kerosakan pada komponen elektronik di bahagian kawalan. Kotak itu juga mengandungi bateri, pemacu motor DC dan penderia kompas. Bateri adalah sumber kuasa untuk menjalankan robot.

Pemacu motor DC berfungsi untuk membekalkan kuasa yang sesuai kepada motor DC. Penderia kompas berfungsi sebagai input untuk menentukan arah mata angin. Penderia kompas berguna untuk navigasi robot mengikut kaedah navigasi robot yang dicadangkan. Di bahagian hadapan robot itu juga terdapat kamera IP, yang berfungsi sebagai input dalam mengesan objek sampah. Bagi memastikan objek sampah yang dikesan adalah nyata (3D), robot itu turut dilengkapi dengan penderia LiDAR. Penderia LiDAR ini akan mengimbas objek sampah yang ditemuinya. Agar proses imbasan berjalan dengan baik, penderia LiDAR dilengkapi dengan motor servo 1 dan motor servo 2. Motor ini berfungsi untuk menggerakkan penderia LiDAR dalam koordinat x dan koordinat z. Di bahagian atas robot juga terdapat Lengan Robot (*Robot Arm*). Robot lengan ini berfungsi untuk mengangkat objek sampah ke dalam bekas yang disediakan. Robot lengan ini terdiri daripada 6 darjah kebebasan (*6 degree of freedom*). Setiap sendi didorong oleh motor servo sendiri. Selain itu, terdapat juga pencengkam di hujung robot (*end of effector*) yang turut digerakkan oleh motor servo. Pencengkam ini berfungsi untuk mencengkam objek sampah apabila objek ingin digerakkan/diangkat. Keseluruhan gambar rajah blok prototaip robot boleh dilihat dalam Rajah 3.3.



Rajah 3.3 Gambar rajah blok robot berautonomi

Dalam rajah blok dalam Rajah 3.3 dapat dilihat prototaip robot yang direka bentuk terbahagi kepada 3 bahagian iaitu; 1. Pengecaman sampah, 2. Navigasi robot, 3. Lengan robot. Setiap robot dikawal oleh pengawal iaitu Arduino Uno. Identifikasi sampah dikawal oleh Uno 1 (*Slave 1*), navigasi dikawal oleh Arduino Uno 2 (*Slave 2*), dan lengan robot dikawal oleh Arduino 3 (*Slave 3*). Ketiga-tiga pengawal ini disambungkan kepada Arduino Mega2560 (*Master*). Fungsi induk mengawal selia kerja setiap *slave* dalam kawalan penyelarasan robot berautonomi. Arduino Mega2560 (*Master*) disambungkan ke komputer melalui rangkaian Wifi. Pemrosesan pengecaman dibuat di komputer bukan di atas robot. Ini kerana kekuatan pemrosesan robot rendah. Pemilihan reka bentuk rangkaian wayarles ini untuk memudahkan kawalan. Selain itu, prototaip sistem gerakan robot akan dibebankan dengan berat komputer. Tidak kurang pentingnya ialah aspek keselamatan komputer dari persekitaran luar seperti cuaca, risiko terjatuh dan kecurian. Berikut menerangkan setiap bahagian dengan lebih terperinci.

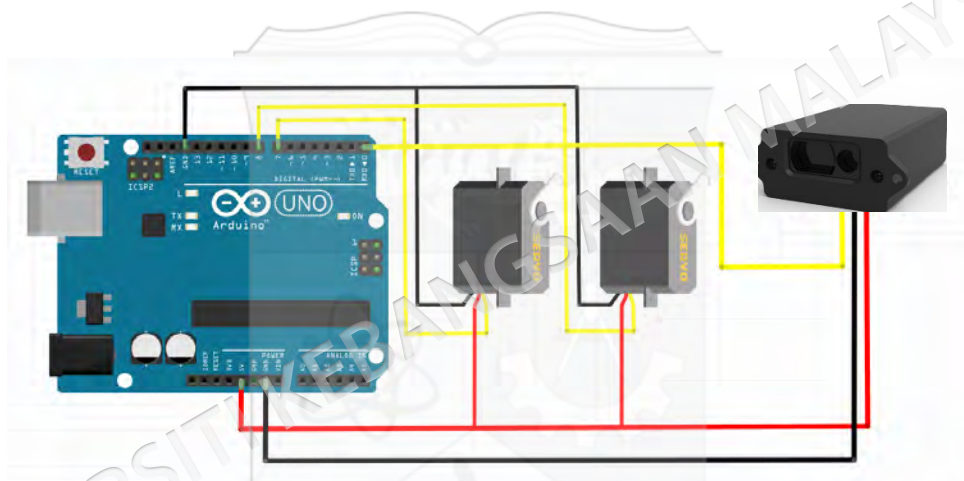
3.2.1 Gambar rajah blok identifikasi sampah

Identifikasi sampah ini dikawal oleh Arduino Uno 1 (*Slave 1*) berdasarkan arahan daripada Arduino Mega 2560 (*Master*). Arahan daripada *Master* diperoleh daripada keputusan yang dihasilkan oleh program Matlab pada komputer. Kali pertama komputer menerima data imej yang dihantar oleh kamera IP. Data imej ini diproses menggunakan program Matlab. Jika ia dikesan sebagai sampah, maka Matlab menghantar arahan melalui wifi kepada Arduino Mega2560 (*Master*) untuk mengaktifkan LiDAR. Data penerima LiDAR disimpan dalam memori Arduino Uno 1 (*Slave 1*). Dalam sistem yang direka bentuk pada Rajah 3.4, penerima LiDAR dipasang di atas motor servo 1 dan motor servo 2.



Rajah 3.4 Susun atur reka bentuk sistem pengimbasan menggunakan penerima LiDAR

Kemudian proses imbasan dijalankan menggunakan penderia LiDAR. Proses pengimbasan dijalankan dalam 2 peringkat iaitu proses pengimbasan pantas dan terperinci. Ini adalah untuk meningkatkan masa imbasan yang cekap. Jika objek yang dikesan adalah rata, maka objek tersebut adalah objek 2D. Jika ia 2D, proses pengimbasan terperinci tidak diperlukan. Walau bagaimanapun, jika objek tidak rata, objek yang dikesan adalah objek 3D. Jika objek adalah 3D, pemprosesan pengimbasan terperinci perlu dilakukan. Ini bertujuan untuk mengetahui dengan pasti bentuk geometri objek tersebut. Dengan data geometri objek, robot penggenggam boleh dengan mudah mengambil objek sampah. Rajah 3.5 merupakan skema skim litar sistem pengecaman sampah.



Rajah 3.5 Skim litar sistem pengecaman sampah

Dalam Rajah 3.5 dapat dilihat bahawa motor servo +5V disambungkan terus ke pin Arduino Uno +5V, negatif (-) motor servo disambungkan ke pin tanah (gnd) Arduino Uno. Isyarat kawalan untuk motor servo 1 disambungkan ke pin 7 Arduino Uno, manakala isyarat kawalan untuk motor servo 2 disambungkan ke pin 8 Arduino Uno. Pemilihan pin 7 dan pin 8 adalah berdasarkan fakta bahawa pin ini adalah pin I/O digital. Pada asasnya, pin I/O digital lain boleh digunakan, selagi ia tidak digunakan untuk komponen input/output lain.

3.2.2 Gambar rajah navigasi robot

Proses navigasi robot dikawal oleh Arduino Uno 2 (*Slave 2*) berdasarkan arahan daripada Arduino Mega2560. Arahan ini mengandungi arahan untuk memacu motor

DC 1 dan motor DC 2 dengan laluan yang ingin dilalui oleh robot. Kedudukan semasa robot ditentukan dengan melaporkan daripada pengekod 1 dan penderia pengekod 2 dan penderia kompas GY-271.

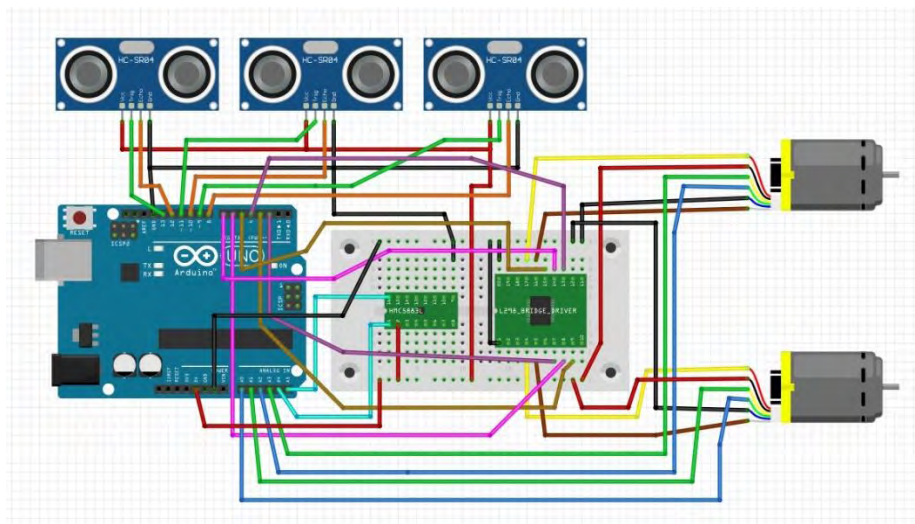
Menentukan arah robot adalah sangat penting. Sebelum kelaziman kompas elektronik dan keupayaannya untuk menjadi padat menggunakan teknologi hibrid, arah yang ditunjukkan oleh robot biasanya ditentukan oleh pengiraan kinematik berdasarkan gerakan atau bergantung pada bacaan penderia pengekod. Walau bagaimanapun, apabila robot berasaskan roda bergerak, mereka diketahui mempunyai kecenderungan untuk tergelincir kerana inersia semasa memulakan atau membrek, atau akibat pelanggaran dengan objek atau robot lain.

Untuk mengesan halangan di hadapan dan di sebelah robot, penderia ultrasonik digunakan. Penderia diletakkan dalam 3 kedudukan, seperti yang ditunjukkan dalam Rajah 3.6.



Rajah 3.6 Susunan reka bentuk penderia ultrasonik

Dalam Rajah 3.6 dapat dilihat bahawa penderia pertama diletakkan di hadapan, penderia kedua diletakkan di sebelah kiri robot dan penderia ketiga diletakkan di sebelah kanan robot. Untuk penderia kedua dan penderia ketiga, terdapat perbezaan tahap kecondongan dengan penderia pertama sebanyak 30 darjah. Skim litar penderia ultrasonik boleh dilihat dalam Rajah 3.7.



Rajah 3.7 Litar sistem navigasi robot

Dalam Rajah 3.7 dapat dilihat bahawa punca voltan positif dan negatif disambungkan kepada punca voltan positif dan negatif Arduino Uno. Input isyarat pencetus untuk penderia ultrasonik 1,2,3 disambungkan kepada pin 13, 11 dan pin 9 Arduino Uno, masing-masing. Input sumber voltan untuk Motor DC disambungkan ke litar pemacu L298. Pemacu ini digunakan untuk mengawal arah putaran motor DC dan mengawal kelajuan putaran motor DC, supaya robot boleh bergerak dan bernavigasi dengan betul. Penderia pengekod disambungkan terus ke pin A3 ke pin A5 Arduino Uno. Modul program mengira bilangan putaran motor DC berdasarkan input pada pin A3 ke pin A5. Penderia kompas GY-271 disambungkan ke pin A0 dan pin A1 Arduino Uno. Pin isyarat yang digunakan pada penderia kompas ialah *Serial Clock* dan *Serial Data*. Daripada isyarat ini dapat dipastikan bahawa data darjah dari penderia kompas dihantar secara bersiri ke Arduino Uno.

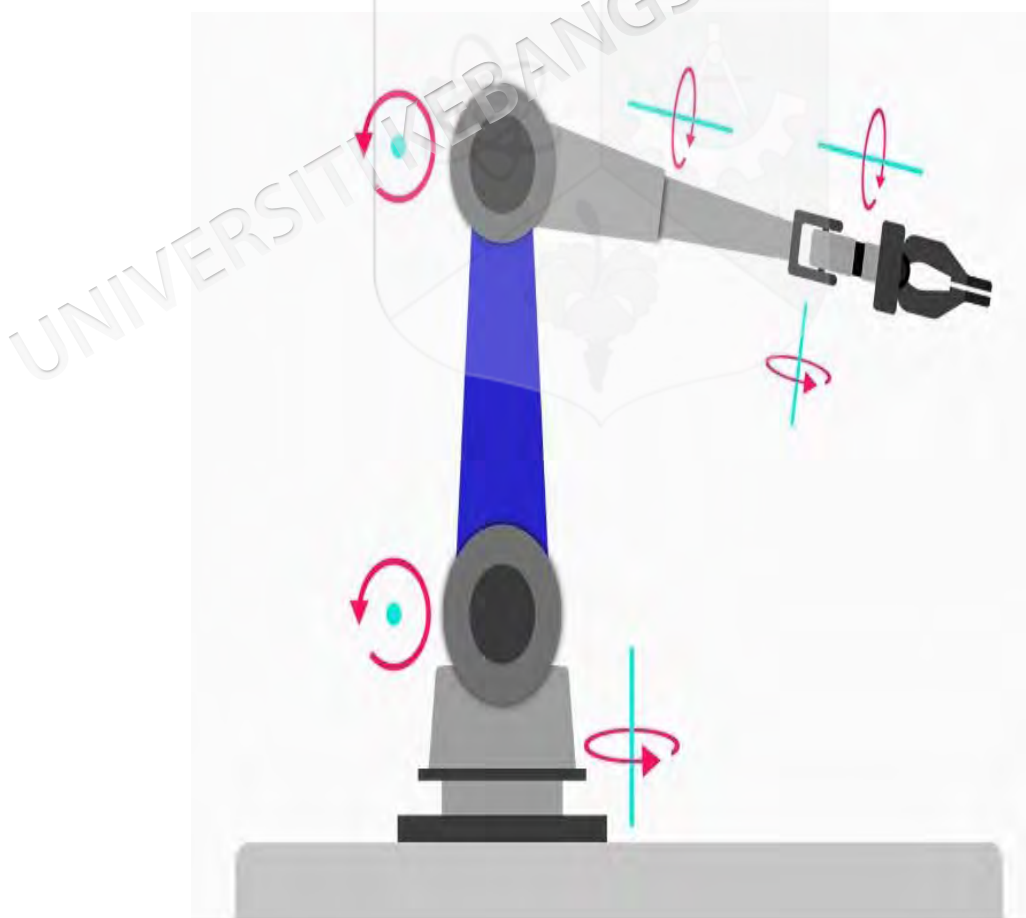
3.2.3 Gambar rajah blok lengan robot

Untuk mengangkat sampah menggunakan lengan robot. Lengan robot ini dikawal oleh Arduino Uno 3 (*Slave 3*) berdasarkan arahan yang diberikan oleh Arduino Mega 2560 (*Master*). Lengan robot menggunakan motor servo 1 hingga 7 motor servo. Fungsi setiap motor servo boleh dilihat dalam Jadual 3.2.

Jadual 3.2 Fungsi dan Spesifikasi Motor Servo Pada Lengan Robot

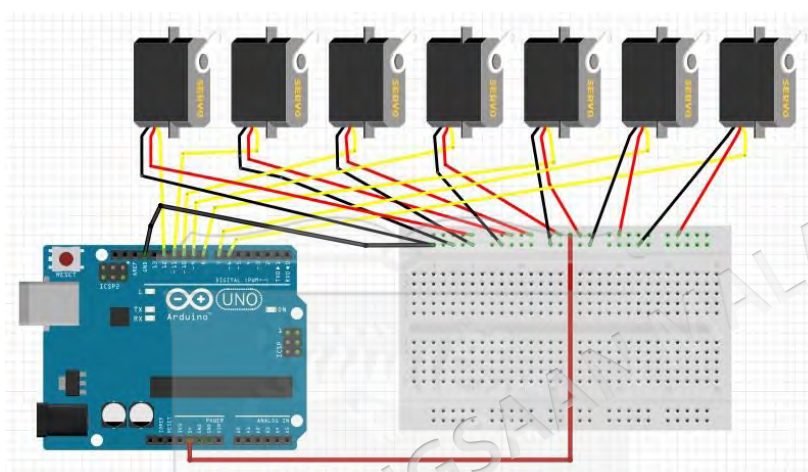
No.	Penerangan	Kod	Penggunaan	Tork
1	Motor Servo 1	MG996R	Joint 1	11kg/cm (6.0V)
2	Motor Servo 2	MG996R	Joint 2	11kg/cm (6.0V)
3	Motor Servo 3	MG996R	Joint 3	11kg/cm (6.0V)
4	Motor Servo 4	MG996R	Joint 4	11kg/cm (6.0V)
5	Motor Servo 5	MG996R	Joint 5	11kg/cm (6.0V)
6	Motor Servo 6	MG996R	Joint 6	11kg/cm (6.0V)
7	Motor Servo 7	MG995R	Gripper	11kg/cm (6.0V)

Dalam Jadual 3.2 Fungsi dan Spesifikasi Motor Servo Pada Lengan Robot dapat dilihat motor servo 1 hingga motor servo 6 mempunyai jenis yang sama iaitu MG996R dengan daya kilas 11kg/cm pada voltan +6VDC. Manakala motor servo 7 pula mempunyai jenis yang berbeza iaitu MG995R dengan daya kilas yang sama. Perbezaan ini disebabkan oleh motor servo 7 dianggap sesuai untuk mengangkat beban dengan jenis MG995R. Susunan reka bentuk lengan robot boleh dilihat dalam Rajah 3.8.



Rajah 3.8 Susunan reka bentuk motor servo pada lengan robot 6 dof

Dalam Rajah 3.8 dapat dilihat bahawa arah anak panah menunjukkan arah pergerakan robot. Terdapat 6 anak panah, yang bermaksud bahawa lengan robot mempunyai 6 darjah kebebasan (dof). Motor servo 1 hingga 6 masing-masing diletakkan pada sambungan 1 hingga 6. Setiap sambungan disambungkan dengan pautan. Manakala motor servo 7 dikhaskan untuk menggerakkan pencengkam robotik. Skim litar lengan robot boleh dilihat dalam Rajah 3.9.



Rajah 3.9 Skema litar sistem lengan robot

Dalam Rajah 3.9 dapat dilihat bahawa motor servo positif (+) 1 kepada motor servo 7 digabungkan dahulu, kemudian disambungkan kepada +5V Arduino Uno. Begitu juga, motor servo negatif (-) 1 kepada motor servo 7 digabungkan terlebih dahulu, kemudian disambungkan ke tanah Arduino Uno (gnd). Sumber voltan ini mungkin menggunakan sumber voltan luaran (dari luar Arduino Uno). Untuk isyarat kawalan motor servo 1 ke motor servo 7 setiap satu disambungkan ke pin 6 hingga pin 12 Arduino Uno. Dalam modul program ujian motor servo, nombor pin yang digunakan mesti sepadan supaya kawalan pergerakan motor servo berjalan dengan baik.

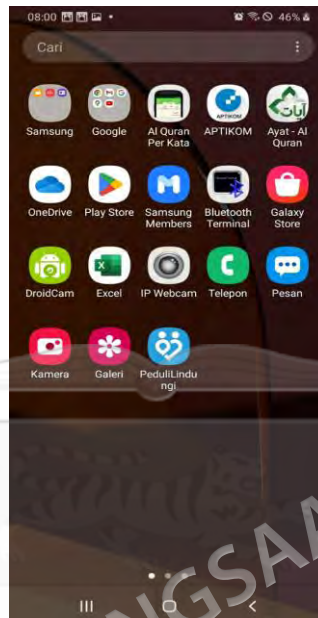
3.2.4 Pembinaan modul program

a. Modul program identifikasi sampah

Untuk memastikan identifikasi sampah berfungsi dengan baik, setiap komponen sokongan proses identifikasi sampah mesti diisi dengan program ujian terlebih dahulu. Berikut ialah program pengujian yang terkandung dalam identifikasi.

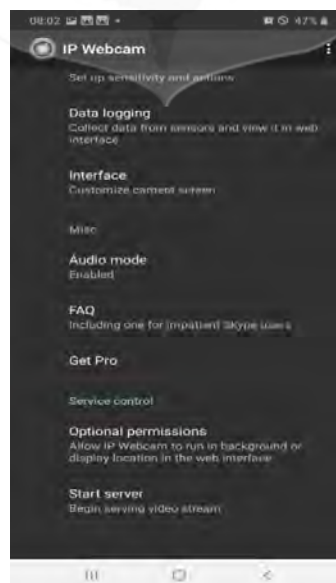
i. Modul ujian program kamera IP

Untuk mengaktifkan kamera IP, telefon pintar mesti memasang aplikasi IP Webcam dengan logo seperti yang ditunjukkan dalam Rajah 3.10.



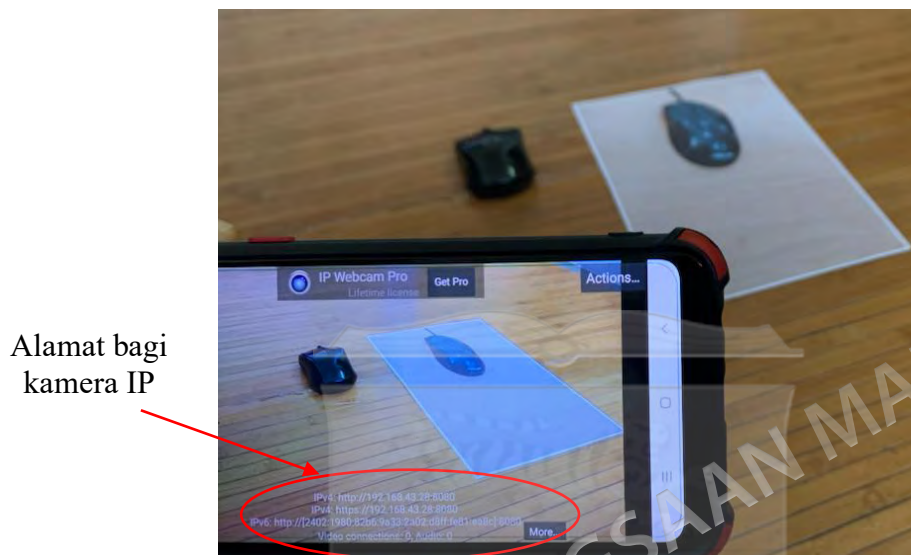
Rajah 3.10 Paparan skrin telefon pintar

Kemudian klik pada ikon IP Webcam (), supaya paparan muncul seperti yang ditunjukkan dalam Rajah 3.11.



Rajah 3.11 Paparan aplikasi Kamera IP

Kemudian klik pada 'Start server' supaya paparan kelihatan seperti Rajah 3.12. Tetapi pastikan telefon pintar disambungkan ke *internet*. Alamat IP daripada kamera IP yang digunakan akan muncul pada skrin telefon pintar.

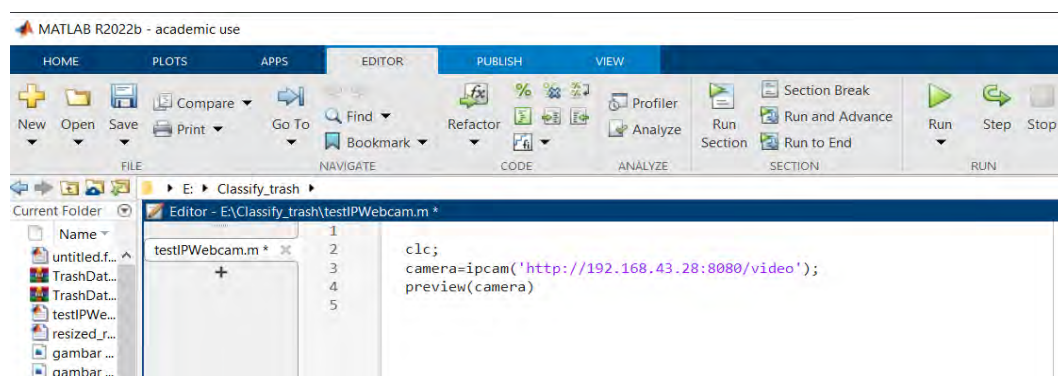


Rajah 3.12 Paparan alamat bagi kamera IP

Supaya paparan pada skrin telefon pintar juga muncul pada skrin monitor komputer, maka dalam perisian MATLAB, nilai IP yang ditetapkan mestilah sama. Perintah untuk tetapan IP adalah seperti berikut.

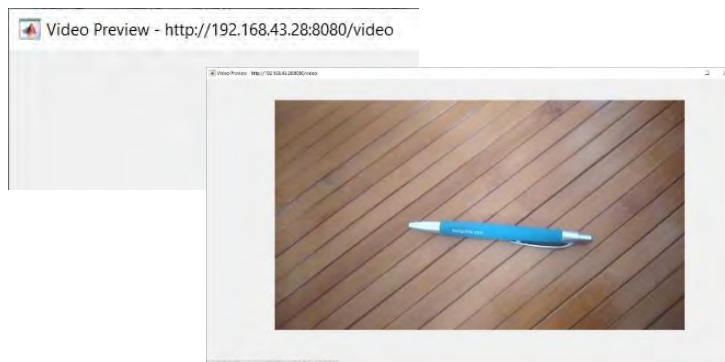
```
Camera = ipcam ('http://192.168.43.28:8080/video');
```

Rajah 3.13 berikut menunjukkan perisian MATLAB untuk menguji kamera IP.



Rajah 3.13 Paparan perisian Matlab untuk ujian kamera IP

Jika modul atur cara berfungsi dengan baik, ia akan muncul seperti dalam Rajah 3.14.



Rajah 3.14 Paparan program ujian kamera IP yang sedang berjalan

Di bahagian atas sebelah kiri dalam Rajah 3.14 terdapat alamat IP daripada kamera IP yang digunakan. Dalam kes ini alamat IP ialah: 'http://192.168.43.28:8080/video'.

ii. Modul program ujian penerima LiDAR

Selepas litar LiDAR dipasang seperti yang ditunjukkan dalam Rajah 3.5, penerima LiDAR diuji menggunakan modul program berikut.

```

TesLiDAR | Arduino 1.8.12
File Edit Serial Monitor Tools Help
TesLiDAR

#include <SoftwareSerial.h> //header file of software serial port
//SoftwareSerial Serial1(2,3); //define software serial port name as Serial1 and define pin2 as RX and pin3
as TX
SoftwareSerial Serial1(10,8); //define software serial port name as Serial1 and define pin2 as RX and pin3
as TX
/* For Arduino boards with multiple serial ports like DUE board, interpret above two pieces of code and
directly use Serial1 serial port*/

unsigned long dist,dist1,dist2,dist3; //actual distance measurements of LiDAR
int i;
int dataFrame[9]; //save data measured by LiDAR

const int FRAME_LEN=0x04; // frame header of data length
const int HEADER_ADDR=0x01; // frame header of data frame
const int HEADER_FUN=0x03; // frame header

/*Commands*/
char CMD_DISABLE_CONTS[8] = {0x01, 0x03, 0x00, 0x0A, 0x00, 0x02, 0xE4, 0x09}; // Disable
continous detection mode
char CMD_ENABLE_CONTS[8] = {0x01, 0x03, 0x00, 0x01, 0x00, 0x02, 0x95, 0xCB}; // Enable
continous detection mode
Done Saving

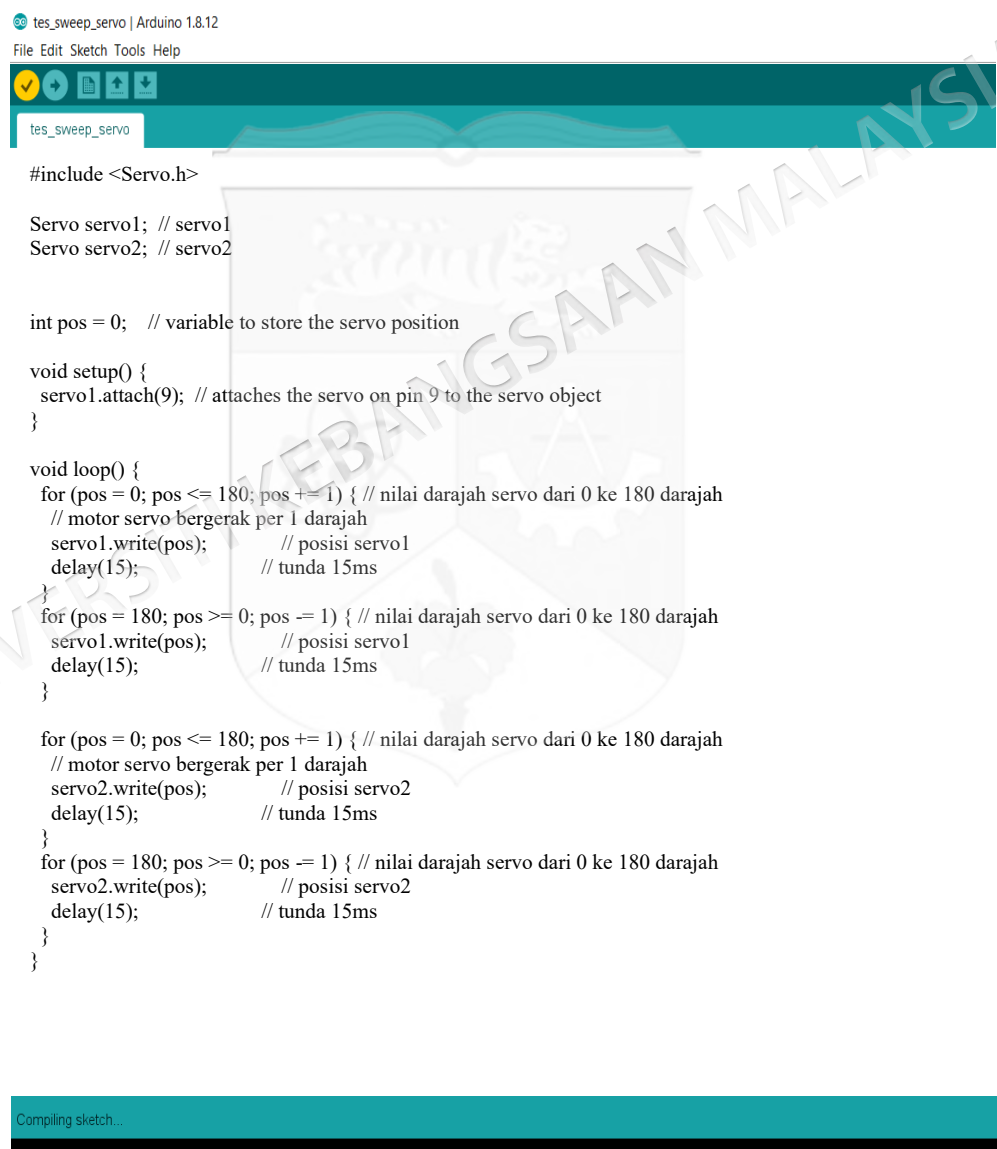
```

Rajah 3.15 Paparan modul program ujian penerima LiDAR pada Arduino IDE

Dalam Rajah 3.15 boleh dilihat program ujian penderia LiDAR menggunakan IDE Arduino. Jika modul program berjalan dengan baik, serial monitor komputer akan memaparkan jarak yang diukur oleh penderia.

iii. Modul program ujian Motor Servo 1 dan Motor Servo 2

Selepas litar motor servo dipasang seperti yang ditunjukkan dalam Rajah 3.5, motor servo diuji menggunakan modul program berikut.



```
tes_sweep_servo | Arduino 1.8.12
File Edit Sketch Tools Help

tes_sweep_servo

#include <Servo.h>

Servo servo1; // servo1
Servo servo2; // servo2

int pos = 0; // variable to store the servo position

void setup() {
  servo1.attach(9); // attaches the servo on pin 9 to the servo object
}

void loop() {
  for (pos = 0; pos <= 180; pos += 1) { // nilai darajah servo dari 0 ke 180 darajah
    // motor servo bergerak per 1 darajah
    servo1.write(pos); // posisi servo1
    delay(15); // tunda 15ms
  }
  for (pos = 180; pos >= 0; pos -= 1) { // nilai darajah servo dari 0 ke 180 darajah
    servo1.write(pos); // posisi servo1
    delay(15); // tunda 15ms
  }

  for (pos = 0; pos <= 180; pos += 1) { // nilai darajah servo dari 0 ke 180 darajah
    // motor servo bergerak per 1 darajah
    servo2.write(pos); // posisi servo2
    delay(15); // tunda 15ms
  }
  for (pos = 180; pos >= 0; pos -= 1) { // nilai darajah servo dari 0 ke 180 darajah
    servo2.write(pos); // posisi servo2
    delay(15); // tunda 15ms
  }
}
```

Compiling sketch...

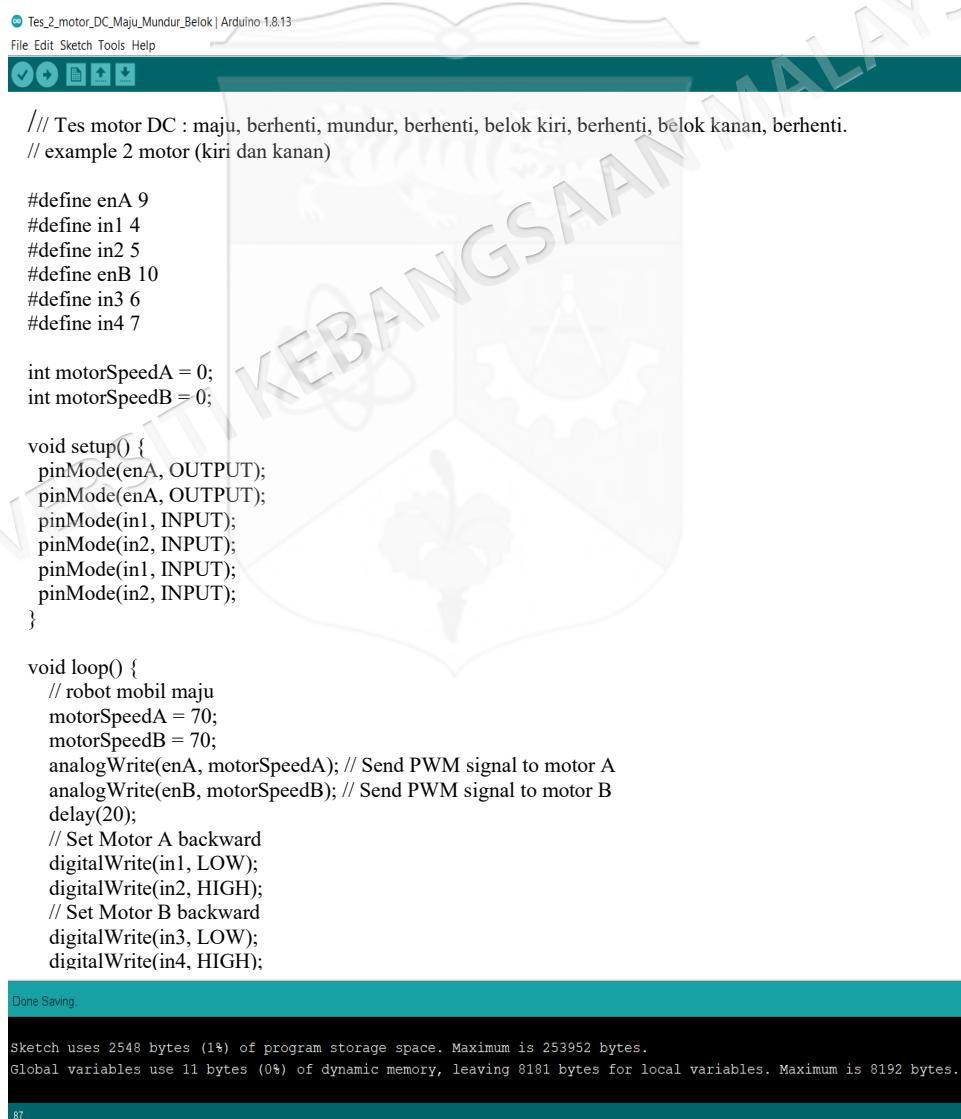
Rajah 3.16 Paparan modul program ujian motor servo 1 dan motor servo 2

Dalam Rajah 3.16 boleh dilihat program ujian motor servo menggunakan Arduino IDE. Jika modul program berjalan dengan baik, maka motor servo 1 dan motor servo 2 akan bergerak secara berselang-seli dari 0 hingga 180 darjah. Selepas mencapai 180 darjah, motor servo akan bergerak semula ke kedudukan 0 darjah.

b. Modul program navigasi robot

i. Modul program ujian Motor DC

Selepas litar motor DC dipasang seperti yang ditunjukkan dalam Rajah 3.7, motor DC diuji menggunakan modul program berikut.



```

Tes_2_motor_DC_Maju_Mundur_Belok | Arduino 1.8.13
File Edit Sketch Tools Help

// Tes motor DC : maju, berhenti, mundur, berhenti, belok kiri, berhenti, belok kanan, berhenti.
// example 2 motor (kiri dan kanan)

#define enA 9
#define in1 4
#define in2 5
#define enB 10
#define in3 6
#define in4 7

int motorSpeedA = 0;
int motorSpeedB = 0;

void setup() {
  pinMode(enA, OUTPUT);
  pinMode(enA, OUTPUT);
  pinMode(in1, INPUT);
  pinMode(in2, INPUT);
  pinMode(in1, INPUT);
  pinMode(in2, INPUT);
}

void loop() {
  // robot mobil maju
  motorSpeedA = 70;
  motorSpeedB = 70;
  analogWrite(enA, motorSpeedA); // Send PWM signal to motor A
  analogWrite(enB, motorSpeedB); // Send PWM signal to motor B
  delay(20);
  // Set Motor A backward
  digitalWrite(in1, LOW);
  digitalWrite(in2, HIGH);
  // Set Motor B backward
  digitalWrite(in3, LOW);
  digitalWrite(in4, HIGH);
}

Done Saving
Sketch uses 2548 bytes (1%) of program storage space. Maximum is 253952 bytes.
Global variables use 11 bytes (0%) of dynamic memory, leaving 8181 bytes for local variables. Maximum is 8192 bytes.
87

```

Rajah 3.17 Paparan modul program ujian motor DC 1 dan motor DC 2

Dalam Rajah 3.17 boleh dilihat program ujian motor DC menggunakan IDE Arduino. Jika modul program berjalan lancar, motor DC 1 dan motor DC 2 akan menggerakkan robot mudah alih ke hadapan dan kemudian ke belakang. Selepas itu belok kiri dan kemudian belok kanan. Di antara setiap keadaan ini, motor DC berhenti seketika. Ini bertujuan untuk mengelakkan kerosakan pada motor DC, terutamanya dalam arah putaran yang berbeza.

ii. Modul program ujian penderia pengekod

Penderia pengekod terletak di bahagian belakang motor DC. Jadi setiap kali motor berputar, penderia pengekod akan mengesan setiap putaran. Selepas kabel penderia pengekod disambungkan seperti yang ditunjukkan dalam Rajah 3.7, penderia pengekod diuji menggunakan modul program berikut.



```
tes_encoder_position1 | Arduino 1.8.13
File Edit Sketch Tools Help

tes_encoder_position1

#define Encoder_A 2 // pin2 of the Arduino
#define Encoder_B 3 // pin 3 of the Arduino
// Dua pin ini mempunyai interupsi hardware.

int hitung_pulses = 0;
void setup() {
  Serial.begin(9600); // activates the serial communication
  pinMode(Encoder_A, INPUT); // sets the Encoder_A pin as the input
  pinMode(Encoder_B, INPUT); // sets the Encoder_B pin as the input
  attachInterrupt(digitalPinToInterrupt(Encoder_A), DC_Motor_Encoder, RISING);
}

void loop() {
  Serial.println("Result: ");
  Serial.println(hitung_pulses);
}

void DC_Motor_Encoder(){
  int b = digitalRead(Encoder_B);
  if(b > 0){
    hitung_pulses++;
  }
  else{
    hitung_pulses--;
  }
}

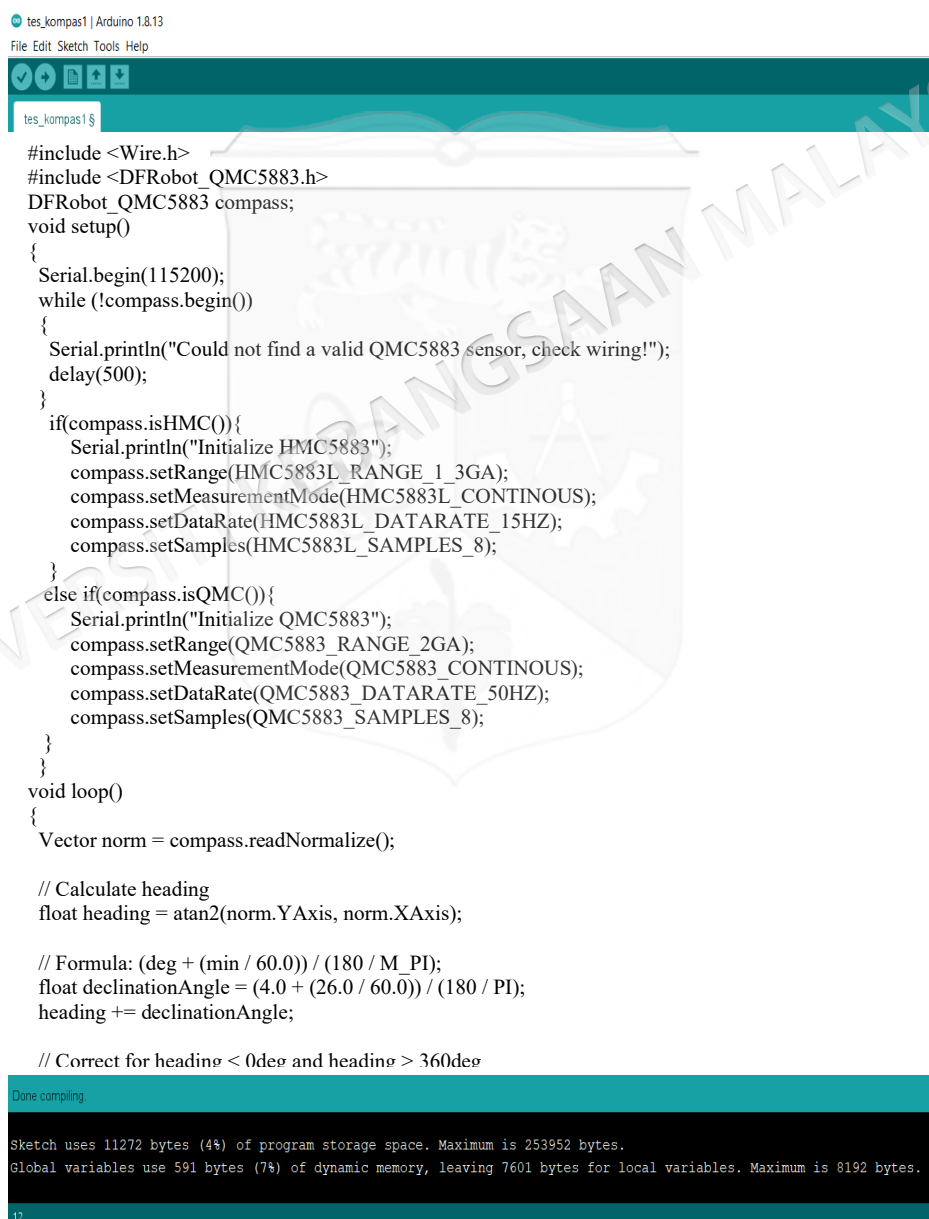
Done compiling.

Sketch uses 3552 bytes (1%) of program storage space. Maximum is 253952 bytes.
Global variables use 214 bytes (2%) of dynamic memory, leaving 7978 bytes for local variables. Maximum is 8192 bytes.
19
```

Rajah 3.18 Paparan modul program ujian penderia pengekod

iii. Modul program ujian penderia kompas

Penderia kompas diletakkan di bahagian kawalan kotak prototaip robot. Ini bertujuan sebagai usaha untuk memastikan penderia tidak mudah rosak. Kedudukan penderia mesti diletakkan pada satah rata supaya setiap anjakan darjah robot, nilai darjah ini diketahui dengan pasti dan betul oleh penderia. Selepas penderia kompas dipasang dan kabel penderia disambungkan seperti yang ditunjukkan dalam Rajah 3.7, penderia kompas diuji menggunakan modul program berikut.



```

tes_kompas1 | Arduino 1.8.13
File Edit Sketch Tools Help

tes_kompas1 $
#include <Wire.h>
#include <DFRobot_QMC5883.h>
DFRobot_QMC5883 compass;
void setup()
{
  Serial.begin(115200);
  while (!compass.begin())
  {
    Serial.println("Could not find a valid QMC5883 sensor, check wiring!");
    delay(500);
  }
  if(compass.isHMC()){
    Serial.println("Initialize HMC5883");
    compass.setRange(HMC5883L_RANGE_1_3GA);
    compass.setMeasurementMode(HMC5883L_CONTINOUS);
    compass.setDataRate(HMC5883L_DATARATE_15HZ);
    compass.setSamples(HMC5883L_SAMPLES_8);
  }
  else if(compass.isQMC()){
    Serial.println("Initialize QMC5883");
    compass.setRange(QMC5883_RANGE_2GA);
    compass.setMeasurementMode(QMC5883_CONTINOUS);
    compass.setDataRate(QMC5883_DATARATE_50HZ);
    compass.setSamples(QMC5883_SAMPLES_8);
  }
}
void loop()
{
  Vector norm = compass.readNormalize();

  // Calculate heading
  float heading = atan2(norm.YAxis, norm.XAxis);

  // Formula: (deg + (min / 60.0)) / (180 / M_PI);
  float declinationAngle = (4.0 + (26.0 / 60.0)) / (180 / PI);
  heading += declinationAngle;

  // Correct for heading < 0deg and heading > 360deg

Done compiling

Sketch uses 11272 bytes (4%) of program storage space. Maximum is 253952 bytes.
Global variables use 591 bytes (7%) of dynamic memory, leaving 7601 bytes for local variables. Maximum is 8192 bytes.
12

```

Rajah 3.19 Paparan modul program ujian penderia kompas

iv. Modul program ujian penderia ultrasonik

Terdapat 3 penderia ultrasonik yang digunakan dalam sistem navigasi robot. Penderia ultrasonik 1 diletakkan di bahagian hadapan. Manakala penderia ultrasonik 2 diletakkan di sebelah kiri dan penderia ultrasonik 3 diletakkan di sebelah kanan robot. Selepas penderia ultrasonik dipasang dan kabel penderia disambungkan seperti yang ditunjukkan dalam Rajah 3.7, penderia ultrasonik diuji menggunakan modul program berikut.



```

tes_sensor_ultrasonik | Arduino 1.8.13
File Edit Sketch Tools Help

tes_sensor_ultrasonik

#define pin_ultra 10
#define red 3
#define green 2

int to_in = 0;
int to_cm = 0;

long ultrasonik(int triggerPin, int echoPin)
{
  pinMode(triggerPin, OUTPUT); //Reset pin trigger
  digitalWrite(triggerPin, LOW);
  delayMicroseconds(2);
  // Trigger pin aktifkan high selama 10 microseconds
  digitalWrite(triggerPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(triggerPin, LOW);
  pinMode(echoPin, INPUT);
  return pulseIn(echoPin, HIGH);
  // Echo Pin membaca pantulan gelombang suara
}

void setup()
{
  Serial.begin(9600);
  pinMode(red, OUTPUT);
  pinMode(green, OUTPUT);
}

void loop()
{
  //Lakukan konversi gelombang ke cm
  to_cm = 0.01723 * ultrasonik(pin_ultra, pin_ultra);
  Serial.print(to_cm);
  Serial.println("cm");
  delay(100);
  if (to_cm < 50) {
    digitalWrite(red, HIGH); //merah nyala
    digitalWrite(green, LOW);
  } else {

```

Done compiling

Sketch uses 3896 bytes (1%) of program storage space. Maximum is 253952 bytes.
Global variables use 192 bytes (2%) of dynamic memory, leaving 8000 bytes for local variables. Maximum is 8192 bytes.

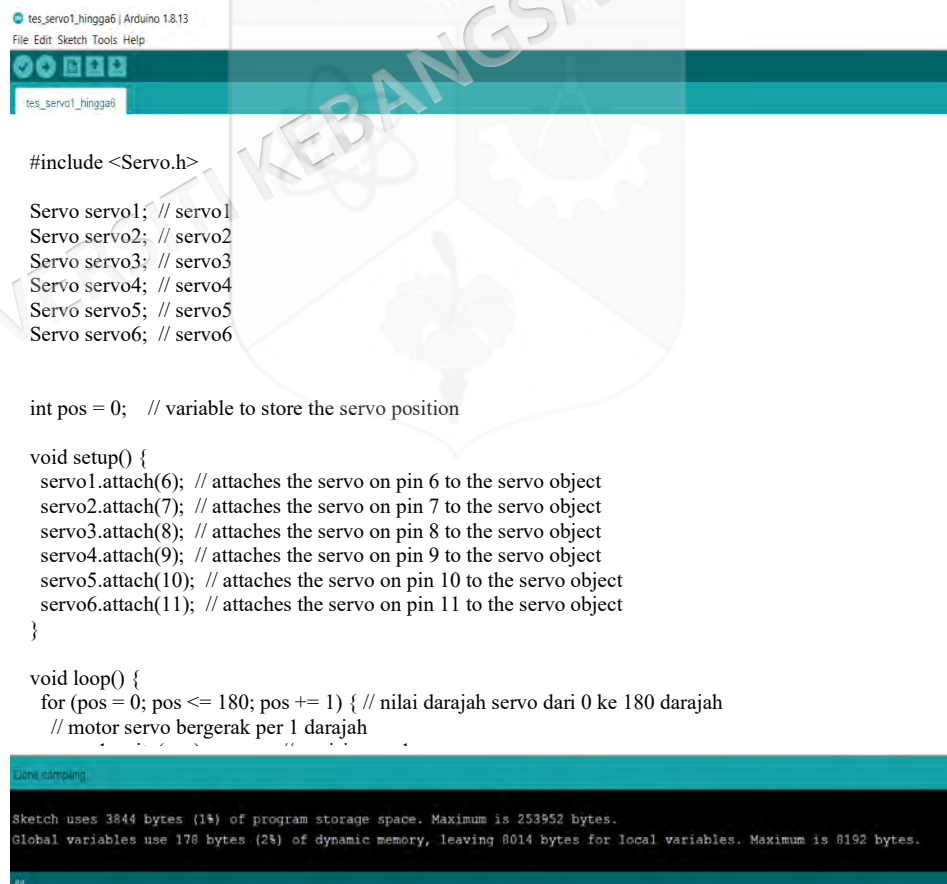
Rajah 3.20 Paparan modul program ujian penderia ultrasonik

c. Modul program lengan robot (*robot arm*)

Lengan robot hanya terdiri daripada 7 motor servo. Fungsi setiap motor servo dapat dilihat pada Jadual 3.2. Pengujian dijalankan secara berasingan iaitu pengujian motor servo 1 hingga 6. Diikuti dengan pengujian motor servo 7. Ini kerana kawasan pengujian motor servo 1 hingga 6 adalah berbeza daripada motor servo 7. Motor servo 1 hingga 6 telah diuji dengan sesaran 180 darjah. Manakala motor servo 7 dikhaskan untuk menggerakkan pencengkam lengan robot, supaya nilai kawasan ujian hanya 60 darjah.

i. Modul program ujian Motor Servo 1 hingga Motor Servo 6

Selepas litar motor servo 1 hingga motor servo 6 dipasang seperti yang ditunjukkan dalam Rajah 3.9, motor servo 1 hingga motor servo 6 diuji menggunakan modul program berikut.



```

tes_servo1_hingga6 | Arduino 1.8.13
File Edit Sketch Tools Help

tes_servo1_hingga6

#include <Servo.h>

Servo servo1; // servo1
Servo servo2; // servo2
Servo servo3; // servo3
Servo servo4; // servo4
Servo servo5; // servo5
Servo servo6; // servo6

int pos = 0; // variable to store the servo position

void setup() {
  servo1.attach(6); // attaches the servo on pin 6 to the servo object
  servo2.attach(7); // attaches the servo on pin 7 to the servo object
  servo3.attach(8); // attaches the servo on pin 8 to the servo object
  servo4.attach(9); // attaches the servo on pin 9 to the servo object
  servo5.attach(10); // attaches the servo on pin 10 to the servo object
  servo6.attach(11); // attaches the servo on pin 11 to the servo object
}

void loop() {
  for (pos = 0; pos <= 180; pos += 1) { // nilai darjah servo dari 0 ke 180 darjah
    // motor servo bergerak per 1 darjah
  }
}

Sketch uses 3844 bytes (1%) of program storage space. Maximum is 253952 bytes.
Global variables use 176 bytes (2%) of dynamic memory, leaving 8014 bytes for local variables. Maximum is 8192 bytes.

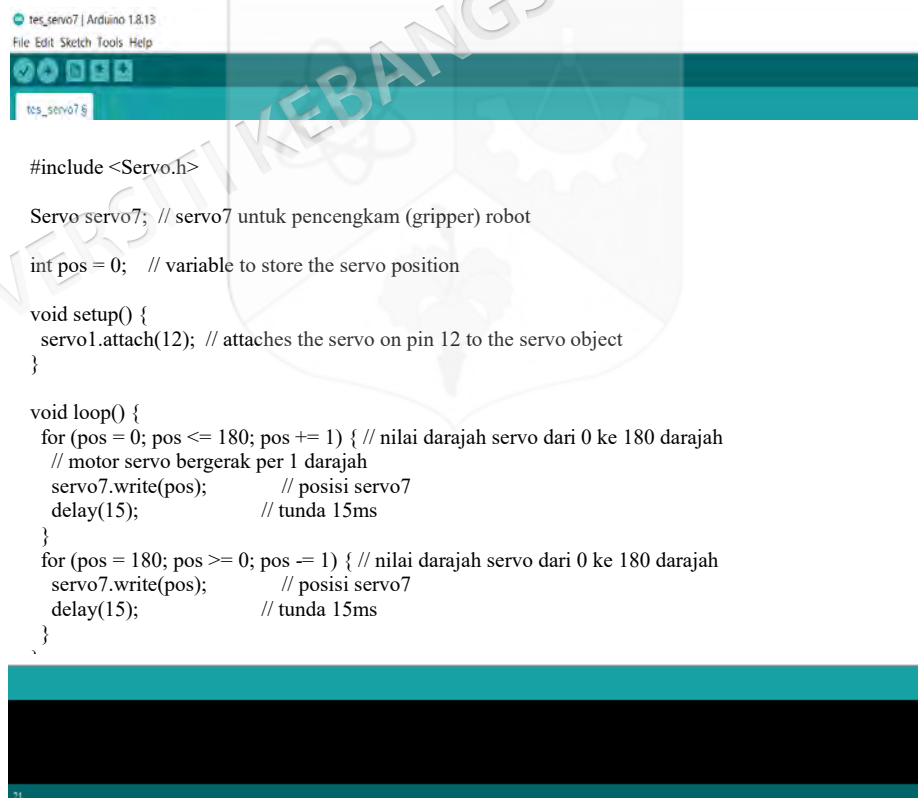
```

Rajah 3.21 Paparan modul program ujian motor servo 1 hingga motor servo 6

Dalam Rajah 3.21 terdapat program ujian motor servo menggunakan IDE Arduino. Jika modul program berjalan dengan baik, maka motor servo 1 ke motor servo 6 akan bergerak secara berurutan. Pada mulanya motor servo 1 bergerak dari 0 hingga 180 darjah. Selepas mencapai 180 darjah, motor servo 1 akan bergerak semula ke kedudukan 0 darjah. Kemudian motor servo 1 berhenti bergerak. Kemudian motor servo 2 mula bergerak dari 0 hingga 180 darjah. Selepas mencapai 180 darjah, motor servo 2 akan bergerak semula ke kedudukan 0 darjah. Kemudian motor servo 2 berhenti bergerak. Perkara yang sama akan digunakan untuk motor servo 3, 4, 5 dan 6.

ii. Modul program ujian Servo Motor 7

Selepas litar motor servo 7 dipasang seperti yang ditunjukkan dalam Rajah 3.8, motor servo 7 diuji menggunakan modul program berikut. Motor servo ini digunakan untuk memacu pencengkam robot. Ujian motor servo ini hanya dijalankan 60 darjah, mengikut nilai bukaan pencengkam (*gripper*).



```

tes_servo7 | Arduino 1.8.13
File Edit Sketch Tools Help

tes_servo7.g

#include <Servo.h>

Servo servo7; // servo7 untuk pencengkam (gripper) robot
int pos = 0; // variable to store the servo position

void setup() {
  servo1.attach(12); // attaches the servo on pin 12 to the servo object
}

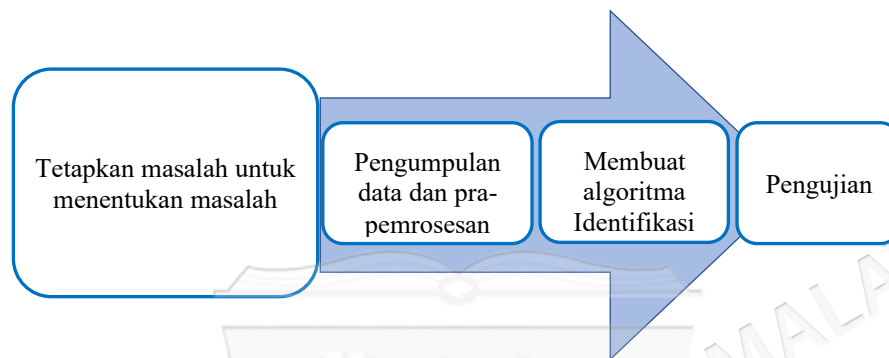
void loop() {
  for (pos = 0; pos <= 180; pos += 1) { // nilai darajah servo dari 0 ke 180 darajah
    // motor servo bergerak per 1 darajah
    servo7.write(pos); // posisi servo7
    delay(15); // tunda 15ms
  }
  for (pos = 180; pos >= 0; pos -= 1) { // nilai darajah servo dari 0 ke 180 darajah
    // posisi servo7
    servo7.write(pos); // posisi servo7
    delay(15); // tunda 15ms
  }
}

```

Rajah 3.22 Paparan modul program ujian motor servo 7

3.3 IDENTIFIKASI SAMPAH DI PERKARANGAN RUMAH

Langkah seterusnya ialah identifikasi sampah di perkarangan rumah. Pada penyelidikan ini, metodologi yang digunakan mengikut amalan terbaik pembangunan Pembelajaran Dalam (*deep learning*) dalam kajian terdahulu seperti diterangkan pada bahagian 1, dengan sedikit pelarasan. metodologi yang dicadangkan terdapat dalam Rajah 3.23.



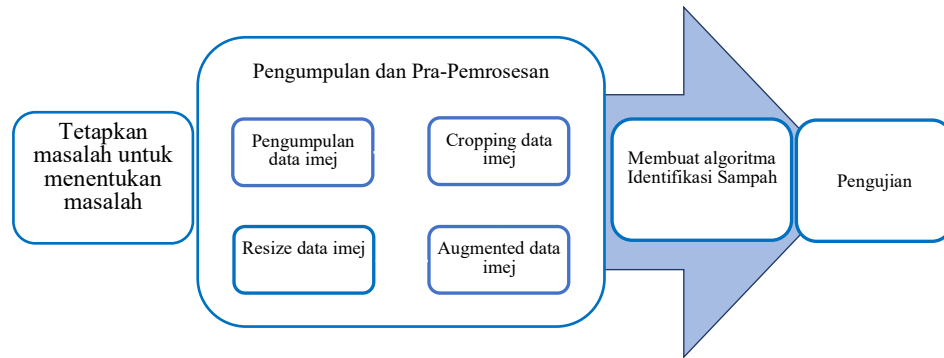
Rajah 3.23 Metodologi yang dicadangkan untuk identifikasi sampah

3.3.1 Penentuan permasalahan

Di awal kajian, masalah dan jurang daripada kajian lepas dianalisis. Masalah kajian perlu ditakrifkan dengan teliti untuk memantapkan permasalahan sebenar. Dalam fasa ini, keberhasilan identifikasi sampah ditentukan oleh pengumpulan data yang benar. Kesalahan dalam pengumpulan data boleh menyebabkan set data tidak dapat diproses dengan betul. Penerangan lanjut berkenaan dengan identifikasi sampah akan diterangkan pada seksyen 3.3.2.

3.3.2 Pengumpulan data dan pra-pemrosesan identifikasi sampah

Seksyen ini menerangkan pelaksanaan pengumpulan data dan pra-pemrosesan identifikasi sampah. Pada Rajah 3.24 menunjukkan beberapa langkah yang terlibat dalam tersebut. Terdapat empat (4) langkah yang dikenal pasti iaitu pengumpulan data, Cropping data, augmented data, resize data.

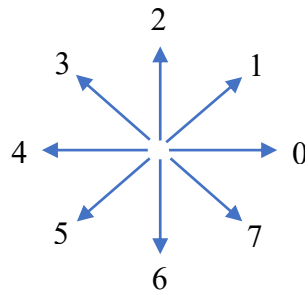


Rajah 3.24 Aliran dalam pengumpulan data dan pra-pemrosesan

a. Pengumpulan Data Imej

Pengumpulan data dijalankan menggunakan data sampah yang dikumpul sendiri. Data ini dikelompokkan kepada 11 jenis data iaitu, Kadbod, Fabrik, Pembungkus Makanan, Buah, Kaca, Daun, Logam, Kertas, Plastik, Getah, dan Kayu. Prosedur untuk mendapatkan set data sendiri dijalankan dengan cara berikut :

1. Terdapat 11 jenis jenis sampah iaitu ; Kadbod, Fabrik, Pembungkusan Makanan, Buah, Kaca, Daun, Logam, Kertas, Plastik, Getah, dan Kayu.
2. Setiap jenis terdiri daripada 10 keping sampah. Setiap jenis sampah disimpan dalam beg plastik yang berbeza. Jumlah sampah adalah 110 keping.
3. Terdapat 4 latar belakang untuk mengambil gambar objek sampah iaitu; Lantai simen, Asfalt, Tanah dan Rumput.
4. Ketinggian kamera dari lantai (simen, asfalt, tanah, rumput) ialah 25 cm. Kedudukan kamera dibengkokkan pada sudut 15 darjah.
5. Mengambil gambar dilakukan dalam 8 arah kompas seperti yang ditunjukkan pada Rajah 3.25.



Rajah 3.25 Lapan arah pengambilan gambar

6. Terdapat 3 jenis jarak kamera dengan sampah iaitu:
 - Jarak antara 30 cm hingga 100 cm
 - Jarak antara 101 cm hingga 150 cm
 - Jarak antara 150 cm hingga 200 cm
7. Selepas prosedur 1 hingga 6 dilakukan, di lokasi yang sama, putarkan sampah mengikut penentuan angin. Sebagai contoh, jika pada mulanya sampah diletakkan ke arah nombor 0, kemudian putarkannya ke arah nombor 1. Ambil foto sekali lagi mengikut arahan dalam mata 2 hingga 6.
8. Berdasarkan perkara 1 hingga 8, jumlah foto sampah yang telah diperolehi ialah:
 - Jenis sampah: 11
 - Jumlah sampah: 10
 - Latar belakang tempat gambar diambil: 4
 - Kedudukan kamera dalam mengambil gambar: 8
 - Jarak menembak: 3
$$\begin{aligned} \text{Jumlah set data sendiri} &= 11 \times 10 \times 4 \times 8 \times 3 \\ &= 10,560 \text{ gambar} \end{aligned}$$
9. Oleh itu setiap jenis sampah mempunyai 960 gambar.

b. Pemangkasan Data Imej

Pemangkasan data imej dilakukan untuk menambah bilangan set data. Ini dilakukan dengan memotong imej kepada beberapa imej. Bilangan imej yang dipangkas boleh berbeza-beza bergantung pada perkara yang dimahukan. Dalam kajian ini pemangkasan

dihadkan kepada maksimum 8 imej. Rajah 3.26 berikut ialah contoh hasil pemangkasan daripada input imej.



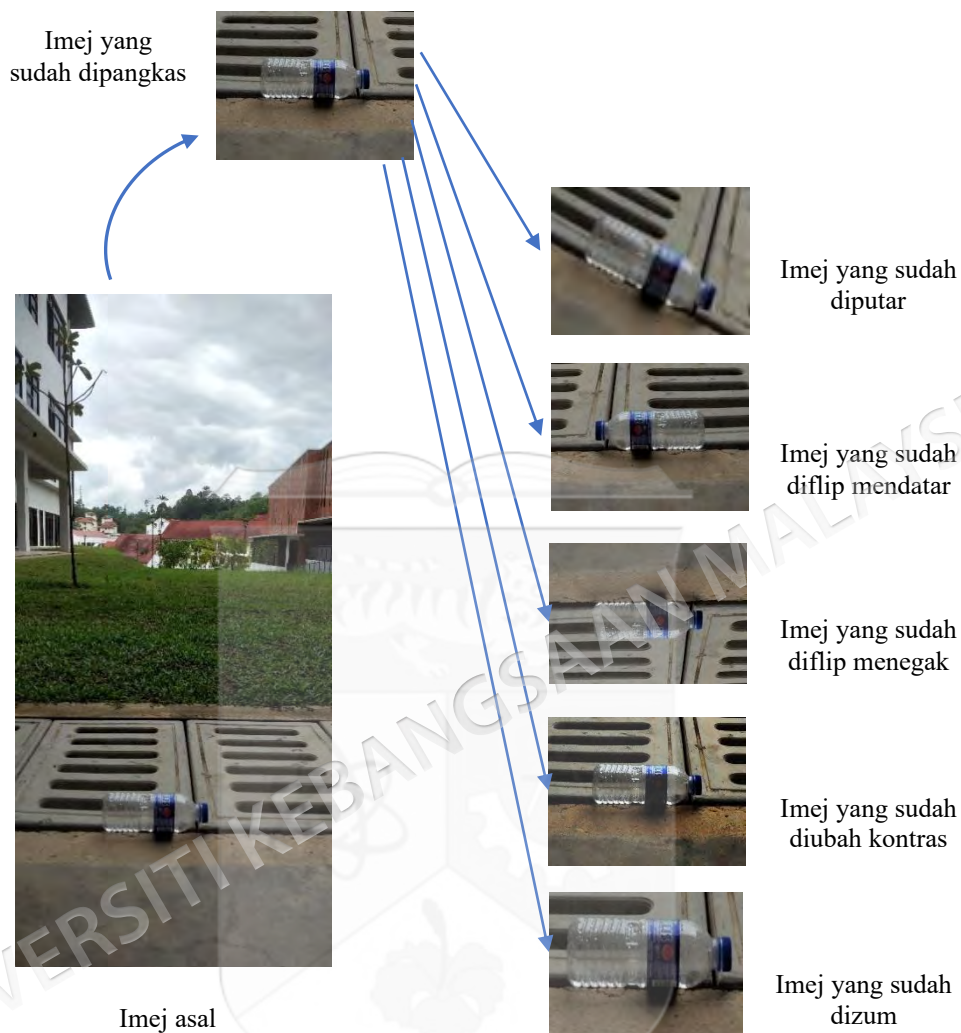
Rajah 3.26 Lapan hasil imej yang dipangkas

Dalam Rajah 3.26 di atas, dapat dilihat bahawa dalam proses pemangkasan imej, objek yang ingin dikenal pasti dipotong dari pelbagai sisi. Ini bertujuan untuk meningkatkan bilangan imej dalam set data. Lebih banyak imej, diharap proses pengenalan akan mendapat hasil yang lebih baik.

c. Imej data tambahan

Pada asasnya pemangkasan imej disertakan dalam data imej tambahan. Walau bagaimanapun, apa yang dimaksudkan di sini ialah proses menambah bilangan set data dengan cara; putar imej, flip imej secara mendatar, flip imej secara menegak, tukar nilai

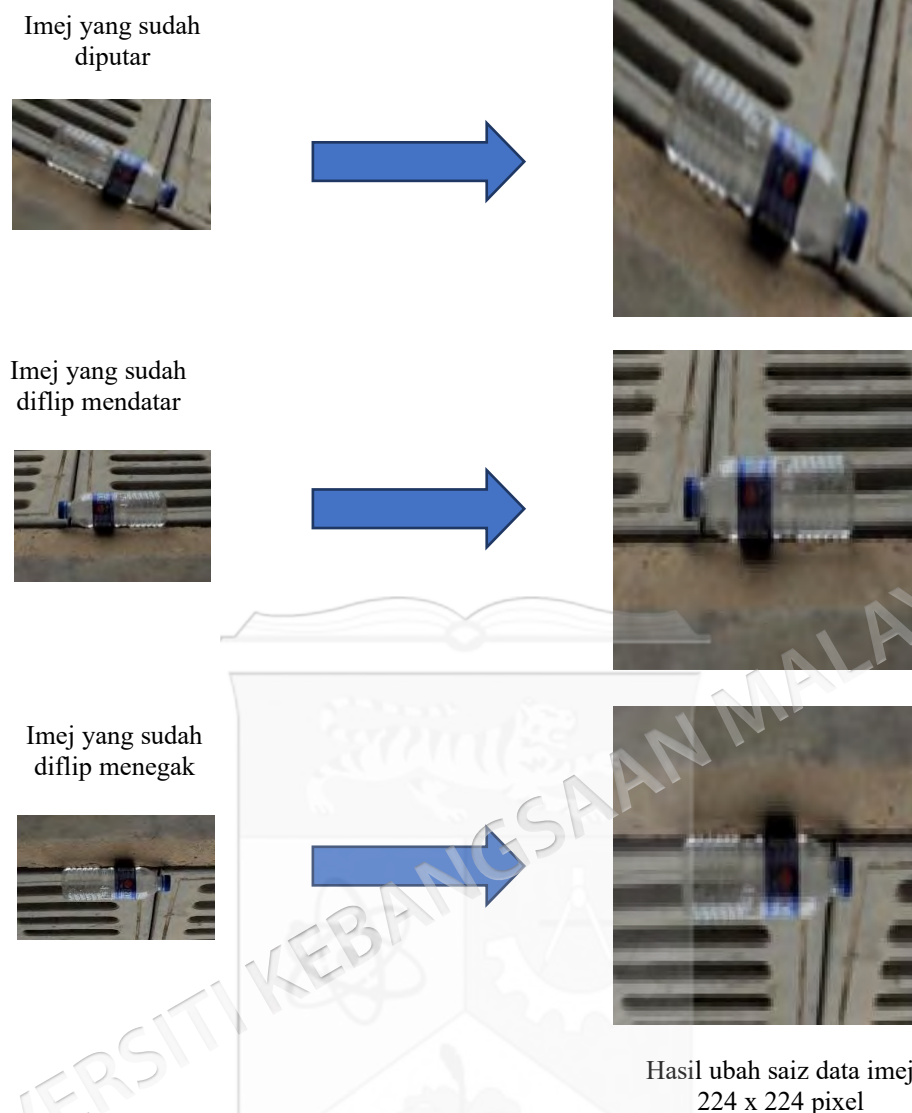
kontras imej dan zum imej. Rajah 3.27 berikut ialah contoh hasil imej tambahan daripada input imej.



Rajah 3.27 Hasil imej tambahan (*augmented image*)

d. Ubah saiz Data Imej

Saiz semula data imej berguna untuk melaraskan saiz imej kepada imej input seni bina Rangkaian Saraf Konvolusi (CNN) tertentu. Seni bina CNN AlexNet dan GoogleNet memerlukan saiz imej 224 x 224 piksel. Sementara itu, seni bina CNN VGG16 dan ResNet18 memerlukan saiz imej 227 x 227 piksel. Rajah 3.28 berikut ialah contoh hasil ubah saiz daripada input imej.

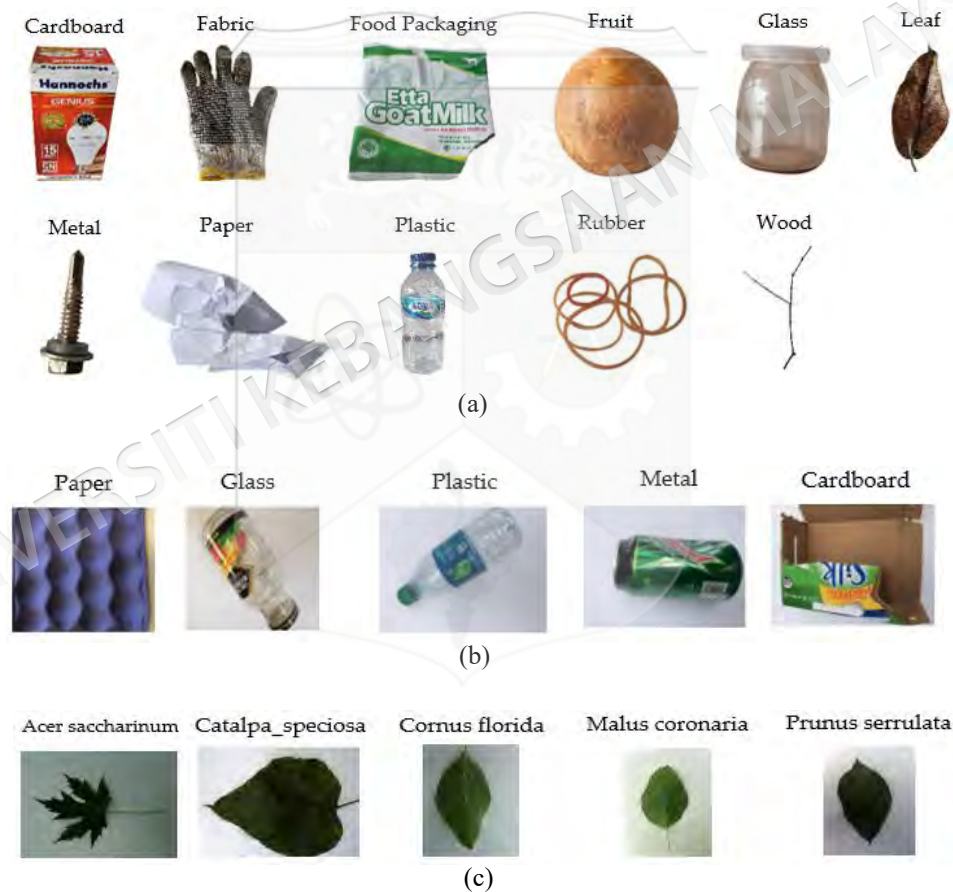


Rajah 3.28 Hasil ubah saiz imej

3.3.3 Pembangunan algoritma identifikasi sampah

Dalam kajian ini, pengumpulan data dilakukan menggunakan dataset sampah sendiri, ditambah dataset TrashNet (M. Yang & Thung 2016), dan dataset Leafsnap (Barré et al. 2017). Ini kerana dataset sampah sendiri tidak mencukupi untuk digunakan. imej dalam dataset TrashNet dan dataset Leafsnap dipilih secara manual. Imej yang diambil adalah imej yang sesuai untuk sampah perkarangan rumah. Data ini dikelompokkan kepada 11 jenis data iaitu, kadbod, kain, pembungkus makanan, buah, kaca, daun, logam, kertas, plastik, getah, dan kayu. Pengelompokan ini berdasarkan data sampah yang boleh diangkat oleh pencengkam robot. Objek sampah yang boleh diangkat oleh

pencengkam robot adalah objek dalam bentuk 3D dengan ketinggian minimum 6 milimeter. Rajah 3.29 menunjukkan imej sampel set data sampah. Dataset *TrashNet* telah dicipta oleh Gary Thung dan Mindy Yang. Ini adalah set data kecil dan mengandungi 1989 imej. Dataset *LeafSnap* telah dicipta oleh Kumar et al. (Kumar et al. 2012). Set data terdiri daripada 7500 imej yang ditambahkan (Adetiba et al. 2021). Set data sampah sendiri telah ditangkap oleh peranti mudah alih (iPhone). Data imej ditambah dengan komputer riba menggunakan perisian MATLAB R2020a. Set data sendiri ini digunakan untuk melengkapkan set data *TrashNet* dan set data *LeafSnap* untuk mendapatkan 11 jenis data.



Rajah 3.29 (a) Contoh imej set data sampah sendiri. (b) Imej contoh set data *TrashNet*. (c) Imej contoh set data *LeafSnap*

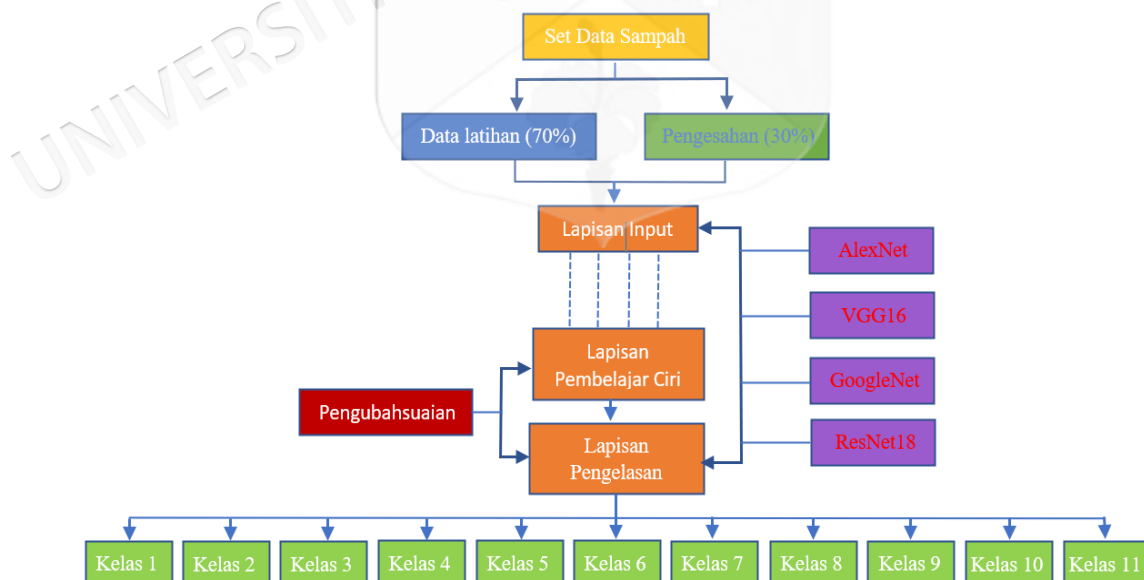
Data telah diuji dengan empat seni bina Rangkaian Neural Convolutional (CNN) pra-terlatih: AlexNet, VGG16, GoogleNet dan ResNet18. Parameter latihan boleh dilihat dalam Jadual 3.3.

Jadual 3.3 Parameter Latihan Digunakan untuk Identifikasi Sampah

Parameter	AlexNet	VGG16	GoogLeNet	Resnet18
Saiz kumpulan mini	5	5	5	5
Kadar pembelajaran	0.0003	0.0003	0.0003	0.0003
Epok maksimum	6	6	6	6
Data diimbuhkan	Ya	Ya	Ya	Ya

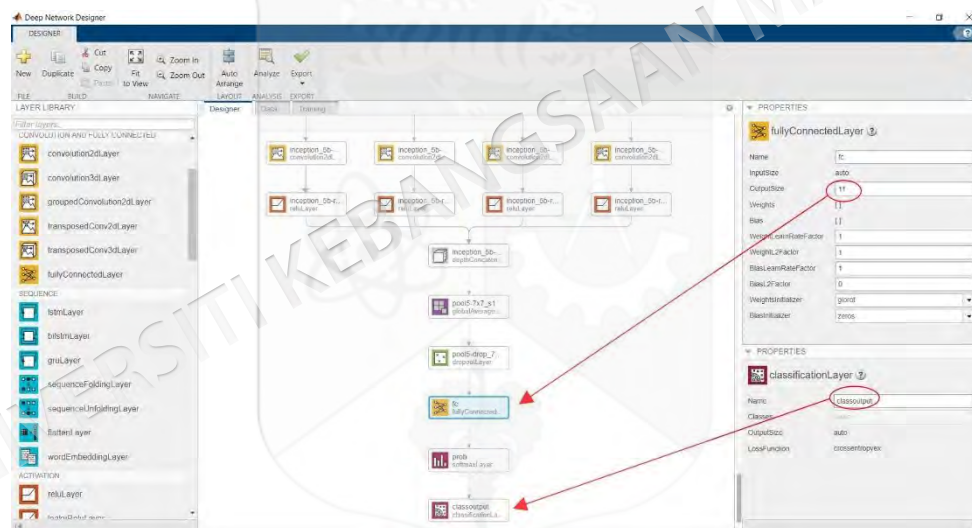
Parameter latihan yang digunakan adalah yang mempunyai nilai piawai yang telah sedia ada dalam seni bina CNN pra-latihan. Selepas data dikumpul, langkah seterusnya ialah menangkap objek sampah.

Selepas kamera IP menangkap data imej, program MATLAB mengubah saiz imej. Ia berguna untuk pemprosesan lanjut mengenai Pengesanan Imej CNN. Seni bina CNN yang digunakan boleh mengklasifikasikan sehingga 1000 jenis objek. Walau bagaimanapun, pengubahsuaian dibuat kepada pelajar ciri untuk mempercepatkan pengelasan objek sampah. Nilai asal ialah 1000 lapisan bersambung sepenuhnya ditukar kepada 11 lapisan bersambung sepenuhnya. Nilai ini ditukar pada setiap seni bina AlexNet, VGG16, GoogLeNet dan ResNet18. Rajah 3.30 menggambarkan pembelajaran pemindahan menggunakan seni bina CNN.

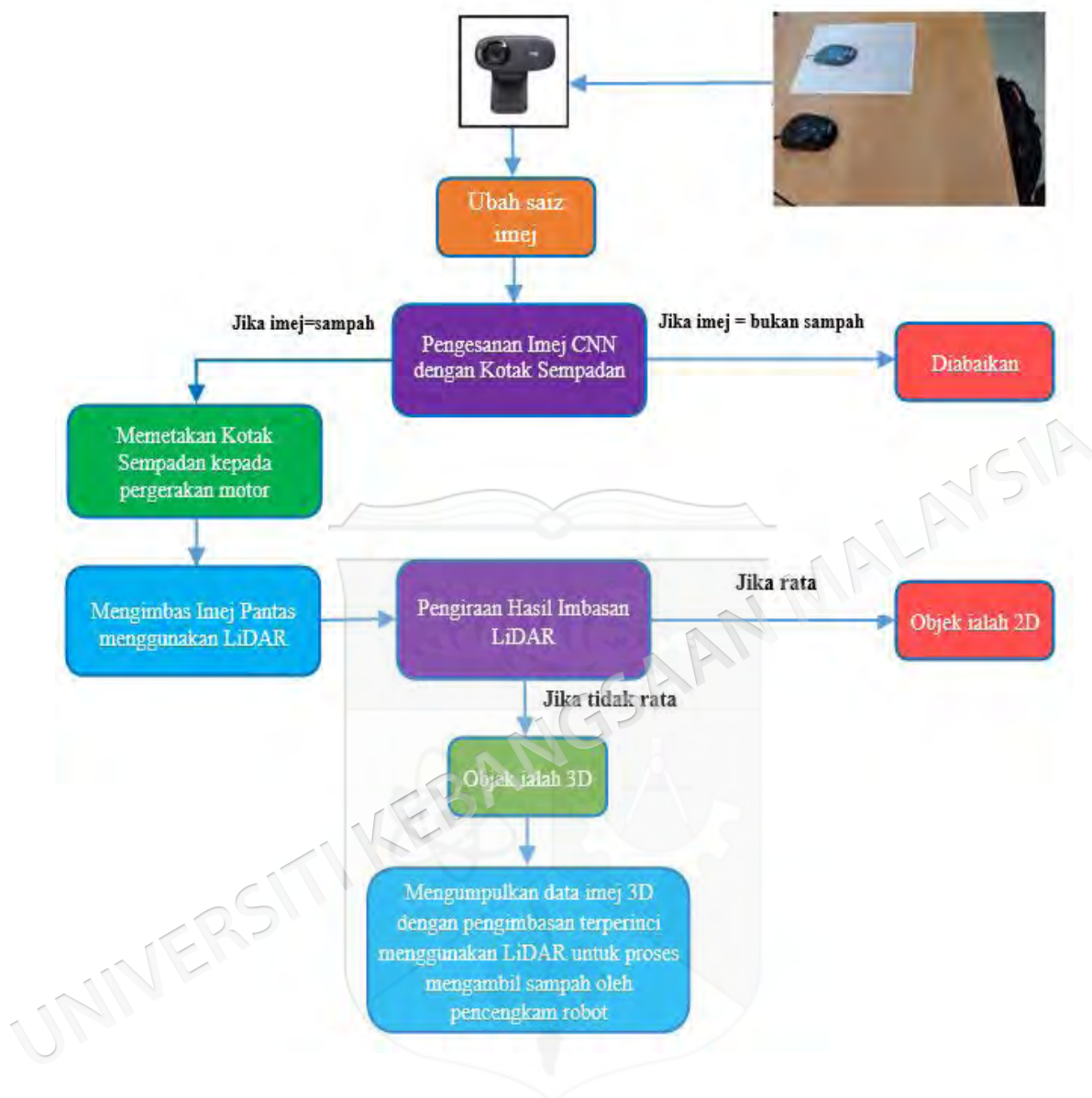


Rajah 3.30 Pembelajaran pemindahan mudah menggunakan seni bina CNN (AlexNet, VGG16, GoogLeNet dan ResNet18)

Dalam MATLAB, proses pengubahsuaian boleh dilakukan dengan menukar reka bentuk seni bina menggunakan Deep Network Designer. Rajah 3.31 menunjukkan proses mengubah suai seni bina GoogleNet.

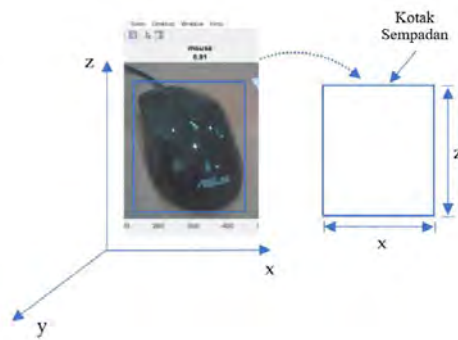


Tambahan pula, program MATLAB memastikan bahawa imej adalah objek sampah. Jika objek itu bukan sampah, ia diabaikan. Jika ia adalah sampah, komputer riba (PC) menghantar arahan kepada Arduino untuk mengaktifkan penggerak (servo1 dan servo2). Perintah ini adalah berdasarkan nilai kotak sempadan objek yang dikesan. Penggerak menggerakkan penderia LiDAR dalam koordinat X dan Z berdasarkan nilai. Selepas itu, data daripada penderia LiDAR dihantar oleh Arduino ke komputer riba (PC). Carta alir yang dicadangkan boleh dilihat dalam Rajah 3.32. Carta alir ini dinamakan SCL (*Sequential Camera LiDAR*).



Rajah 3.32 Carta alir kajian yang dicadangkan, SCL (*Sequential Camera LiDAR*)

Mapping bounding box digunakan untuk menukar pergerakan motor servo agar sepadan dengan saiz objek yang ingin dikesan, seperti yang ditunjukkan dalam Rajah 3.33. Selepas mendapat x dan z , ia dikira/dipetakan kepada tahap pergerakan motor servo. Dalam proses pengimbasan, motor servo 1 (x) dan motor servo 2 (z) bergerak pada sempadan dalam kotak sempadan.



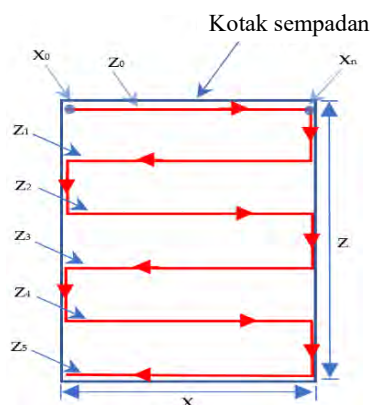
Rajah 3.33 Menukar kotak sempadan kepada koordinat x, y, dan z

Dalam sistem yang direka bentuk, penderia LiDAR dipasang di atas motor servo

1. Kemudian proses pengimbasan dijalankan menggunakan penderia LiDAR. Proses imbasan dijalankan dalam 2 peringkat iaitu proses imbasan yang pantas dan terperinci. Ini adalah untuk mencapai masa pengimbasan yang cekap. Jika objek yang dikesan adalah rata, maka objek tersebut adalah objek 2D. Jika ia 2D, proses pengimbasan terperinci tidak diperlukan. Walau bagaimanapun, jika objek tidak rata, objek yang dikesan adalah objek 3D. Jika objek adalah 3D, pemprosesan terperinci perlu dilakukan. Ini bertujuan untuk mengetahui dengan pasti bentuk geometri objek tersebut. Dengan data geometri objek, robot penggenggam boleh dengan mudah mengambil objek sampah.

3.3.4 Imbas imej pantas menggunakan LiDAR

Proses pengimbasan pantas, dari kiri atas ke kanan bawah. Proses pengimbasan pantas dijalankan sebanyak lima kali mengikut laluan dalam Rajah 3.34.



Rajah 3.34 Laluan proses imbasan imej pantas

Ketinggian kotak pembatas (z) mungkin berbeza bergantung pada saiz objek sampah yang dikesan. Walau bagaimanapun, nilai darjah pergerakan motor servo 2 (z) dalam sistem imbasan pantas ini diperoleh daripada ketinggian kotak sempadan dibahagikan dengan lima, seperti yang ditunjukkan dalam formula ...(3.1) berikut.

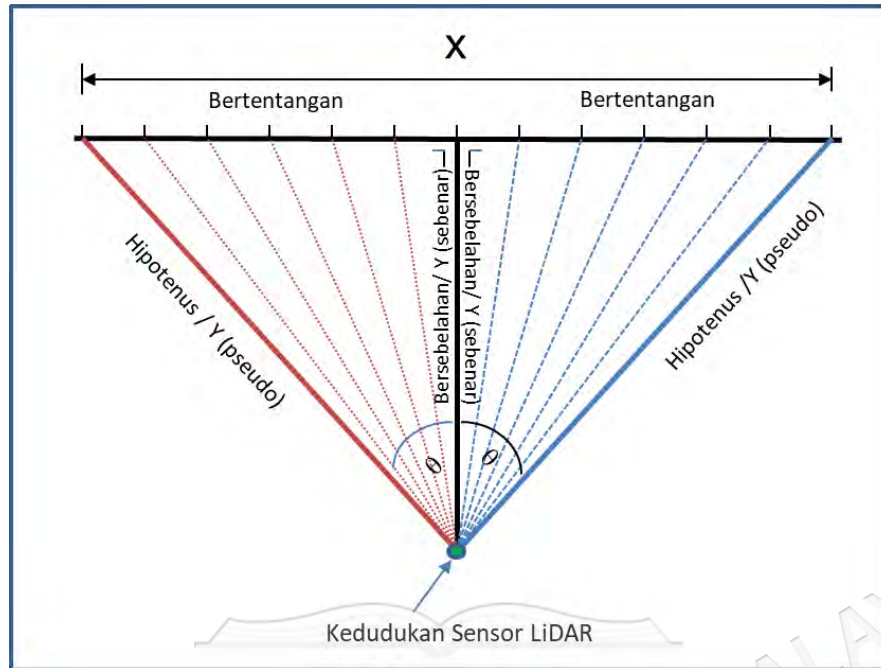
$$\text{Servo motor 2 (z)darjah} = \frac{\text{Tinggi kotak sempadan}}{5} \quad \dots(3.1)$$

Selepas mendapat nilai x dan z berdasarkan kotak sempadan, ia dikira (dipetakan) kepada tahap pergerakan motor servo. Titik x_0 garis z_0 dalam Rajah 3.34 ialah titik permulaan untuk proses pengimbasan imej pantas. Jika kedudukan motor servo 1 dan motor servo 2 tidak berada pada titik itu, maka program yang terkandung dalam Arduino akan mengalihkannya ke kedudukan tersebut.

Selepas itu, motor servo 1 (x) dan motor servo 2 (z) bergerak seperti yang ditunjukkan dalam Rajah 3.34 pada garisan z_0 . Dalam keadaan awal, motor servo akan bergerak dari titik x_0 garis z_0 ke titik x_n garis z_0 . Kemudian motor servo akan bergerak ke bawah sepanjang motor servo 2 (z) darjah. Contohnya ketinggian z ialah 100 darjah, maka motor servo 2 (z) akan bergerak ke bawah 20 darjah ($100 \text{ darjah} / 5 = 20 \text{ darjah}$). Langkah seterusnya ialah motor servo 1 (x) akan bergerak sepanjang z_1 ke arah pergerakan bertentangan dengan garisan z_0 . Pergerakan motor servo diteruskan dengan mengikut garisan z_1, z_2, z_3 , dan z_4 dan tiba di garisan z_5 pada titik x_0 .

Dalam Rajah 3.35 menunjukkan kedudukan dan pergerakan penderia LiDAR semasa proses pengimbasan. Algoritma berikut menentukan sama ada objek yang dikesan mempunyai sisi rata, cekung atau cembung.

- Jika nilai_penderia = hipotenus, maka garis/titik pada imej ialah satah rata;
- Jika nilai_penderia > hipotenus, maka garis/titik dalam imej menjadi cekung;
- Jika nilai_penderia < hipotenus, maka garis/titik dalam imej menjadi cembung.



Rajah 3.35 Kedudukan penderia LiDAR semasa proses imbasan

Berdasarkan Rajah 3.35 tersebut, terdapat tiga rumus yang digunakan iaitu:

Jika objek lurus di hadapan penderia LiDAR, maka koordinat-y:

$$y = \text{nilai jarak yang diukur oleh penderia LiDAR} \quad \dots(3.2)$$

Jika objek berada di sebelah kiri hadapan/sebelah kanan hadapan penderia LiDAR, nilai koordinat-y boleh dikira dengan formula $\dots(3.3)$.

$$y = \cos \theta \times \text{jarak diukur oleh penderia LiDAR} \quad \dots(3.3)$$

Formula boleh mengira nilai koordinat x:

$$x = \text{Darjah motor servo1} - 90 \quad \dots(3.4)$$

Nilai 90 adalah disebabkan oleh motor servo 1 ditetapkan pada 90 darjah. Selepas semua nilai x dibaca dan disimpan dalam pembolehubah matriks, maka nilai x ditambah kepada nilai x maksimum dengan formula:

$$x = x + \text{nilai maksimum } x \quad \dots(3.5)$$

Formula ini bertujuan untuk menjadikan nilai koordinat-x semuanya positif.

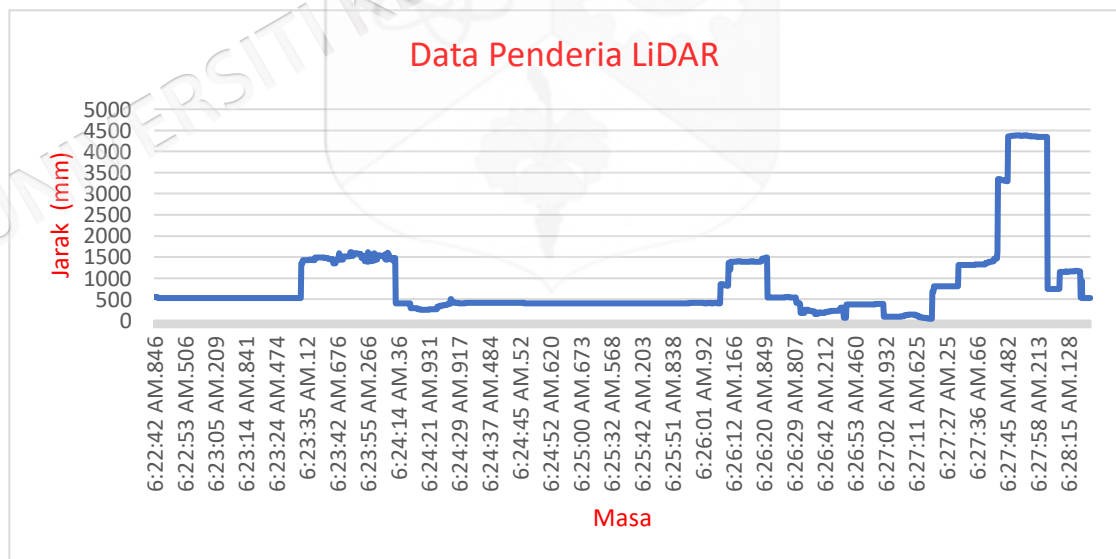
Selepas nilai koordinat-x dan koordinat-y diperoleh, maka nilai-z diperoleh dengan formula:

$$z = \text{Darjah Motor servo2} - 90 \quad \dots(3.6)$$

Selepas semua nilai z dibaca dan disimpan dalam pembolehubah matriks, maka nilai z ditambah kepada nilai z maksimum dengan formula:

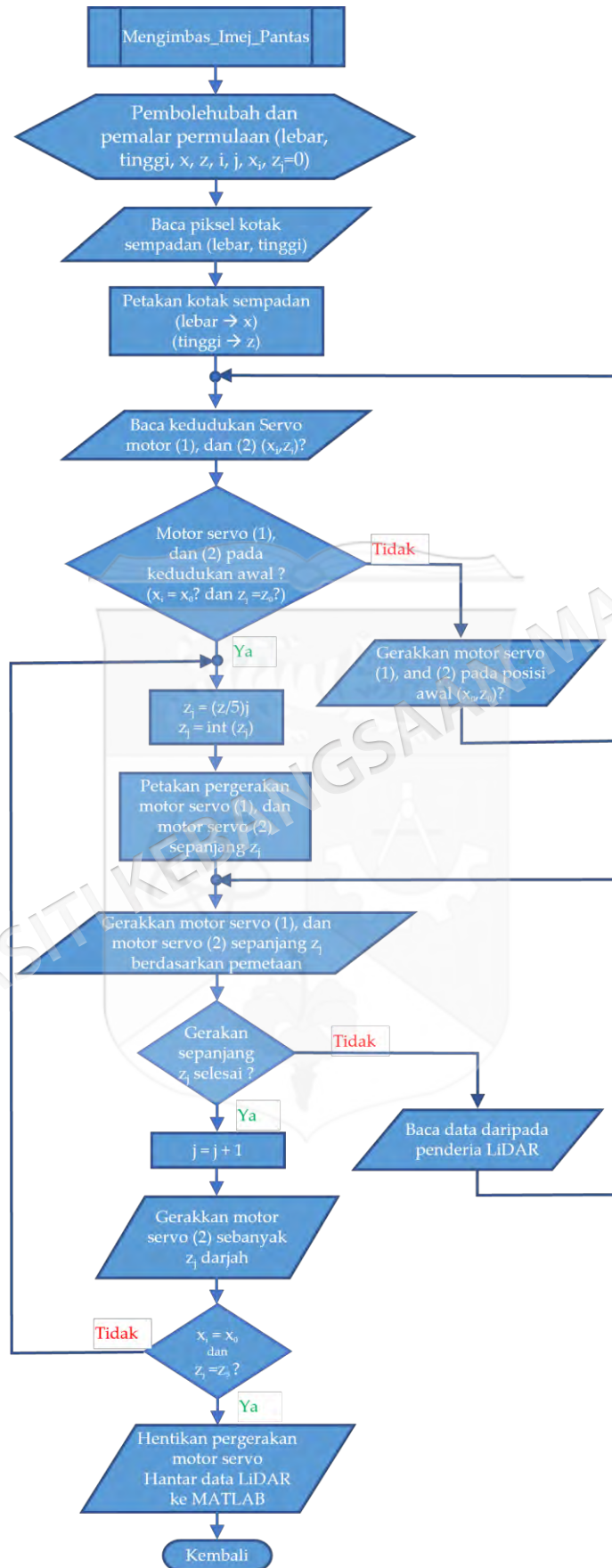
$$z = z + \text{nilai maksimum } z \quad \dots(3.7)$$

Dalam proses pengimbasan pantas, data yang dibaca ialah data masa (*Time Stamp*) dan data jarak (*Distance*). Rajah 3.36 ialah contoh graf bacaan penderia LiDAR TF40.



Rajah 3.36 Contoh bacaan penderia LiDAR

Rajah 3.37 ialah carta alir bagi imej imbasan pantas. Carta alir ini ialah prosedur yang dipanggil selepas program MATLAB mengesan objek sampah.



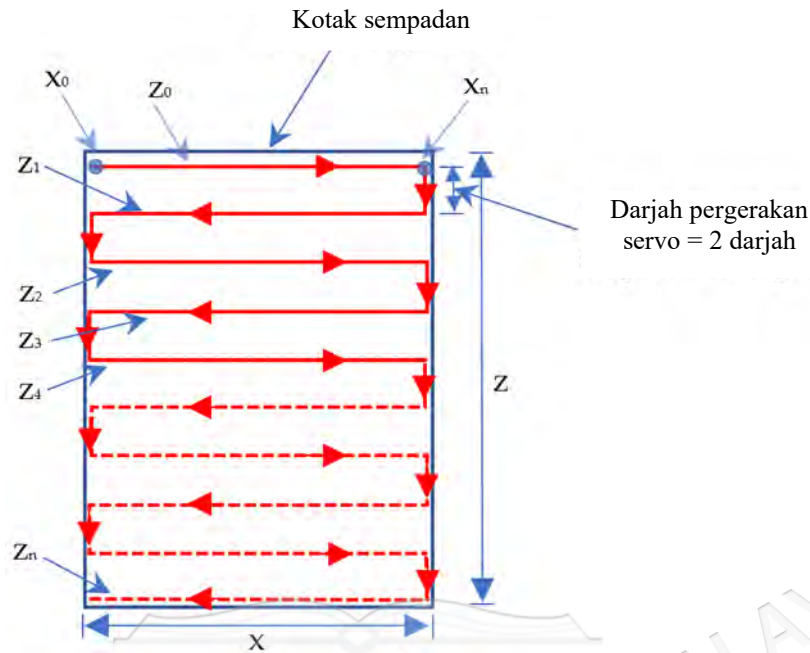
Rajah 3.37 Carta alir imej pengimbasan pantas

Carta alir lukisan imbasan pantas ini digunakan dalam Arduino. Pada peringkat awal, pembolehubah dan pemalar yang akan digunakan dimulakan. Seterusnya, program akan membaca nilai piksel kotak sempadan (lebar, tinggi). Setelah nilai ini diketahui, pemetaan kotak sempadan dilakukan. Hasil pemetaan ini adalah dalam bentuk nilai x dan z yang merupakan nilai asas bagi darjah pergerakan motor servo 1 dan motor servo 2. Langkah seterusnya ialah membaca kedudukan motor servo 1 dan motor servo 2. Jika kedudukan tidak berada pada kedudukan awal, maka motor servo 1 dan motor servo 2 akan dialihkan ke kedudukan awal tersebut. Kemudian nilai z_j dibahagikan dengan 5. Jika nilai z_j ialah nombor pecahan, maka nilai ini akan dibundarkan dengan perintah $\text{int}(z_j)$. Motor servo hanya boleh bergerak dalam darjah *integer* bukan darjah titik terapung. Langkah seterusnya ialah memetakan pergerakan motor servo 1 dan motor servo 2.

Motor servo menggunakan pemetaan ini untuk pergi dari titik x_0 garis z_0 ke titik x_0 garis z_5 . Serentak dengan pergerakan motor servo di sepanjang laluan z_0, z_1, z_2, z_3, z_4 , dan z_5 , data penerima LiDAR juga dibaca dan disimpan dalam memori Arduino. Selepas motor servo 1 dan motor servo 2 tiba di titik x_0 garis z_5 , data penerima LiDAR dihantar ke program MATLAB. Dalam MATLAB, data penerima LiDAR diproses. Hasilnya boleh 2D atau 3D. Dengan carta alir *Sequential Camera LiDAR (SCL)* yang dicadangkan dalam Rajah 3.32, jika ia adalah 2D, ia diabaikan. Jika ia dalam 3D, maka langkah seterusnya ialah mengimbas imej terperinci.

3.3.5 Imbas imej terperinci menggunakan LiDAR

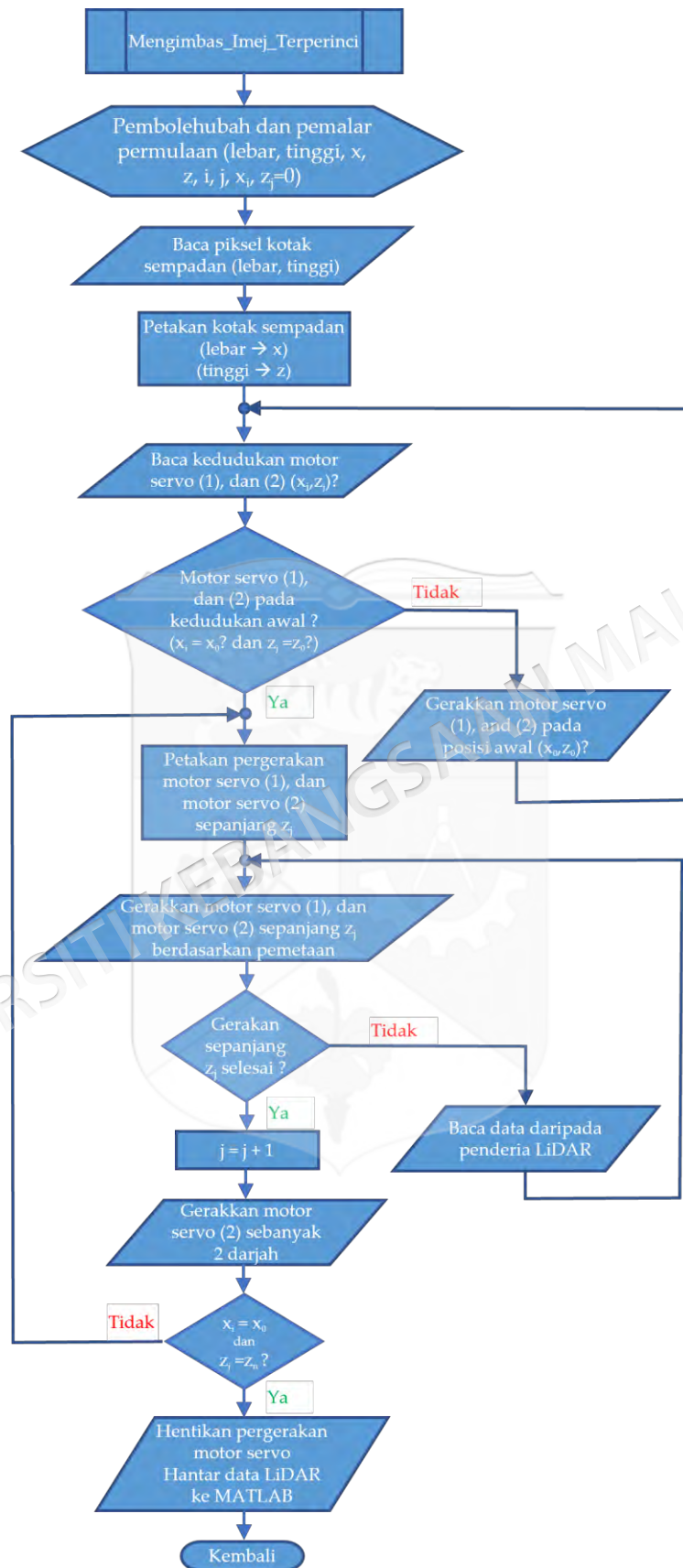
Pengimbasan terperinci sangat membantu untuk proses mengambil objek menggunakan robot penggenggam. Jika pencengkam tidak diletakkan dengan betul, proses mengangkat mungkin gagal. Proses pengimbasan hampir sama dengan pengimbasan pantas, tetapi tahap pergerakan motor servo telah ditentukan dari awal, iaitu 2 darjah. Butirannya boleh dilihat dalam Rajah 3.38.



Rajah 3.38 Laluan proses imbasan imej terperinci

Perbezaan laluan proses antara imej imbasan pantas dan imej imbasan terperinci hanya terletak pada bilangan baris (z). Dalam imej imbasan pantas, bilangan baris dihadkan kepada 5 baris. Sementara itu, bilangan garisan minimum untuk imej imbasan terperinci ialah lima. Sekiranya jarak antara baris dihadkan kepada 2 darjah, maka tahap pergerakan motor servo 2 adalah sekurang-kurangnya 10 darjah.

Rajah 3.39 ialah carta alir imej pengimbasan terperinci. Carta alir ini ialah prosedur yang dipanggil selepas program MATLAB mengesan objek sampah. Carta alir ini sama dengan carta alir dalam Rajah 3.37. Bezanya proses imbasan akan berhenti jika $x_i = x_0$, dan $z_j = z_n$. Perbezaan lain ialah tahap pergerakan motor servo 2 dalam pengimbasan pantas ditentukan oleh nilai z dibahagikan dengan 5. Sementara itu, dalam pengimbasan imej terperinci, tahap pergerakan adalah sama dengan 2 darjah.



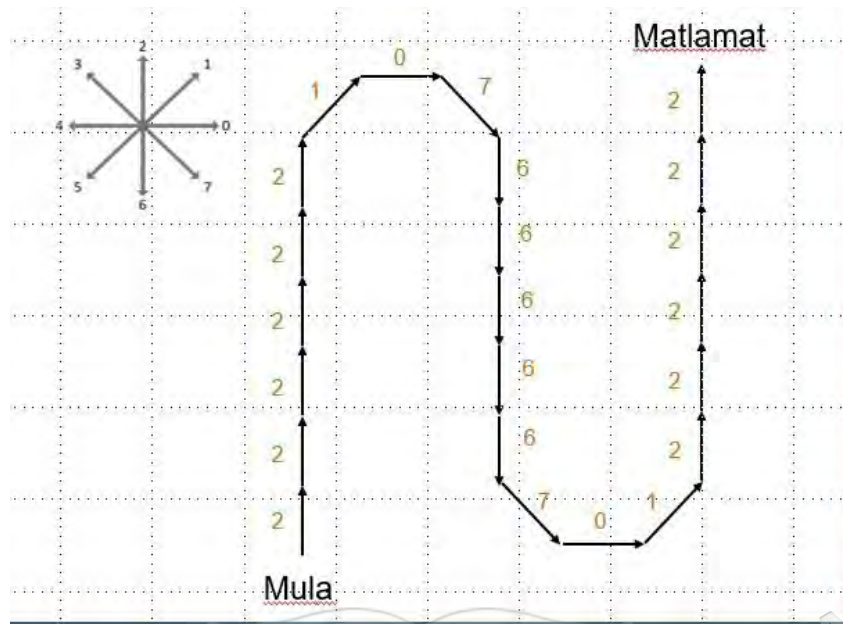
Rajah 3.39 Carta alir mengimbas imej terperinci

Dengan ketepatan LiDAR ± 2 mm, hasil imej pengimbasan terperinci yang dipaparkan menggunakan MATLAB adalah serupa dengan objek asal. Pada mulanya, data yang dihasilkan oleh penerima LiDAR masih dalam bentuk nombor yang mengandungi data koordinat y selepas menggabungkan data dengan koordinat x (motor servo 1) dan koordinat z (motor servo 2). Walau bagaimanapun, kerana apa yang diimbas hanyalah permukaan di hadapan penerima LiDAR, imbasan terperinci hanya menunjukkan permukaan sahaja. Namun ini boleh digunakan sebagai patokan untuk menentukan posisi daripada pencengkam robot dalam membuang sampah.

3.4 NAVIGASI ROBOT AUTONOMI DI PERKARANGAN RUMAH

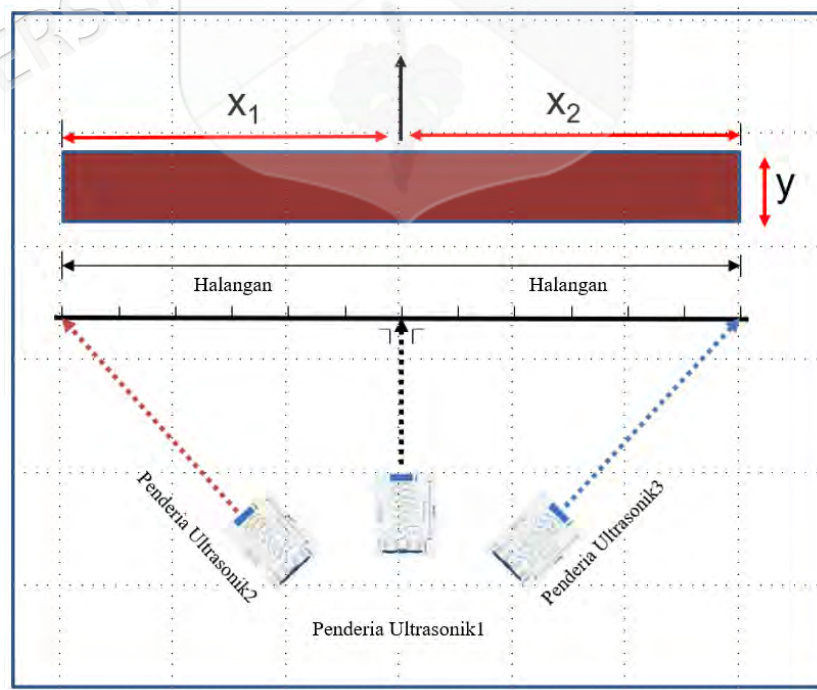
Terdapat banyak kaedah dalam navigasi robot seperti kaedah navigasi menggunakan odometri (pengekod). Kaedah lain ialah kaedah navigasi berasaskan imej. Setiap kaedah mempunyai kelemahannya sendiri. Dalam kaedah navigasi menggunakan odometri, kelemahannya ialah ia tidak dapat menentukan kedudukan mutlak robot tetapi hanya boleh menentukan kedudukan relatif robot berdasarkan pengiraan putaran roda robot. Begitu juga dengan kaedah navigasi berasaskan imej, kelemahannya ialah pemprosesan imej boleh diganggu oleh kesan perubahan cahaya.

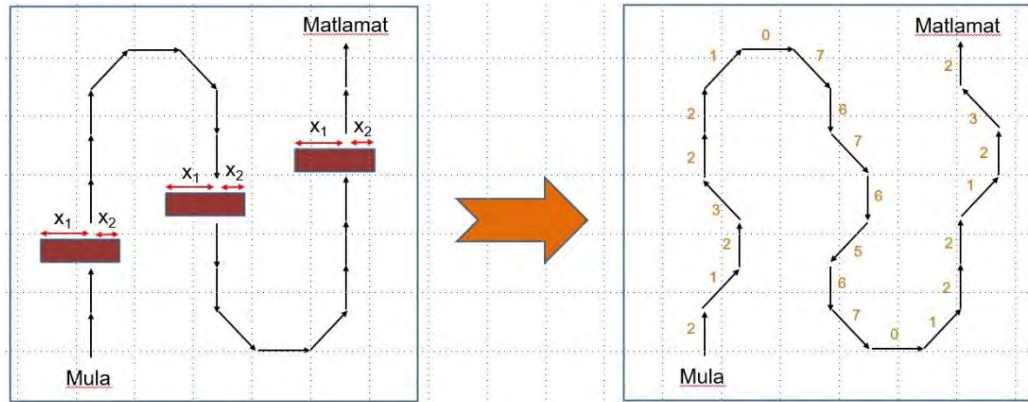
Penyelidikan ini mencadangkan kaedah yang menggabungkan imejan dan odometri menggunakan Kod Rantaian Freeman. Imej dikesan menggunakan kamera, manakala navigasi dicadangkan kaedah *Modified Freeman Chain Code Path Planning* (MFCC-PP). Kaedah ini mengubah suai bilangan kod rantai Freeman yang pada asalnya hanya maksimum 8 kod, diubah suai kepada 16 dan 32 kod. Gambar rajah blok sistem navigasi ditunjukkan dalam Rajah 3.3. Dalam gambar rajah blok ini dapat dilihat bahawa robot mengemudi berdasarkan input kamera, penerima LiDAR, penerima ultrasonik, penerima pengekod dan penerima kompas. Robot mempunyai kod rantai yang telah disimpan dalam ingatan pengawal. Dalam keadaan biasa, robot bergerak dari awal ke matlamat berdasarkan kod dengan corak kod rantai Freeman yang disimpan dalam ingatan. Rajah 3.40 menunjukkan laluan biasa pergerakan robot menggunakan 8 kod rantai Freeman.



Rajah 3.40 Laluan biasa pergerakan robot

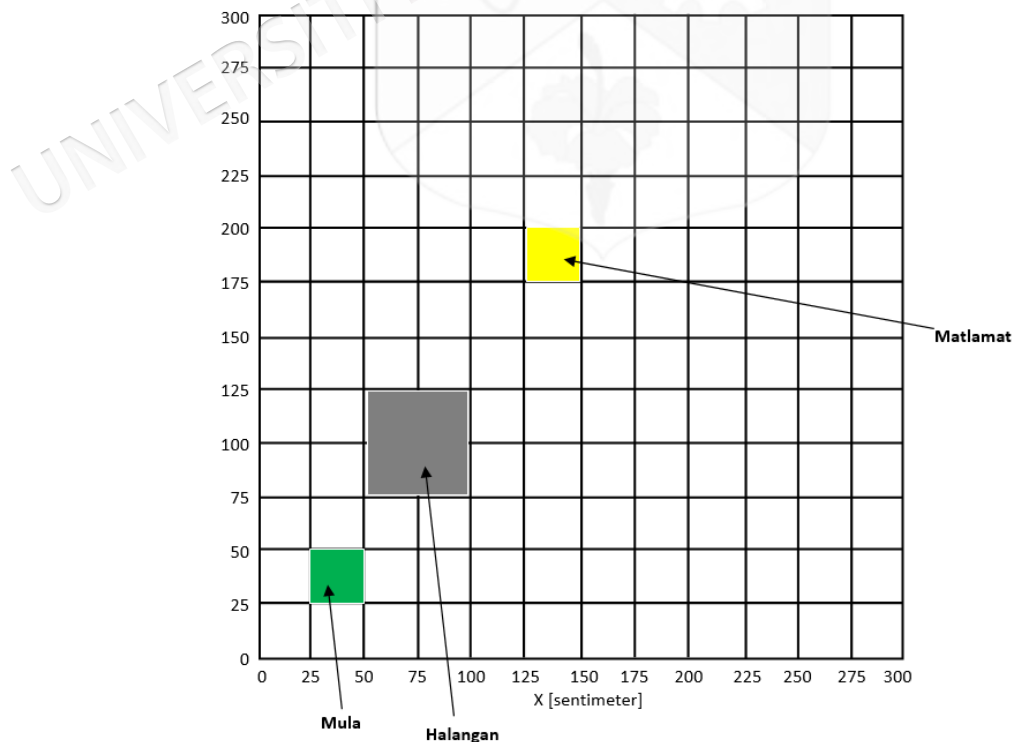
Dalam Kod Rantaian Freeman terdapat 2 model, 4 arah pertama dan 8 arah kedua. Di sini kita mengambil model kedua, yang mempunyai 8 arah, termasuk; 0 di Timur, 2 di Utara, 4 di Barat, 6 di Selatan. Jika halangan dikesan oleh kamera, robot akan mengira nilai x_1 , x_2 dan y menggunakan ultrasonik. Ilustrasi boleh dilihat Rajah 3.41.

Rajah 3.41 Pengesanan X_1 dan X_2 menggunakan penderia ultrasonik



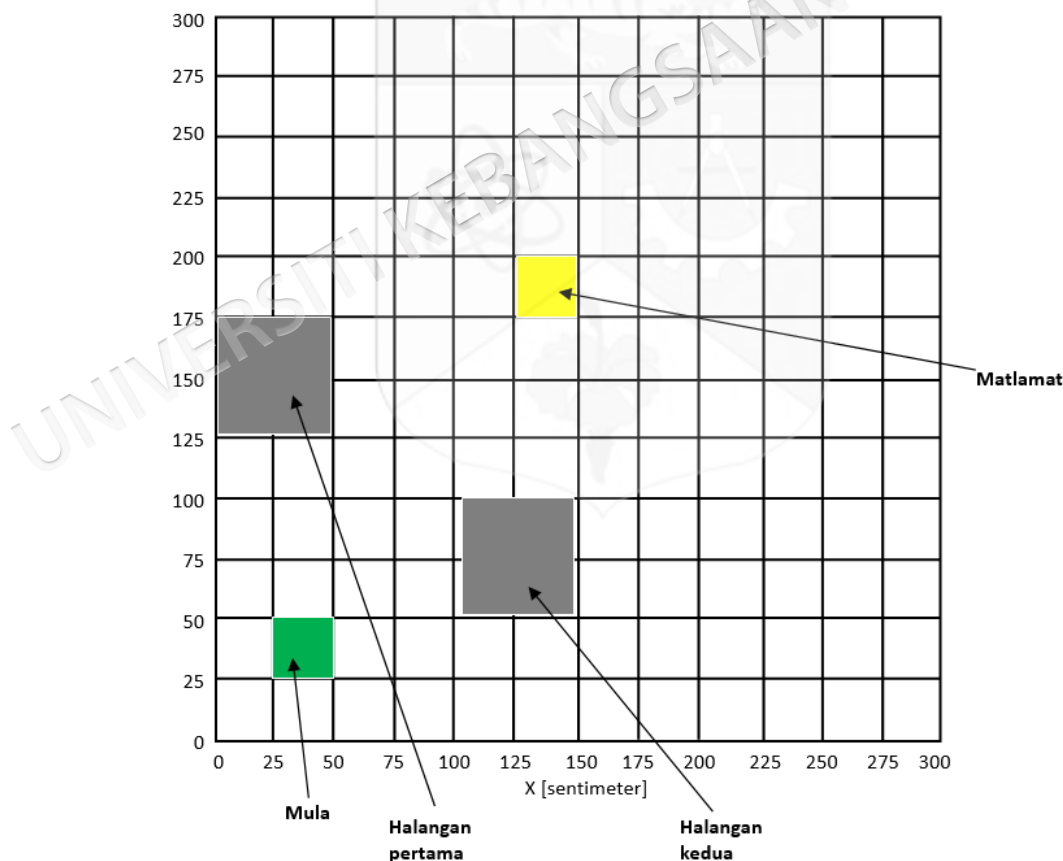
Rajah 3.44 Laluan pergerakan robot jika nilai $X_1 > X_2$

Dalam semua keadaan ini, nilai rantai kod dalam memori akan berubah secara automatik mengikut syarat ini. Oleh itu robot boleh bergerak dari Mula ke Matlamat dalam dua keadaan, dengan satu halangan dan dengan dua halangan. Selepas kaedah MFCC-PP dilaksanakan dalam mencari sampah di perkarangan rumah, kaedah ini kemudiannya dibandingkan dengan kaedah navigasi algoritma A*, algoritma Dijkstra, dan algoritma FDCC-PP. Pengujian dilakukan dengan menyediakan 2 jenis peta yang berbeza. Peta pertama iaitu peta laluan robot dengan satu halangan. Halangannya ialah satu buah kotak berbentuk kubus, yang dapat dilihat pada Rajah 3.45.



Rajah 3.45 Peta untuk menguji algoritma navigasi robot dengan satu halangan

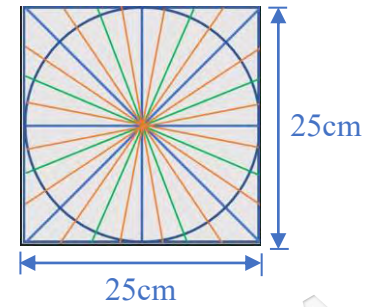
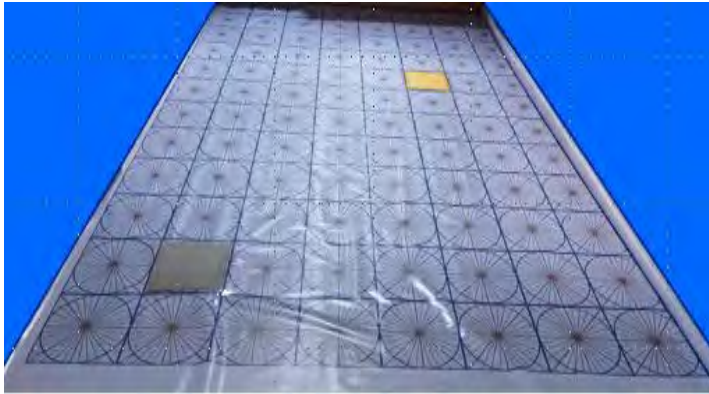
Selepas peta dibuat, program simulasi navigasi robot dibuat menggunakan algoritma A*. Robot akan bergerak dari Mula ke Matlamat. Sekiranya terdapat halangan, robot akan mengelak halangan tersebut berdasarkan algoritma yang digunakan. Terdapat dua parameter yang direkodkan iaitu purata masa perjalanan (*average time*) dan panjang laluan (*path length*). Setelah semua parameter direkodkan dalam algoritma A*, maka algoritma seterusnya diuji secara berturut-turut iaitu algoritma Dijkstra, *Freeman Direction Chain Code - Path Planning* (FDCC-PP), serta algoritma yang dicadangkan, iaitu **Modified Freeman Chain Code - Path Planning (MFCC-PP)**. Selepas mendapat parameter kesemua algoritma ini, maka pengujian dijalankan menggunakan peta kedua. Peta ini ialah peta navigasi robot dengan dua halangan. Halangannya ialah dua buah kotak berbentuk kubus Peta kedua boleh dilihat dalam Rajah 3.46.



Rajah 3.46 Peta untuk menguji navigasi robot dengan dua halangan

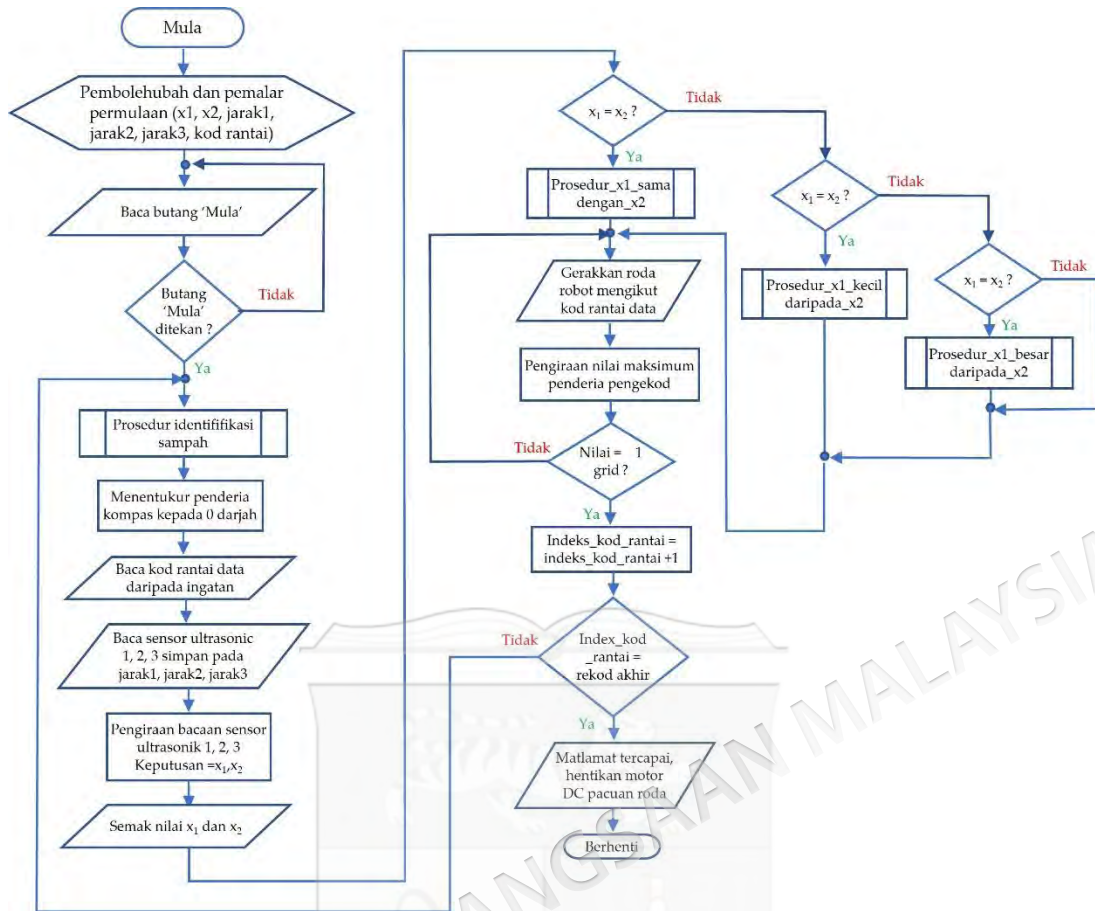
Bentuk fizikal peta grid boleh dilihat dalam Rajah 3.47. Setiap grid berukuran 25 cm x 25 cm. Pemilihan saiz ini bersesuaian dengan saiz robot. Pada grid, garisan

dibuat dalam bentuk bulatan dibahagikan dengan 32, supaya setiap baris menunjukkan nilai sudut dengan gandaan 11.25 darjah.



Rajah 3.47 Bentuk Fizikal Peta Grid Navigasi Robot

Dengan menggunakan grid ini, pergerakan robot dapat diperhatikan mengikut program simulasi yang telah dibuat. Jika sesuai, maka wujud perkaitan antara simulasi dan hasil pergerakan robot dalam masa nyata. Carta alir *Modified Freeman Chain Code – Path Planning* (MFCC-PP) boleh dilihat dalam Rajah 3.48.



Rajah 3.48 Carta alir MFCC-PP

3.5 KESIMPULAN

Pada Bab III ini telah dibincangkan metodologi yang digunakan dalam mencapai objektif penyelidikan yang telah dirumuskan. Objektif pertama, iaitu membangunkan prototaip robot berautonomi. Kaedah yang digunakan untuk mencapai objektif ini adalah dengan reka bentuk robot berautonomi berasaskan tiga reka bentuk, iaitu ; reka bentuk identifikasi robot, reka bentuk navigasi robot dan reka bentuk lengan robot. Selepas itu semua komponen elektronik yang menyokong reka bentuk tersebut diuji bagi memastikan 2 perkara, iaitu kejutuan dan kejayaan penderia dan penggerak (*actuator*) untuk bergerak sesuai dengan senario.

Objektif kedua iaitu membangunkan algoritma identifikasi sampah. Untuk mencapai objektif kedua ini, algoritma identifikasi sampah berdasarkan pada

Rangkaian Saraf Konvolusi (CNN) terlatih yaitu AlexNet, VGG16, GoogleNet, dan ResNet18. Untuk menguji keakuratan pengesanan ini, digunakan matriks konfusi. Sementara itu, untuk menguji sama ada objek itu bentuk 3D atau tidak (2D) digunakan penderia LiDAR. Dalam kajian ini, kaedah *Sequential_Camera_LiDAR* (SCL) dicadangkan. Kaedah ini menggabungkan kamera dan penderia LiDAR. CNN memproses output kamera, diikuti dengan proses pengimbasan oleh penderia LiDAR.

Objektif ketiga iaitu membangunkan algoritma navigasi robot. Kaedah yang digunakan untuk mencapai objektif ini ialah mencipta algoritma baharu dan membandingkannya dengan algoritma sedia ada. Algoritma yang diusulkan adalah ***Modified Freeman Chain Code-Path Planning (MFCC-PP)***. Algoritma ini memodifikasi *Freeman Chain Code* dengan menambahkan jumlah kod iaitu 16 kod dan 32 kod. Dengan penambahan kod ini memungkinkan robot bergerak lebih lancar. Ujian dijalankan dengan membuat dua peta iaitu peta laluan dengan satu halangan dan peta laluan dengan dua halangan. Daripada kedua-dua peta ini, pencapaian setiap algoritma navigasi robot akan diperolehi.

BAB IV

HASIL DAN PERBINCANGAN

4.1 PENGENALAN

Keupayaan robot berautonomi untuk mengidentifikasi sampah di perkarangan rumah dan bernavigasi di sekitar rumah adalah satu cabaran bagi robot. Jika kemampuan ini tidak dapat direalisasikan, maka robot tersebut dianggap belum dapat membantu manusia dalam hal membersihkan sampah di perkarangan rumah.

Bab ini akan membentangkan hasil penyelidikan dan perbincangan secara lebih mendalam. Pembentangan hasil dan pembincangan ini adalah merujuk kepada objektif yang telah dinyatakan dalam Bab I.

Sub Bab 4.2 membincangkan hasil daripada prestasi robot berautonomi. Pada Sub Bab ini rekabentuk dan pembangunan robot berautonomi serta prestasinya akan dibentangkan dan dibincangkan. Seterusnya, Sub Bab 4.3 akan membincangkan hasil daripada prestasi algoritma identifikasi sampah. Pengecaman ini amat penting kerana sampah tersebut perlu dikesan dengan baik. Kesilapan dalam proses pengecaman menyebabkan kesilapan dalam robot mengambil tindakan untuk membuang sampah. Dalam Sub-Bab 4.4, akan dibentangkan dan dibincangkan hasil pencapaian algoritma navigasi robot berautonomi. Navigasi robot yang baik akan menyebabkan robot bergerak dengan lebih lancar dan dapat bergerak mengikut laluan yang telah ditetapkan. Selain itu, navigasi robot yang baik boleh membuat keputusan untuk mengelakkan halangan di hadapan robot dan kembali ke laluan yang telah ditetapkan secepat mungkin.

4.2 HASIL REKA BENTUK ROBOT AUTONOMI

Prototaip robot berautonomi dibuat berdasarkan reka bentuk yang telah dibuat dalam Rajah 3.2, Sub Bab 3.2. Dalam reka bentuk ini, robot berautonomi terdiri daripada bahagian identifikasi sampah, bahagian navigasi robot dan bahagian lengan robot. Sub bab 4.2.1 menerangkan hasil reka bentuk bahagian identifikasi sampah. Kemudian pada Sub Bab 4.2.2 akan diterangkan tentang hasil reka bentuk bahagian navigasi robot di perkarangan rumah. Sub-bab 4.2.3 menerangkan hasil reka bentuk lengan robot. Tambahan pula, dalam Sub-Bab 4.2.4, diterangkan hasil reka bentuk keseluruhan robot berautonomi.

4.2.1 Hasil reka bentuk bahagian identifikasi sampah

Selepas reka bentuk identifikasi sampah dibuat seperti dalam Rajah 3.4, maka komponen-komponen yang terkandung dalam reka bentuk tersebut dipasang mengikut susun atur sedia ada. Gambar sebenar reka bentuk boleh dilihat dalam Rajah 4.1.



Rajah 4.1 Gambar sebenar reka bentuk sistem pengimbasan menggunakan penderia LiDAR

Dalam Rajah 4.1 dapat dilihat bahawa sistem pengimbasan penderia LiDAR menggunakan motor servo untuk menggerakkan penderia dalam arah mendatar dan menegak. Motor servo 1 untuk arah mendatar dan motor servo 2 untuk arah pergerakan menegak. Adalah sangat penting sebelum menggerakkan penderia LiDAR adalah untuk memastikan kedudukan darjah motor servo mengikut arahan yang diberikan dalam modul program ujian motor servo. Jika ia tidak sepadan, maka motor servo perlu ditentukan supaya arahan darjah yang diberikan sepadan dengan keadaan

sebenar. Selepas ujian, darjah yang ditunjukkan oleh motor servo adalah berbeza seperti yang ditunjukkan dalam Jadual 4.1.

Jadual 4.1 Selisih nilai kitaran tugas (*duty cycle*) antara teori dan praktikal

No.	Darjah Motor Servo	Kitaran Tugas (teori (%))	Kitaran Tugas (praktikal (%))	Selisih (%)
1	0	2	3	1
2	90	7.5	8	0.5
8	180	10	13	3

Dalam Jadual 4.1 dapat dilihat terdapat perbezaan nilai Kitaran Tugas (*Duty Cycle*) bagi mengawal motor servo ini. Nilai ini perlu ditentukan supaya nilai darjah motor servo sepadan dengan apa yang telah ditentukan dalam atur cara. Pengiraan untuk penentuan adalah seperti berikut.

$$\frac{\text{Kitaran Tugas}-3}{13-3} = \frac{\alpha-0}{180-0} \quad (10)$$

$$\frac{\text{Kitaran Tugas}-3}{13} = \frac{\alpha}{180} \quad (11)$$

$$\text{Kitaran Tugas} = \frac{10 \times \alpha}{180} + 3 \quad (12)$$

$$\text{Kitaran Tugas} = \frac{\alpha}{18} + 3 \quad (13)$$

Selepas formula Kitaran Tugas (*Duty Cycle*) diperoleh, proses penentuan selesai dan formula dimasukkan ke dalam modul program kawalan motor servo. Tambahan pula, motor servo diuji untuk menentukan prestasi motor servo apabila tiada beban dan apabila terdapat beban. Keputusan ujian boleh dilihat dalam Jadual 4.2.

Pada Jadual 4.2 dapat dilihat bahwa pada saat motor servo dengan kondisi tiada beban dan dengan beban. Beban yang digunakan di sini ialah penerima LiDAR seberat 60 gram. Dalam motor servo 1, terdapat perbezaan dalam pergerakan motor apabila terdapat beban dan apabila tiada beban, iaitu 0.01 saat hingga 0.02 saat. Manakala pada motor servo 2 terdapat perbezaan 0.02 saat hingga 0.03 saat. Daripada Jadual 4.2

juga boleh dilihat bahawa motor servo 2 mempunyai masa pergerakan yang lebih perlahan berbanding motor servo 1, iaitu 0.02 saat hingga 0.04 saat.

Jadual 4.2 Hasil Uji Kinerja Motor Servo

No.	Status Beban	Pergerakan Motor Servo (darjah)	Motor Servo 1 (mendatar sesaat)	Motor Servo 2 (menegak sesaat)
1	Tiada beban	30	0.35	0.37
2	Tiada beban	45	0.47	0.50
3	Tiada beban	90	0.85	0.89
4	Tiada beban	120	1.12	1.15
5	Dengan beban	30	0.36	0.39
6	Dengan beban	45	0.49	0.53
7	Dengan beban	90	0.87	0.91
8	Dengan beban	120	1.13	1.17

Pergerakan motor servo 2 boleh disebabkan oleh pergerakan menegak motor servo, tetapi ini dianggap tidak menjejaskan prestasi motor servo dengan ketara. Ini kerana motor servo tidak bergerak secara automatik dari 0 darjah ke 45 darjah atau dari 0 darjah ke 120 darjah. Dalam pengimbasan terperinci, motor servo 2 hanya bergerak 2 darjah. Begitu juga, dalam pengimbasan pantas, pergerakan motor servo tertinggi dalam setiap kedudukan pengimbasan hanya 10 darjah.

4.2.2 Hasil reka bentuk bahagian navigasi robot

Selepas reka bentuk navigasi robot dibuat seperti dalam Rajah 3.6, maka komponen-komponen yang terkandung dalam reka bentuk tersebut dipasang mengikut susun atur sedia ada. Gambar sebenar reka bentuk boleh dilihat dalam Rajah 4.2.



Rajah 4.2 Gambar sebenar reka bentuk sistem navigasi robot

Dalam Rajah 4.2 dapat dilihat bahawa imej sistem navigasi robot sebenar menyerupai imej reka bentuk navigasi robot dalam Rajah 3.6. Penderia pengekod terletak di bahagian belakang motor DC 1 dan motor DC 2. Manakala penderia kompas pula terletak pada kotak kawalan robot. Penderia ultrasonik 1, 2 dan 3 diletakkan di hadapan dan kiri dan kanan robot. Selepas modul program ujian motor DC dilaksanakan, keputusan ujian motor DC boleh dilihat dalam Jadual 4.3.

Jadual 4.3 Hasil Uji Kinerja Motor DC

No.	Motor DC 1	Motor DC 2	Kondisi Robot
1	Berhenti	Berhenti	Berhenti
2	Searah jarum jam (CW)	Berlawanan arah jarum jam (CCW)	Maju
3	Berlawanan arah jarum jam (CCW)	Searah jarum jam (CW)	Mundur
4	Searah jarum jam (CW)	Berhenti	Belok kanan
5	Berhenti	Searah jarum jam (CW)	Belok kiri

Dalam Jadual 4.3 terdapat 5 keadaan robot berdasarkan status motor DC 1 dan motor DC 2. Dalam keadaan pertama, dapat dilihat apabila motor DC 1 dan motor DC 2 berada dalam keadaan berhenti, robot berhenti bergerak. Dalam keadaan kedua, jika motor DC 1 bergerak mengikut arah jam (CW/*Clock Wise*) dan motor DC2 bergerak melawan arah jam (CCW/*Counter Clock Wise*), maka robot akan bergerak ke hadapan. Dalam keadaan ketiga, jika motor DC 1 bergerak melawan arah jam (CCW/*Counter Clock Wise*) dan motor DC2 bergerak mengikut arah jam (CW/*Clock Wise*), maka robot akan bergerak ke belakang. Kemudian, jika motor DC 1 bergerak mengikut arah jam (CW/*Clock Wise*) dan motor DC2 berhenti bergerak, robot akan membelok ke kanan. Sebaliknya, jika motor DC 1 bergerak melawan arah jam (CCW/*Counter Clock Wise*) dan motor DC2 berhenti bergerak, maka robot akan membelok ke kiri.

Untuk memastikan pengekod berfungsi dengan betul, modul program ujian pengekod dalam Rajah 3.18 dilaksanakan. Jejari roda robot bersaiz 2 cm dan bilangan denyutan penderia pengekod untuk satu putaran ialah 18 denyutan. Keputusan ujian boleh dilihat dalam Jadual 4.4.

Jadual 4.4 Hasil Uji Kinerja Penderia Pengekod

No.	Jumlah Denyut penderia pengkod 1	Jumlah Denyut penderia pengkod 2	Jarak Tempuh Robot Sebenarnya (cm)	Jarak Tempuh Robot pada serial monitor (cm)	Ralat (error) (%)
1	28	29	19.9	19.90	0.00
2	40	39	27.6	27.59	0.05
3	79	79	55.1	55.17	0.14
4	87	88	61.2	61.11	0.15
5	93	94	65.3	65.30	0.00
6	105	105	73.3	73.33	0.05
7	111	112	77.9	77.87	0.03
8	127	127	88.7	88.70	0.00
9	146	146	102.1	101.97	0.13
10	158	158	110.3	110.35	0.04
11	174	112	99.6	99.87	0.27
12	198	125	112.5	112.79	0.26
13	209	115	113.4	113.14	0.23
14	228	100	114.3	114.54	0.21
15	252	80	116.2	115.94	0.23
16	196	258	158.7	158.54	0.10
17	188	262	157.1	157.14	0.03
18	186	279	162.7	162.38	0.20
19	156	284	152.1	153.65	1.01
20	122	293	145.4	144.92	0.33

Dalam Jadual 4.4 dapat dilihat bahawa ralat yang diperoleh dalam ujian adalah antara 0% hingga 1.01%. Nilai ralat yang paling rendah iaitu 0% terdapat pada ujian nombor 1, nombor 5 dan nombor 8. Manakala nilai ralat yang paling tinggi terdapat pada ujian nombor 19 iaitu 1.01%.

Dalam ujian nombor 1 hingga nombor 1, nilai ralat yang agak rendah diperolehi berbanding ujian dalam nombor seterusnya. Nilai ralat rendah berlaku apabila nilai bilangan denyutan penderia pengkod 1 dan pengkod 2 adalah hampir sama. Dalam keadaan ini, robot berada dalam keadaan bergerak lurus ke hadapan. Sehingga daya geseran yang diterima oleh setiap roda adalah hampir sama. Manakala dalam ujian nombor 11 hingga nombor 20 terdapat perbezaan yang agak besar antara nilai bilangan denyutan penderia pengkod 1 dan nilai bilangan denyutan penderia pengkod 2. Nilai ini berlaku apabila robot membelok ke kiri atau membelok. betul.

Pada masa pusingan, satu roda berputar kurang. Akibatnya, nilai bilangan denyutan pada salah satu penderia pengekod menjadi kurang. Nilai ralat yang lebih tinggi boleh disebabkan apabila membelok, roda mendapat daya geseran yang lebih besar berbanding dengan bergerak lurus.

Selepas penderia pengekod diuji dan keputusan diperolehi, kemudian penderia kompas diuji dengan menjalankan modul program pengujian penderia kompas dalam Rajah 3.19. Keputusan ujian boleh dilihat dalam Jadual 4.5.

Jadual 4.5 Hasil Uji Kinerja Penderia Kompas

No.	Ukuran Sudut Sebenarnya (darjah)	Tampilan pada serial monitor (darjah)	Beza (darjah)	Ralat (<i>error</i>) (%)
1	0	0	0	0.00
2	15	15	0	0.00
3	30	31	1	3.33
4	45	45	0	0.00
5	60	61	1	1.67
6	75	77	2	2.67
7	90	90	0	0.00
8	105	106	1	0.95
9	120	121	1	0.83
10	135	135	0	0.00
11	150	152	2	1.33
12	165	166	1	0.61
13	180	180	0	0.00
14	195	196	1	0.51
15	210	211	1	0.48
16	225	226	1	0.44
17	240	241	1	0.42
18	255	256	1	0.39
19	270	270	0	0.00
20	285	286	1	0.35
21	300	302	2	0.67
22	315	315	0	0.00
23	330	330	0	0.00
24	345	346	1	0.29

Untuk memastikan bahawa penderia ultrasonik berfungsi dengan betul, modul program ujian penderia ultrasonik dalam Rajah 3.20 dilaksanakan. Keputusan ujian boleh dilihat dalam Jadual 4.6.

Jadual 4.6 Hasil Uji Kinerja Penderia Ultrasonik

No.	Ukuran Jarak Sebenarnya (cm)	Tampilan pada serial monitor (cm)	Beza (cm)	Ralat (<i>error</i>) (%)
1	5	5	0	0.00
2	10	10.1	0.1	1.00
3	15	15.2	0.2	1.33
4	20	20.3	0.3	1.50
5	25	25.2	0.2	0.80
6	30	30.3	0.3	1.00
7	35	35.3	0.3	0.86
8	40	40.5	0.5	1.25
9	45	45.3	0.3	0.67
10	50	50.1	0.1	0.20
11	55	55.6	0.6	1.09
12	60	60.7	0.7	1.17
13	70	71.2	1.2	1.71
14	75	75.3	0.3	0.40
15	85	84.4	0.6	0.71
16	90	90.5	0.5	0.56
17	100	100.2	0.2	0.20
18	115	114.7	0.3	0.26
19	125	125.3	0.3	0.24
20	150	150.2	0.2	0.13

Daripada keadaan pengukuran dalam Jadual 4.6 terdapat perbezaan antara saiz sebenar dan paparan pada siri monitor komputer riba, tetapi hanya terletak pada nombor perpuluhan. Ini tidak benar-benar menjejaskan nilai pembolehubah jarak dalam proses navigasi robot nanti. Dengan itu dapat dipastikan bahawa litar penderia ultrasonik telah dapat berfungsi dengan baik.

4.2.3 Hasil reka bentuk bahagian lengan robot

Selepas reka bentuk lengan robot dibuat seperti dalam Rajah 3.8, maka komponen-komponen yang terkandung dalam reka bentuk tersebut dipasang mengikut susun atur sedia ada. Gambar sebenar reka bentuk boleh dilihat dalam Rajah 4.3.



Rajah 4.3 Gambar sebenar reka bentuk sistem lengan robot

4.2.4 Hasil reka bentuk keseluruhan robot berautonomi

Pada asasnya reka bentuk keseluruhan robot berautonomi dalam Rajah 3.2 adalah gabungan reka bentuk pengenalan robot dan navigasi robot dan lengan robot. Selepas reka bentuk robot keseluruhan dibuat seperti yang ditunjukkan dalam Rajah 3.2, bentuk sebenar robot berautonomi secara keseluruhannya juga merupakan gabungan ketiga-tiga bahagian tersebut. Pertama sekali, bahagian pengenalan sampah diletakkan di atas bahagian navigasi robot. Tambahan pula, lengan robot juga diletakkan di atas bahagian navigasi robot. Bagaimanapun, kedudukannya berada di belakang bahagian identifikasi sampah. Gambar sebenar reka bentuk boleh dilihat dalam Rajah 4.4.



Rajah 4.4 Gambar sebenar reka bentuk sistem robot berautonomi

4.3 HASIL IDENTIFIKASI SAMPAH MENGGUNAKAN RANGKAIAN SARAF KONVOLUSI (CNN)

Selepas algoritma identifikasi sampah dibuat seperti Rajah 3.29, barulah data imej diuji untuk 11 jenis sampah menggunakan empat seni bina CNN, iaitu AlexNet, VGG16, GoogleNet dan ResNet18. Bilangan imej untuk setiap jenis sampah ialah 150 keping. Data ditambah kepada 1200 imej, jadi jumlah imej ialah 13,200. Data dibahagikan kepada 70% data latihan dan 30% data ujian. Keputusan ujian akan diterangkan dalam setiap sub-bab. Dalam sub-bab 4.3.1, hasil identifikasi sampah menggunakan AlexNet akan diterangkan. Seterusnya, dalam sub-bab 4.3.2, hasil identifikasi sampah menggunakan VGG16 akan diterangkan. Selepas itu, dalam sub-bab 4.3.3, hasil identifikasi sampah menggunakan GoogleNet akan diterangkan. Akhirnya dalam sub-bab 4.3.4, hasil identifikasi sampah menggunakan ResNet18 akan diterangkan. Pilihan model pra-latih, AlexNet, VGG16, GoogleNet dan ResNet untuk identifikasi sampah adalah kerana model ini telah berjaya dalam cabaran ImageNet.

4.3.1 Identifikasi sampah menggunakan AlexNet

Selepas menguji pengesanan sampah menggunakan seni bina pra-latih AlexNet, keputusan kemajuan latihan diperolehi seperti ditunjukkan dalam Rajah 4.5.



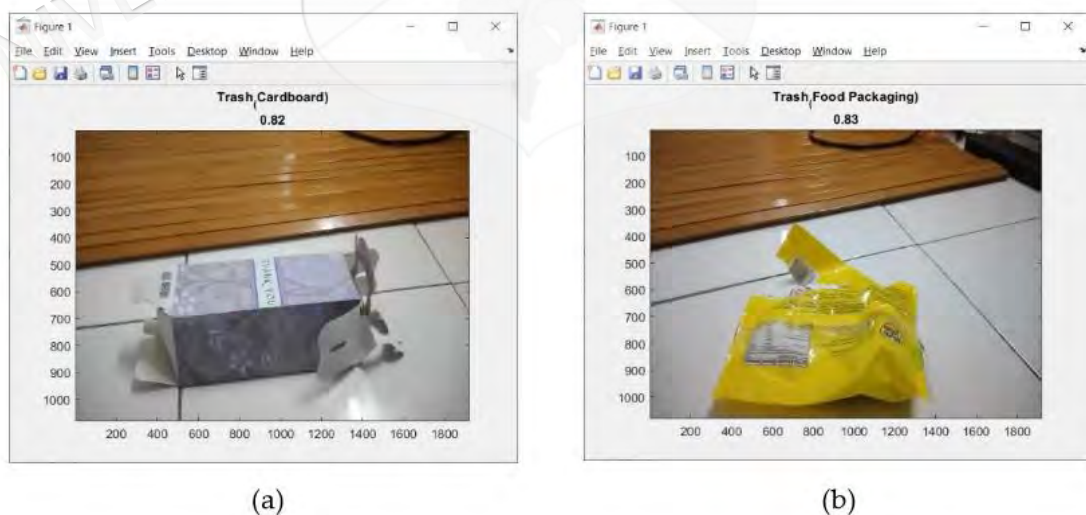
Rajah 4.5 Hasil Latihan Kemajuan Menggunakan Seni Bina Pra-latih AlexNet

Pada Rajah 4.5 tersebut dapat dilihat bahawa Masa Latihan untuk AlexNet ialah 229 saat, dan Ketepatan Pengesanan (*validation*) ialah 77.54%. Setelah hasil kemajuan latihan diperolehi, selanjutnya sistem identifikasi sampah ini untuk prestasinya menggunakan matriks kekeliruan. Rajah 4.6 menunjukkan keputusan matriks kekeliruan dengan pra-latih AlexNet.

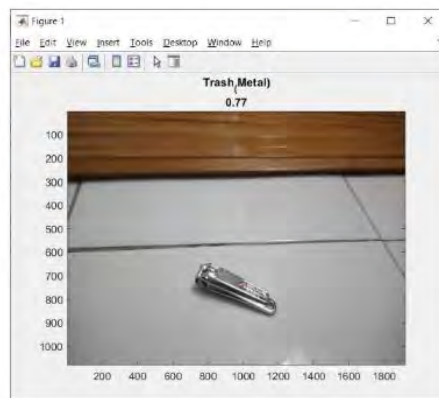
		Confusion Matrix											
Output Class	Trash_Cardboard)	360 9.1%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
	Trash_Fabric)	0 0.0%	360 9.1%	0 0.0%	0 0.0%	4 0.1%	4 0.1%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	97.8% 2.2%
	Trash_Food Packaging)	0 0.0%	0 0.0%	354 8.9%	0 0.0%	32 0.8%	0 0.0%	0 0.0%	0 0.0%	23 0.6%	0 0.0%	0 0.0%	86.6% 13.4%
	Trash_Fruit)	0 0.0%	0 0.0%	0 0.0%	360 9.1%	0 0.0%	0 0.0%	4 0.1%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	98.9% 1.1%
	Trash_Glass)	0 0.0%	0 0.0%	1 0.0%	242 6.1%	0 0.0%	9 0.2%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	96.0% 4.0%
	Trash_Leaf)	0 0.0%	0 0.0%	1 0.0%	0 0.0%	3 0.1%	352 8.9%	0 0.0%	8 0.2%	0 0.0%	0 0.0%	0 0.0%	96.7% 3.3%
	Trash_Metal)	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	347 8.8%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
	Trash_Paper)	0 0.0%	0 0.0%	2 0.1%	0 0.0%	58 1.5%	0 0.0%	0 0.0%	352 8.9%	0 0.0%	0 0.0%	0 0.0%	85.4% 14.6%
	Trash_Plastic)	0 0.0%	0 0.0%	2 0.1%	0 0.0%	7 0.2%	0 0.0%	0 0.0%	0 0.0%	337 8.5%	0 0.0%	0 0.0%	97.4% 2.6%
	Trash_Rubber)	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	360 9.1%	0 0.0%	100% 0.0%
	Trash_Wood)	0 0.0%	0 0.0%	0 0.0%	0 0.0%	14 0.4%	4 0.1%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	360 9.1%	95.2% 4.8%
		100% 0.0%	100% 0.0%	98.3% 1.7%	100% 0.0%	67.2% 32.8%	97.8% 2.2%	96.4% 3.6%	97.8% 2.2%	93.6% 6.4%	100% 0.0%	100% 0.0%	95.6% 4.4%
		Target Class											

Rajah 4.6 Hasil matriks kekeliruan dengan pra-latih AlexNet

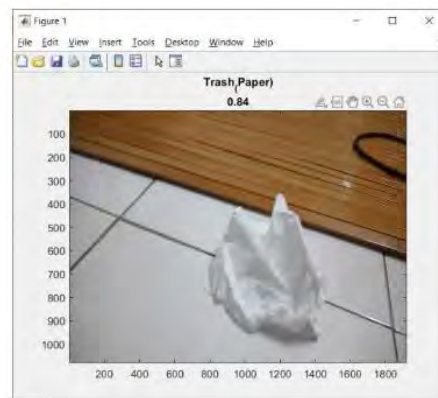
Selepas proses latihan selesai, ujian pengecaman sampah dijalankan dalam masa nyata menggunakan kamera IP. Prosedur pemasangan kamera IP telah dijelaskan dalam sub-bab 3.2.4 dalam bahagian modul program ujian kamera IP. Sampel keputusan ujian boleh dilihat dalam Rajah 4.7 hingga Rajah 4.10.



Rajah 4.7 Hasil identifikasi sampah menggunakan kamera IP. (a) Kadbod; (b)Pembungkus makanan

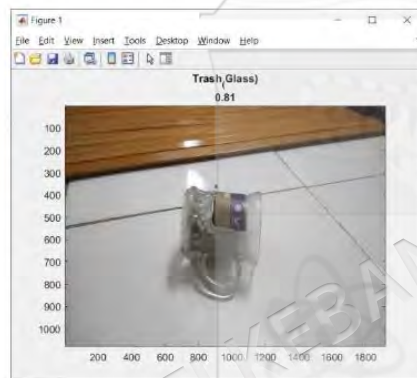


(a)

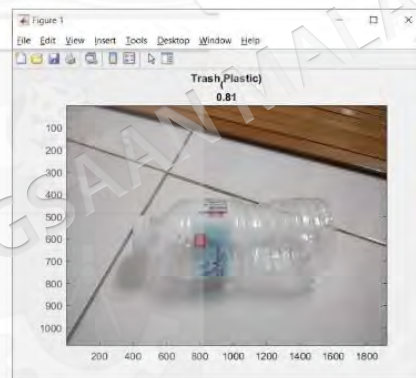


(b)

Rajah 4.8 Hasil identifikasi sampah menggunakan kamera IP. (a) Logam; (b) Kertas



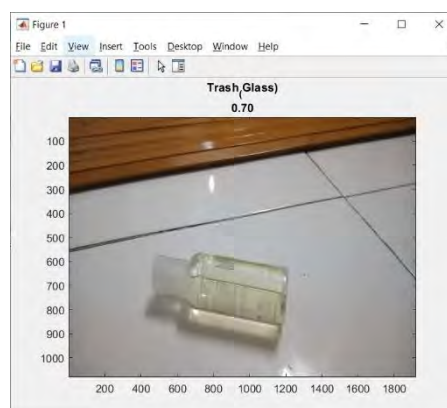
(a)



(b)

Rajah 4.9 Hasil identifikasi sampah menggunakan kamera IP. (a) Kaca; (b) Plastik

Pada masa ujian, terdapat juga ralat dalam identifikasi sampah, seperti yang ditunjukkan dalam Rajah 4.10. Objek sampah diidentifikasi sebagai kaca, walaupun objek sampah adalah plastik.



Rajah 4.10 Hasil identifikasi sampah yang salah

Ujian yang dilakukan dalam Rajah 4.7 hingga Rajah 4.10 digunakan untuk menguji prestasi seni bina CNN pra-latihan iaitu AlexNet dalam identifikasi sampah dalam masa nyata. Bilangan yang diuji ialah 411 keping sampah. Kemudian, rekod jumlah sampah yang dikesan untuk setiap jenis sampah. Ujian ini menggunakan sampah yang tidak pernah digunakan dalam latihan. Jadual 4.7 menunjukkan keputusan ketepatan yang diperolehi oleh setiap seni bina CNN pra-latihan.

Jadual 4.7 Hasil ketepatan yang dicapai oleh seni bina CNN pra-latihan AlexNet

No.	Jenis Sampah	Bilangan Ujian Sampah Sebenar	Jumlah Sampah Dikesan Dengan Betul	Ketepatan
1	Kadbod (<i>Cardboard</i>)	45	37	82.22222222
2	Fabrik (<i>Fabric</i>)	17	13	76.47058824
3	Pembungkusan Makanan (<i>Food Packaging</i>)	63	52	82.53968254
4	Buah (<i>Fruit</i>)	21	17	80.95238095
5	Kaca (<i>Glass</i>)	34	27	79.41176471
6	Daun (<i>Leaf</i>)	56	46	82.14285714
7	Logam (<i>Metal</i>)	21	16	76.19047619
8	Kertas (<i>Paper</i>)	61	51	83.60655738
9	Plastik (<i>Plastic</i>)	52	42	80.76923077
10	Getah (<i>Rubber</i>)	14	10	71.42857143
11	Kayu (<i>Wood</i>)	27	21	77.77777778

4.3.2 Identifikasi sampah menggunakan VGG16

Selepas menguji pengesanan sampah menggunakan seni bina pra-latih VGG16, keputusan kemajuan latihan diperolehi seperti ditunjukkan dalam Rajah 4.11.



Rajah 4.11 Hasil Latihan Kemajuan Menggunakan Seni Bina Pra-latih VGG16

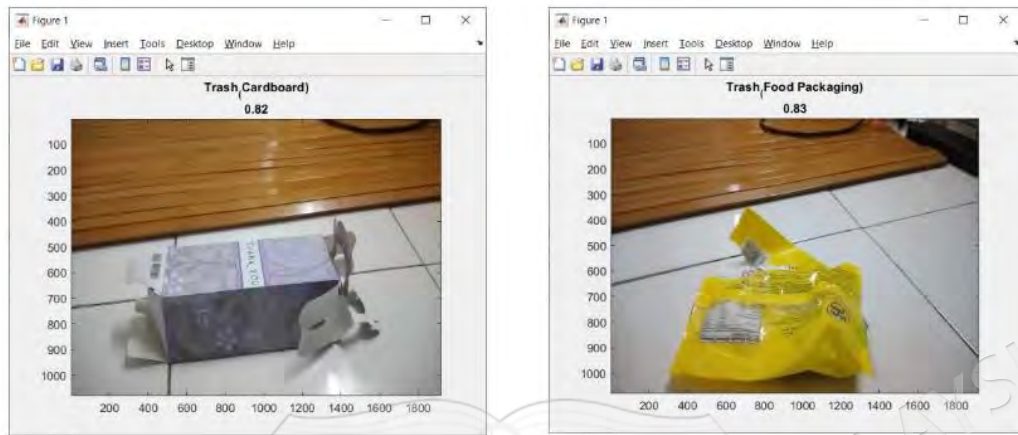
Pada Rajah 4.11 tersebut dapat dilihat bahawa masa latihan untuk VGG16 ialah 894 saat, dan Ketepatan Pengesanan (*validation*) ialah 79.09%. Setelah hasil kemajuan latihan diperoleh, selanjutnya sistem identifikasi sampah ini untuk prestasinya menggunakan matriks kekeliruan. Rajah 4.12 menunjukkan keputusan matriks kekeliruan dengan pra-latih VGG16.

		Confusion Matrix										
Output Class	Trash (Cardboard)	360 9.1%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
	Trash (Fabric)	0 0.0%	360 9.1%	0 0.0%	0 0.0%	4 0.1%	4 0.1%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	97.8% 2.2%
	Trash (Food Packaging)	0 0.0%	0 0.0%	354 8.9%	0 0.0%	32 0.8%	0 0.0%	0 0.0%	23 0.6%	0 0.0%	0 0.0%	86.6% 13.4%
	Trash (Fruit)	0 0.0%	0 0.0%	0 0.0%	360 9.1%	0 0.0%	0 0.0%	4 0.1%	0 0.0%	0 0.0%	0 0.0%	98.9% 1.1%
	Trash (Glass)	0 0.0%	0 0.0%	1 0.0%	0 0.0%	242 6.1%	0 0.0%	9 0.2%	0 0.0%	0 0.0%	0 0.0%	96.0% 4.0%
	Trash (Leaf)	0 0.0%	0 0.0%	1 0.0%	0 0.0%	3 0.1%	352 8.9%	0 0.0%	8 0.2%	0 0.0%	0 0.0%	96.7% 3.3%
	Trash (Metal)	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	347 8.8%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
	Trash (Paper)	0 0.0%	0 0.0%	2 0.1%	0 0.0%	58 1.5%	0 0.0%	0 0.0%	352 8.9%	0 0.0%	0 0.0%	85.4% 14.6%
	Trash (Plastic)	0 0.0%	0 0.0%	2 0.1%	0 0.0%	7 0.2%	0 0.0%	0 0.0%	0 0.0%	337 8.5%	0 0.0%	97.4% 2.6%
	Trash (Rubber)	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	360 9.1%	100% 0.0%
	Trash (Wood)	0 0.0%	0 0.0%	0 0.0%	0 0.0%	14 0.4%	4 0.1%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	95.2% 4.8%
		100% 0.0%	100% 0.0%	98.3% 1.7%	100% 0.0%	67.2% 32.8%	97.8% 2.2%	96.4% 3.6%	97.8% 2.2%	93.6% 6.4%	100% 0.0%	95.8% 4.4%
		Trash (Cardboard)	Trash (Fabric)	Trash (Food Packaging)	Trash (Fruit)	Trash (Glass)	Trash (Leaf)	Trash (Metal)	Trash (Paper)	Trash (Plastic)	Trash (Rubber)	Trash (Wood)
		Target Class										

Rajah 4.12 Hasil matriks kekeliruan dengan pra-latih VGG16

Selepas proses latihan selesai, ujian pengecaman sampah dijalankan dalam masa nyata menggunakan kamera IP. Prosedur pemasangan kamera IP telah dijelaskan

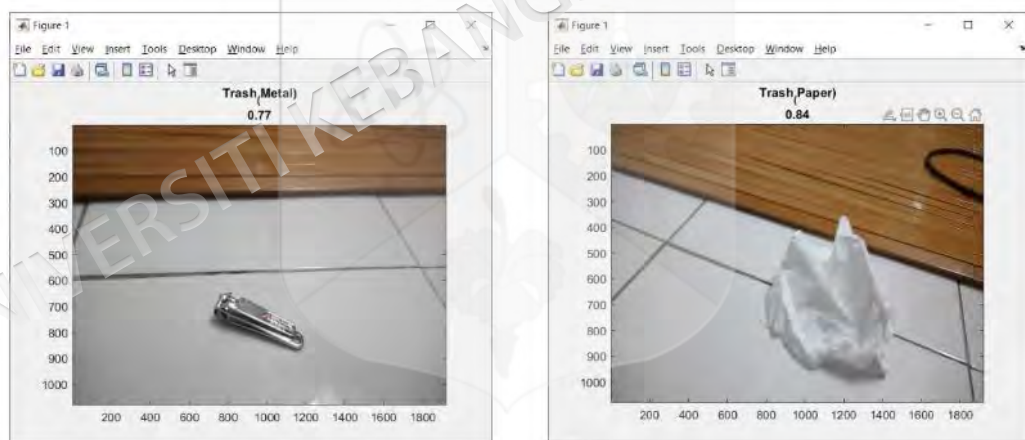
dalam sub-bab 3.2.4 dalam bahagian modul program ujian kamera IP. Sampel keputusan ujian boleh dilihat dalam Rajah 4.13 hingga Rajah 4.16.



(a)

(b)

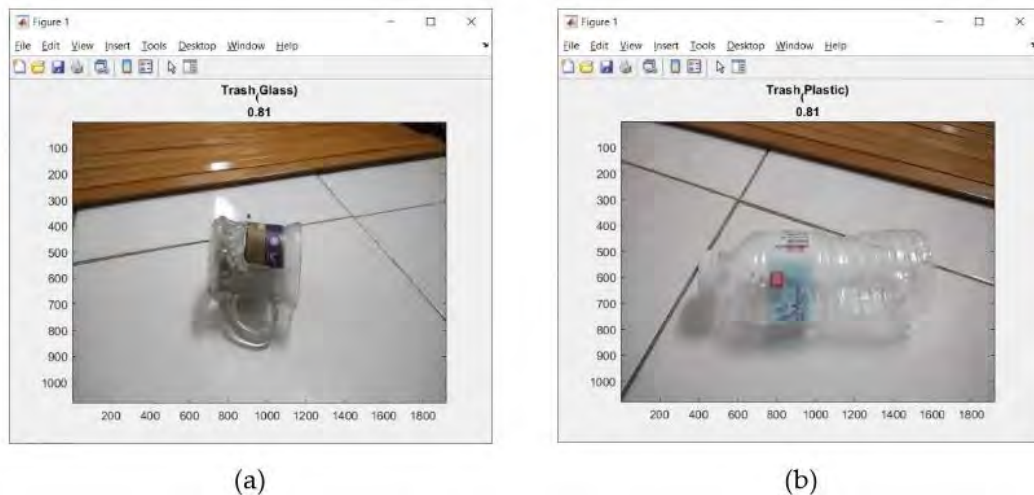
Rajah 4.13 Hasil identifikasi sampah menggunakan kamera IP. (a) Kadbod; (b)Pembungkus makanan



(a)

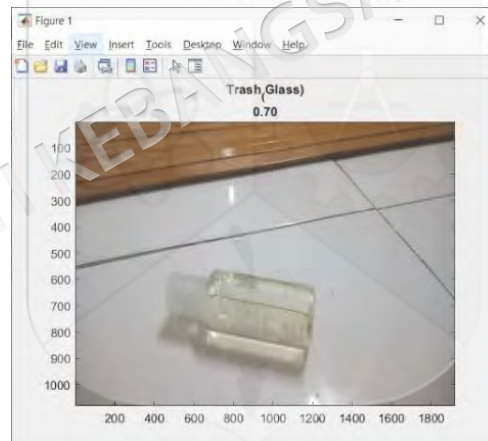
(b)

Rajah 4.14 Hasil identifikasi sampah menggunakan kamera IP. (a) Logam; (b) Kertas



Rajah 4.15 Hasil identifikasi sampah menggunakan kamera IP. (a) Kaca; (b) Plastik

Pada masa ujian, terdapat juga ralat dalam identifikasi sampah, seperti yang ditunjukkan dalam Rajah 4.16. Objek sampah diidentifikasi sebagai kaca, walaupun objek sampah adalah plastik.



Rajah 4.16 Hasil identifikasi sampah yang salah

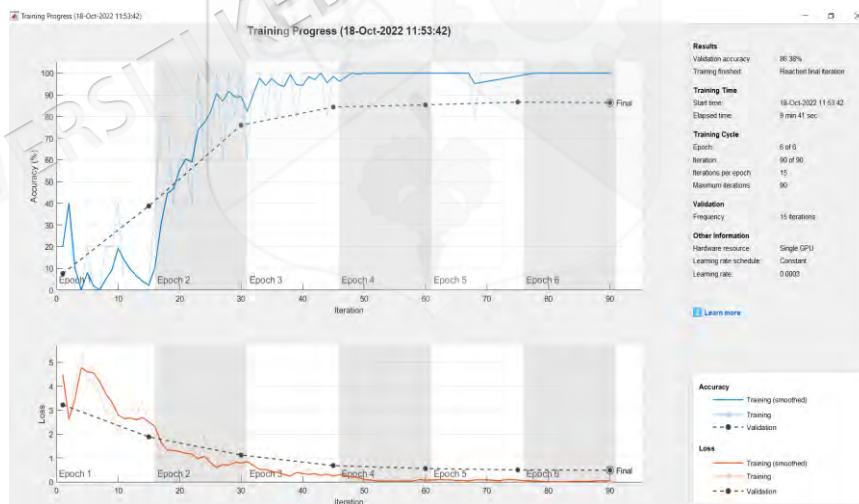
Ujian yang dilakukan dalam Rajah 4.13 hingga Rajah 4.16 digunakan untuk menguji prestasi seni bina CNN pra-latihan iaitu VGG16 dalam identifikasi sampah dalam masa nyata. Bilangan yang diuji ialah 411 keping sampah. Kemudian, rekod jumlah sampah yang dikesan untuk setiap jenis sampah. Ujian ini menggunakan sampah yang tidak pernah digunakan dalam latihan. Jadual 4.8 menunjukkan keputusan ketepatan yang diperolehi oleh setiap seni bina CNN pra-latihan.

Jadual 4.8 Hasil ketepatan yang dicapai oleh seni bina CNN pra-latihan VGG16

No.	Jenis Sampah	Bilangan Ujian Sampah Sebenar	Jumlah Sampah Dikesan Dengan Betul	Ketepatan
1	Kadbod (<i>Cardboard</i>)	45	40	88.88888889
2	Fabrik (<i>Fabric</i>)	17	14	82.35294118
3	Pembungkusan Makanan (<i>Food Packaging</i>)	63	57	90.47619048
4	Buah (<i>Fruit</i>)	21	19	90.47619048
5	Kaca (<i>Glass</i>)	34	30	88.23529412
6	Daun (<i>Leaf</i>)	56	49	87.50000000
7	Logam (<i>Metal</i>)	21	18	85.71428571
8	Kertas (<i>Paper</i>)	61	54	88.52459016
9	Plastik (<i>Plastic</i>)	52	45	86.53846154
10	Getah (<i>Rubber</i>)	14	11	78.57142857
11	Kayu (<i>Wood</i>)	27	23	85.18518519

4.3.3 Identifikasi sampah menggunakan GoogleNet

Selepas menguji pengesanan sampah menggunakan seni bina GoogleNet pra-latih, keputusan kemajuan latihan diperolehi seperti ditunjukkan dalam Rajah 4.17.



Rajah 4.17 Hasil Latihan Kemajuan Menggunakan Seni Bina Pra-latih GoogleNet

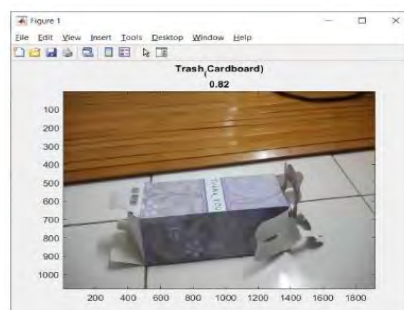
Pada Rajah 4.17 tersebut dapat dilihat bahawa masa latihan untuk GoogleNet ialah 581 saat, dan Ketepatan Pengesanan (*validation*) ialah 86.38%. Setelah hasil kemajuan latihan diperolehi, selanjutnya sistem identifikasi sampah ini untuk

prestasinya menggunakan matriks kekeliruan. Rajah 4.18 menunjukkan keputusan matriks kekeliruan dengan pra-latih GoogleNet.

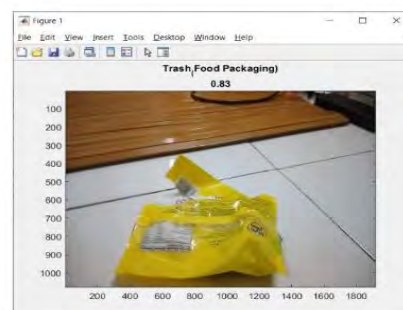
		Confusion Matrix											
Output Class	Trash (Cardboard)	360 9.1%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
	Trash (Fabric)	0 0.0%	360 9.1%	0 0.0%	4 0.1%	4 0.1%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	97.8% 2.2%
	Trash (Food Packaging)	0 0.0%	0 0.0%	354 8.9%	0 0.0%	32 0.8%	0 0.0%	0 0.0%	23 0.6%	0 0.0%	0 0.0%	0 0.0%	86.6% 13.4%
	Trash (Fruit)	0 0.0%	0 0.0%	0 0.0%	360 9.1%	0 0.0%	0 0.0%	4 0.1%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	98.9% 1.1%
	Trash (Glass)	0 0.0%	0 0.0%	1 0.0%	0 0.0%	242 6.1%	0 0.0%	9 0.2%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	96.0% 4.0%
	Trash (Leaf)	0 0.0%	0 0.0%	1 0.0%	0 0.0%	3 0.1%	352 8.9%	0 0.0%	8 0.2%	0 0.0%	0 0.0%	0 0.0%	96.7% 3.3%
	Trash (Metal)	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	347 8.8%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
	Trash (Paper)	0 0.0%	0 0.0%	2 0.1%	0 0.0%	58 1.5%	0 0.0%	0 0.0%	352 8.9%	0 0.0%	0 0.0%	0 0.0%	85.4% 14.6%
	Trash (Plastic)	0 0.0%	0 0.0%	2 0.1%	0 0.0%	7 0.2%	0 0.0%	0 0.0%	0 0.0%	337 8.5%	0 0.0%	0 0.0%	97.4% 2.6%
	Trash (Rubber)	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	360 9.1%	0 0.0%	0 0.0%	100% 0.0%
	Trash (Wood)	0 0.0%	0 0.0%	0 0.0%	0 0.0%	14 0.4%	4 0.1%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	360 9.1%	95.2% 4.8%
		100% 0.0%	100% 0.0%	98.3% 1.7%	100% 0.0%	67.2% 32.8%	97.8% 2.2%	96.4% 3.6%	97.8% 2.2%	93.8% 6.4%	100% 0.0%	100% 0.0%	95.6% 4.4%
		Trash (Cardboard)	Trash (Fabric)	Trash (Food Packaging)	Trash (Fruit)	Trash (Glass)	Trash (Leaf)	Trash (Metal)	Trash (Paper)	Trash (Plastic)	Trash (Rubber)	Trash (Wood)	
		Target Class											

Rajah 4.18 Hasil matriks kekeliruan dengan pra-latih GoogleNet

Selepas proses latihan selesai, ujian pengecaman sampah dijalankan dalam masa nyata menggunakan kamera IP. Prosedur pemasangan kamera IP telah dijelaskan dalam sub-bab 3.2.4 dalam bahagian modul program ujian kamera IP. Sampel keputusan ujian boleh dilihat dalam Rajah 4.19 hingga Rajah 4.22.

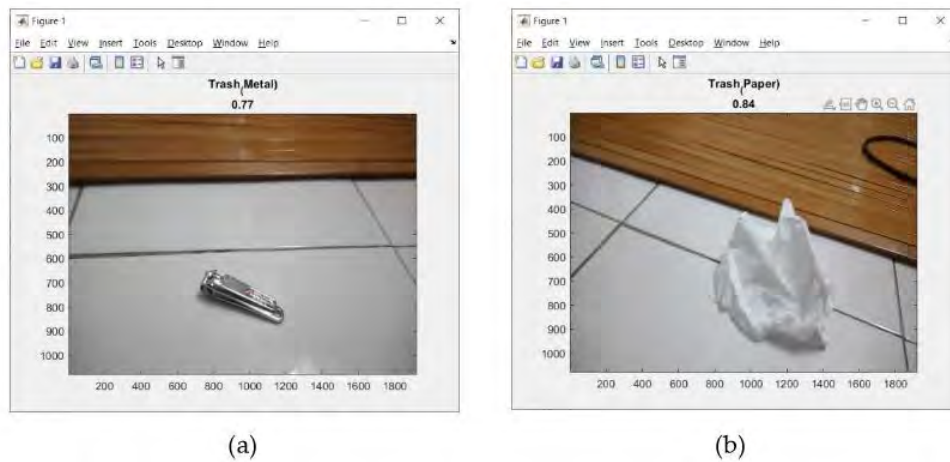


(a)

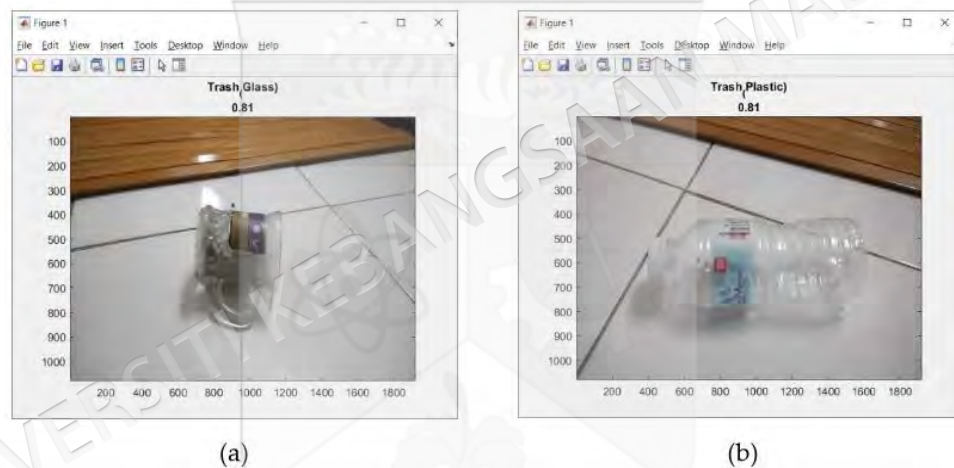


(b)

Rajah 4.19 Hasil identifikasi sampah menggunakan kamera IP. (a) Kadbod; (b)Pembungkus makanan

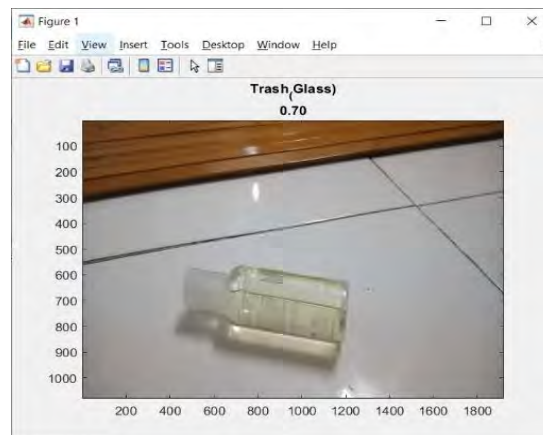


Rajah 4.20 Hasil identifikasi sampah menggunakan kamera IP. (a) Logam; (b) Kertas



Rajah 4.21 Hasil identifikasi sampah menggunakan kamera IP. (a) Kaca; (b) Plastik

Pada masa ujian, terdapat juga ralat dalam identifikasi sampah, seperti yang ditunjukkan dalam Rajah 4.22. Objek sampah diidentifikasi sebagai kaca, walaupun objek sampah adalah plastik.



Rajah 4.22 Hasil identifikasi sampah yang salah

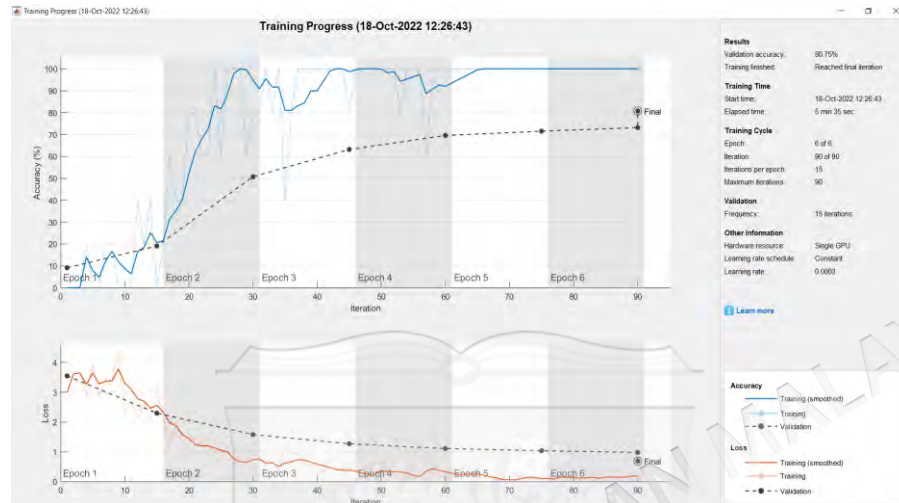
Ujian yang dilakukan dalam Rajah 4.19 hingga Rajah 4.22 digunakan untuk menguji prestasi seni bina CNN pra-latihan iaitu VGG16 dalam identifikasi sampah dalam masa nyata. Bilangan yang diuji ialah 411 keping sampah. Kemudian, rekod jumlah sampah yang dikesan untuk setiap jenis sampah. Ujian ini menggunakan sampah yang tidak pernah digunakan dalam latihan. Jadual 4.9 menunjukkan keputusan ketepatan yang diperolehi oleh setiap seni bina CNN pra-latihan.

Jadual 4.9 Hasil ketepatan yang dicapai oleh seni bina CNN pra-latihan GoogleNet

No.	Jenis Sampah	Bilangan Ujian Sampah Sebenar	Jumlah Sampah Dikesan Dengan Betul	Ketepatan
1	Kadbod (<i>Cardboard</i>)	45	44	97.77777778
2	Fabrik (<i>Fabric</i>)	17	16	94.11764706
3	Pembungkusan Makanan (<i>Food Packaging</i>)	63	62	98.41269841
4	Buah (<i>Fruit</i>)	21	20	95.23809524
5	Kaca (<i>Glass</i>)	34	33	97.05882353
6	Daun (<i>Leaf</i>)	56	55	98.21428571
7	Logam (<i>Metal</i>)	21	20	95.23809524
8	Kertas (<i>Paper</i>)	61	60	98.36065574
9	Plastik (<i>Plastic</i>)	52	51	98.07692308
10	Getah (<i>Rubber</i>)	14	13	92.85714286
11	Kayu (<i>Wood</i>)	27	26	96.29629630

4.3.4 Identifikasi sampah menggunakan ResNet18

Selepas menguji pengesanan sampah menggunakan seni bina pra-latih ResNet18, keputusan kemajuan latihan diperolehi seperti ditunjukkan dalam Rajah 4.23.



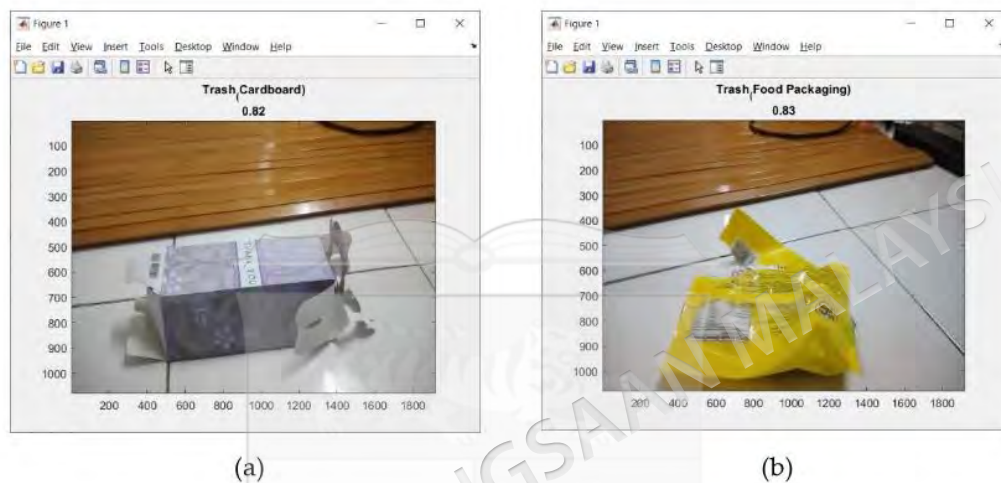
Rajah 4.23 Hasil Latihan Kemajuan Menggunakan Seni Bina Pra-latih ResNet18

Pada Rajah 4.23 tersebut dapat dilihat bahawa Masa Latihan untuk ResNet18 ialah 335 saat, dan Ketepatan Pengesanan (*validation*) ialah 80.75%. Setelah hasil kemajuan latihan diperolehi, selanjutnya sistem identifikasi sampah ini untuk prestasinya menggunakan matriks kekeliruan. Rajah 4.24 menunjukkan keputusan matriks kekeliruan dengan pra-latih ResNet18.

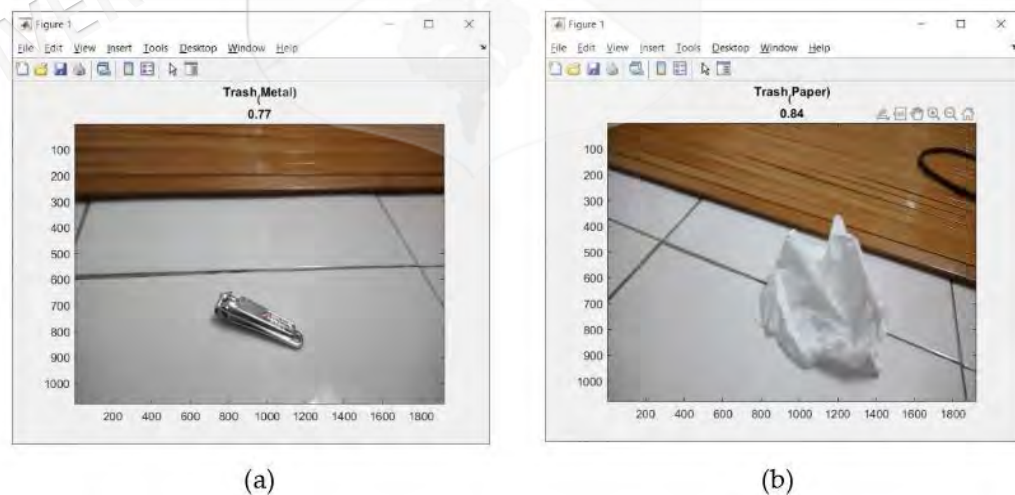
Confusion Matrix	
Output Class	Target Class
Trash (Cardboard)	360 9.1% 0.0% 0.0% 0.0% 0.0% 0.0% 0.0% 0.0% 0.0% 0.0% 0.0% 100% 0.0%
Trash (Fabric)	0 0.0% 360 9.1% 0.0% 0.0% 0.1% 0.1% 0.0% 0.0% 0.0% 0.0% 0.0% 97.8% 2.2%
Trash (Food Packaging)	0 0.0% 0 0.0% 354 8.9% 0.0% 32 0.8% 0 0.0% 0 0.0% 23 0.6% 0 0.0% 86.6% 13.4%
Trash (Fruit)	0 0.0% 0 0.0% 0 0.0% 360 9.1% 0.0% 0.0% 0.0% 0.0% 0.0% 0.0% 98.9% 1.1%
Trash (Glass)	0 0.0% 0 0.0% 1 0.0% 0 0.0% 242 6.1% 0.0% 0.2% 0 0.0% 0 0.0% 0 0.0% 96.0% 4.0%
Trash (Leaf)	0 0.0% 0 0.0% 1 0.0% 0 0.0% 3 0.1% 362 8.9% 0.0% 8 0.2% 0 0.0% 0 0.0% 96.7% 3.3%
Trash (Metal)	0 0.0% 0 0.0% 0 0.0% 0 0.0% 0 0.0% 0 0.0% 347 8.8% 0 0.0% 0 0.0% 0 0.0% 100% 0.0%
Trash (Paper)	0 0.0% 0 0.0% 2 0.1% 0 0.0% 58 1.6% 0 0.0% 0 0.0% 352 8.9% 0 0.0% 0 0.0% 85.4% 14.6%
Trash (Plastic)	0 0.0% 0 0.0% 2 0.1% 0 0.0% 7 0.2% 0 0.0% 0 0.0% 0 0.0% 337 8.5% 0 0.0% 97.4% 2.6%
Trash (Rubber)	0 0.0% 0 0.0% 0 0.0% 0 0.0% 0 0.0% 0 0.0% 0 0.0% 0 0.0% 360 9.1% 0 0.0% 100% 0.0%
Trash (Wood)	0 0.0% 0 0.0% 0 0.0% 0 0.0% 14 0.4% 4 0.1% 0 0.0% 0 0.0% 0 0.0% 360 9.1% 95.6% 4.4%
	100% 100% 88.3% 100% 67.2% 97.8% 96.4% 97.8% 93.8% 100% 100% 95.6% 4.4%
	0.0% 0.0% 1.7% 0.0% 32.8% 2.2% 3.0% 2.2% 6.4% 0.0% 0.0%
Trash (Cardboard)	Trash (Fabric)
Trash (Food Packaging)	Trash (Fruit)
Trash (Glass)	Trash (Leaf)
Trash (Metal)	Trash (Paper)
Trash (Plastic)	Trash (Rubber)
Trash (Wood)	

Rajah 4.24 Hasil matriks kekeliruan dengan pra-latih ResNet18

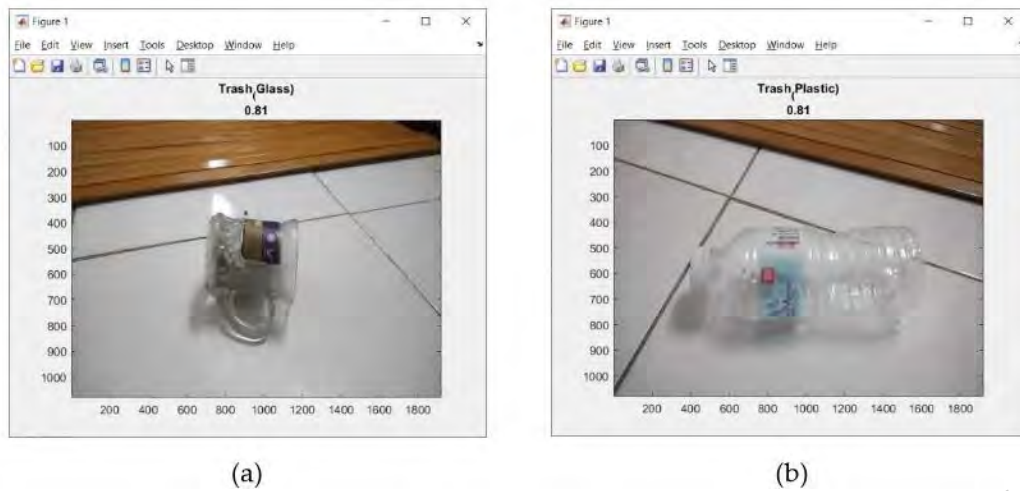
Selepas proses latihan selesai, ujian pengecaman sampah dijalankan dalam masa nyata menggunakan kamera IP. Prosedur pemasangan kamera IP telah dijelaskan dalam sub-bab 3.2.4 dalam bahagian modul program ujian kamera IP. Sampel keputusan ujian boleh dilihat dalam Rajah 4.25 hingga Rajah 4.28.



Rajah 4.25 Hasil identifikasi sampah menggunakan kamera IP. (a) Kadbod; (b)Pembungkus makanan

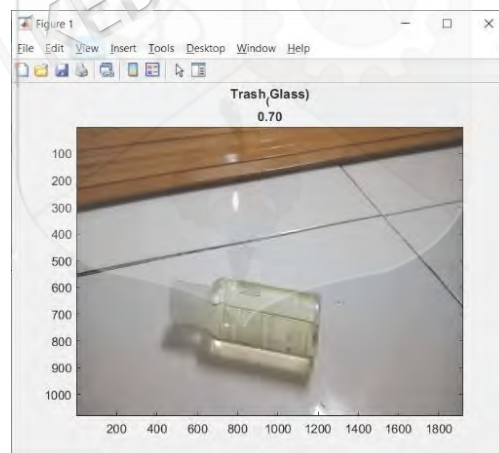


Rajah 4.26 Hasil identifikasi sampah menggunakan kamera IP. (a) Logam; (b) Kertas



Rajah 4.27 Hasil identifikasi sampah menggunakan kamera IP. (a) Kaca; (b) Plastik

Pada masa ujian, terdapat juga ralat dalam identifikasi sampah, seperti yang ditunjukkan dalam Rajah 4.28. Objek sampah diidentifikasi sebagai kaca, walaupun objek sampah adalah plastik.



Rajah 4.28 Hasil identifikasi sampah yang salah

Ujian yang dilakukan dalam Rajah 4.24 hingga Rajah 4.28 digunakan untuk menguji prestasi seni bina CNN pra-latihan iaitu VGG16 dalam identifikasi sampah dalam masa nyata. Bilangan yang diuji ialah 411 keping sampah. Kemudian, rekod jumlah sampah yang dikesan untuk setiap jenis sampah. Ujian ini menggunakan

sampah yang tidak pernah digunakan dalam latihan. Jadual 4.10 menunjukkan keputusan ketepatan yang diperolehi oleh setiap seni bina CNN pra-latihan.

Jadual 4.10 Hasil ketepatan yang dicapai oleh seni bina CNN pra-latihan ResNet18

No.	Jenis Sampah	Bilangan Ujian Sampah Sebenar	Jumlah Sampah Dikesan Dengan Betul	Ketepatan
1	Kadbod (<i>Cardboard</i>)	45	43	95.55555556
2	Fabrik (<i>Fabric</i>)	17	15	88.23529412
3	Pembungkusan Makanan (<i>Food Packaging</i>)	63	61	96.82539683
4	Buah (<i>Fruit</i>)	21	20	95.23809524
5	Kaca (<i>Glass</i>)	34	33	97.05882353
6	Daun (<i>Leaf</i>)	56	54	96.42857143
7	Logam (<i>Metal</i>)	21	20	95.23809524
8	Kertas (<i>Paper</i>)	61	59	96.72131148
9	Plastik (<i>Plastic</i>)	52	50	96.15384615
10	Getah (<i>Rubber</i>)	14	13	92.85714286
11	Kayu (<i>Wood</i>)	27	26	96.2962963

Jadual 4.11 menunjukkan keputusan kemajuan latihan yang boleh dicapai oleh setiap seni bina CNN pra-latihan. Manakala Jadual 4.12 menunjukkan purata ketepatan yang boleh dicapai oleh setiap seni bina CNN pra-latihan.

Jadual 4.11 Keputusan kemajuan latihan

No.	Seni Bina CNN	Masa Latihan (saat)	Ketepatan Pengesahan (<i>Validation</i>) (%)
1	AlexNet	229	77.54
2	VGG16	894	79.09
3	GoogleNet	581	86.38
4	ResNet18	335	80.75

Jadual 4.12 Purata Ketepatan CNN Pra-Latihan dalam identifikasi Sampah Masa Nyata

No.	Seni Bina CNN	Purata Ketepatan (%)
1	AlexNet	79.410
2	VGG16	86.588
3	GoogleNet	96.513
4	ResNet18	95.146

Daripada Jadual 4.11 dapat dilihat bahawa hasil kemajuan latihan tertinggi dimiliki oleh seni bina GoogleNet iaitu bersamaan dengan 86.38% dan hasil kemajuan

latihan terendah dimiliki oleh seni bina AlexNet. Namun dari segi masa latihan, keputusan terendah dimiliki oleh AlexNet iaitu 229 saat dan masa latihan tertinggi dimiliki oleh VGG16 iaitu 894 saat.

Daripada Jadual 4.12 dapat dilihat bahawa purata ketepatan tertinggi dalam identifikasai sampah masa nyata dimiliki oleh GoogleNet, iaitu 96,513%, dan ketepatan terendah dimiliki oleh AlexNet, iaitu 79,410%. Walau bagaimanapun, apabila dilihat daripada hasil kemajuan latihan dan ketepatan purata, seni bina ResNet18 mempunyai hasil yang terbaik. Ini dapat dilihat dari panjang masa latihan yang terhasil iaitu bersamaan 335 saat, tidak jauh bezanya dengan AlexNet yang mempunyai masa latihan selama 229 saat. Walau bagaimanapun, purata ketepatan yang terhasil ialah 95.146%, tidak jauh ketinggalan berbanding GoogleNet.

4.4 HASIL IMBASAN PENDERIA LIDAR

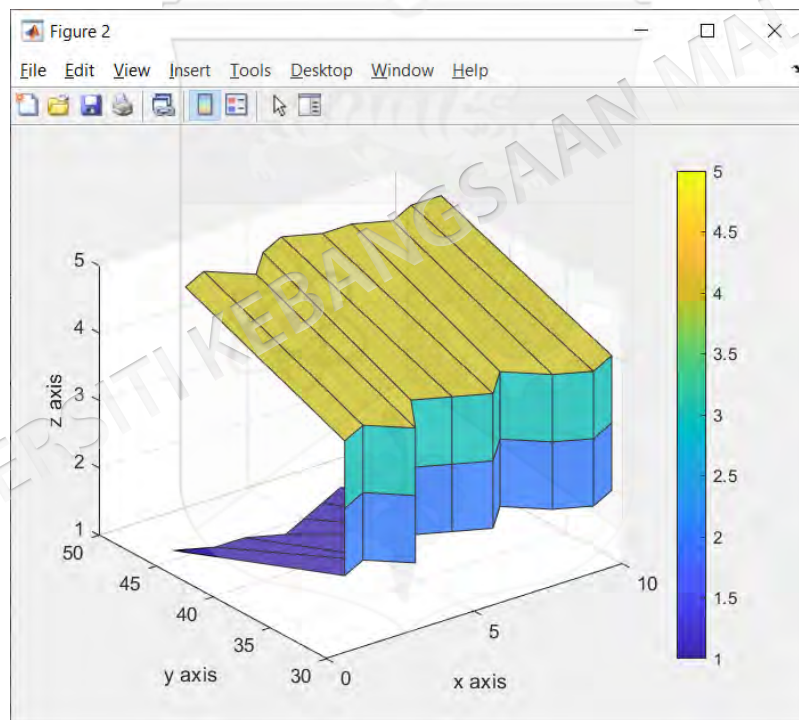
Hasil imbasan penderia LiDAR ini adalah sangat penting, kerana ini adalah sebahagian daripada menentukan sama ada sampah itu dikategorikan sebagai objek 3D atau tidak (2D). Mengimbas imej terperinci agak memakan masa kerana keupayaan kadar bingkai penderia LiDAR hanya 5 Hz. Jika setiap kali ia menjumpai objek yang dianggap sampah, robot mesti mengimbas imej terperinci, jadi keputusan untuk mengangkat objek akan mengambil masa yang lebih lama. Atas sebab ini, sistem ini dilengkapi dengan pengimbasan pantas. Dengan pengimbasan pantas ini, jika objek yang dianggap sampah tidak dalam bentuk 3D, maka proses pengangkatan oleh pencengkam robot diabaikan. Kecekapan masa ini boleh dilakukan, dan robot boleh mencari objek sampah seterusnya.

Dalam kajian ini, kaedah Sequential_Camera_LiDAR (SCL) dicadangkan. Kaedah ini agak mudah, menggabungkan kamera dan penderia LiDAR. CNN memproses output kamera, diikuti dengan proses pengimbasan pantas oleh penderia LiDAR. Perbandingan kaedah yang dicadangkan dengan kaedah yang terdahulu adalah seperti di Jadual 2.4. Namun begitu, kajian yang dicadangkan ini boleh digunakan di perkarangan rumah. Kaedah yang dicadangkan berbeza daripada kaedah sebelumnya dalam Jadual 2.4 (Penyatuan Awal, Gabungan Berjujukan, Gabungan Lewat). Ketiga-tiga kaedah ini digunakan untuk kenderaan yang mesti mempunyai

kelajuan yang agak tinggi berbanding dengan kelajuan robot yang digunakan. Selain itu, tindak balas kenderaan terhadap halangan di hadapan kamera secara relatifnya lebih pantas berbanding pergerakan robot di perkarangan rumah. Jadi kaedah yang dicadangkan sesuai digunakan di perumahan, tetapi perlu menyesuaikan parameter sedia ada untuk digunakan di lebuhraya.

4.4.1 Hasil imbasan imej pantas

Pengimbasan imej pantas digunakan untuk memastikan imej yang diidentifikasi adalah imej dalam 2D atau 3D. Sampel keputusan imej pengimbasan pantas boleh dilihat dalam Rajah 4.29.

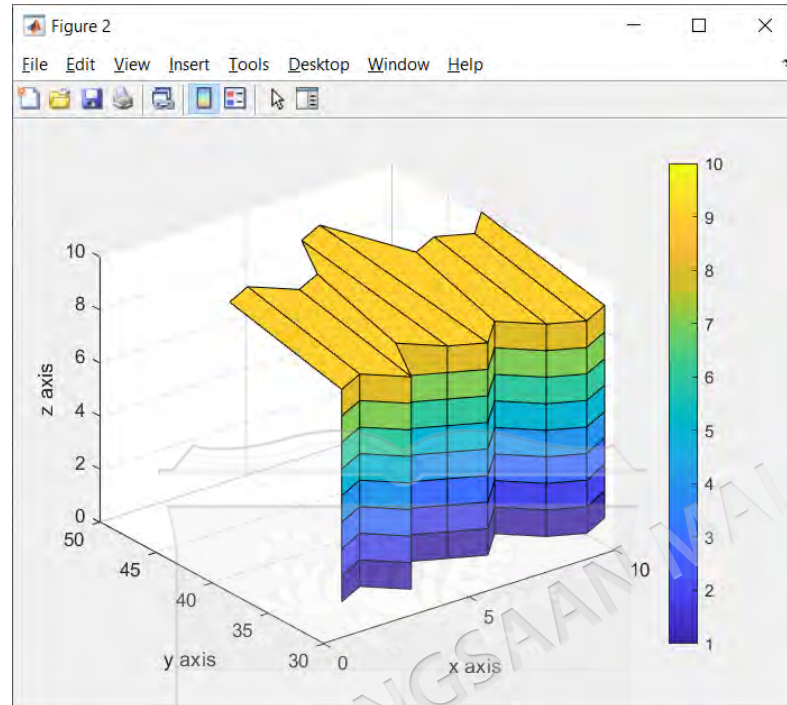


Rajah 4.29 Sampel hasil pengimbasan imej objek sampah pantas

4.4.2 Hasil imbasan imej terperinci

Selepas proses pengimbasan pantas menyatakan objek yang diperhatikan ialah objek dalam 3D, proses imej pengimbasan terperinci dijalankan. Fungsi imej pengimbasan terperinci ini adalah untuk memastikan kedudukan pencengkam robot semasa

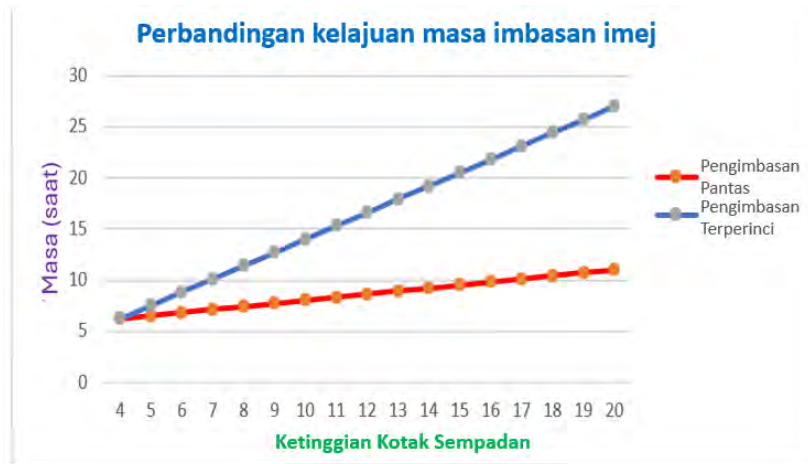
mengangkat sampah. Sampel keputusan imej pengimbasan pantas boleh dilihat dalam Rajah 4.30.



Rajah 4.30 Sampel hasil pengimbasan imej objek sampah terperinci

Dalam Rajah 4.30, dapat dilihat bahawa keputusan imej pengimbasan terperinci adalah lebih baik daripada keputusan imej pengimbasan pantas. Ini kerana jumlah data pada paksi-y dan paksi-z adalah lebih daripada jumlah data pada paksi-y dan z dalam imej pengimbasan pantas.

Salah satu pengoptimuman dalam proses identifikasi sampah adalah untuk menjadikan masa cekap dalam identifikasi objek sampah. Oleh itu, adalah perlu untuk mengkaji perbezaan antara imej pengimbasan pantas dan terperinci, seperti yang digambarkan dalam Rajah 4.31. Kecekapan penggunaan masa yang digunakan dalam proses identifikasi sampah akan diperoleh daripada keputusan ujian ini.



Rajah 4.31 Perbandingan masa antara pengimbasan pantas dan terperinci

Dalam Rajah 4.31, dapat dilihat keputusan perbandingan masa antara pengimbasan pantas dan terperinci. Kecekapan masa berjulat dari 13.33% hingga 59.26%, bergantung pada saiz objek yang dikesan. Lebih besar objek, lebih besar kecekapan masa. Lebih besar objek, lebih banyak masa yang diperlukan. Pengujian terhadap kepada objek bersaiz 15 cm × 20 cm. Nilai saiz ini disebabkan oleh saiz dan geometri pencengkam robot.

4.5 HASIL NAVIGASI ROBOT BERAUTONOMI

Ujian navigasi robot autonomi telah dijalankan menggunakan dua buah peta. Setiap peta telah diuji menggunakan algoritma: Dijkstra, A*, dan FDCC-PP. Terdapat dua parameter yang direkodkan iaitu purata masa perjalanan dan panjang laluan.

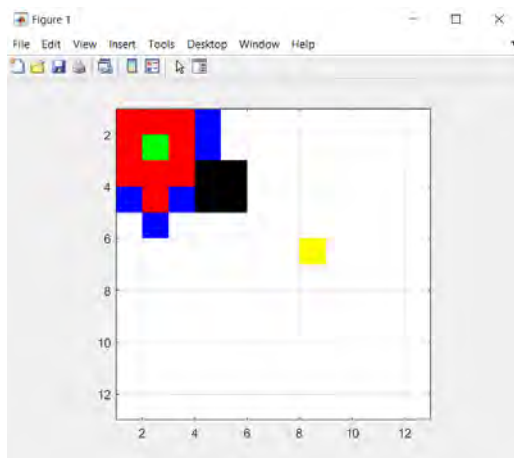
4.5.1 Hasil Pengujian Menggunakan Peta Pertama

Peta pertama menggunakan satu halangan di hadapan robot. Robot akan bergerak dari mula ke matlamat mengikut algoritma yang digunakan pada robot.

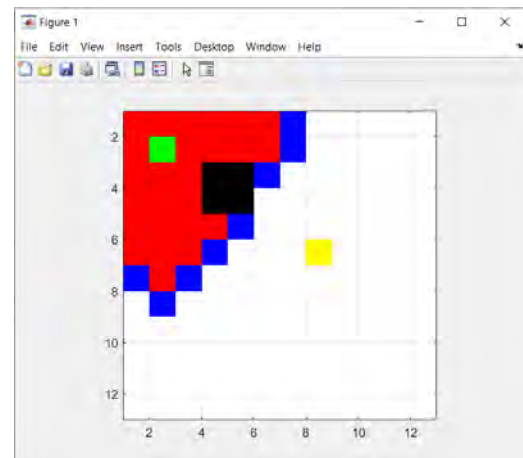
a. Hasil Pengujian Navigasi Menggunakan Algoritma Dijkstra

Rajah 4.32 ialah hasil ujian peta pertama menggunakan algoritma Dijkstra dengan simulasi dalam Matlab. Manakala, Rajah 4.33 adalah hasil daripada ujian algoritma Dijkstra dalam masa nyata.

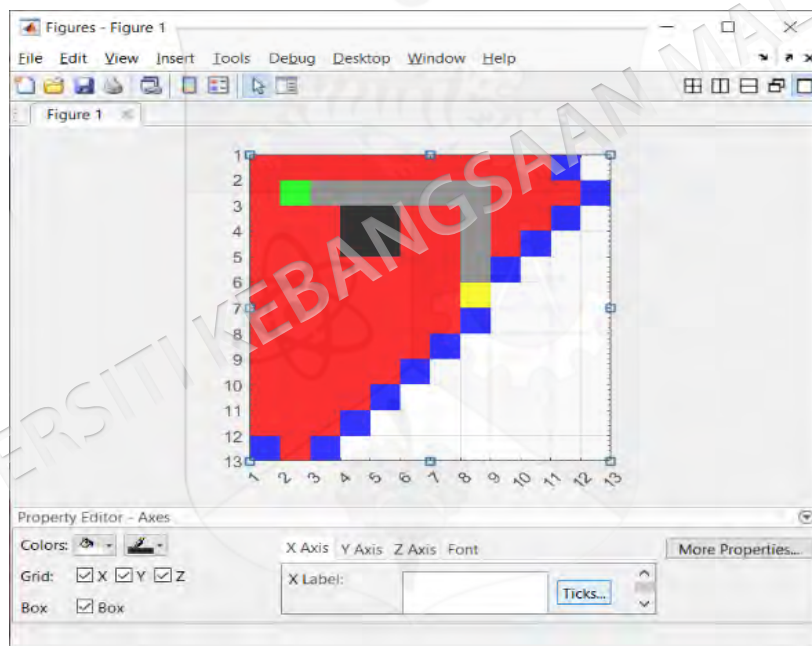
i. Hasil simulasi algoritma Dijkstra dengan Matlab



(a)



(b)



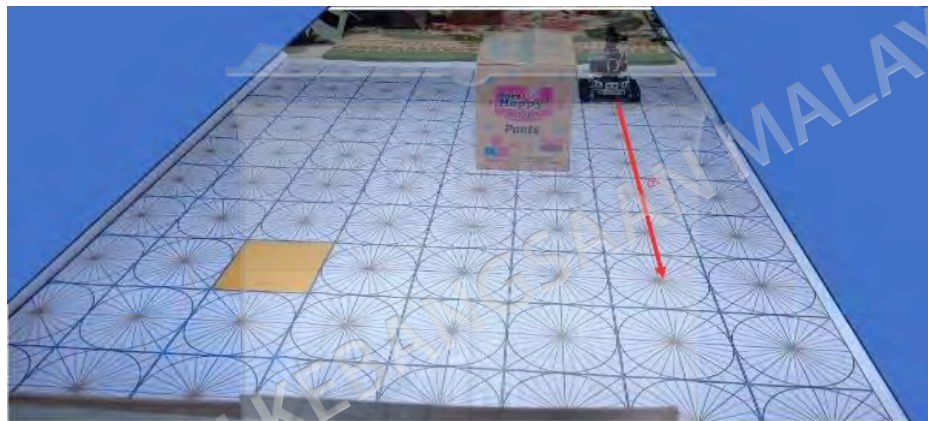
(c)

Rajah 4.32 Hasil simulasi algoritma Dijkstra dengan Matlab

Dalam Rajah 4.32 dapat dilihat bahawa setiap grid sel peta robot ditandakan dengan warna (*color map*). Warna hijau menunjukkan titik mula, warna kuning adalah titik matlamat. Setiap grid sel yang telah dilawati ditandakan dengan warna merah. Warna biru ialah grid sel yang akan dilawati, manakala warna putih ialah grid sel yang tidak dilawati. Ini berlaku kerana algoritma Dijkstra telah menemui grid sel untuk matlamat robot.

Dalam Rajah 4.32 (a), algoritma Dijkstra mengesan grid sel sekitar titik permulaan robot. Seterusnya, proses melawat grid sel ini terus diulang seperti yang ditunjukkan dalam Rajah 4.32 (b) sehingga titik matlamat ditemui. Setiap grid sel ditandakan dan jika grid sel matlamat belum ditemui dan masih terdapat ruang kosong, maka grid sel itu akan dilawati sehingga titik matlamat dilawati. Jika matlamat telah ditemui, algoritma Dijkstra akan mencari laluan terpendek dari titik mula ke matlamat. Dalam Rajah 4.32 (c) dapat dilihat bahawa bilangan grid sel yang dilawati ialah 75 grid sel. Selepas matlamat ditemui, laluan terpendek dari mula ke matlamat ialah 10 grid sel.

ii. Hasil pengujian algoritma Dijkstra dalam masa nyata



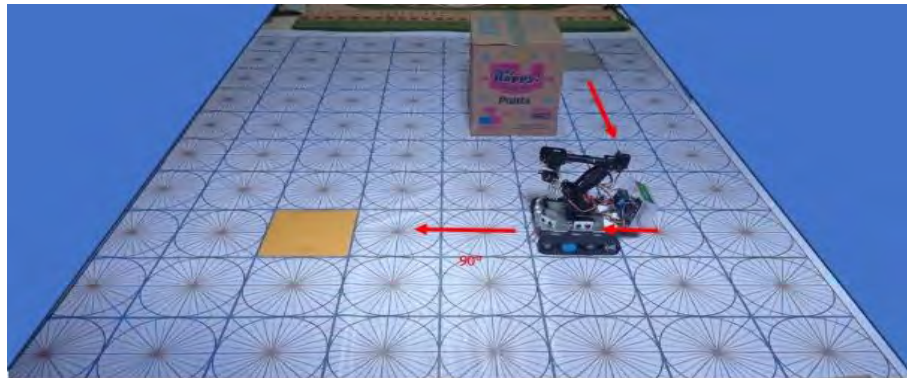
(a)



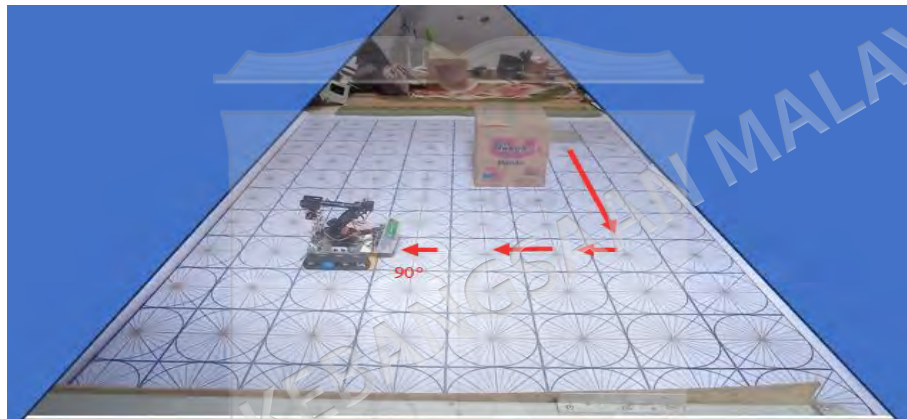
(b)

bersambung...

sambungan...



(c)



(d)

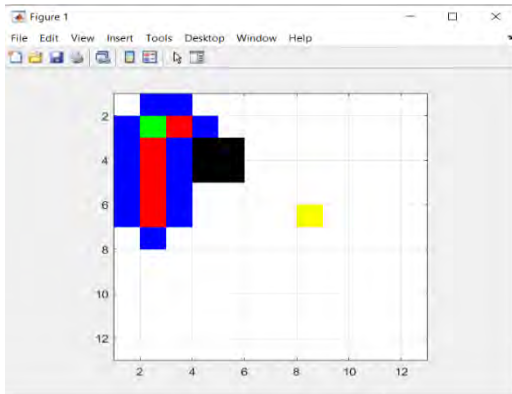
Rajah 4.33 Hasil pengujian algoritma Dijkstra dalam masa nyata

Pengujian algoritma dalam masa nyata diperlukan untuk membuktikan bahwa laluan robot pada masa nyata sesuai dengan hasil simulasi sebelumnya. Dalam Rajah 4.33 dapat dilihat bahawa robot bergerak lurus ke hadapan untuk 6 grid sel. Selepas itu robot membelok 90 darjah ke kanan. Kemudian robot bergerak ke hadapan 4 grid sel dan akhirnya menemui matlamat. Jumlah panjang laluan ialah 2.5 m.

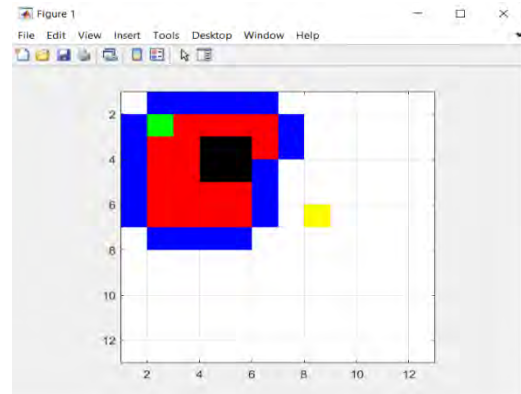
b. Hasil Pengujian Navigasi Menggunakan Algoritma A*

Rajah 4.34 menunjukkan hasil ujian peta pertama menggunakan algoritma A* dengan simulasi dalam Matlab. Jika dilihat daripada keputusan ujian menggunakan algoritma Dijkstra, algoritma A* mempunyai lebih sedikit grid sel yang dilawati iaitu 30 grid sel.

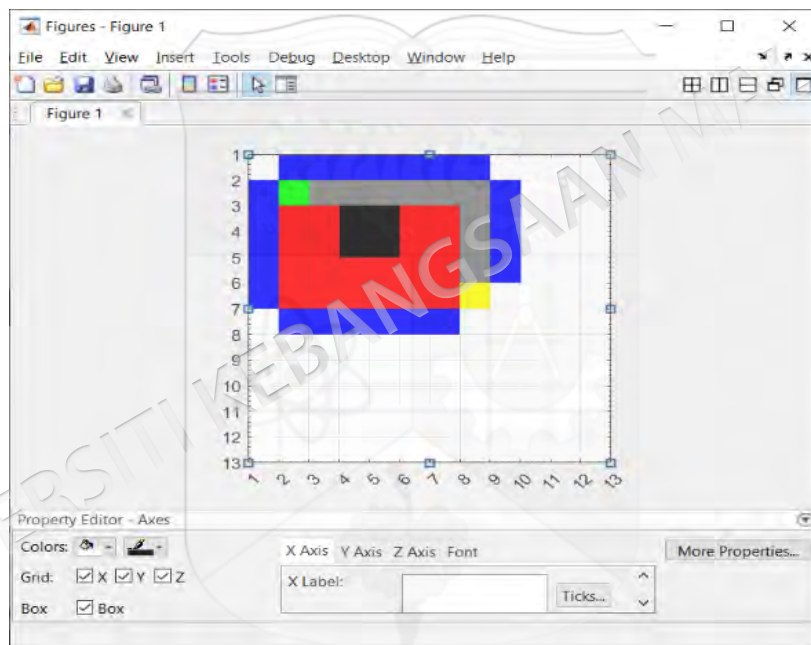
i. Hasil simulasi algoritma A* dengan Matlab



(a)



(b)



(c)

Rajah 4.34 Hasil simulasi algoritma A* dengan Matlab

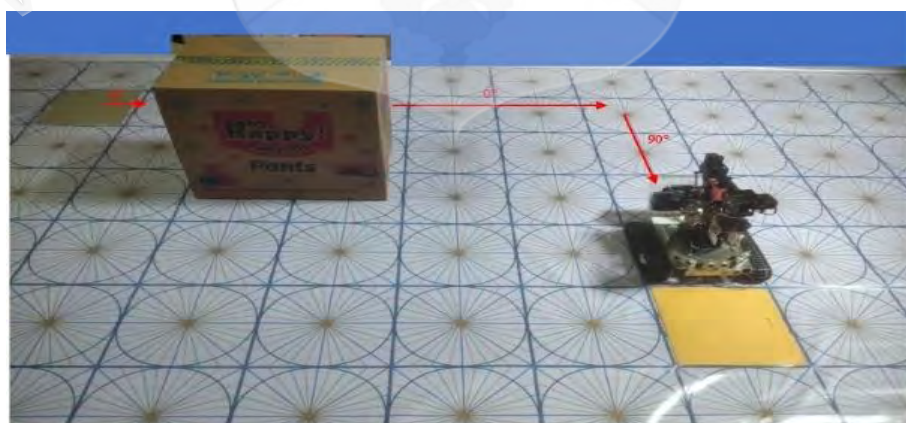
ii. Hasil pengujian algoritma A* dalam masa nyata



(a)



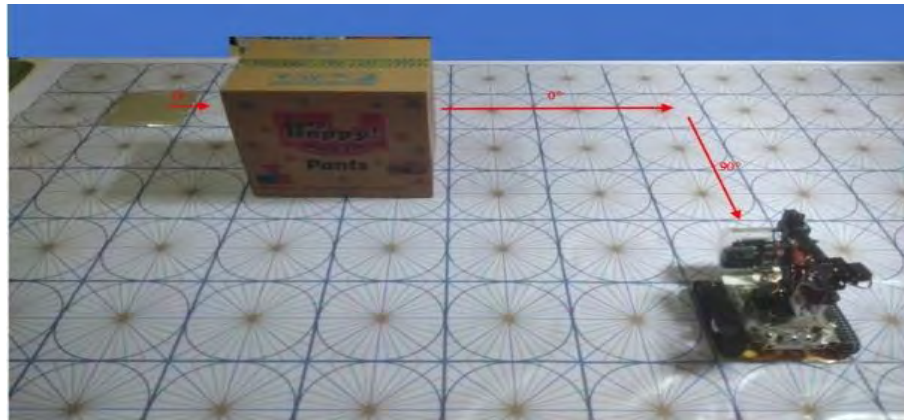
(b)



(c)

bersambung...

sambungan...



(d)

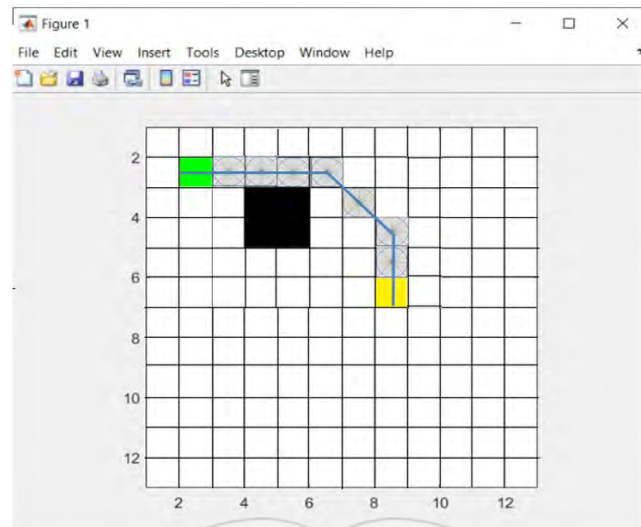
Rajah 4.35 Hasil pengujian algoritma A* dalam masa nyata

Selepas ujian simulasi dijalankan, robot diuji pada laluan masa nyata. Rajah 4.35 menunjukkan hasil daripada ujian algoritma A* dalam masa nyata. Robot bergerak lurus ke hadapan untuk 6 grid sel. Selepas itu robot membelok 90 darjah ke kanan. Kemudian robot bergerak lurus ke hadapan 4 grid sel dan akhirnya menemui matlamat. Jumlah panjang laluan ialah 2.5 m. Laluan pergerakan robot adalah sama seperti algoritma Dijkstra.

c. Hasil Pengujian Navigasi Menggunakan Algoritma FDCC-PP

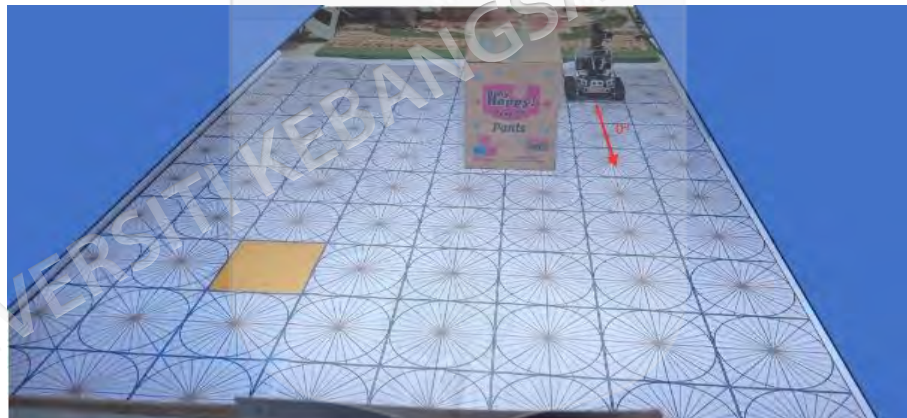
Rajah 4.36 ialah hasil ujian peta pertama menggunakan algoritma FDCC-PP. Dalam Rajah 4.36 ini dapat dilihat bahawa algoritma FDCC-PP tidak memerlukan lawatan ke grid sel di sekitar titik permulaan robot. Di sini laluan terus diperoleh berdasarkan kod rantai pada robot, supaya robot boleh bergerak terus dari titik mula ke matlamat. Jumlah grid sel yang dilawati iaitu 8 grid sel.

i. Hasil Pergerakan Robot Dalam Bentuk Garis

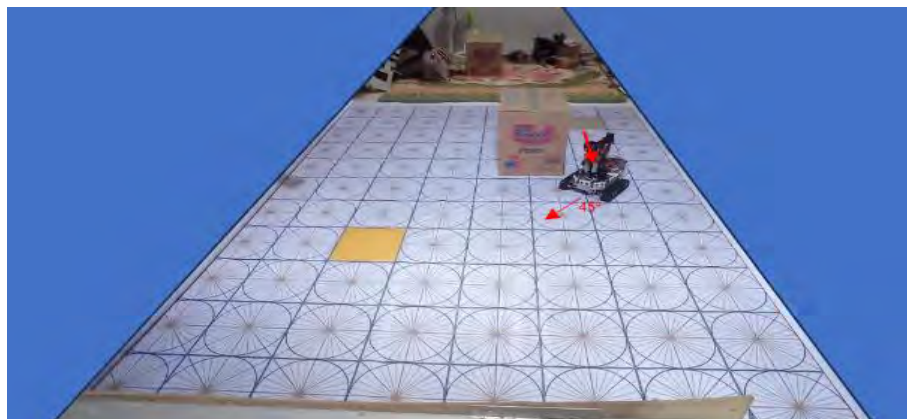


Rajah 4.36 Hasil pergerakan robot dalam bentuk garis

ii. Hasil pengujian algoritma FDCC-PP dalam masa nyata



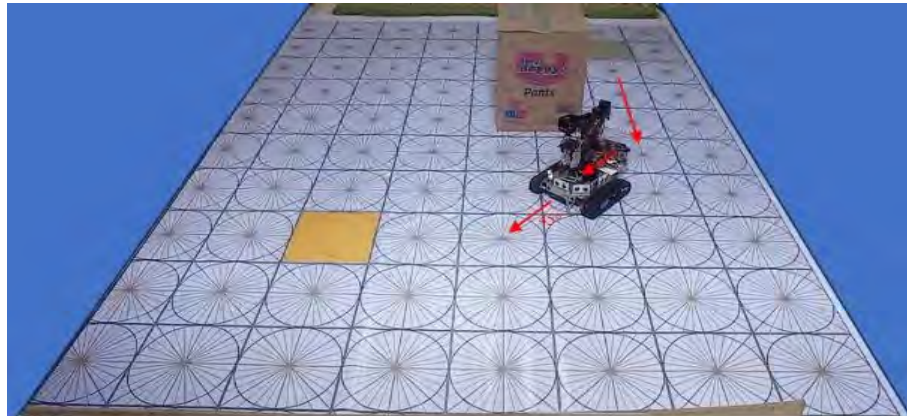
(a)



(b)

bersambung...

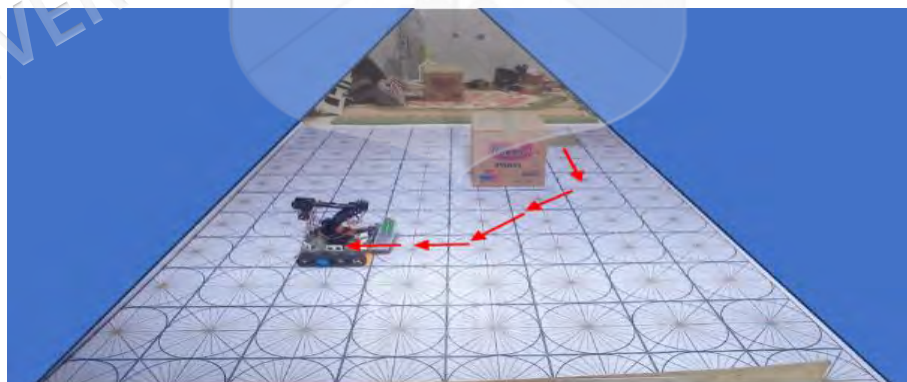
sambungan...



(c)



(d)



(e)

Rajah 4.37 Hasil pengujian algoritma FDCC-PP dalam masa nyata

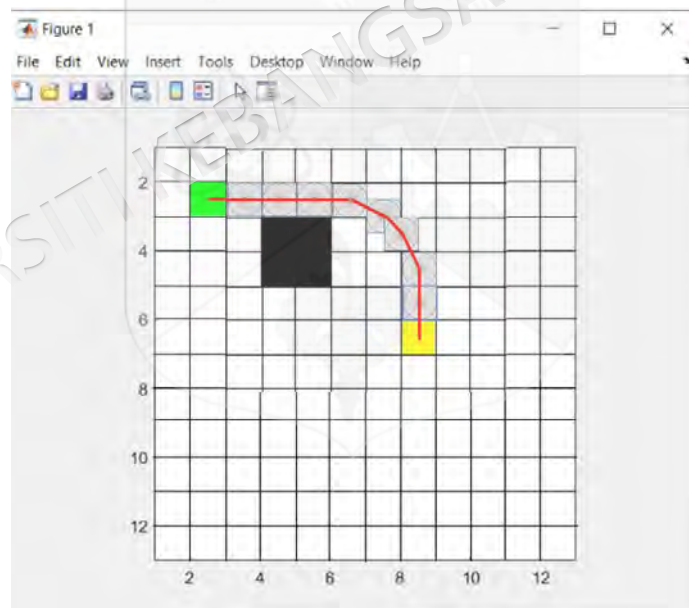
Rajah 4.37 adalah hasil daripada ujian algoritma FDCC-PP dalam masa nyata. Robot bergerak lurus ke hadapan untuk 4 grid sel. Selepas itu robot membelok 45 darjah ke kanan. Selepas melalui 2 grid sel, robot membelok 45 darjah. Kemudian robot berjalan

ke hadapan dalam jarak 2 grid sel dan akhirnya menemui matlamat. Jumlah panjang laluan ialah 2.15 m.

d. Hasil Pengujian Navigasi Menggunakan Algoritma MFCC-PP

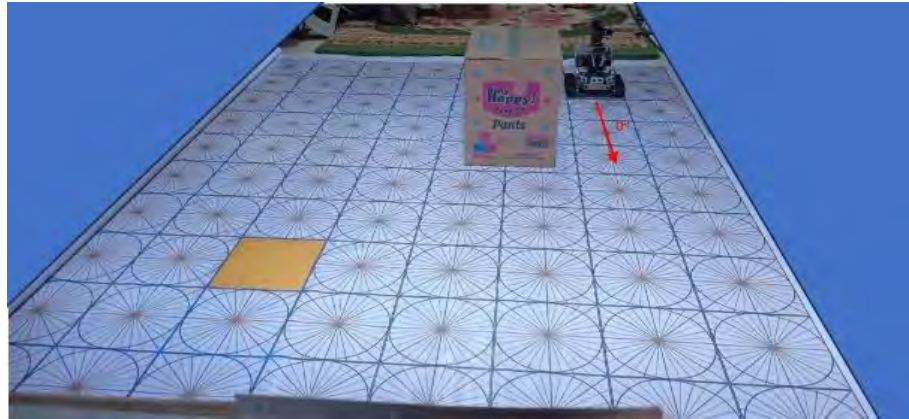
Rajah 4.38 ialah hasil ujian peta pertama menggunakan algoritma MFCC-PP. Algoritma MFCC-PP adalah sama dengan algoritma FDCC-PP, iaitu tidak memerlukan lawatan ke grid sel di sekitar titik permulaan robot. Di sini laluan terus diperoleh berdasarkan kod rantai pada robot, supaya robot boleh bergerak terus dari titik mula ke matlamat. Algoritma MFCC-PP mempunyai keupayaan untuk membelok ke kiri atau kanan sehingga 11.25 darjah. Jadi algoritma ini boleh membelok dengan lebih lancar. Walau bagaimanapun, panjang laluan adalah lebih panjang daripada algoritma FDCC-PP, iaitu 9 grid sel.

i. Hasil Pergerakan Robot Dalam Bentuk Garis

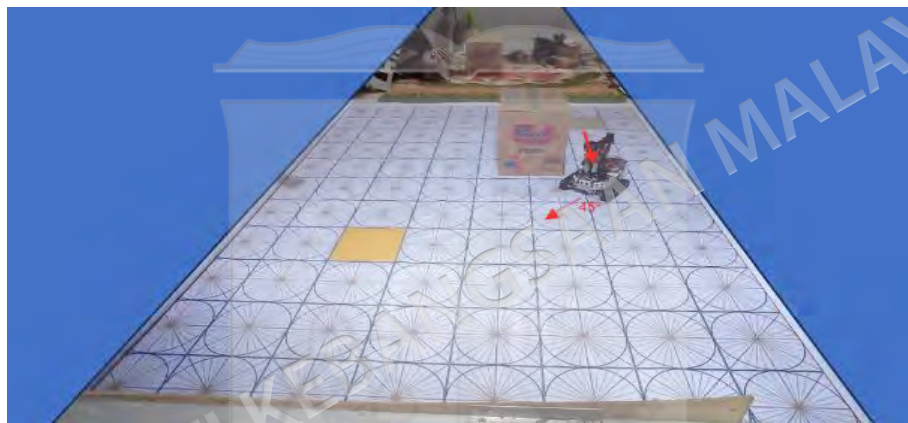


Rajah 4.38 Hasil pergerakan robot dalam bentuk garis

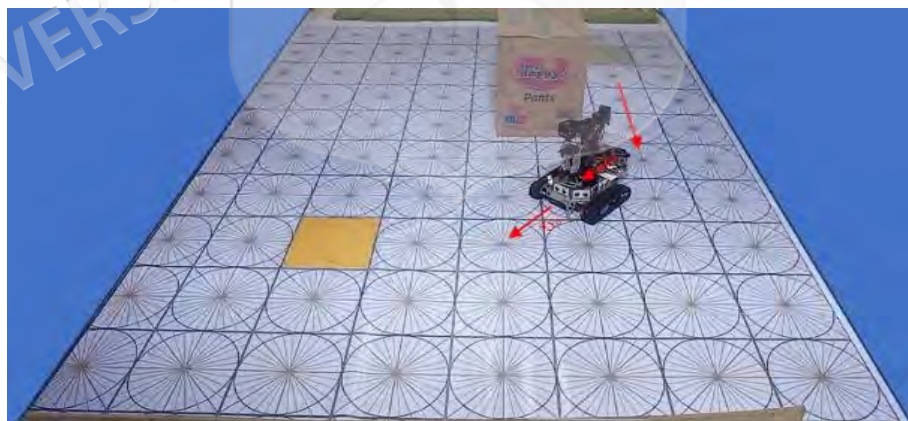
ii. Hasil pengujian algoritma MFCC-PP dalam masa nyata



(a)



(b)



(c)

bersambung...

Rajah 4.39 adalah hasil daripada ujian algoritma MFCC-PP dalam masa nyata. Robot bergerak lurus ke hadapan untuk 4 grid sel. Selepas itu robot membelok 45 darjah ke kanan. Selepas melalui 1 grid sel, robot membelok 20 darjah ke kanan. Kemudian robot membelok 15 darjah ke kanan. Kemudian robot berjalan dalam jarak 2 grid sel dan akhirnya menemui matlamat. Jumlah panjang laluan ialah 2.2 m.

Perbandingan yang ditunjukkan dalam Jadual 4.13 menggambarkan bahawa disebabkan penggunaan algoritma MFCC-PP, kecekapan perancangan laluan robot boleh ditingkatkan sebanyak 71.36% berbanding dengan algoritma A* dalam peta dengan lebih banyak ruang kosong, dan menjadi malah meningkat sebanyak 84.67% berbanding dengan algoritma Dijkstra.

Walau bagaimanapun, walaupun kecekapan carian telah bertambah baik, panjang laluan yang dirancang oleh algoritma MFCC-PP adalah lebih panjang daripada yang dirancang oleh algoritma A*, algoritma Dijkstra dan algoritma FDCC-PP. Bagi algoritma A* dan algoritma Dijkstra, strategi perancangan yang melibatkan perbandingan kos pergerakan antara nod, pengemaskinian berterusan hubungan antara nod membantu robot mengelak daripada memintas banyak halangan, sekaligus mengurangkan panjang laluan.

Berdasarkan peta pertama, panjang laluan yang dirancang oleh algoritma MFCC-PP ialah 5.69% lebih panjang daripada laluan terpendek yang dirancang oleh algoritma Dijkstra. Tetapi pengubahsuaian kod rantai menghasilkan pusingan yang lebih lancar berbanding dengan algoritma lain.

Jadual 4.13 Perbandingan antara algoritma MFCC-PP dan tiga algoritma navigasi lain berdasarkan peta pertama

No.	Algoritma	Grid sel yang dilawati (sel)	Masa Perancangan Purata (saat)	Panjang Laluan (cm)
1	Dijkstra	75	0.7390	250
2	A*	30	0.2970	250
3	FDCC-PP	8	0.0079	215
4	MFCC-PP	9	0.0089	225

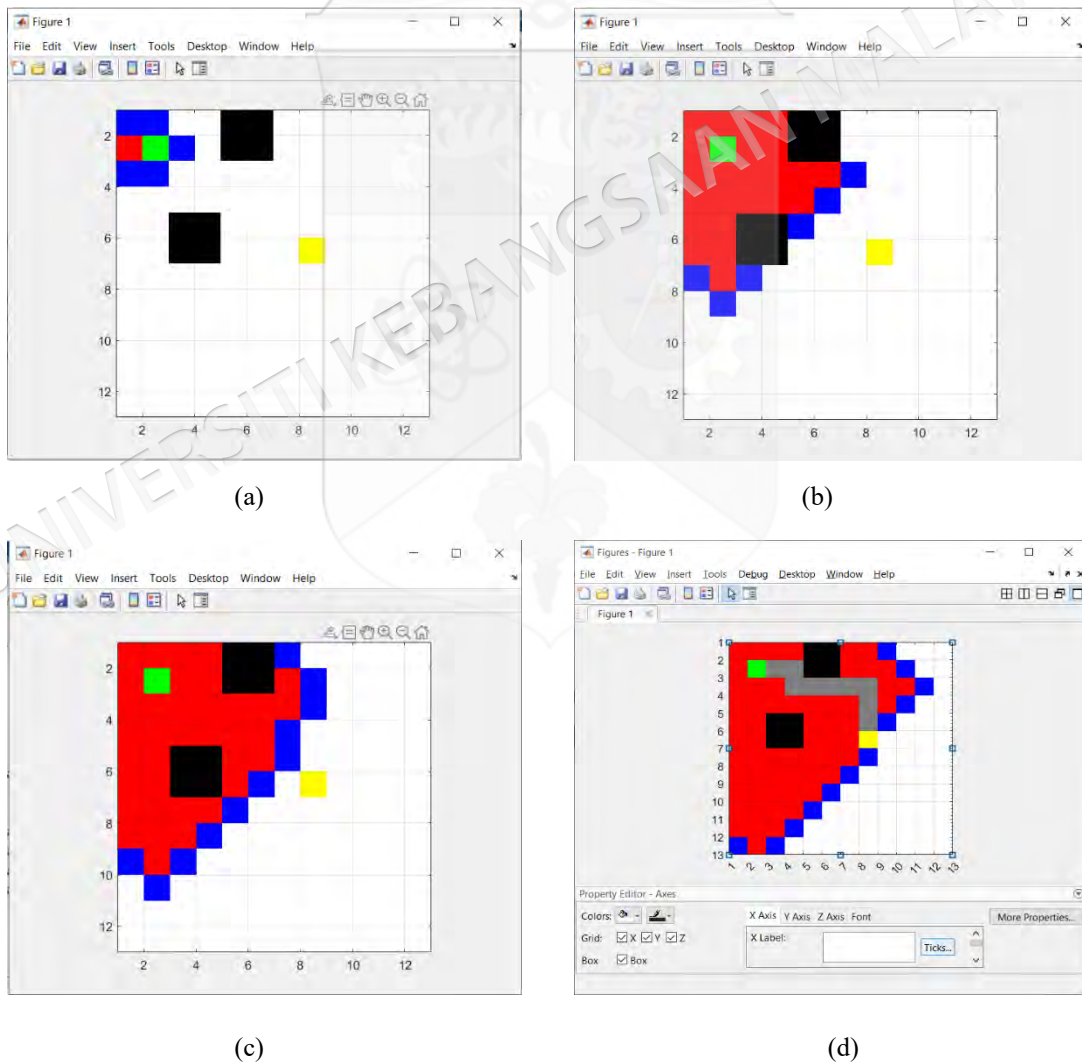
4.5.2 Hasil Pengujian Menggunakan Peta Kedua

Peta kedua menggunakan dua halangan di hadapan robot. Robot akan bergerak dari mula ke matlamat mengikut algoritma yang digunakan pada robot.

a. Hasil Pengujian Navigasi Menggunakan Algoritma Dijkstra

Rajah 4.40 ialah hasil ujian peta kedua menggunakan algoritma Dijkstra dengan simulasi dalam Matlab. Manakala, Rajah 4.41 adalah hasil daripada ujian algoritma Dijkstra dalam masa nyata.

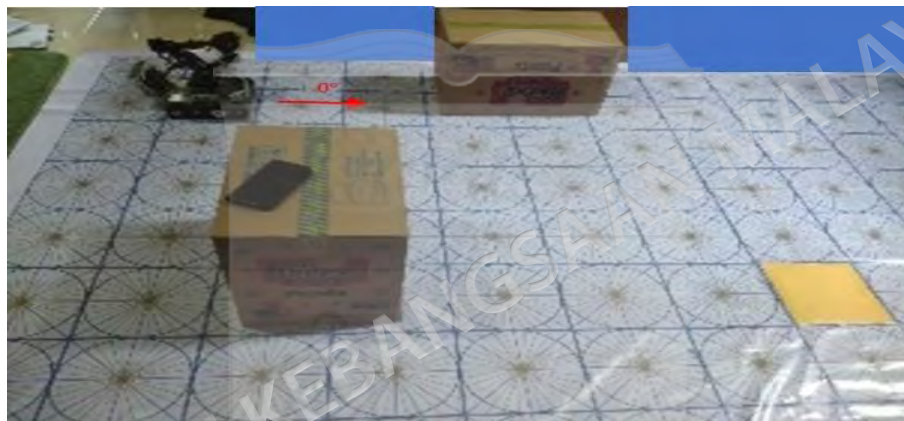
i. Hasil simulasi algoritma Dijkstra dengan Matlab



Rajah 4.40 Hasil simulasi algoritma Dijkstra dengan Matlab

Dalam Rajah 4.40 (a), algoritma Dijkstra mengesan grid sel sekitar titik permulaan robot. Seterusnya, proses melawat grid sel ini terus diulang Rajah 4.40 (b) dan Rajah 4.40 (c) sehingga titik matlamat ditemui. Setiap grid sel ditandakan dan jika grid sel matlamat belum ditemui dan masih terdapat ruang kosong, maka grid sel itu akan dilawati sehingga titik matlamat dilawati. Jika matlamat telah ditemui, algoritma Dijkstra akan mencari laluan terpendek dari titik mula ke matlamat. Dalam Rajah 4.40 (d) dapat dilihat bahawa bilangan grid sel yang dilawati ialah 69 grid sel. Selepas matlamat ditemui, laluan terpendek dari mula ke matlamat ialah 10 grid sel.

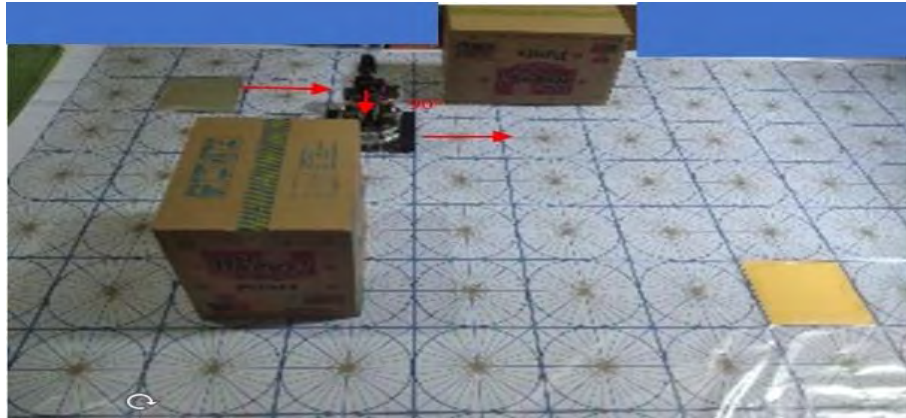
ii. Hasil pengujian algoritma Dijkstra dalam masa nyata



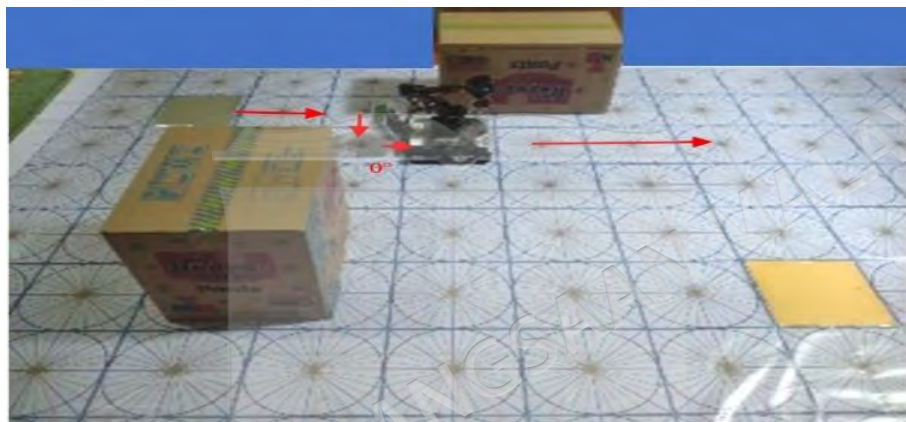
(a)



(b)



(c)



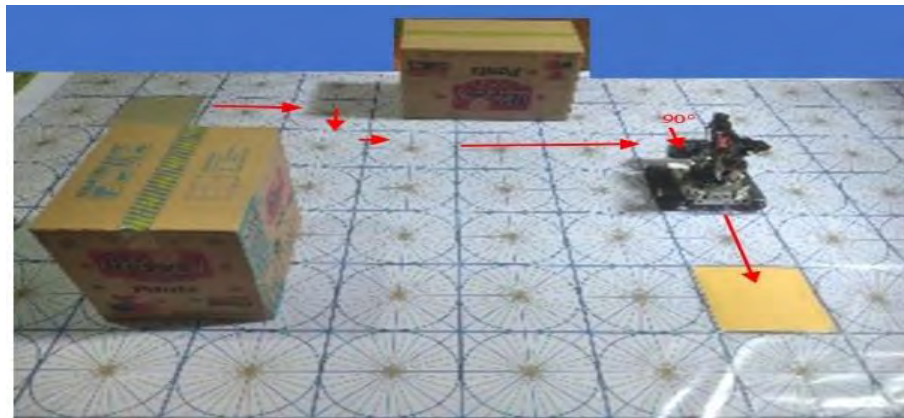
(d)



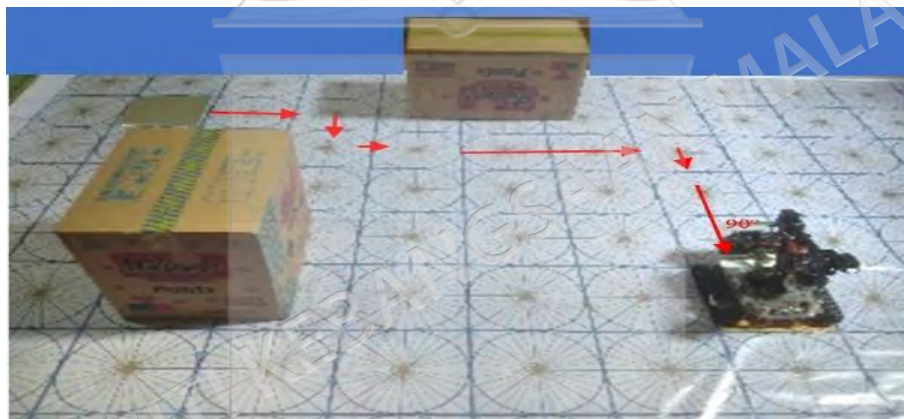
(e)

bersambung...

sambungan...



(f)



(g)

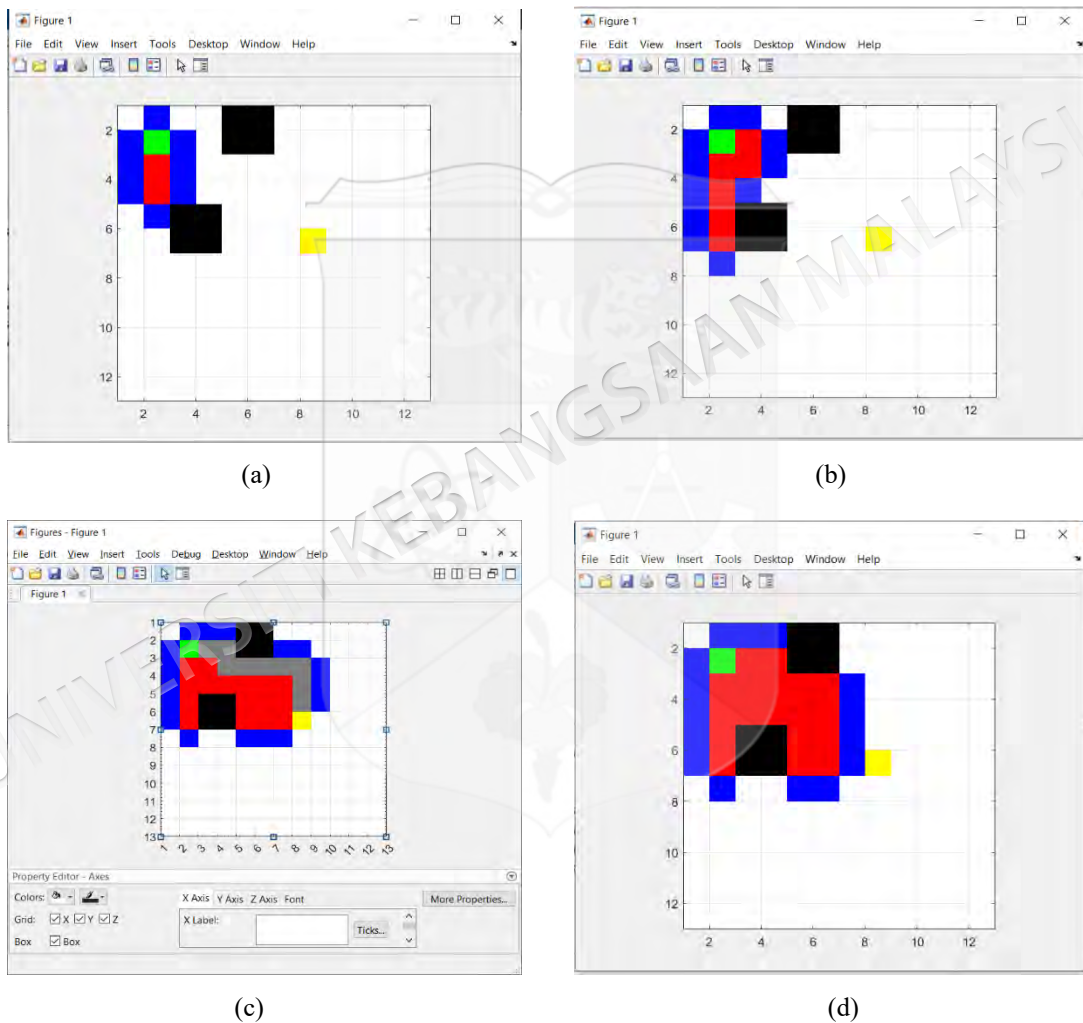
Rajah 4.41 Hasil pengujian algoritma Dijkstra dalam masa nyata

Rajah 4.41 ialah pengujian algoritma Dijkstra dalam masa nyata. Dalam Rajah 4.41 dapat dilihat bahawa robot bergerak lurus ke hadapan untuk 2 grid sel. Selepas itu robot membelok 90 darjah ke kanan. Kemudian robot bergerak ke hadapan 1 grid sel. Selepas itu robot membelok 90 darjah ke kiri. Kemudian robot bergerak ke hadapan 4 grid sel. Selepas itu robot membelok 90 darjah ke kanan. Kemudian robot bergerak ke hadapan 3 grid sel dan akhirnya menemui matlamat. Jumlah panjang laluan ialah 2.5 m.

b. Hasil Pengujian Navigasi Menggunakan Algoritma A*

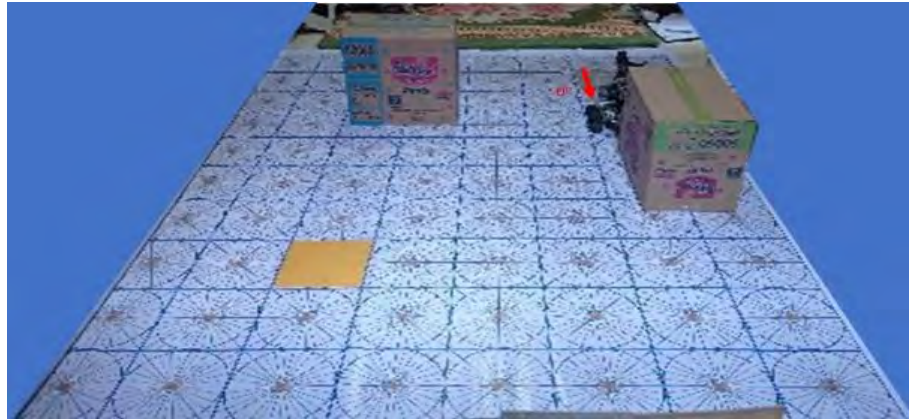
Rajah 4.42 ialah hasil ujian peta kedua menggunakan algoritma A* dengan simulasi dalam Matlab. Jika dilihat daripada keputusan ujian peta kedua menggunakan algoritma Dijkstra, algoritma A* mempunyai lebih sedikit grid sel yang dilawati iaitu 26 grid sel.

i. Hasil simulasi algoritma A* dengan Matlab



Rajah 4.42 Hasil simulasi algoritma A* dengan Matlab

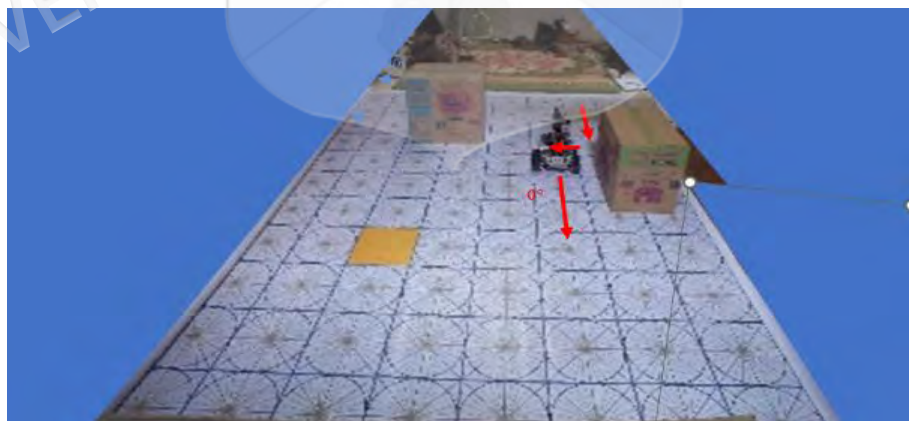
ii. Hasil pengujian algoritma A* dalam masa nyata



(a)



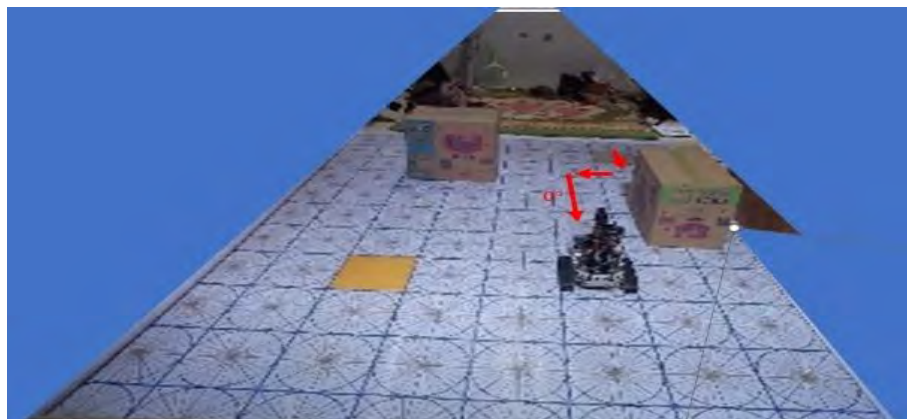
(b)



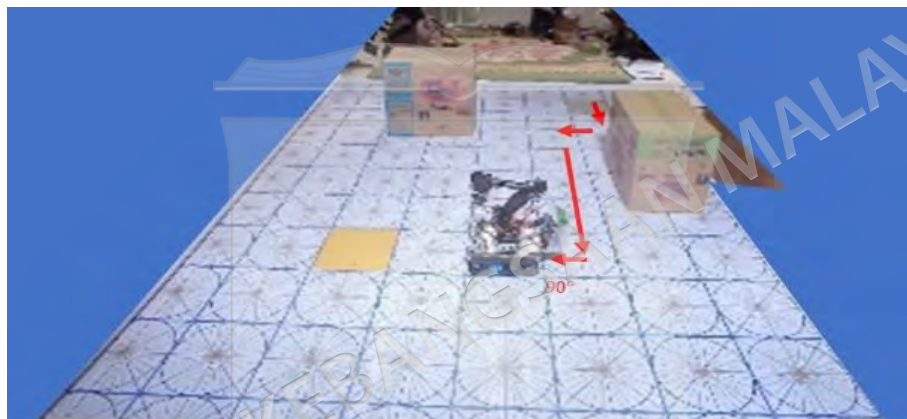
(c)

bersambung...

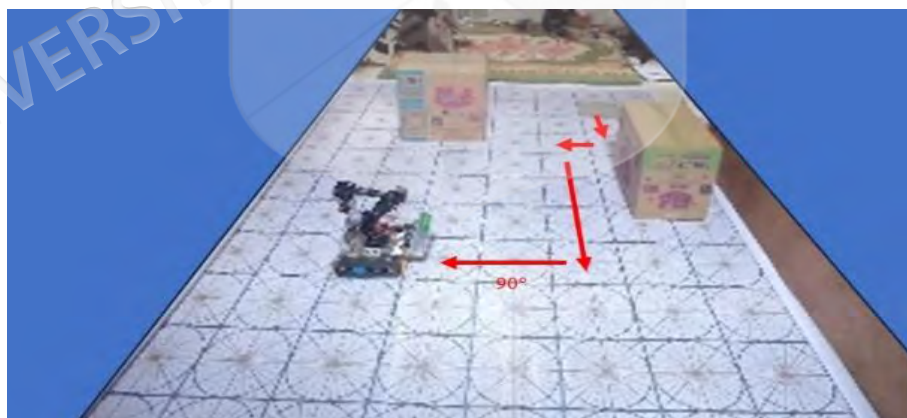
sambungan...



(d)



(e)



(f)

Rajah 4.43 Hasil pengujian algoritma A* dalam masa nyata

Rajah 4.43 ialah pengujian algoritma A* dalam masa nyata untuk peta kedua. Peta kedua ini mempunyai 2 halangan. Dalam Rajah 4.43 dapat dilihat bahawa robot bergerak lurus ke hadapan untuk 2 grid sel. Selepas itu robot membelok 90 darjah ke

kanan. Kemudian robot bergerak ke hadapan 1 grid sel. Selepas itu robot membelok 90 darjah ke kiri. Kemudian robot bergerak ke hadapan 4 grid sel. Selepas itu robot membelok 90 darjah ke kanan. Kemudian robot bergerak ke hadapan 3 grid sel dan akhirnya menemui matlamat. Jumlah panjang laluan ialah 2.5 m.

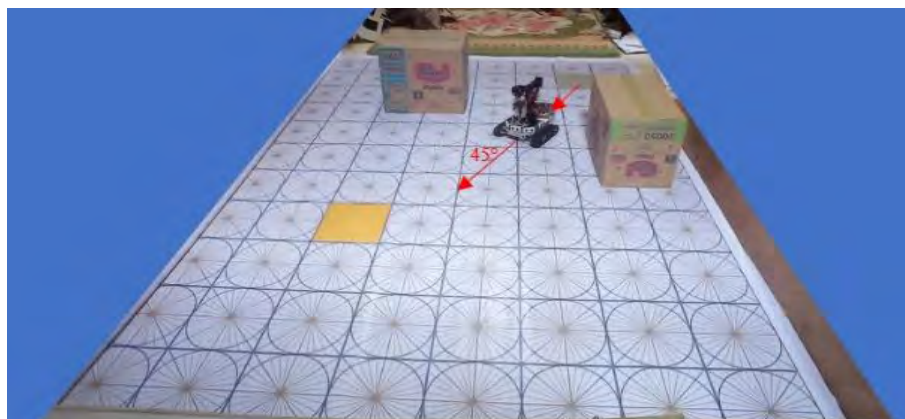
c. Hasil Pengujian Navigasi Menggunakan Algoritma FDCC-PP

Rajah 4.44 adalah hasil daripada ujian peta kedua menggunakan algoritma FDCC-PP dalam masa nyata. Peta kedua ini mempunyai 2 halangan. Dalam Rajah 4.44 ini dapat dilihat bahawa algoritma FDCC-PP tidak memerlukan lawatan ke grid sel di sekitar titik permulaan robot. Di sini laluan terus diperoleh berdasarkan kod rantai pada robot, supaya robot boleh bergerak terus dari titik mula ke matlamat. Jumlah grid sel yang dilawati iaitu 8 grid sel.

i. Hasil pengujian algoritma FDCC-PP dalam masa nyata



(a)

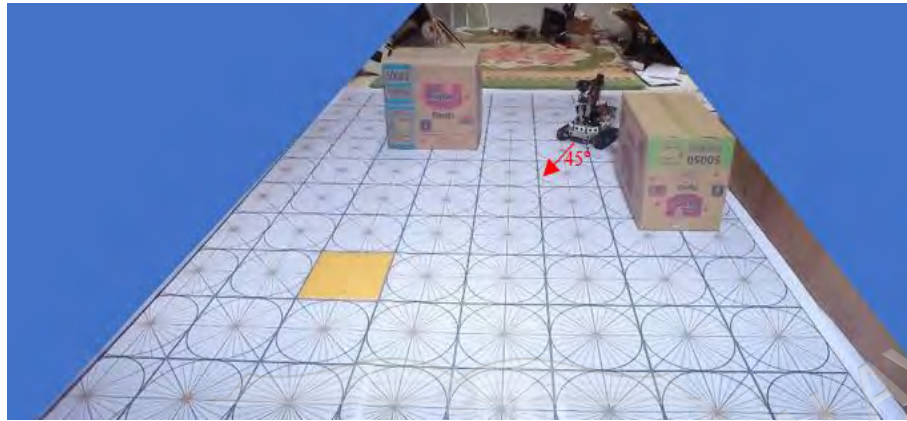


(b)

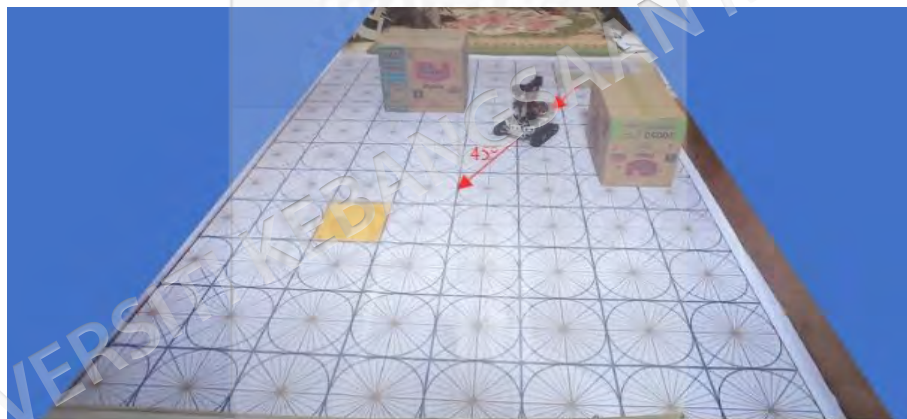
bersambung...

11 derajat ke kanan. Kemudian robot berjalan dalam jarak 1 grid sel dan akhirnya menemui matlamat. Jumlah panjang laluan ialah 2.2 m.

i. Hasil pengujian algoritma MFCC-PP dalam masa nyata



(a)



(b)



(c)

bersambung...

sambungan...



(d)

Rajah 4.45 Hasil pengujian algoritma MFCC-PP dalam masa nyata

Perbandingan yang ditunjukkan dalam Jadual 4.14 menggambarkan perbandingan antara algoritma MFCC-PP dan tiga algoritma navigasi lain berdasarkan peta kedua. Daripada perbandingan ini dapat dilihat bahawa disebabkan penggunaan algoritma MFCC-PP, kecekapan perancangan laluan robot boleh ditingkatkan berbanding dengan algoritma A* dalam peta dengan lebih banyak ruang kosong, dan menjadi malah meningkat berbanding dengan algoritma Dijkstra.

Jadual 4.14 Perbandingan antara algoritma MFCC-PP dan tiga algoritma navigasi lain berdasarkan peta kedua

No.	Algoritma	Grid sel yang dilawati (sel)	Masa Perancangan Purata (saat)	Panjang Laluan (cm)
1	Dijkstra	69	0.6870	250
2	A*	26	0.2530	250
3	FDCC-PP	8	0.0074	215
4	MFCC-PP	9	0.0085	225

Walau bagaimanapun, walaupun kecekapan carian telah bertambah baik, panjang laluan yang dirancang oleh algoritma MFCC-PP adalah lebih panjang daripada yang dirancang oleh algoritma A*, algoritma Dijkstra dan algoritma FDCC-PP. Bagi algoritma A* dan algoritma Dijkstra, strategi perancangan yang melibatkan perbandingan kos pergerakan antara nod, pengemaskinian berterusan hubungan antara nod membantu robot mengelak daripada memintas banyak halangan, sekaligus mengurangkan panjang laluan. Berdasarkan peta kedua, panjang laluan yang

dirancang oleh algoritma MFCC-PP ialah 18.32% lebih panjang daripada laluan terpendek yang dirancang oleh algoritma Dijkstra. Tetapi pengubahsuaian kod rantai menghasilkan pusingan yang lebih lancar berbanding dengan algoritma lain. Selepas menjalankan beberapa ujian, dapat disimpulkan bahawa navigasi robot mencapai keputusan terbaik menggunakan kaedah MFCC-PP. Ini bersesuaian dengan objektif navigasi robot iaitu : Robot hendaklah bergerak lancar dan menghindari selekoh tajam. Ini boleh dilihat dalam Jadual 4.15.

Jadual 4.15 Pencapaian Matlamat Navigasi untuk Setiap Kaedah

No.	Kaedah	Penerangan	Sudut Pusingan yang boleh dicapai	Maklumat
1	Dijkstra	Robot berbelok pada selekoh tajam (membuat pusingan tajam). Tidak sesuai dengan matlamat navigasi dan boleh merosakkan motor pacuan roda.	0, 90, 180, 270, 360	Matlamat tidak tercapai.
2	A*	Robot berbelok pada selekoh tajam (membuat pusingan tajam). Tidak sesuai dengan matlamat navigasi dan boleh merosakkan motor pacuan roda.	0, 90, 180, 270, 360	Matlamat tidak tercapai.
3	FDCC-PP	Robot tidak lagi membuat selekoh tajam, tetapi sudut minimum hanya gandaan 45 darjah.	0, 45, 90, 135, 180, 225, 270, 315, 360	Matlamat tidak tercapai sepenuhnya dan perlu dioptimumkan.
4	MFCC-PP	Pusingan robot lancar berbanding kaedah FDCC-PP, kerana sudut minimum gandaan 11.25 darjah.	0, 11.25, 22.5, 33.75, 45, 56.25, 67.5, 78.75, 90, 101.25, 112.5, 123.75, 135, 146.25, 157.5, 168.75, 180, 191.25, 202.5, 213.75, 225, 236.25, 247.5, 258.75, 270, 281.25, 292.5, 303.75, 315, 326.25, 337.5, 348.75, 360.	Matlamat tercapai.

Dalam algoritma Dijkstra dan A* matlamat tidak tercapai, kerana robot mesti berhenti jika mahu berputar 90 darjah. Ini sudah tentu meningkatkan masa robot untuk sampai ke matlamatnya. Dalam algoritma FDCC-PP, matlamat dicapai tetapi tidak

sempurna. Ini kerana algoritma FDCC-PP hanya mampu berpusing serendah 45 darjah, jadi robot tidak boleh berpusing dengan lancar. Dalam algoritma MFCC-PP, keupayaan memusing robot dipertingkatkan lagi, supaya ia boleh bertukar di bawah 45 darjah. Keupayaan memusing terendah robot ialah 11.25 darjah.

4.6 KESIMPULAN

Keputusan dan perbincangan daripada penyelidikan telah dihuraikan secara rinci. Ada tiga perkara utama dalam keputusan dan perbincangan dalam Bab IV ini, iaitu menguji keberkesanan dan prestasi keseluruhan sistem yang dibangunkan (objektif 4). Sistem yang telah dibangunkan iaitu pembinaan reka bentuk robot berautonomi (objektif 1), algoritma identifikasi sampah (objektif 2), serta algoritma navigasi robot (objektif 3).

Pada penyelidikan ini telah berjaya dihasilkan pembinaan reka bentuk robot berautonomi yang boleh bergerak secara autonomi mengikut laluan yang telah ditetapkan. Ini sudah tentu kerana ia disokong oleh komponen penggerak, penderia, serta pengawal yang digunakan. Dari segi penggerak, motor DC boleh menggerakkan roda mengikut arahan yang diberikan. Dalam motor servo memang ada ralat dalam mengawal darjah. Jika ralat ini tidak diperbetulkan, ia akan mengakibatkan percanggahan antara arahan dan keputusan yang diperolehi. Walau bagaimanapun, ralat ini boleh ditentukan supaya arahan pergerakan darjah yang diberikan adalah mengikut yang dihasilkan oleh motor servo. Daripada keputusan ujian memang terdapat beberapa ralat pada setiap penderia, tetapi nilainya masih agak rendah. Untuk penderia kompas ralat maksimum yang diperolehi ialah 2 darjah. Ini adalah selaras dengan lembaran data penderia kompas. Tetapi untuk penderia lain seperti penderia ultrasonik, penderia pengekod mempunyai nilai ralat kurang daripada 1.5%. Dalam sistem kawalan, jika nilai ralat masih di bawah 10%, maka sistem kawalan masih dikategorikan berfungsi dengan baik.

Algoritma identifikasi sampah juga telah berjaya dilaksanakan. Identifikasi sampah menggunakan Rangkaian Saraf Konvolusi terlatih iaitu AlexNet, VGG16, GoogleNet dan ResNet18. Pengujian dilakukan dengan membandingkan keputusan ketepatan matriks kekeliruan dengan keputusan dalam masa nyata. Terdapat

perbezaan antara 2% hingga 5%. Nilai ini masih dikategorikan sebagai baik dalam sistem kawalan. Untuk menguji sama ada sampah itu dalam bentuk 3D atau 2D, penderia LiDAR digunakan. Penderia ini mempunyai ketepatan $\pm 2\text{mm}$ supaya resolusi objek sampah 3D yang terhasil secara relatifnya lebih baik. Keputusan ujian menunjukkan bahawa penderia LiDAR mampu menghasilkan bentuk 3D daripada objek sampah yang dikesan dalam bentuk grafik. Graf adalah asas untuk menentukan kedudukan pencengkam robot dalam mengangkat objek sampah. Kejayaan ini tidak dapat dipisahkan daripada kaedah Sequential_Camera_LiDAR (SCL) yang dicadangkan. Kaedah ini menggabungkan kamera dan penderia LiDAR. CNN memproses output kamera, diikuti (*sequential*) dengan proses pengimbasan oleh penderia LiDAR. Dengan kaedah ini pengoptimuman proses pengesanan dapat dicapai. Selepas proses imbasan pantas dijalankan, jika objek dalam bentuk 3D, maka proses imbasan terperinci dijalankan. Walau bagaimanapun, jika ia dalam 2D, maka proses pengimbasan diabaikan. Oleh itu kecekapan masa yang diperolehi antara 13.33% hingga 59.26%.

Seterusnya, algoritma navigasi robot berautonomi juga telah berjaya dilaksanakan. Algoritma yang diusulkan adalah *Modified Freeman Chain Code-Path Planning* (MFCC-PP). Algoritma ini mengubah suai Kod Rantaian *Freeman* dengan menambah bilangan kod iaitu 16 kod dan 32 kod. Selepas kaedah MFCC-PP dilaksanakan dalam mencari sampah di Perkarangan rumah, kaedah ini kemudiannya dibandingkan dengan kaedah navigasi algoritma A*, algoritma Dijkstra, dan algoritma FDCC-PP. Pengujian dilakukan dengan menyediakan 2 jenis peta yang berbeza. Berdasarkan dua jenis peta yang berbeza, panjang laluan yang dirancang oleh algoritma MFCC-PP ialah 5.69% dan 18.32% lebih panjang daripada laluan terpendek yang dirancang oleh algoritma Dijkstra. Tetapi pengubahsuaian kod rantai menghasilkan pusingan yang lebih lancar berbanding dengan algoritma lain.

BAB V

KESIMPULAN DAN KAJIAN MASA DEPAN

5.1 PENGENALAN

Pada masa ini, robot telah digunakan secara meluas dalam pelbagai sektor seperti sektor pendidikan, perindustrian, kesihatan, pertahanan dan keselamatan, pembinaan sesebuah bangunan. Salah satu sebab memilih robot ini ialah robot tersebut mampu bekerja mengikut tugas yang diberikan tanpa batasan dan tanpa penat. Itu juga sebab penyelidikan dalam membina robot untuk membantu manusia terutama dalam mengenal pasti sampah di perkarangan rumah. Walau bagaimanapun, pembangunan robot mesti dilengkapi dengan kecerdasan buatan. Jika robot ingin mencari sampah, maka robot tersebut mestilah mempunyai kecerdasan dalam mengenal pasti sampah tersebut. Selain itu, robot tersebut mestilah mempunyai kecerdasan buatan untuk dapat menentukan sama ada sampah tersebut adalah objek bentuk 3D atau tidak (2D). Kesilapan dalam menentukan 3D dan 2D boleh menyebabkan penggenggam robot hanya mengangkat objek tidak nyata, kerana objek hanya dalam bentuk foto di atas kertas (2D). Memang sampah boleh berbentuk 2D, namun keadaan ini agak jarang berlaku.

Pada tesis ini telah dibincangkan ikhtiar-ikhtiar untuk memecahkan permasalahan tersebut dengan disertai pengujian prestasi dari ikhtiar yang dicadangkan, mulai dari pembinaan robot berautonomi, algoritma identifikasi sampah, algoritma navigasi robot berautonomi. Bab ini menyimpulkan tesis secara keseluruhan hasil penting dan sumbangan utama tesis ini terkait perkara-perkara di atas. Kerja-kerja masa depan untuk melanjutkan penyelidikan akan dibincangkan.

5.2 KESIMPULAN

Salah satu penyokong untuk mengoptimumkan sistem identifikasi sampah dan navigasi adalah ketersediaan beberapa komponen dengan ketepatan tinggi dan juga keupayaan penggerak untuk memacu sistem penderia. Bacaan penderia dan pergerakan penggerak dikawal oleh sistem berdasarkan program yang terdapat pada pengawal dan komputer riba. Kesemua komponen penyokong ini bersatu untuk membentuk robot berautonomi. Dalam penyelidikan ini, prototaip robot berautonomi telah direka bentuk dan dibina dengan betul. Reka bentuk dan pembinaan robot ini adalah objektif pertama daripada penyelidikan. Daripada keputusan ujian, robot berautonomi telah berjaya mengidentifikasi sampah dan menavigasi di perkarangan rumah. Ketepatan robot dapat dilihat dari nilai ralat yang dihasilkan oleh setiap penderia. Nilai ralat masing-masing adalah di bawah 5%. Dalam sistem kawalan, sekiranya nilai ralat masih di bawah 10%, maka sistem kawalan tersebut masih dikategorikan sebagai telah berjaya menjalankan fungsinya. Dengan keputusan ini objektif pertama telah berjaya dicapai.

Algoritma identifikasi sampah juga berjaya dilaksanakan. Algoritma ini adalah objektif kedua dalam penyelidikan. Identifikasi sampah menggunakan rangkaian saraf konvolusi pra-latih (AlexNet, VGG16, GoogleNet, ResNet18). Semuanya mempunyai ketepatan identifikasi objek sampah masing-masing sebanyak 80.5%, 95.6%, 98.3% dan 97.5%. Pengujian dilakukan dengan membandingkan keputusan ketepatan matriks kekeliruan dengan keputusan masa nyata. Terdapat perbezaan 2% hingga 5%. Nilai ini masih diklasifikasikan sebagai baik dalam sistem kawalan. Penderia LiDAR digunakan untuk menguji sama ada sampah yang diidentifikasi adalah nyata (3D) atau tidak (2D). Ketepatan penderia ini ialah $\pm 2\text{mm}$, jadi resolusi objek sampah 3D yang dihasilkan adalah agak baik. Keputusan ujian menunjukkan bahawa penderia LiDAR boleh menjana bentuk 3D daripada objek sampah yang dikesan. Bentuk 3D ini diperlukan oleh pencengkam robot sebagai panduan mengutip sampah. Kejayaan ini tidak boleh dipisahkan daripada kaedah Sequential_Camera_LiDAR (SCL) yang dicadangkan. Kaedah ini menggabungkan kamera dan penderia LiDAR. CNN memproses output kamera, diikuti (berurutan) oleh penderia LiDAR untuk pengimbasan. Kaedah ini membolehkan pengoptimuman proses pengesanan dicapai.

Seterusnya, algoritma navigasi robot berautonomi juga berjaya dilaksanakan. Algoritma ini adalah objektif ketiga dalam penyelidikan. Algoritma yang dicadangkan ialah *Modified Freeman Chain Code-Path Planning* (MFCC-PP). Algoritma ini mengubah suai kod rantai Freeman dengan menambah bilangan kod antara 16 dan 32 kod. Kaedah ini diuji di perkarangan rumah menggunakan dua peta berbeza. Peta pertama ialah peta laluan dengan satu halangan dan peta kedua ialah peta laluan dengan dua halangan. Algoritma ini dalam kaedah ini dibandingkan dengan kaedah navigasi algoritma A*, Dijkstra dan FDCC-PP. Berdasarkan dua jenis peta yang berbeza, panjang laluan yang dirancang oleh algoritma MFCC-PP ialah 5.69% dan 18.32% lebih panjang daripada laluan terpendek yang dirancang oleh algoritma Dijkstra. Walau bagaimanapun, menukar kod rantai menjadikan pusingan lebih lancar berbanding dengan algoritma lain. Dengan itu dapat dibuktikan bahawa algoritma yang dicadangkan telah berjaya dijalankan.

5.3 SUMBANGAN KAJIAN

Kajian yang telah dijalankan dalam tesis ini telah memberikan sekurang-kurangnya tiga sumbangan dalam bidang robotik dan Pembelajaran Dalam. Tiga sumbangan tersebut ialah:

1. Reka bentuk robot berautonomi untuk mengidentifikasi sampah dan bernavigasi di perkarangan rumah.
2. Algoritma untuk identifikasi sampah iaitu : *Sequential Camera LiDAR* (SCL).
3. Algoritma untuk navigasi robot di perkarangan rumah, iaitu : *Modified Freeman Chain Code – Path Planning* (MFCC-PP).

5.4 CADANGAN KAJIAN LANJUTAN

Penyelidikan yang dijalankan dalam tesis ini masih terhad kepada identifikasi sampah dan navigasi robot di perkarangan rumah. Dalam reka bentuk robot berautonomi memang sudah dilengkapi dengan lengan robot. Bagaimanapun dalam penyelidikan ini masih belum dilengkapi dengan sistem untuk mengutip sampah berdasarkan bentuk 3D sampah tersebut.

Selain itu, jika terdapat sampah dalam bentuk 2D, pencengkam robot mesti dilengkapi dengan sistem sedutan untuk mengangkat objek berbentuk kertas 2D dengan disedut. Proses pengimbasan masih menggunakan satu penderia LiDAR. Untuk meningkatkan kelajuan proses pengimbasan, bilangan penderia LiDAR perlu ditambah supaya masa pengimbasan juga lebih pantas.

Robot yang digunakan masih dalam bentuk prototaip, jadi jika perkarangan rumah agak besar, maka saiz robot perlu diperbesarkan, dan dilengkapi dengan tempat untuk menampung sementara sampah yang diletakkan bersama robot.



RUJUKAN

- Abdel-Shafy, H. I. & Mansour, M. S. M. 2018. Solid waste issue: Sources, composition, disposal, recycling, and valorization. *Egyptian Journal of Petroleum* 27(4): 1275–1290. doi:10.1016/j.ejpe.2018.07.003
- Abdi, M. & Nahavandi, S. 2016. Multi-Residual Networks: Improving the Speed and Accuracy of Residual Networks. Retrieved from <http://arxiv.org/abs/1609.05672>
- Adedeji, O. & Wang, Z. 2019. Intelligent waste classification system using deep learning convolutional neural network. *Procedia Manufacturing* 35: 607–612. doi:10.1016/j.promfg.2019.05.086
- Adetiba, E., Ajayi, O. T., Kala, J. R., Badejo, J. A., Ajala, S., Abayomi, A., Badejo, J. A., et al. 2021. LeafsnapNet: An Experimentally Evolved Deep Learning Model for Recognition of Plant Species based on Leafsnap Image Dataset. *Journal of Computer Science* 17(3): 349–363. doi:10.3844/jcssp.2021.349.363
- Agarwal, R., Mittal, N. & Madasu, H. 2021. Convolutional Neural Network Based Facial Expression Recognition Using Image Filtering Techniques. *International Journal of Intelligent Engineering and Systems* 14(5): 78–87. doi:10.22266/ijies2021.1031.08
- Ahmadzadeh, Sedigheh and Ghanavati, M. 2012. Navigation of mobile robot using the PSO particle swarm optimization. *Journal of Academic and Applied Studies (JAAS)* 2: 32--38.
- Ahmed, A., Shaalan, K., Toral, S. & Hifny, Y. 2021. A Multimodal Approach to Improve Performance Evaluation of Call Center Agent. *Sensors* 21(8): 2720. doi:10.3390/s21082720
- Alajlan, M., Koubaa, A., Chaari, I., Bennaceur, H. & Ammar, A. 2013. Global path planning for mobile robots in large-scale grid environments using genetic algorithms. *2013 International Conference on Individual and Collective Behaviors in Robotics (ICBR)*, hlm. 1–8. IEEE. doi:10.1109/ICBR.2013.6729271
- Alom, M. Z., Alam, M., Taha, T. M. & Iftexharuddin, K. M. 2017. Object recognition using cellular simultaneous recurrent networks and convolutional neural network. *2017 International Joint Conference on Neural Networks (IJCNN)*, hlm. 2873–2880. IEEE. doi:10.1109/IJCNN.2017.7966211
- Alom, M. Z., Hasan, M., Yakopcic, C., Taha, T. M. & Asari, V. K. 2020. Improved inception-residual convolutional neural network for object recognition. *Neural Computing and Applications* 32(1): 279–293. doi:10.1007/s00521-018-3627-6
- Alzubaidi, L., Zhang, J., Humaidi, A. J., Al-Dujaili, A., Duan, Y., Al-Shamma, O., Santamaria, J., et al. 2021. Review of deep learning: concepts, CNN architectures, challenges, applications, future directions. *Journal of Big Data* 8(1): 53. doi:10.1186/s40537-021-00444-8

- Ambina, D. G. 2019. Tinjauan Pemilahan Sampah Menurut Undang-Undang Nomor 18 Tahun 2008 Tentang Pengelolaan Sampah. *Bina Huk Lingkung* 3(2): 171–185. doi:10.24970/jbhl.v3n2.13
- Anderson, C. B. 2018. The CCB-ID approach to tree species mapping with airborne imaging spectroscopy. *PeerJ* 2018(10): 1–18. doi:10.7717/peerj.5666
- Ara, R. K., Matiolański, A., Dziech, A., Baran, R., Domin, P. & Wieczorkiewicz, A. 2022. Fast and Efficient Method for Optical Coherence Tomography Images Classification Using Deep Learning Approach. *Sensors* 22(13): 4675. doi:10.3390/s22134675
- Arvanitis, G., Lalos, A. S. & Moustakas, K. 2020. Image-Based 3D MESH Denoising Through A Block Matching 3D Convolutional Neural Network Filtering Approach. *2020 IEEE International Conference on Multimedia and Expo (ICME)*, hlm. 1–6. IEEE. doi:10.1109/ICME46284.2020.9102938
- Asad, M. U., Gu, J., Farooq, U., Dey, R., Balas, V. E. & Abbas, G. 2022. Intelligent Obstacle Avoidance Controller for QBot2. *IFAC-PapersOnLine* 55(1): 120–125. doi:10.1016/j.ifacol.2022.04.020
- Barré, P., Stöver, B. C., Müller, K. F. & Steinhage, V. 2017. LeafNet: A computer vision system for automatic plant species identification. *Ecological Informatics* 40: 50–56. doi:10.1016/j.ecoinf.2017.05.005
- Baud, I., Post, J. & Furedy, C. (Eds.). 2004. Solid Waste Management and Recycling. GeoJournal Library, hlm. Vol. 76. Dordrecht: Springer Netherlands. doi:10.1007/1-4020-2529-7
- Benewake, T. 2022. TF40 Datasheet. https://my.mouser.com/datasheet/2/1099/Benewake_10152020_TF40-1954048.pdf [20 June 2022].
- Bhatia, S., Devi, A., Alsuwailem, R. I. & Mashat, A. 2022. Convolutional Neural Network Based Real Time Arabic Speech Recognition to Arabic Braille for Hearing and Visually Impaired. *Frontiers in Public Health* 10. doi:10.3389/fpubh.2022.898355
- Cao, X., Zhou, F., Xu, L., Meng, D., Xu, Z. & Paisley, J. 2018. Hyperspectral Image Classification With Markov Random Fields and a Convolutional Neural Network. *IEEE Transactions on Image Processing* 27(5): 2354–2367. doi:10.1109/TIP.2018.2799324
- Chai, J., Zeng, H., Li, A. & Ngai, E. W. T. 2021. Deep learning in computer vision: A critical review of emerging techniques and application scenarios. *Machine Learning with Applications* 6: 100134. doi:10.1016/j.mlwa.2021.100134
- Chaudhuri, A., Mandaviya, K., Badelia, P. & Ghosh, S. K. 2017. Optical Character Recognition Systems, hlm. 9–41. doi:10.1007/978-3-319-50252-6_2
- Chauhan, S., Vig, L., De Filippo De Grazia, M., Corbetta, M., Ahmad, S. & Zorzi, M.

2019. A Comparison of Shallow and Deep Learning Methods for Predicting Cognitive Performance of Stroke Patients From MRI Lesion Images. *Frontiers in Neuroinformatics* 13. doi:10.3389/fninf.2019.00053
- Chen, C.-H., Lin, C.-J., Jeng, S.-Y., Lin, H.-Y. & Yu, C.-Y. 2021. Using Ultrasonic Sensors and a Knowledge-Based Neural Fuzzy Controller for Mobile Robot Navigation Control. *Electronics* 10(4): 466. doi:10.3390/electronics10040466
- Cheng, X., Wang, P., Guan, C. & Yang, R. 2019. CSPN++: Learning Context and Resource Aware Convolutional Spatial Propagation Networks for Depth Completion. Retrieved from <http://arxiv.org/abs/1911.05377>
- Coltman, J. W. 2002. The transformer [historical overview]. *IEEE Industry Applications Magazine* 8(1): 8–15. doi:10.1109/2943.974352
- Dargan, S., Kumar, M., Ayyagari, M. R. & Kumar, G. 2020. A Survey of Deep Learning and Its Applications: A New Paradigm to Machine Learning. *Archives of Computational Methods in Engineering* 27(4): 1071–1092. doi:10.1007/s11831-019-09344-w
- Deng, L. 2014. Deep Learning: Methods and Applications. *Foundations and Trends® in Signal Processing* 7(3–4): 197–387. doi:10.1561/20000000039
- Dijkstra, E. W. 1959. A note on two problems in connexion with graphs. *Numerische Mathematik* 1(1): 269–271. doi:10.1007/BF01386390
- Divya Vani, G., Karumuri, S. R. & Chinnaiah, M. C. 2022. Hardware Schemes for Autonomous Navigation of Cooperative-type Multi-robot in Indoor Environment. *Journal of The Institution of Engineers (India): Series B* 103(2): 449–460. doi:10.1007/s40031-021-00659-6
- Domingues, P. & Rosário, A. F. 2019. Deep Learning-based Facial Detection and Recognition in Still Images for Digital Forensics. *Proceedings of the 14th International Conference on Availability, Reliability and Security*, hlm. 1–10. New York, NY, USA: ACM. doi:10.1145/3339252.3340107
- Dougherty, G. 2013. Pattern Recognition and Classification, An Introduction, hlm. 2013th Edisi . New York: Springer Science+Business Media.
- Duchoň, F., Babinec, A., Kajan, M., Beňo, P., Florek, M., Fico, T. & Jurišica, L. 2014. Path Planning with Modified a Star Algorithm for a Mobile Robot. *Procedia Engineering* 96: 59–69. doi:10.1016/j.proeng.2014.12.098
- Eldesokey, A., Felsberg, M., Holmquist, K. & Persson, M. 2020. Uncertainty-Aware CNNs for Depth Completion: Uncertainty from Beginning to End. Retrieved from <http://arxiv.org/abs/2006.03349>
- Engedy, I. & Horvath, G. 2010. Artificial neural network based local motion planning of a wheeled mobile robot. *2010 11th International Symposium on Computational Intelligence and Informatics (CINTI)*, hlm. 213–218. IEEE. doi:10.1109/CINTI.2010.5672245

- Fadlilah, A. F. & Djamal, E. C. 2019. Speaker and Speech Recognition Using Hierarchy Support Vector Machine and Backpropagation. *2019 6th International Conference on Electrical Engineering, Computer Science and Informatics (EECSI)*, hlm. 404–409. IEEE. doi:10.23919/EECSI48112.2019.8976919
- Faisal, M., Chaudhury, S., Sankaran, K. S., Raghavendra, S., Chitra, R. J., Eswaran, M. & Boddu, R. 2022. Faster R-CNN Algorithm for Detection of Plastic Garbage in the Ocean: A Case for Turtle Preservation. (P. N. Mahalle, Ed.) *Mathematical Problems in Engineering* 2022: 1–11. doi:10.1155/2022/3639222
- Fu, S., Liu, W., Zhang, K. & Zhou, Y. 2021. Example-feature graph convolutional networks for semi-supervised classification. *Neurocomputing* 461: 63–76. doi:10.1016/j.neucom.2021.07.048
- Fuchikawa, Y., Nishida, T., Kurogi, S., Kondo, T., Ohkawa, F., Obata, M., Miyagawa, H., et al. 2005. Development of a Vision System for an Outdoor Service Robot to Collect Trash on Streets, hlm. 100–105.
- Fukushima, K. 1988. Neocognitron: A hierarchical neural network capable of visual pattern recognition. *Neural Networks* 1(2): 119–130. doi:10.1016/0893-6080(88)90014-7
- Funch, O. I., Marhaug, R., Kohtala, S. & Steinert, M. 2021. Detecting glass and metal in consumer trash bags during waste collection using convolutional neural networks. *Waste Management* 119: 30–38. doi:10.1016/j.wasman.2020.09.032
- Garcia, M. A. P., Montiel, O., Castillo, O., Sepúlveda, R. & Melin, P. 2009. Path planning for autonomous mobile robot navigation with ant colony optimization and fuzzy cost function evaluation. *Applied Soft Computing* 9(3): 1102–1110. doi:10.1016/j.asoc.2009.02.014
- Ghorbani, A., Shiry, S. & Nodehi, A. 2009. Using Genetic Algorithm for a Mobile Robot Path Planning. *2009 International Conference on Future Computer and Communication*, hlm. 164–166. IEEE. doi:10.1109/ICFCC.2009.28
- Gollapudi, S. 2019. Deep Learning for Computer Vision. *Learn Computer Vision Using OpenCV*, hlm. 51–69. Apress. doi:10.1007/978-1-4842-4261-2_3
- Gomez, E. J., Santa, F. M. & Sarmiento, F. H. M. 2013. A comparative study of geometric path planning methods for a mobile robot: Potential field and voronoi diagrams. *2013 II International Congress of Engineering Mechatronics and Automation (CIIMA)*, hlm. 1–6. IEEE. doi:10.1109/CIIMA.2013.6682776
- González-Jorge, H., Martínez-Sánchez, J., Bueno, M. & Arias, and P. 2017. Unmanned Aerial Systems for Civil Applications: A Review. *Drones* 1(1): 2. doi:10.3390/drones1010002
- Gul, F., Rahiman, W. & Nazli Alhady, S. S. 2019. A comprehensive study for robot navigation techniques. (K. Chen, Ed.) *Cogent Engineering* 6(1). doi:10.1080/23311916.2019.1632046

- Guo, X., Grushka-Cockayne, Y. & De Reyck, B. 2020. London Heathrow Airport Uses Real-Time Analytics for Improving Operations. *INFORMS Journal on Applied Analytics* 50(5): 325–339. doi:10.1287/inte.2020.1044
- Halupka, K. J., Antony, B. J., Lee, M. H., Lucy, K. A., Rai, R. S., Ishikawa, H., Wollstein, G., et al. 2018. Retinal optical coherence tomography image enhancement via deep learning. *Biomedical Optics Express* 9(12): 6205. doi:10.1364/BOE.9.006205
- Hart, P., Nilsson, N. & Raphael, B. 1968. A Formal Basis for the Heuristic Determination of Minimum Cost Paths. *IEEE Transactions on Systems Science and Cybernetics* 4(2): 100–107. doi:10.1109/TSSC.1968.300136
- He, K., Zhang, X., Ren, S. & Sun, J. 2016. Deep Residual Learning for Image Recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, hlm. 770–778. IEEE. doi:10.1109/CVPR.2016.90
- Hinton, G. E., Osindero, S. & Teh, Y.-W. 2006. A Fast Learning Algorithm for Deep Belief Nets. *Neural Computation* 18(7): 1527–1554. doi:10.1162/neco.2006.18.7.1527
- Hirner, D. & Fraundorfer, F. 2021. FC-DCNN: A densely connected neural network for stereo estimation. *2020 25th International Conference on Pattern Recognition (ICPR)*, hlm. 2482–2489. IEEE. doi:10.1109/ICPR48806.2021.9413281
- Hoang, T. T., Quan, N. N. A., Thanh, V. C. & Dong, T. L. T. 2022. Navigation for Two-Wheeled Differential Mobile Robot in the Special Environment, hlm. 167–183. doi:10.1007/978-3-031-08878-0_12
- Hu, M., Wang, S., Li, B., Ning, S., Fan, L. & Gong, X. 2021. PENet: Towards Precise and Efficient Image Guided Depth Completion. Retrieved from <http://arxiv.org/abs/2103.00783>
- Hui, N. B. & Pratihari, D. K. 2010. Design and Development of Intelligent Autonomous Robots, hlm. 29–56. doi:10.1007/978-3-642-11676-6_3
- Hulyalkar S., Deshpande R., Makode K., K. S. 2018. Implementation of Smartbin Using Convolutional Neural Networks. *International Research Journal of Engineering and Technology* 5(4): 3352–3358. Retrieved from www.irjet.net
- Iordan, Daniela and Popescu, G. and others. 2015. The accuracy of lidar measurements for the different land cover categories. *Scientific Papers. Series E. Land Reclamation, Earth Observation & Surveying, Environmental Engineering* 4: 158–164.
- Jaritz, M., de Charette, R., Wirbel, E., Perrotton, X. & Nashashibi, F. 2018. Sparse and Dense Data with CNNs: Depth Completion and Semantic Segmentation. Retrieved from <http://arxiv.org/abs/1808.00769>
- Jing, Y., Yang, Y., Feng, Z., Ye, J., Yu, Y. & Song, M. 2020. Neural Style Transfer: A Review. *IEEE Transactions on Visualization and Computer Graphics* 26(11):

3365–3385. doi:10.1109/TVCG.2019.2921336

Khan, M. A. U., Nazir, D., Pagani, A., Mokayed, H., Liwicki, M., Stricker, D. & Afzal, M. Z. 2022. A Comprehensive Survey of Depth Completion Approaches. *Sensors* 22(18): 6969. doi:10.3390/s22186969

Khurana, D., Koli, A., Khatter, K. & Singh, S. 2023. Natural language processing: state of the art, current trends and challenges. *Multimedia Tools and Applications* 82(3): 3713–3744. doi:10.1007/s11042-022-13428-4

Kim, B., Kim, J., Lee, J.-G., Kim, D. H., Park, S. H. & Ye, J. C. 2019. Unsupervised Deformable Image Registration Using Cycle-Consistent CNN, hlm. 166–174. doi:10.1007/978-3-030-32226-7_19

Kim, D., Jung, Y., Toh, K.-A., Son, B. & Kim, J. 2016. An empirical study on iris recognition in a mobile phone. *Expert Systems with Applications* 54: 328–339. doi:10.1016/j.eswa.2016.01.050

Kim, S., Jun, D., Kim, B.-G., Lee, H. & Rhee, E. 2021. Single Image Super-Resolution Method Using CNN-Based Lightweight Neural Networks. *Applied Sciences* 11(3): 1092. doi:10.3390/app11031092

Klančar, G., Zdešar, A., Blažič, S. & Škrjanc, I. 2017. Path Planning. *Wheeled Mobile Robotics*, hlm. 161–206. Elsevier. doi:10.1016/B978-0-12-804204-5.00004-4

Koodtalang, W., Mangkalajan, S., Supsatien, T., Thammalangka, P., Darawan, W. & Sangsuwan, T. 2022. Monitoring and Controlling the Multiple Mobile Robots in Outdoor Environments Based on IoT. *2022 22nd International Conference on Control, Automation and Systems (ICCAS)*, hlm. 1447–1452. IEEE. doi:10.23919/ICCAS55662.2022.10003713

Krizhevsky, A., Sutskever, I. & Hinton, G. E. 2017. ImageNet classification with deep convolutional neural networks. *Communications of the ACM* 60(6): 84–90. doi:10.1145/3065386

Kshirsagar, P. R., Kumar, N., Almulihi, A. H., Alassery, F., Khan, A. I., Islam, S., Rothe, J. P., et al. 2022. Artificial Intelligence-Based Robotic Technique for Reusable Waste Materials. (Z. Uddin, Ed.) *Computational Intelligence and Neuroscience* 2022: 1–9. doi:10.1155/2022/2073482

Kumar, N., Belhumeur, P. N., Biswas, A., Jacobs, D. W., Kress, W. J., Lopez, I. C. & Soares, J. V. B. 2012. Leafsnap: A Computer Vision System for Automatic Plant Species Identification, hlm. 502–516. doi:10.1007/978-3-642-33709-3_36

Kusetogullari, H., Yavariabdi, A., Hall, J. & Lavesson, N. 2021. DIGITNET: A Deep Handwritten Digit Detection and Recognition Method Using a New Historical Handwritten Digit Dataset. *Big Data Research* 23: 100182. doi:10.1016/j.bdr.2020.100182

LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W. & Jackel, L. D. 1989. Backpropagation applied to digit recognition. *Neural*

computation.

- Lecun, Y., Bottou, L., Bengio, Y. & Haffner, P. 1998. Gradient-based learning applied to document recognition. *Proceedings of the IEEE* 86(11): 2278–2324. doi:10.1109/5.726791
- Lee, C. Y., Gallagher, P. & Tu, Z. 2018. Generalizing Pooling Functions in CNNs: Mixed, Gated, and Tree. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 40(4): 863–875. doi:10.1109/TPAMI.2017.2703082
- Lee, J.-G., Jun, S., Cho, Y.-W., Lee, H., Kim, G. B., Seo, J. B. & Kim, N. 2017. Deep Learning in Medical Imaging: General Overview. *Korean Journal of Radiology* 18(4): 570. doi:10.3348/kjr.2017.18.4.570
- Liao, H., Yuan, L., Wu, M., Zhong, L., Jin, G. & Xiong, N. 2022. Face Gender and Age Classification Based on Multi-Task, Multi-Instance and Multi-Scale Learning. *Applied Sciences* 12(23): 12432. doi:10.3390/app122312432
- Litjens, G., Kooi, T., Bejnordi, B. E., Setio, A. A. A., Ciompi, F., Ghafoorian, M., van der Laak, J. A. W. M., et al. 2017. A survey on deep learning in medical image analysis. *Medical Image Analysis* 42: 60–88. doi:10.1016/j.media.2017.07.005
- Liu, C., Xie, N., Yang, X., Chen, R., Chang, X., Zhong, R. Y., Peng, S., et al. 2022. A Domestic Trash Detection Model Based on Improved YOLOX. *Sensors (Basel, Switzerland)* 22(18). doi:10.3390/s22186974
- Liu, L., Song, X., Lyu, X., Diao, J., Wang, M., Liu, Y. & Zhang, L. 2020. FCFR-Net: Feature Fusion based Coarse-to-Fine Residual Learning for Depth Completion. Retrieved from <http://arxiv.org/abs/2012.08270>
- Long, L. & Zeng, X. 2022. Beginning Deep Learning with TensorFlow. Berkeley, CA: Apress. doi:10.1007/978-1-4842-7915-1
- Longo, E., Sahin, F. A., Redondi, A. E. C., Bolzan, P., Bianchini, M. & Maffei, S. 2021. A 5G-Enabled Smart Waste Management System for University Campus. *Sensors* 21(24): 8278. doi:10.3390/s21248278
- Lye, Z., Nisar, H., Lai, K. & Yeap, K. 2014. Localization and feature recognition: Implementation on an indoor navigator robot. *2014 10th France-Japan/ 8th Europe-Asia Congress on Mechatronics (MECATRONICS2014- Tokyo)*, hlm. 238–243. IEEE. doi:10.1109/MECATRONICS.2014.7018561
- Ma, S., Liu, J. & Chen, C. W. 2017. A-Lamp: Adaptive Layout-Aware Multi-Patch Deep Convolutional Neural Network for Photo Aesthetic Assessment. Retrieved from <http://arxiv.org/abs/1704.00248>
- Malgheet, J. R., Manshor, N. B. & Affendey, L. S. 2021. Iris Recognition Development Techniques: A Comprehensive Review. (R. M. Lopez Gutierrez, Ed.) *Complexity* 2021: 1–32. doi:10.1155/2021/6641247
- Mao, W. L., Chen, W. C., Wang, C. T. & Lin, Y. H. 2021. Recycling waste

- classification using optimized convolutional neural network. *Resources, Conservation and Recycling* 164(July 2020): 105132. doi:10.1016/j.resconrec.2020.105132
- Marra, F., Poggi, G., Sansone, C. & Verdoliva, L. 2018. A deep learning approach for iris sensor model identification. *Pattern Recognition Letters* 113: 46–53. doi:10.1016/j.patrec.2017.04.010
- Masehian, E. & Amin-Naseri, M. R. 2004. A voronoi diagram-visibility graph-potential field compound algorithm for robot path planning. *Journal of Robotic Systems* 21(6): 275–300. doi:10.1002/rob.20014
- Mehlhorn, Kurt and Sanders, Peter and Sanders, P. 2008. Algorithms and Data Structures. Berlin, Heidelberg: Springer Berlin Heidelberg. doi:10.1007/978-3-540-77978-0
- Mo, X., Ge, W., Miraglia, M., Inglese, F., Zhao, D., Stefanini, C. & Romano, D. 2020. Jumping Locomotion Strategies: From Animals to Bioinspired Robots. *Applied Sciences* 10(23): 8607. doi:10.3390/app10238607
- Moeskops, P., Viergever, M. A., Mendrik, A. M., de Vries, L. S., Benders, M. J. N. L. & Isgum, I. 2016. Automatic Segmentation of MR Brain Images With a Convolutional Neural Network. *IEEE Transactions on Medical Imaging* 35(5): 1252–1261. doi:10.1109/TMI.2016.2548501
- Mohamed, Abdel-rahman and Dahl, George and Hinton, G. and others. 2009. Deep belief networks for phone recognition, hlm. 39.
- Montaner, M. B. & Ramirez-Serrano, A. 1998. Fuzzy knowledge-based controller design for autonomous robot navigation. *Expert Systems with Applications* 14(1–2): 179–186. doi:10.1016/S0957-4174(97)00059-6
- Mosavi, A. & Rabczuk, T. 2017. Learning and Intelligent Optimization for Material Design Innovation, hlm. 358–363. doi:10.1007/978-3-319-69404-7_31
- Mosavi, A. & Vaezipour, A. 2012. Reactive Search Optimization; Application to Multiobjective Optimization Problems. *Applied Mathematics* 03(10): 1572–1582. doi:10.4236/am.2012.330217
- Nagatani, K., Ishigami, G. & Okada, Y. 2016. Modeling and Control of Robots on Rough Terrain, hlm. 1267–1284. doi:10.1007/978-3-319-32552-1_50
- Najafabadi, M. M., Villanustre, F., Khoshgoftaar, T. M., Seliya, N., Wald, R. & Muharemagic, E. 2015. Deep learning applications and challenges in big data analytics. *Journal of Big Data* 2(1): 1. doi:10.1186/s40537-014-0007-7
- Nazir, D., Liwicki, M., Stricker, D. & Afzal, M. Z. 2022. SemAttNet: Towards Attention-based Semantic Aware Guided Depth Completion. Retrieved from <http://arxiv.org/abs/2204.13635>
- Oh, J.-M., Lee, P.-K., Kim, S.-G. & Lee, W.-H. 2018. A Study on the Development of

- Indoor Autonomous Mobile Robots. *Journal of Institute of Control, Robotics and Systems* 24(9): 814–821. doi:10.5302/J.ICROS.2018.18.0121
- Omrane, H., Masmoudi, M. S. & Masmoudi, M. 2016. Fuzzy Logic Based Control for Autonomous Mobile Robot Navigation. *Computational Intelligence and Neuroscience* 2016: 1–10. doi:10.1155/2016/9548482
- Onishi, J. & Ono, T. 2011. Contour pattern recognition through auditory labels of freeman chain codes for people with visual impairments. *2011 IEEE International Conference on Systems, Man, and Cybernetics*, hlm. 1088–1093. IEEE. doi:10.1109/ICSMC.2011.6083819
- Park, K.-H., Kim, Y.-J. & Kim, J.-H. 2001. Modular Q-learning based multi-agent cooperation for robot soccer. *Robotics and Autonomous Systems* 35(2): 109–122. doi:10.1016/S0921-8890(01)00114-2
- Patil, V., Madgi, M. & Kiran, A. 2022. Early prediction of Alzheimer's disease using convolutional neural network: a review. *The Egyptian Journal of Neurology, Psychiatry and Neurosurgery* 58(1): 130. doi:10.1186/s41983-022-00571-w
- Păvăloaia, V.-D. & Husac, G. 2022. Tracking Unauthorized Access Using Machine Learning and PCA for Face Recognition Developments. *Information* 14(1): 25. doi:10.3390/info14010025
- Pekala, M., Joshi, N., Liu, T. Y. A., Bressler, N. M., DeBuc, D. C. & Burlina, P. 2019. Deep learning based retinal OCT segmentation. *Computers in Biology and Medicine* 114: 103445. doi:10.1016/j.compbiomed.2019.103445
- Phueakthong, P., Varagul, J. & Pinrath, N. 2022. Deep Reinforcement Learning Based Mobile Robot Navigation in Unknown Environment with Continuous Action Space. *2022 5th International Conference on Intelligent Autonomous Systems (ICoIAS)*, hlm. 154–158. IEEE. doi:10.1109/ICoIAS56028.2022.9931272
- poorva, A., Gautam, R. & Kala, R. 2018. Motion Planning for a Chain of Mobile Robots Using A* and Potential Field. *Robotics* 7(2): 20. doi:10.3390/robotics7020020
- Priti & Mandal, K. 2022. Correction to: Review on evolution of municipal solid waste management in India: practices, challenges and policy implications. *Journal of Material Cycles and Waste Management* 24(5): 2088–2088. doi:10.1007/s10163-022-01444-3
- Qiu, J., Cui, Z., Zhang, Y., Zhang, X., Liu, S., Zeng, B. & Pollefeys, M. 2019. DeepLiDAR: Deep Surface Normal Guided Depth Prediction for Outdoor Scene From Sparse LiDAR Data and Single Color Image. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, hlm. 3308–3317. IEEE. doi:10.1109/CVPR.2019.00343
- Rahman, M. M., Islam, M. S., Sassi, R. & Aktaruzzaman, M. 2019. Convolutional neural networks performance comparison for handwritten Bengali numerals recognition. *SN Applied Sciences* 1(12): 1660. doi:10.1007/s42452-019-1682-y

- Rahman, M. W., Islam, R., Hasan, A., Bithi, N. I., Hasan, M. M. & Rahman, M. M. 2022. Intelligent waste management system using deep learning with IoT. *Journal of King Saud University - Computer and Information Sciences* 34(5): 2072–2087. doi:10.1016/j.jksuci.2020.08.016
- Raza, S. M., Hassan, S. M. G., Hassan, S. A. & Shin, S. Y. 2021. Real-Time Trash Detection for Modern Societies using CCTV to Identifying Trash by utilizing Deep Convolutional Neural Network. Retrieved from <http://arxiv.org/abs/2109.09611>
- Ren, C., Jung, H., Lee, S. & Jeong, D. 2021. Coastal waste detection based on deep convolutional neural networks. *Sensors* 21(21). doi:10.3390/s21217269
- Rosique, F., Navarro, P. J., Fernández, C. & Padilla, A. 2019. A Systematic Review of Perception System and Simulators for Autonomous Vehicles Research. *Sensors* 19(3): 648. doi:10.3390/s19030648
- Rubio, F., Valero, F. & Llopis-Albert, C. 2019. A review of mobile robots: Concepts, methods, theoretical framework, and applications. *International Journal of Advanced Robotic Systems* 16(2): 172988141983959. doi:10.1177/1729881419839596
- Salimi, I., Bayu Dewantara, B. S. & Wibowo, I. K. 2019. Visual-based trash detection and classification system for smart trash bin robot. *International Electronics Symposium on Knowledge Creation and Intelligent Computing, IES-KCIC 2018 - Proceedings* 378–383. doi:10.1109/KCIC.2018.8628499
- Salvini, P., Teti, G., Spadoni, E., Laschi, C., Mazzolai, B. & Dario, P. 2011. The Robot DustCart. *IEEE Robotics & Automation Magazine* 18(1): 59–67. doi:10.1109/MRA.2010.940153
- Sarker, I. H. 2021. Deep Learning: A Comprehensive Overview on Techniques, Taxonomy, Applications and Research Directions. *SN Computer Science* 2(6): 420. doi:10.1007/s42979-021-00815-1
- Sedighi, K. H., Ashenayi, K., Manikas, T. W., Wainwright, R. L. & Heng-Ming Tai. (n.d.). Autonomous local path planning for a mobile robot using a genetic algorithm. *Proceedings of the 2004 Congress on Evolutionary Computation (IEEE Cat. No.04TH8753)*, hlm. 1338–1345. IEEE. doi:10.1109/CEC.2004.1331052
- Sejnowski, T. J. 2018. The deep learning revolution. MIT press.
- Sermanet, P., Eigen, D., Zhang, X., Mathieu, M., Fergus, R. & LeCun, Y. 2013. OverFeat: Integrated Recognition, Localization and Detection using Convolutional Networks. Retrieved from <http://arxiv.org/abs/1312.6229>
- Simonyan, K. & Zisserman, A. 2014. Very Deep Convolutional Networks for Large-Scale Image Recognition. Retrieved from <http://arxiv.org/abs/1409.1556>
- Soltani, A. R., Tawfik, H., Goulernas, J. Y. & Fernando, T. 2002. Path planning in construction sites: performance evaluation of the Dijkstra, A*, and GA search algorithms. *Advanced Engineering Informatics* 16(4): 291–303.

doi:10.1016/S1474-0346(03)00018-1

- Srinivas, S., Sarvadevabhatla, R. K., Mopuri, K. R., Prabhu, N., Kruthiventi, S. S. S. & Babu, R. V. 2017. An Introduction to Deep Convolutional Neural Nets for Computer Vision. *Deep Learning for Medical Image Analysis*, hlm. 25–52. Elsevier. doi:10.1016/B978-0-12-810408-8.00003-1
- Subramanian, P. . 2000. Plastics recycling and waste management in the US. *Resources, Conservation and Recycling* 28(3–4): 253–263. doi:10.1016/S0921-3449(99)00049-X
- Sze, V., Chen, Y.-H., Yang, T.-J. & Emer, J. 2017. Efficient Processing of Deep Neural Networks: A Tutorial and Survey. Retrieved from <http://arxiv.org/abs/1703.09039>
- Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J. & Wojna, Z. 2016. Rethinking the Inception Architecture for Computer Vision. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, hlm. 2818–2826. IEEE. doi:10.1109/CVPR.2016.308
- Szegedy, C., Wei Liu, Yangqing Jia, Sermanet, P., Reed, S., Anguelov, D., Erhan, D., et al. 2015. Going deeper with convolutions. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, hlm. 1–9. IEEE. doi:10.1109/CVPR.2015.7298594
- Szeliski, R. 2021. Computer Vision: Algorithms and Applications 2nd Edition. *Springer* 1196.
- Tan, J. H., Acharya, U. R., Bhandary, S. V., Chua, K. C. & Sivaprasad, S. 2017. Segmentation of optic disc, fovea and retinal vasculature using a single convolutional neural network. *Journal of Computational Science* 20: 70–79. doi:10.1016/j.jocs.2017.02.006
- Tan, M. & Le, Q. V. 2019. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. Retrieved from <http://arxiv.org/abs/1905.11946>
- The Mathworks. 2018. Introducing Deep Learning with MATLAB: “What is Deep Learning?” *The MathWorks, Inc. MATLAB and Simulink are registered trademarks of The MathWorks, Inc* 1–14.
- Tran, P. V. 2016. A Fully Convolutional Neural Network for Cardiac Segmentation in Short-Axis MRI. Retrieved from <http://arxiv.org/abs/1604.00494>
- Treiber, M. A. 2013. Optimization for Computer Vision. *Optimization for Computer Vision*. doi:10.1007/978-1-4471-5283-5
- Van Gansbeke, W., Neven, D., De Brabandere, B. & Van Gool, L. 2019. Sparse and Noisy LiDAR Completion with RGB Guidance and Uncertainty. *2019 16th International Conference on Machine Vision Applications (MVA)*, hlm. 1–6. IEEE. doi:10.23919/MVA.2019.8757939
- Veit, Andreas and Wilber, Michael J and Belongie, S. 2016. Residual Networks Behave

- Like Ensembles of Relatively Shallow Networks. Dlm. Garnett (pnyt.). *Advances in Neural Information Processing Systems*. Curran Associates, Inc. Retrieved from <https://proceedings.neurips.cc/paper/2016/file/37bc2f75bf1bcfe8450a1a41c200364c-Paper.pdf>
- Wu, X., Sahoo, D. & Hoi, S. C. H. 2020. Recent advances in deep learning for object detection. *Neurocomputing* 396: 39–64. doi:10.1016/j.neucom.2020.01.085
- Xiang Liu & Daoxiong Gong. 2011. A comparative study of A-star algorithms for search and rescue in perfect maze. *2011 International Conference on Electric Information and Control Engineering*, hlm. 24–27. IEEE. doi:10.1109/ICEICE.2011.5777723
- Xie, S., Girshick, R., Dollar, P., Tu, Z. & He, K. 2017. Aggregated Residual Transformations for Deep Neural Networks. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, hlm. 5987–5995. IEEE. doi:10.1109/CVPR.2017.634
- Xing, J., Li, K., Hu, W., Yuan, C. & Ling, H. 2017. Diagnosing deep learning models for high accuracy age estimation from a single image. *Pattern Recognition* 66: 106–116. doi:10.1016/j.patcog.2017.01.005
- Xu, J., Zhang, Y. & Miao, D. 2020. Three-way confusion matrix for classification: A measure driven view. *Information Sciences* 507: 772–794. doi:10.1016/j.ins.2019.06.064
- Xu, Y., Kong, Q., Huang, Q., Wang, W. & Plumbley, M. D. 2017. Convolutional Gated Recurrent Neural Network Incorporating Spatial Features for Audio Tagging. Retrieved from <http://arxiv.org/abs/1702.07787>
- Yan, Z., Wang, K., Li, X., Zhang, Z., Li, J. & Yang, J. 2022. RigNet: Repetitive Image Guided Network for Depth Completion, hlm. 214–230. doi:10.1007/978-3-031-19812-0_13
- Yang, M. & Thung, G. 2016. Classification of Trash for Recyclability Status. *CS229Project Report* (1): 3.
- Yang, W., Che, J., Zhang, L. & Ma, M. 2022. Research of garbage salvage system based on deep learning. Dlm. Lu (pnyt.) & Cheng (pnyt.). *International Conference on Computer Application and Information Security (ICCAIS 2021)*, hlm. 19. SPIE. doi:10.1117/12.2637385
- Yuan, Z. & Liu, J. 2022. A Hybrid Deep Learning Model for Trash Classification Based on Deep Trasnsfer Learning. (G. Yang, Ed.) *Journal of Electrical and Computer Engineering* 2022: 1–9. doi:10.1155/2022/7608794
- Zagoruyko, S. & Komodakis, N. 2016. Wide Residual Networks. Retrieved from <http://arxiv.org/abs/1605.07146>
- Zeiler, M. D. & Fergus, R. 2014. Visualizing and Understanding Convolutional Networks, hlm. 818–833. doi:10.1007/978-3-319-10590-1_53

- Zeng, Z., Lian, L., Sammut, K., He, F., Tang, Y. & Lammas, A. 2015. A survey on path planning for persistent autonomy of autonomous underwater vehicles. *Ocean Engineering* 110: 303–313. doi:10.1016/j.oceaneng.2015.10.007
- Zhang, C., Sargent, I., Pan, X., Gardiner, A., Hare, J. & Atkinson, P. M. 2018. VPRS-Based Regional Decision Fusion of CNN and MRF Classifications for Very Fine Resolution Remotely Sensed Images. *IEEE Transactions on Geoscience and Remote Sensing* 56(8): 4507–4521. doi:10.1109/TGRS.2018.2822783
- Zhang, X., Li, Z., Loy, C. C. & Lin, D. 2016. PolyNet: A Pursuit of Structural Diversity in Very Deep Networks. Retrieved from <http://arxiv.org/abs/1611.05725>
- Zhang, Z., Xie, Y., Xing, F., McGough, M. & Yang, L. 2017. MDNet: A Semantically and Visually Interpretable Medical Image Diagnosis Network. Retrieved from <http://arxiv.org/abs/1707.02485>
- Zhao, M. & Lu, H. 2019. Robot Path Planning Based on Freeman Direction Chain Code. *2019 IEEE Symposium Series on Computational Intelligence (SSCI)*, hlm. 1967–1973. IEEE. doi:10.1109/SSCI44817.2019.9002747
- Zhao, Z.-Q., Zheng, P., Xu, S.-T. & Wu, X. 2019. Object Detection With Deep Learning: A Review. *IEEE Transactions on Neural Networks and Learning Systems* 30(11): 3212–3232. doi:10.1109/TNNLS.2018.2876865
- Zhu, A. & Yang, S. X. 2007. Neurofuzzy-Based Approach to Mobile Robot Navigation in Unknown Environments. *IEEE Transactions on Systems, Man and Cybernetics, Part C (Applications and Reviews)* 37(4): 610–621. doi:10.1109/TSMCC.2007.897499

LAMPIRAN A**SENARAI PENERBITAN****Jurnal berindeks WOS/Scopus**

Naf'an, E., Sulaiman, R., Ali, N.M, 2021. Improving the Modelling of Robot Bunker With Camera. Journal of Physics: Conference Series. **DOI** 10.1088/1742-6596/1339/1/012014.

Naf'an, E., Sulaiman, R., Ali, N.M, 2023. Optimization of Trash Identification on The House Compound Using a Convolutional Neural Network (CNN) and Sensor System, MDPI Sensors. <https://doi.org/10.3390/s23031499>.

Seminar antara bangsa

Naf'an, E., Sulaiman, R., Ali, N.M, 2018. Modelling of Robot Bunker Based on Fuzzy Logic. VIIS 2018 – Proceeding, 2018. Can be accessed on <https://ivi.ukm.my/wp-content/uploads/Pages-from-Buku-VIIS-with-cover.pdf>.

Naf'an, E., Sulaiman, R., Ali, N.M, 2023. Improving autonomous robot gripper position on lifting trash object based on Object Geometry Parameters and Centroid Modification. Springer: Conference Series, IVIC 2023 – Proceeding, 2023. **DOI** 10.1007/978-981-99-7339-2_6.

LAMPIRAN B

KEPUTUSAN UJIAN SENI BINA ALEXNET

Identifikasi Sampah Dengan Seni Bina CNN Pra-Latihan AlexNet

Jenis Sampah :

1.Kadbod

No	Label Sebenar	Keputusan Ujian			Sampah Dikesan Betul		
		Label Ramalan Ke-1	Label Ramalan Ke-2	Label Ramalan Ke-3	Ke-1	Ke-2	Ke-3
1	Kadbod	Kadbod	Kadbod	Kadbod	1	1	1
2	Kadbod	Kadbod	Kadbod	Kadbod	1	1	1
3	Kadbod	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	0	0	0
4	Kadbod	Kadbod	Kadbod	Kadbod	1	1	1
5	Kadbod	Kadbod	Kadbod	Kadbod	1	1	1
6	Kadbod	Kadbod	Kadbod	Kadbod	1	1	1
7	Kadbod	Kadbod	Kadbod	Kadbod	1	1	1
8	Kadbod	Kadbod	Kadbod	Kadbod	1	1	1
9	Kadbod	Kadbod	Kadbod	Kadbod	1	1	1
10	Kadbod	Kadbod	Kadbod	Kadbod	1	1	1
11	Kadbod	Kadbod	Kadbod	Kadbod	1	1	1
12	Kadbod	Kertas	Kadbod	Kertas	0	1	0
13	Kadbod	Kadbod	Kadbod	Kadbod	1	1	1
14	Kadbod	Kadbod	Kadbod	Kadbod	1	1	1
15	Kadbod	Kadbod	Kadbod	Kadbod	1	1	1
16	Kadbod	Kadbod	Kadbod	Kadbod	1	1	1
17	Kadbod	Kadbod	Kadbod	Kadbod	1	1	1
18	Kadbod	Kadbod	Kadbod	Kadbod	1	1	1
19	Kadbod	Kadbod	Kertas	Kadbod	1	0	1
20	Kadbod	Kadbod	Kadbod	Kadbod	1	1	1
21	Kadbod	Kadbod	Pembungkus makanan	Kadbod	1	0	1
22	Kadbod	Kadbod	Kadbod	Kadbod	1	1	1
23	Kadbod	Kadbod	Kadbod	Kadbod	1	1	1
24	Kadbod	Kadbod	Kadbod	Kadbod	1	1	1
25	Kadbod	Kadbod	Kadbod	Kertas	1	1	0
26	Kadbod	Kertas	Kadbod	Kadbod	0	1	1
27	Kadbod	Kadbod	Kadbod	Kadbod	1	1	1
28	Kadbod	Kertas	Pembungkus makanan	Kertas	0	0	0
29	Kadbod	Kadbod	Kadbod	Kadbod	1	1	1
30	Kadbod	Kadbod	Kadbod	Kadbod	1	1	1

31	Kadbod	Kadbod	Kadbod	Kadbod	1	1	1
32	Kadbod	Kadbod	Kadbod	Kadbod	1	1	1
33	Kadbod	Kertas	Kertas	Kertas	0	0	0
34	Kadbod	Kadbod	Kadbod	Kadbod	1	1	1
35	Kadbod	Kadbod	Kertas	Kadbod	1	0	1
36	Kadbod	Kadbod	Kadbod	Kadbod	1	1	1
37	Kadbod	Kadbod	Kertas	Kadbod	1	0	1
38	Kadbod	Kadbod	Kadbod	Kadbod	1	1	1
39	Kadbod	Kadbod	Kadbod	Kadbod	1	1	1
40	Kadbod	Kadbod	Kadbod	Kadbod	1	1	1
41	Kadbod	Kadbod	Kadbod	Kadbod	1	1	1
42	Kadbod	Kadbod	Kadbod	Pembungkus makanan	1	1	0
43	Kadbod	Kadbod	Kadbod	Kadbod	1	1	1
44	Kadbod	Kadbod	Kertas	Kadbod	1	0	1
45	Kadbod	Kadbod	Kadbod	Kadbod	1	1	1
Jumlah objek yang dikesan betul					40	37	39
Ketepatan (<i>Accuracy</i>)					88.89	82.22	86.67

Ketepatan terendah = 82.2222222

**Identifikasi Sampah Dengan Seni Bina CNN Pra-Latihan
AlexNet**

Jenis Sampah : 2.Fabrik

No	Label Sebenarnya	Keputusan Ujian			Sampah Dikesan Betul		
		Label Ramalan Ke-1	Label Ramalan Ke-2	Label Ramalan Ke-3	Ke-1	Ke-2	Ke-3
1	Fabrik	Fabrik	Fabrik	Fabrik	1	1	1
2	Fabrik	Kadbod	Fabrik	Fabrik	0	1	1
3	Fabrik	Fabrik	Fabrik	Fabrik	1	1	1
4	Fabrik	Fabrik	Fabrik	Fabrik	1	1	1
5	Fabrik	Kadbod	Kadbod	Kadbod	0	0	0
6	Fabrik	Fabrik	Fabrik	Fabrik	1	1	1
7	Fabrik	Fabrik	Fabrik	Fabrik	1	1	1
8	Fabrik	Fabrik	Fabrik	Fabrik	1	1	1
9	Fabrik	Kayu	Kayu	Getah	0	0	0
10	Fabrik	Fabrik	Fabrik	Fabrik	1	1	1
11	Fabrik	Fabrik	Fabrik	Fabrik	1	1	1
12	Fabrik	Fabrik	Fabrik	Kayu	1	1	0
13	Fabrik	Fabrik	Fabrik	Fabrik	1	1	1
14	Fabrik	Getah	Fabrik	Fabrik	0	1	1
15	Fabrik	Fabrik	Fabrik	Fabrik	1	1	1
16	Fabrik	Fabrik	Fabrik	Fabrik	1	1	1
17	Fabrik	Fabrik	Fabrik	Fabrik	1	1	1
Jumlah objek yang dikesan betul					13	15	14
Ketepatan (Accuracy)					76.47	88.24	82.35

Ketepatan terendah = 76.4705882

Identifikasi Sampah Dengan Seni Bina CNN Pra-Latihan AlexNet

Jenis Sampah : 3. Pembungkus Makanan

No	Label Sebenarnya	Keputusan Ujian			Sampah Dikesan Betul		
		Label Ramalan Ke-1	Label Ramalan Ke-2	Label Ramalan Ke-3	Ke-1	Ke-2	Ke-3
1	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	1	1	1
2	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	1	1	1
3	Pembungkus makanan	Kadbod	Kadbod	Kadbod	0	0	0
4	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	1	1	1
5	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	1	1	1
6	Pembungkus makanan	Kadbod	Pembungkus makanan	Kadbod	0	1	0
7	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	1	1	1
8	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	1	1	1
9	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	1	1	1
10	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	1	1	1
11	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	1	1	1
12	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	1	1	1
13	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	1	1	1
14	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	1	1	1
15	Pembungkus makanan	Kertas	Kertas	Kertas	0	0	0
16	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	1	1	1
17	Pembungkus makanan	Kayu	Pembungkus makanan	Kayu	0	1	0
18	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	1	1	1
19	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	1	1	1
20	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	1	1	1
21	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	1	1	1

47	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	1	1	1
48	Pembungkus makanan	Pembungkus makanan	Kayu	Pembungkus makanan	1	0	1
49	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	1	1	1
50	Pembungkus makanan	Pembungkus makanan	Kadbod	Pembungkus makanan	1	0	1
51	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	1	1	1
52	Pembungkus makanan	Kertas	Pembungkus makanan	Kertas	0	1	0
53	Pembungkus makanan	Kertas	Pembungkus makanan	Pembungkus makanan	0	1	1
54	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	1	1	1
55	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	1	1	1
56	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	1	1	1
57	Pembungkus makanan	Kadbod	Pembungkus makanan	Pembungkus makanan	0	1	1
58	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	1	1	1
59	Pembungkus makanan	Kadbod	Pembungkus makanan	Kadbod	0	1	0
60	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	1	1	1
61	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	1	1	1
62	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	1	1	1
63	Pembungkus makanan	Pembungkus makanan	Kertas	Kertas	1	0	0
Jumlah objek yang dikesan betul					52	56	55
Ketepatan (Accuracy)					82.54	88.89	87.3

Ketepatan terendah = 82.5396825

Identifikasi Sampah Dengan Seni Bina CNN Pra-Latihan AlexNet

Jenis Sampah : 4. Buah

No	Label Sebenar	Keputusan Ujian			Sampah Dikesan Betul		
		Label Ramalan Ke-1	Label Ramalan Ke-2	Label Ramalan Ke-3	Ke-1	Ke-2	Ke-3
1	Buah	Buah	Buah	Buah	1	1	1
2	Buah	Buah	Buah	Buah	1	1	1
3	Buah	Daun	Buah	Daun	0	1	0
4	Buah	Buah	Buah	Buah	1	1	1
5	Buah	Fabrik	Fabrik	Fabrik	0	0	0
6	Buah	Buah	Buah	Buah	1	1	1
7	Buah	Buah	Buah	Buah	1	1	1
8	Buah	Buah	Buah	Buah	1	1	1
9	Buah	Buah	Buah	Buah	1	1	1
10	Buah	Buah	Buah	Buah	1	1	1
11	Buah	Kertas	Kertas	Buah	0	0	1
12	Buah	Buah	Buah	Buah	1	1	1
13	Buah	Buah	Buah	Buah	1	1	1
14	Buah	Buah	Buah	Buah	1	1	1
15	Buah	Fabrik	Fabrik	Buah	0	0	1
16	Buah	Buah	Buah	Daun	1	1	0
17	Buah	Buah	Buah	Buah	1	1	1
18	Buah	Buah	Buah	Buah	1	1	1
19	Buah	Buah	Buah	Buah	1	1	1
20	Buah	Buah	Buah	Buah	1	1	1
21	Buah	Buah	Buah	Buah	1	1	1
Jumlah objek yang dikesan betul					17	18	18
Ketepatan (Accuracy)					80.95	85.71	85.71

Ketepatan terendah = 80.952381

Identifikasi Sampah Dengan Seni Bina CNN Pra-Latihan AlexNet

Jenis Sampah : 5. Kaca

No	Label Sebenarnya	Keputusan Ujian			Sampah Dikesan Betul		
		Label Ramalan Ke-1	Label Ramalan Ke-2	Label Ramalan Ke-3	Ke-1	Ke-2	Ke-3
1	Kaca	Kaca	Kaca	Kaca	1	1	1
2	Kaca	Logam	Logam	Kaca	0	0	1
3	Kaca	Kaca	Kaca	Kaca	1	1	1
4	Kaca	Plastik	Plastik	Plastik	0	0	0
5	Kaca	Kaca	Kaca	Kaca	1	1	1
6	Kaca	Kaca	Kaca	Kaca	1	1	1
7	Kaca	Kaca	Kaca	Kaca	1	1	1
8	Kaca	Kaca	Kaca	Kaca	1	1	1
9	Kaca	Kaca	Pembungkus Makanan	Kaca	1	0	1
10	Kaca	Kaca	Kaca	Kaca	1	1	1
11	Kaca	Kaca	Kaca	Kaca	1	1	1
12	Kaca	Kaca	Kaca	Pembungkus Makanan	1	1	0
13	Kaca	Kaca	Kaca	Kaca	1	1	1
14	Kaca	Kaca	Kaca	Kaca	1	1	1
15	Kaca	Kaca	Kaca	Kaca	1	1	1
16	Kaca	Kaca	Kaca	Kaca	1	1	1
17	Kaca	Logam	Logam	Kaca	0	0	1
18	Kaca	Kaca	Kaca	Kaca	1	1	1
19	Kaca	Kaca	Kaca	Kaca	1	1	1
20	Kaca	Kaca	Kaca	Kaca	1	1	1
21	Kaca	Plastik	Plastik	Plastik	0	0	0
22	Kaca	Kaca	Kaca	Kaca	1	1	1
23	Kaca	Kaca	Kaca	Kaca	1	1	1
24	Kaca	Kaca	Kaca	Kaca	1	1	1
25	Kaca	Kaca	Kaca	Kaca	1	1	1
26	Kaca	Logam	Logam	Logam	0	0	0
27	Kaca	Kaca	Kaca	Kaca	1	1	1
28	Kaca	Pembungkus Makanan	Pembungkus Makanan	Kaca	0	0	1
29	Kaca	Kaca	Kaca	Kaca	1	1	1
30	Kaca	Kaca	Kaca	Kaca	1	1	1
31	Kaca	Kaca	Kaca	Kaca	1	1	1
32	Kaca	Kaca	Kaca	Kaca	1	1	1
33	Kaca	Kaca	Kaca	Kaca	1	1	1

34	Kaca	Kaca	Kaca	Kaca	1	1	1
Jumlah objek yang dikesan betul					28	27	30
Ketepatan (<i>Accuracy</i>)					82.35	79.41	88.24



Identifikasi Sampah Dengan Seni Bina CNN Pra-Latihan AlexNet

Jenis Sampah : 6. Daun

No	Label Sebenarnya	Keputusan Ujian			Sampah Dikesan Betul		
		Label Ramalan Ke-1	Label Ramalan Ke-2	Label Ramalan Ke-3	Ke-1	Ke-2	Ke-3
1	Daun	Daun	Daun	Daun	1	1	1
2	Daun	Daun	Daun	Daun	1	1	1
3	Daun	Kadbod	Kadbod	Daun	0	0	1
4	Daun	Daun	Daun	Daun	1	1	1
5	Daun	Daun	Daun	Daun	1	1	1
6	Daun	Daun	Daun	Daun	1	1	1
7	Daun	Daun	Daun	Daun	1	1	1
8	Daun	Daun	Daun	Daun	1	1	1
9	Daun	Daun	Daun	Daun	1	1	1
10	Daun	Daun	Daun	Daun	1	1	1
11	Daun	Daun	Daun	Daun	1	1	1
12	Daun	Fabrik	Fabrik	Fabrik	0	0	0
13	Daun	Daun	Daun	Daun	1	1	1
14	Daun	Daun	Daun	Daun	1	1	1
15	Daun	Pembungkus Makanan	Pembungkus Makanan	Daun	0	0	1
16	Daun	Daun	Daun	Daun	1	1	1
17	Daun	Daun	Daun	Daun	1	1	1
18	Daun	Buah	Buah	Buah	0	0	0
19	Daun	Daun	Daun	Daun	1	1	1
20	Daun	Daun	Daun	Daun	1	1	1
21	Daun	Buah	Buah	Buah	0	0	0
22	Daun	Daun	Daun	Daun	1	1	1
23	Daun	Daun	Daun	Daun	1	1	1
24	Daun	Daun	Daun	Daun	1	1	1
25	Daun	Kertas	Kertas	Daun	0	0	1
26	Daun	Daun	Daun	Daun	1	1	1
27	Daun	Daun	Daun	Daun	1	1	1
28	Daun	Daun	Daun	Daun	1	1	1
29	Daun	Daun	Daun	Daun	1	1	1
30	Daun	Daun	Daun	Daun	1	1	1
31	Daun	Daun	Daun	Daun	1	1	1
32	Daun	Daun	Daun	Daun	1	1	1
33	Daun	Daun	Daun	Daun	1	1	1
34	Daun	Daun	Daun	Daun	1	1	1
35	Daun	Daun	Daun	Daun	1	1	1
36	Daun	Daun	Daun	Daun	1	1	1

37	Daun	Daun	Daun	Daun	1	1	1
38	Daun	Daun	Daun	Daun	1	1	1
39	Daun	Daun	Daun	Daun	1	1	1
40	Daun	Daun	Daun	Daun	1	1	1
41	Daun	Daun	Daun	Daun	1	1	1
42	Daun	Daun	Daun	Daun	1	1	1
43	Daun	Daun	Daun	Daun	1	1	1
44	Daun	Daun	Daun	Daun	1	1	1
45	Daun	Kertas	Kertas	Kertas	0	0	0
46	Daun	Daun	Daun	Daun	1	1	1
47	Daun	Daun	Daun	Daun	1	1	1
48	Daun	Pembungkus Makanan	Pembungkus Makanan	Pembungkus Makanan	0	0	0
49	Daun	Daun	Daun	Daun	1	1	1
50	Daun	Daun	Daun	Daun	1	1	1
51	Daun	Daun	Buah	Buah	1	0	0
52	Daun	Daun	Daun	Daun	1	1	1
53	Daun	Daun	Buah	Buah	1	0	0
54	Daun	Daun	Daun	Daun	1	1	1
55	Daun	Daun	Daun	Daun	1	1	1
56	Daun	Daun	Daun	Daun	1	1	1
Jumlah objek yang dikesan betul					48	46	49
Ketepatan (Accuracy)					85.71	82.14	87.5

Ketepatan terendah = 82.1428571

Identifikasi Sampah Dengan Seni Bina CNN Pra-Latihan AlexNet

Jenis Sampah : 7.Logam

No	Label Sebenarnya	Keputusan Ujian			Sampah Dikesan Betul		
		Label Ramalan Ke-1	Label Ramalan Ke-2	Label Ramalan Ke-3	Ke-1	Ke-2	Ke-3
1	Logam	Logam	Logam	Logam	1	1	1
2	Logam	Logam	Logam	Logam	1	1	1
3	Logam	Pembungkus Makanan	Pembungkus Makanan	Logam	0	0	1
4	Logam	Logam	Logam	Logam	1	1	1
5	Logam	Kadbod	Kadbod	Kadbod	0	0	0
6	Logam	Logam	Logam	Logam	1	1	1
7	Logam	Logam	Logam	Logam	1	1	1
8	Logam	Logam	Logam	Logam	1	1	1
9	Logam	Logam	Logam	Logam	1	1	1
10	Logam	Logam	Logam	Logam	1	1	1
11	Logam	Logam	Kayu	Kayu	1	0	0
12	Logam	Logam	Logam	Logam	1	1	1
13	Logam	Logam	Logam	Logam	1	1	1
14	Logam	Logam	Logam	Logam	1	1	1
15	Logam	Logam	Pembungkus Makanan	Pembungkus Makanan	1	0	0
16	Logam	Logam	Logam	Logam	1	1	1
17	Logam	Logam	Logam	Logam	1	1	1
18	Logam	Pembungkus Makanan	Pembungkus Makanan	Pembungkus Makanan	0	0	0
19	Logam	Logam	Logam	Logam	1	1	1
20	Logam	Logam	Logam	Logam	1	1	1
21	Logam	Logam	Logam	Logam	1	1	1
Jumlah objek yang dikesan betul					18	16	17
Ketepatan (Accuracy)					85.71	76.19	80.95

Ketepatan terendah = 76.1904762

Identifikasi Sampah Dengan Seni Bina CNN Pra-Latihan AlexNet

Jenis Sampah : 8.Kertas

No	Label Sebenarnya	Keputusan Ujian			Sampah Dikesan Betul		
		Label Ramalan Ke-1	Label Ramalan Ke-2	Label Ramalan Ke-3	Ke-1	Ke-2	Ke-3
1	Kertas	Kertas	Kertas	Kertas	1	1	1
2	Kertas	Kertas	Kertas	Kertas	1	1	1
3	Kertas	Kadbod	Kadbod	Kadbod	0	0	0
4	Kertas	Kertas	Kertas	Kertas	1	1	1
5	Kertas	Kertas	Kertas	Kertas	1	1	1
6	Kertas	Kayu	Kayu	Kayu	0	0	0
7	Kertas	Kertas	Kertas	Kertas	1	1	1
8	Kertas	Kertas	Kertas	Kertas	1	1	1
9	Kertas	Kertas	Kertas	Kertas	1	1	1
10	Kertas	Kertas	Kertas	Kertas	1	1	1
11	Kertas	Kertas	Kertas	Kertas	1	1	1
12	Kertas	Kertas	Kertas	Kertas	1	1	1
13	Kertas	Kertas	Kertas	Kertas	1	1	1
14	Kertas	Kertas	Kertas	Kertas	1	1	1
15	Kertas	Kertas	Kertas	Kertas	1	1	1
16	Kertas	Pembungkus Makanan	Pembungkus Makanan	Kertas	0	0	1
17	Kertas	Kertas	Kertas	Kertas	1	1	1
18	Kertas	Kertas	Kertas	Kertas	1	1	1
19	Kertas	Kertas	Kertas	Kertas	1	1	1
20	Kertas	Kertas	Kertas	Kertas	1	1	1
21	Kertas	Kertas	Pembungkus Makanan	Pembungkus Makanan	1	0	0
22	Kertas	Kertas	Kertas	Kertas	1	1	1
23	Kertas	Kertas	Kertas	Kertas	1	1	1
24	Kertas	Kadbod	Kadbod	Kadbod	0	0	0
25	Kertas	Kadbod	Kadbod	Kertas	0	0	1
26	Kertas	Kertas	Kertas	Kertas	1	1	1
27	Kertas	Kertas	Kertas	Kertas	1	1	1
28	Kertas	Kertas	Kertas	Kertas	1	1	1
29	Kertas	Kertas	Kertas	Kertas	1	1	1
30	Kertas	Kertas	Kertas	Kertas	1	1	1
31	Kertas	Kertas	Kertas	Kertas	1	1	1
32	Kertas	Kertas	Kertas	Kertas	1	1	1
33	Kertas	Kertas	Kertas	Kertas	1	1	1
34	Kertas	Kertas	Kertas	Kertas	1	1	1
35	Kertas	Kertas	Kertas	Kertas	1	1	1

36	Kertas	Kertas	Kertas	Kertas	1	1	1
37	Kertas	Kertas	Kertas	Kertas	1	1	1
38	Kertas	Kertas	Kertas	Kertas	1	1	1
39	Kertas	Kertas	Kertas	Kertas	1	1	1
40	Kertas	Kertas	Kertas	Kertas	1	1	1
41	Kertas	Kertas	Kertas	Kertas	1	1	1
42	Kertas	Kertas	Kertas	Kertas	1	1	1
43	Kertas	Kertas	Kertas	Kertas	1	1	1
44	Kertas	Kertas	Kertas	Kertas	1	1	1
45	Kertas	Kertas	Kertas	Kertas	1	1	1
46	Kertas	Kertas	Kertas	Kertas	1	1	1
47	Kertas	Kadbod	Kadbod	Kadbod	0	0	0
48	Kertas	Kertas	Kertas	Kertas	1	1	1
49	Kertas	Kertas	Kertas	Kertas	1	1	1
50	Kertas	Kertas	Kertas	Kertas	1	1	1
51	Kertas	Kertas	Kertas	Kertas	1	1	1
52	Kertas	Kertas	Pembungkus Makanan	Pembungkus Makanan	1	0	0
53	Kertas	Kertas	Kertas	Kertas	1	1	1
54	Kertas	Kadbod	Kadbod	Kertas	0	0	1
55	Kertas	Kertas	Kertas	Kertas	1	1	1
56	Kertas	Kertas	Kertas	Kertas	1	1	1
57	Kertas	Kayu	Kertas	Kayu	0	1	0
58	Kertas	Kertas	Kadbod	Kadbod	1	0	0
59	Kertas	Kertas	Kertas	Kertas	1	1	1
60	Kertas	Kertas	Kertas	Kertas	1	1	1
61	Kertas	Kertas	Kertas	Kertas	1	1	1
Jumlah objek yang dikesan betul					53	51	53
Ketepatan (Accuracy)					86.89	83.61	86.89

Ketepatan terendah = 83.6065574

Identifikasi Sampah Dengan Seni Bina CNN Pra-Latihan AlexNet

Jenis Sampah : 9.Plastik

No	Label Sebenar	Keputusan Ujian			Sampah Dikesan Betul		
		Label Ramalan Ke-1	Label Ramalan Ke-2	Label Ramalan Ke-3	Ke-1	Ke-2	Ke-3
1	Plastik	Plastik	Plastik	Plastik	1	1	1
2	Plastik	Plastik	Plastik	Plastik	1	1	1
3	Plastik	Pembungkus Makanan	Pembungkus Makanan	Plastik	0	0	1
4	Plastik	Plastik	Plastik	Plastik	1	1	1
5	Plastik	Plastik	Plastik	Plastik	1	1	1
6	Plastik	Plastik	Plastik	Plastik	1	1	1
7	Plastik	Plastik	Plastik	Plastik	1	1	1
8	Plastik	Pembungkus Makanan	Pembungkus Makanan	Pembungkus Makanan	0	0	0
9	Plastik	Plastik	Plastik	Plastik	1	1	1
10	Plastik	Plastik	Plastik	Plastik	1	1	1
11	Plastik	Plastik	Plastik	Plastik	1	1	1
12	Plastik	Plastik	Plastik	Plastik	1	1	1
13	Plastik	Plastik	Plastik	Plastik	1	1	1
14	Plastik	Plastik	Logam	Logam	1	0	0
15	Plastik	Plastik	Plastik	Plastik	1	1	1
16	Plastik	Pembungkus Makanan	Pembungkus Makanan	Pembungkus Makanan	0	0	0
17	Plastik	Plastik	Plastik	Plastik	1	1	1
18	Plastik	Plastik	Plastik	Plastik	1	1	1
19	Plastik	Plastik	Plastik	Plastik	1	1	1
20	Plastik	Plastik	Plastik	Plastik	1	1	1
21	Plastik	Plastik	Plastik	Plastik	1	1	1
22	Plastik	Plastik	Plastik	Plastik	1	1	1
23	Plastik	Plastik	Plastik	Plastik	1	1	1
24	Plastik	Plastik	Plastik	Plastik	1	1	1
25	Plastik	Plastik	Pembungkus Makanan	Pembungkus Makanan	1	0	0
26	Plastik	Plastik	Plastik	Plastik	1	1	1
27	Plastik	Plastik	Plastik	Plastik	1	1	1
28	Plastik	Plastik	Plastik	Plastik	1	1	1
29	Plastik	Plastik	Plastik	Plastik	1	1	1
30	Plastik	Kaca	Kaca	Plastik	0	0	1
31	Plastik	Plastik	Plastik	Plastik	1	1	1
32	Plastik	Plastik	Kaca	Kaca	1	0	0
33	Plastik	Plastik	Plastik	Plastik	1	1	1
34	Plastik	Plastik	Plastik	Plastik	1	1	1

35	Plastik	Plastik	Plastik	Plastik	1	1	1
36	Plastik	Plastik	Plastik	Plastik	1	1	1
37	Plastik	Plastik	Plastik	Plastik	1	1	1
38	Plastik	Plastik	Plastik	Plastik	1	1	1
39	Plastik	Plastik	Plastik	Plastik	1	1	1
40	Plastik	Kaca	Kaca	Kaca	0	0	0
41	Plastik	Plastik	Plastik	Plastik	1	1	1
42	Plastik	Plastik	Plastik	Plastik	1	1	1
43	Plastik	Plastik	Plastik	Plastik	1	1	1
44	Plastik	Plastik	Kaca	Kaca	1	0	0
45	Plastik	Plastik	Plastik	Plastik	1	1	1
46	Plastik	Plastik	Plastik	Plastik	1	1	1
47	Plastik	Kaca	Kaca	Kaca	0	0	0
48	Plastik	Plastik	Plastik	Plastik	1	1	1
49	Plastik	Pembungkus Makanan	Plastik	Pembungkus Makanan	0	1	0
50	Plastik	Plastik	Plastik	Plastik	1	1	1
51	Plastik	Plastik	Plastik	Plastik	1	1	1
52	Plastik	Plastik	Plastik	Plastik	1	1	1
Jumlah objek yang dikesan betul					45	42	43
Ketepatan (Accuracy)					86.54	80.77	82.69

Ketepatan terendah = 80.7692308

Identifikasi Sampah Dengan Seni Bina CNN Pra-Latihan AlexNet

Jenis Sampah :

10. Getah

No	Label Sebenarnya	Keputusan Ujian			Sampah Dikesan Betul		
		Label Ramalan Ke-1	Label Ramalan Ke-2	Label Ramalan Ke-3	Ke-1	Ke-2	Ke-3
1	Getah	Getah	Getah	Getah	1	1	1
2	Getah	Getah	Getah	Getah	1	1	1
3	Getah	Fabrik	Fabrik	Fabrik	0	0	0
4	Getah	Getah	Getah	Getah	1	1	1
5	Getah	Getah	Getah	Getah	1	1	1
6	Getah	Pembungkus Makanan	Pembungkus Makanan	Getah	0	0	1
7	Getah	Getah	Getah	Getah	1	1	1
8	Getah	Getah	Getah	Getah	1	1	1
9	Getah	Getah	Getah	Getah	1	1	1
10	Getah	Fabrik	Fabrik	Getah	0	0	1
11	Getah	Getah	Getah	Getah	1	1	1
12	Getah	Getah	Getah	Getah	1	1	1
13	Getah	Getah	Getah	Getah	1	1	1
14	Getah	Fabrik	Fabrik	Fabrik	0	0	0
Jumlah objek yang dikesan betul					10	10	12
Ketepatan (Accuracy)					71.43	71.43	85.71

Ketepatan terendah = 71.4285714

Identifikasi Sampah Dengan Seni Bina CNN Pra-Latihan AlexNet

Jenis Sampah : 11.Kayu

No	Label Sebenarnya	Keputusan Ujian			Sampah Dikesan Betul		
		Label Ramalan Ke-1	Label Ramalan Ke-2	Label Ramalan Ke-3	Ke-1	Ke-2	Ke-3
1	Kayu	Kayu	Kayu	Kayu	1	1	1
2	Kayu	Kayu	Kayu	Kayu	1	1	1
3	Kayu	Kadbod	Kadbod	Kadbod	0	0	0
4	Kayu	Kayu	Kayu	Kayu	1	1	1
5	Kayu	Pembungkus Makanan	Pembungkus Makanan	Kayu	0	0	1
6	Kayu	Daun	Daun	Daun	0	0	0
7	Kayu	Kayu	Kayu	Kayu	1	1	1
8	Kayu	Kayu	Kayu	Kayu	1	1	1
9	Kayu	Kayu	Kayu	Kayu	1	1	1
10	Kayu	Kayu	Kayu	Kayu	1	1	1
11	Kayu	Kertas	Kertas	Kertas	0	0	0
12	Kayu	Kayu	Kayu	Kayu	1	1	1
13	Kayu	Kayu	Kayu	Kayu	1	1	1
14	Kayu	Kayu	Kayu	Kayu	1	1	1
15	Kayu	Kayu	Kayu	Kayu	1	1	1
16	Kayu	Kayu	Kayu	Kayu	1	1	1
17	Kayu	Kayu	Daun	Daun	1	0	0
18	Kayu	Kayu	Kayu	Kayu	1	1	1
19	Kayu	Kayu	Kayu	Kayu	1	1	1
20	Kayu	Kayu	Kayu	Kayu	1	1	1
21	Kayu	Kertas	Kertas	Kertas	0	0	0
22	Kayu	Kayu	Kayu	Kayu	1	1	1
23	Kayu	Kayu	Kayu	Kayu	1	1	1
24	Kayu	Kayu	Kayu	Kayu	1	1	1
25	Kayu	Kayu	Kayu	Kayu	1	1	1
26	Kayu	Kayu	Kayu	Kayu	1	1	1
27	Kayu	Kayu	Kayu	Kayu	1	1	1
Jumlah objek yang dikesan betul					22	21	22
Ketepatan (Accuracy)					81.48	77.78	81.48

Ketepatan terendah = 77.777778

LAMPIRAN C

KEPUTUSAN UJIAN SENI BINA VGG16

Identifikasi Sampah Dengan Seni Bina CNN Pra-Latihan VGG16

Jenis Sampah :

1.Kadbod

No	Label Sebenarnya	Keputusan Ujian			Sampah Dikesan Betul		
		Label Ramalan Ke-1	Label Ramalan Ke-2	Label Ramalan Ke-3	Ke-1	Ke-2	Ke-3
1	Kadbod	Kadbod	Kadbod	Kadbod	1	1	1
2	Kadbod	Kadbod	Kadbod	Kadbod	1	1	1
3	Kadbod	Kadbod	Kadbod	Kadbod	1	1	1
4	Kadbod	Kadbod	Kadbod	Kadbod	1	1	1
5	Kadbod	Kadbod	Kadbod	Kadbod	1	1	1
6	Kadbod	Kadbod	Pembungkus makanan	Pembungkus makanan	1	0	0
7	Kadbod	Kadbod	Kadbod	Kadbod	1	1	1
8	Kadbod	Kadbod	Kadbod	Kadbod	1	1	1
9	Kadbod	Kadbod	Kadbod	Kadbod	1	1	1
10	Kadbod	Kadbod	Kadbod	Kadbod	1	1	1
11	Kadbod	Kadbod	Kadbod	Kadbod	1	1	1
12	Kadbod	Kertas	Kadbod	Kertas	0	1	0
13	Kadbod	Kadbod	Kadbod	Kadbod	1	1	1
14	Kadbod	Kadbod	Kadbod	Kadbod	1	1	1
15	Kadbod	Kadbod	Kadbod	Kadbod	1	1	1
16	Kadbod	Kadbod	Kadbod	Kadbod	1	1	1
17	Kadbod	Kadbod	Kadbod	Kadbod	1	1	1
18	Kadbod	Kadbod	Kadbod	Kadbod	1	1	1
19	Kadbod	Kadbod	Kertas	Kadbod	1	0	1
20	Kadbod	Kadbod	Kadbod	Kadbod	1	1	1
21	Kadbod	Kadbod	Pembungkus makanan	Kadbod	1	0	1
22	Kadbod	Kadbod	Kadbod	Kadbod	1	1	1
23	Kadbod	Kadbod	Kadbod	Kadbod	1	1	1
24	Kadbod	Kadbod	Kadbod	Kadbod	1	1	1
25	Kadbod	Kadbod	Kadbod	Kadbod	1	1	1
26	Kadbod	Kertas	Pembungkus makanan	Kadbod	0	0	1
27	Kadbod	Kadbod	Kadbod	Kadbod	1	1	1
28	Kadbod	Kertas	Kadbod	Kertas	0	1	0
29	Kadbod	Kadbod	Kadbod	Kadbod	1	1	1
30	Kadbod	Kadbod	Kadbod	Kadbod	1	1	1
31	Kadbod	Kadbod	Kadbod	Kadbod	1	1	1
32	Kadbod	Kadbod	Kadbod	Kadbod	1	1	1
33	Kadbod	Kertas	Kadbod	Kertas	0	1	0
34	Kadbod	Kadbod	Kadbod	Kadbod	1	1	1

35	Kadbod	Kadbod	Kadbod	Kadbod	1	1	1
36	Kadbod	Kadbod	Kadbod	Kadbod	1	1	1
37	Kadbod	Kadbod	Kadbod	Kadbod	1	1	1
38	Kadbod	Kadbod	Kadbod	Kadbod	1	1	1
39	Kadbod	Kadbod	Kertas	Kadbod	1	0	1
40	Kadbod	Kadbod	Kadbod	Kadbod	1	1	1
41	Kadbod	Kadbod	Kadbod	Kadbod	1	1	1
42	Kadbod	Kadbod	Kadbod	Kadbod	1	1	1
43	Kadbod	Kadbod	Kadbod	Kadbod	1	1	1
44	Kadbod	Kadbod	Kadbod	Kadbod	1	1	1
45	Kadbod	Kadbod	Kadbod	Kadbod	1	1	1
Jumlah objek yang dikesan betul					41	40	41
Ketepatan (<i>Accuracy</i>)					91.11	88.89	91.11

Ketepatan terendah = 88.888889



**Identifikasi Sampah Dengan Seni Bina CNN Pra-Latihan
VGG16**

Jenis Sampah : 2.Fabrik

No	Label Sebenar	Keputusan Ujian			Sampah Dikesan Betul		
		Label Ramalan Ke-1	Label Ramalan Ke-2	Label Ramalan Ke-3	Ke-1	Ke-2	Ke-3
1	Fabrik	Fabrik	Fabrik	Fabrik	1	1	1
2	Fabrik	Kadbod	Fabrik	Fabrik	0	1	1
3	Fabrik	Fabrik	Fabrik	Fabrik	1	1	1
4	Fabrik	Fabrik	Fabrik	Fabrik	1	1	1
5	Fabrik	Kadbod	Kadbod	Kadbod	0	0	0
6	Fabrik	Fabrik	Fabrik	Fabrik	1	1	1
7	Fabrik	Fabrik	Fabrik	Fabrik	1	1	1
8	Fabrik	Fabrik	Fabrik	Fabrik	1	1	1
9	Fabrik	Fabrik	Fabrik	Getah	1	1	0
10	Fabrik	Getah	Fabrik	Fabrik	0	1	1
11	Fabrik	Fabrik	Fabrik	Fabrik	1	1	1
12	Fabrik	Fabrik	Kadbod	Kayu	1	0	0
13	Fabrik	Fabrik	Fabrik	Fabrik	1	1	1
14	Fabrik	Fabrik	Fabrik	Fabrik	1	1	1
15	Fabrik	Fabrik	Fabrik	Fabrik	1	1	1
16	Fabrik	Fabrik	Fabrik	Fabrik	1	1	1
17	Fabrik	Fabrik	Fabrik	Fabrik	1	1	1
Jumlah objek yang dikesan betul					14	15	14
Ketepatan (Accuracy)					82.35	88.24	82.35

Ketepatan terendah = 82.3529412

Identifikasi Sampah Dengan Seni Bina CNN Pra-Latihan VGG16
Jenis Sampah : 3. Pembungkus Makanan

No	Label Sebenarnya	Keputusan Ujian			Sampah Dikesan Betul		
		Label Ramalan Ke-1	Label Ramalan Ke-2	Label Ramalan Ke-3	Ke-1	Ke-2	Ke-3
1	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	1	1	1
2	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	1	1	1
3	Pembungkus makanan	Kadbod	Kadbod	Kadbod	0	0	0
4	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	1	1	1
5	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	1	1	1
6	Pembungkus makanan	Kadbod	Pembungkus makanan	Kadbod	0	1	0
7	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	1	1	1
8	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	1	1	1
9	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	1	1	1
10	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	1	1	1
11	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	1	1	1
12	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	1	1	1
13	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	1	1	1
14	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	1	1	1
15	Pembungkus makanan	Kertas	Kertas	Kertas	0	0	0
16	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	1	1	1
17	Pembungkus makanan	Kayu	Pembungkus makanan	Kayu	0	1	0
18	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	1	1	1
19	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	1	1	1
20	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	1	1	1
21	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	1	1	1

[illegible]

47	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	1	1	1
48	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	1	1	1
49	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	1	1	1
50	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	1	1	1
51	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	1	1	1
52	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	1	1	1
53	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	1	1	1
54	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	1	1	1
55	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	1	1	1
56	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	1	1	1
57	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	1	1	1
58	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	1	1	1
59	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	1	1	1
60	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	1	1	1
61	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	1	1	1
62	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	1	1	1
63	Pembungkus makanan	Pembungkus makanan	Kertas	Pembungkus makanan	1	0	1
Jumlah objek yang dikesan betul					58	57	59
Ketepatan (Accuracy)					92.06	90.48	93.65

Ketepatan terendah = 90.4761905

Identifikasi Sampah Dengan Seni Bina CNN Pra-Latihan

VGG16

Jenis Sampah : 4. Buah

No	Label Sebenar	Keputusan Ujian			Sampah Dikesan Betul		
		Label Ramalan Ke-1	Label Ramalan Ke-2	Label Ramalan Ke-3	Ke-1	Ke-2	Ke-3
1	Buah	Buah	Buah	Buah	1	1	1
2	Buah	Buah	Buah	Buah	1	1	1
3	Buah	Buah	Buah	Daun	1	1	0
4	Buah	Buah	Buah	Buah	1	1	1
5	Buah	Buah	Fabrik	Buah	1	0	1
6	Buah	Buah	Buah	Buah	1	1	1
7	Buah	Buah	Buah	Buah	1	1	1
8	Buah	Buah	Buah	Buah	1	1	1
9	Buah	Buah	Buah	Buah	1	1	1
10	Buah	Buah	Buah	Buah	1	1	1
11	Buah	Buah	Buah	Buah	1	1	1
12	Buah	Buah	Buah	Buah	1	1	1
13	Buah	Buah	Buah	Buah	1	1	1
14	Buah	Buah	Buah	Buah	1	1	1
15	Buah	Fabrik	Buah	Buah	0	1	1
16	Buah	Buah	Buah	Daun	1	1	0
17	Buah	Buah	Buah	Buah	1	1	1
18	Buah	Buah	Buah	Buah	1	1	1
19	Buah	Buah	Buah	Buah	1	1	1
20	Buah	Buah	Buah	Buah	1	1	1
21	Buah	Buah	Buah	Buah	1	1	1
Jumlah objek yang dikesan betul					20	20	19
Ketepatan (<i>Accuracy</i>)					95.24	95.24	90.48

Ketepatan terendah = 90.4761905

Identifikasi Sampah Dengan Seni Bina CNN Pra-Latihan VGG16

Jenis Sampah : 5. Kaca

No	Label Sebenarnya	Keputusan Ujian			Sampah Dikesan Betul		
		Label Ramalan Ke-1	Label Ramalan Ke-2	Label Ramalan Ke-3	Ke-1	Ke-2	Ke-3
1	Kaca	Kaca	Kaca	Kaca	1	1	1
2	Kaca	Kaca	Kaca	Kaca	1	1	1
3	Kaca	Kaca	Kaca	Kaca	1	1	1
4	Kaca	Kaca	Kaca	Plastik	1	1	0
5	Kaca	Kaca	Logam	Logam	1	0	0
6	Kaca	Kaca	Kaca	Kaca	1	1	1
7	Kaca	Kaca	Kaca	Pembungkus Makanan	1	1	0
8	Kaca	Kaca	Kaca	Kaca	1	1	1
9	Kaca	Kaca	Kaca	Kaca	1	1	1
10	Kaca	Kaca	Kaca	Kaca	1	1	1
11	Kaca	Kaca	Kaca	Kaca	1	1	1
12	Kaca	Kaca	Kaca	Kaca	1	1	1
13	Kaca	Kaca	Kaca	Kaca	1	1	1
14	Kaca	Kaca	Logam	Kaca	1	0	1
15	Kaca	Kaca	Kaca	Kaca	1	1	1
16	Kaca	Kaca	Kaca	Kaca	1	1	1
17	Kaca	Kaca	Kaca	Kaca	1	1	1
18	Kaca	Kaca	Kaca	Kaca	1	1	1
19	Kaca	Kaca	Kaca	Kaca	1	1	1
20	Kaca	Kaca	Kaca	Kaca	1	1	1
21	Kaca	Plastik	Plastik	Kaca	0	0	1
22	Kaca	Kaca	Kaca	Kaca	1	1	1
23	Kaca	Logam	Logam	Kaca	0	0	1
24	Kaca	Kaca	Kaca	Kaca	1	1	1
25	Kaca	Kaca	Kaca	Kaca	1	1	1
26	Kaca	Kaca	Kaca	Kaca	1	1	1
27	Kaca	Kaca	Kaca	Kaca	1	1	1
28	Kaca	Kaca	Kaca	Kaca	1	1	1
29	Kaca	Kaca	Kaca	Kaca	1	1	1
30	Kaca	Kaca	Kaca	Kaca	1	1	1
31	Kaca	Kaca	Kaca	Kaca	1	1	1
32	Kaca	Kaca	Kaca	Kaca	1	1	1
33	Kaca	Kaca	Kaca	Kaca	1	1	1
34	Kaca	Kaca	Kaca	Kaca	1	1	1
Jumlah objek yang dikesan betul					32	30	31
Ketepatan (<i>Accuracy</i>)					94.12	88.24	91.18

Ketepatan terendah = 88.2352941

Identifikasi Sampah Dengan Seni Bina CNN Pra-Latihan VGG16
Jenis Sampah : 6. Daun

No	Label Sebenarnya	Keputusan Ujian			Sampah Dikesan Betul		
		Label Ramalan Ke-1	Label Ramalan Ke-2	Label Ramalan Ke-3	Ke-1	Ke-2	Ke-3
1	Daun	Daun	Daun	Daun	1	1	1
2	Daun	Daun	Daun	Daun	1	1	1
3	Daun	Daun	Daun	Daun	1	1	1
4	Daun	Daun	Daun	Daun	1	1	1
5	Daun	Daun	Daun	Daun	1	1	1
6	Daun	Daun	Daun	Daun	1	1	1
7	Daun	Fabrik	Fabrik	Daun	0	0	1
8	Daun	Daun	Daun	Daun	1	1	1
9	Daun	Daun	Daun	Daun	1	1	1
10	Daun	Daun	Daun	Daun	1	1	1
11	Daun	Daun	Kadbod	Kadbod	1	0	0
12	Daun	Daun	Daun	Daun	1	1	1
13	Daun	Daun	Daun	Daun	1	1	1
14	Daun	Daun	Daun	Daun	1	1	1
15	Daun	Pembungkus Makanan	Pembungkus Makanan	Daun	0	0	1
16	Daun	Daun	Daun	Daun	1	1	1
17	Daun	Daun	Daun	Daun	1	1	1
18	Daun	Buah	Daun	Buah	0	1	0
19	Daun	Daun	Daun	Daun	1	1	1
20	Daun	Daun	Daun	Daun	1	1	1
21	Daun	Buah	Daun	Buah	0	1	0
22	Daun	Daun	Daun	Daun	1	1	1
23	Daun	Daun	Daun	Daun	1	1	1
24	Daun	Daun	Daun	Daun	1	1	1
25	Daun	Daun	Daun	Daun	1	1	1
26	Daun	Daun	Daun	Daun	1	1	1
27	Daun	Daun	Daun	Daun	1	1	1
28	Daun	Kertas	Daun	Daun	0	1	1
29	Daun	Daun	Daun	Daun	1	1	1
30	Daun	Daun	Kertas	Daun	1	0	1
31	Daun	Daun	Daun	Daun	1	1	1
32	Daun	Daun	Kertas	Daun	1	0	1
33	Daun	Daun	Daun	Daun	1	1	1
34	Daun	Daun	Daun	Daun	1	1	1
35	Daun	Daun	Daun	Daun	1	1	1
36	Daun	Daun	Daun	Daun	1	1	1
37	Daun	Daun	Daun	Daun	1	1	1
38	Daun	Daun	Daun	Daun	1	1	1
39	Daun	Daun	Daun	Daun	1	1	1
40	Daun	Daun	Daun	Daun	1	1	1
41	Daun	Daun	Daun	Buah	1	1	0

42	Daun	Daun	Daun	Daun	1	1	1
43	Daun	Daun	Daun	Daun	1	1	1
44	Daun	Daun	Daun	Daun	1	1	1
45	Daun	Kertas	Daun	Kertas	0	1	0
46	Daun	Daun	Daun	Daun	1	1	1
47	Daun	Daun	Daun	Daun	1	1	1
48	Daun	Daun	Daun	Daun	1	1	1
49	Daun	Daun	Daun	Daun	1	1	1
50	Daun	Daun	Daun	Daun	1	1	1
51	Daun	Daun	Daun	Daun	1	1	1
52	Daun	Daun	Daun	Daun	1	1	1
53	Daun	Daun	Daun	Daun	1	1	1
54	Daun	Daun	Daun	Daun	1	1	1
55	Daun	Daun	Daun	Daun	1	1	1
56	Daun	Buah	Daun	Buah	0	1	0
Jumlah objek yang dikesan betul					49	51	50
Ketepatan (<i>Accuracy</i>)					87.5	91.07	89.29

Ketepatan terendah = 87.5

Identifikasi Sampah Dengan Seni Bina CNN Pra-Latihan VGG16

Jenis Sampah : 7.Logam

No	Label Sebenarnya	Keputusan Ujian			Sampah Dikesan Betul		
		Label Ramalan Ke-1	Label Ramalan Ke-2	Label Ramalan Ke-3	Ke-1	Ke-2	Ke-3
1	Logam	Logam	Logam	Logam	1	1	1
2	Logam	Logam	Logam	Logam	1	1	1
3	Logam	Logam	Logam	Logam	1	1	1
4	Logam	Logam	Logam	Logam	1	1	1
5	Logam	Logam	Logam	Kadbod	1	1	0
6	Logam	Logam	Pembungkus Makanan	Logam	1	0	1
7	Logam	Logam	Logam	Logam	1	1	1
8	Logam	Logam	Kadbod	Logam	1	0	1
9	Logam	Logam	Logam	Logam	1	1	1
10	Logam	Logam	Logam	Logam	1	1	1
11	Logam	Pembungkus Makanan	Logam	Kayu	0	1	0
12	Logam	Logam	Logam	Logam	1	1	1
13	Logam	Logam	Logam	Logam	1	1	1
14	Logam	Logam	Logam	Logam	1	1	1
15	Logam	Logam	Logam	Logam	1	1	1
16	Logam	Logam	Logam	Logam	1	1	1
17	Logam	Logam	Logam	Logam	1	1	1
18	Logam	Pembungkus Makanan	Logam	Logam	0	1	1
19	Logam	Logam	Logam	Logam	1	1	1
20	Logam	Logam	Logam	Logam	1	1	1
21	Logam	Logam	Pembungkus Makanan	Logam	1	0	1
Jumlah objek yang dikesan betul					19	18	19
Ketepatan (Accuracy)					90.48	85.71	90.48

Ketepatan terendah = 85.7142857

Identifikasi Sampah Dengan Seni Bina CNN Pra-Latihan VGG16

Jenis Sampah : 8.Kertas

No	Label Sebenarnya	Keputusan Ujian			Sampah Dikesan Betul		
		Label Ramalan Ke-1	Label Ramalan Ke-2	Label Ramalan Ke-3	Ke-1	Ke-2	Ke-3
1	Kertas	Kertas	Kertas	Kertas	1	1	1
2	Kertas	Kertas	Kertas	Kertas	1	1	1
3	Kertas	Kertas	Kayu	Kayu	1	0	0
4	Kertas	Kertas	Kertas	Kertas	1	1	1
5	Kertas	Kertas	Kertas	Kertas	1	1	1
6	Kertas	Kertas	Kertas	Kertas	1	1	1
7	Kertas	Kertas	Kertas	Kertas	1	1	1
8	Kertas	Kertas	Kadbod	Kadbod	1	0	0
9	Kertas	Kertas	Kertas	Kertas	1	1	1
10	Kertas	Kertas	Kertas	Kertas	1	1	1
11	Kertas	Kertas	Kertas	Kertas	1	1	1
12	Kertas	Kertas	Kertas	Kertas	1	1	1
13	Kertas	Kertas	Kertas	Kertas	1	1	1
14	Kertas	Kertas	Kertas	Kertas	1	1	1
15	Kertas	Kertas	Kertas	Kertas	1	1	1
16	Kertas	Kertas	Pembungkus Makanan	Kertas	1	0	1
17	Kertas	Kertas	Kertas	Kertas	1	1	1
18	Kertas	Kertas	Kertas	Kertas	1	1	1
19	Kertas	Kertas	Kertas	Kertas	1	1	1
20	Kertas	Kertas	Kertas	Kertas	1	1	1
21	Kertas	Kertas	Pembungkus Makanan	Kertas	1	0	1
22	Kertas	Kertas	Kertas	Kertas	1	1	1
23	Kertas	Kertas	Kertas	Kertas	1	1	1
24	Kertas	Kadbod	Kadbod	Kadbod	0	0	0
25	Kertas	Kadbod	Kadbod	Kertas	0	0	1
26	Kertas	Kertas	Kertas	Kertas	1	1	1
27	Kertas	Kertas	Kertas	Kertas	1	1	1
28	Kertas	Kertas	Kertas	Kertas	1	1	1
29	Kertas	Kertas	Kertas	Kertas	1	1	1
30	Kertas	Kertas	Kertas	Kertas	1	1	1
31	Kertas	Kertas	Kertas	Pembungkus Makanan	1	1	0
32	Kertas	Kertas	Kertas	Kertas	1	1	1
33	Kertas	Kertas	Kertas	Kertas	1	1	1
34	Kertas	Kertas	Kertas	Kertas	1	1	1
35	Kertas	Kertas	Kertas	Kertas	1	1	1
36	Kertas	Kertas	Kertas	Kertas	1	1	1
37	Kertas	Kertas	Kertas	Kertas	1	1	1
38	Kertas	Kertas	Kertas	Kertas	1	1	1
39	Kertas	Kertas	Kertas	Kertas	1	1	1

40	Kertas	Kertas	Kertas	Kertas	1	1	1
41	Kertas	Kadbod	Kertas	Kertas	0	1	1
42	Kertas	Kertas	Kertas	Kertas	1	1	1
43	Kertas	Kertas	Kertas	Kertas	1	1	1
44	Kertas	Kayu	Kertas	Kertas	0	1	1
45	Kertas	Kertas	Kertas	Kertas	1	1	1
46	Kertas	Kertas	Kertas	Kertas	1	1	1
47	Kertas	Kertas	Kertas	Kertas	1	1	1
48	Kertas	Kadbod	Pembungkus Makanan	Kertas	0	0	1
49	Kertas	Kertas	Kertas	Kertas	1	1	1
50	Kertas	Kertas	Kertas	Kertas	1	1	1
51	Kertas	Kertas	Kertas	Kertas	1	1	1
52	Kertas	Kertas	Kertas	Kertas	1	1	1
53	Kertas	Kertas	Kertas	Kertas	1	1	1
54	Kertas	Kertas	Kertas	Kertas	1	1	1
55	Kertas	Kertas	Kertas	Kertas	1	1	1
56	Kertas	Kertas	Kertas	Kertas	1	1	1
57	Kertas	Kertas	Kertas	Kayu	1	1	0
58	Kertas	Kertas	Kertas	Pembungkus Makanan	1	1	0
59	Kertas	Kertas	Kertas	Kertas	1	1	1
60	Kertas	Kertas	Kertas	Kertas	1	1	1
61	Kertas	Kertas	Kertas	Kertas	1	1	1
Jumlah objek yang dikesan betul					56	54	55
Ketepatan (Accuracy)					91.8	88.52	90.16

Ketepatan terendah = 88.5245902

Identifikasi Sampah Dengan Seni Bina CNN Pra-Latihan VGG16
Jenis Sampah : 9.Plastik

No	Label Sebenarnya	Keputusan Ujian			Sampah Dikesan Betul		
		Label Ramalan Ke-1	Label Ramalan Ke-2	Label Ramalan Ke-3	Ke-1	Ke-2	Ke-3
1	Plastik	Plastik	Plastik	Plastik	1	1	1
2	Plastik	Plastik	Plastik	Plastik	1	1	1
3	Plastik	Pembungkus Makan	Pembungkus Makanan	Plastik	0	0	1
4	Plastik	Plastik	Plastik	Plastik	1	1	1
5	Plastik	Plastik	Plastik	Plastik	1	1	1
6	Plastik	Plastik	Plastik	Plastik	1	1	1
7	Plastik	Plastik	Plastik	Plastik	1	1	1
8	Plastik	Pembungkus Makanan	Pembungkus Makanan	Pembungkus Makanan	0	0	0
9	Plastik	Plastik	Plastik	Plastik	1	1	1
10	Plastik	Plastik	Plastik	Plastik	1	1	1
11	Plastik	Plastik	Plastik	Pembungkus Makanan	1	1	0
12	Plastik	Plastik	Pembungkus Makanan	Plastik	1	0	1
13	Plastik	Plastik	Plastik	Plastik	1	1	1
14	Plastik	Plastik	Plastik	Plastik	1	1	1
15	Plastik	Plastik	Plastik	Plastik	1	1	1
16	Plastik	Pembungkus Makanan	Plastik	Pembungkus Makanan	0	1	0
17	Plastik	Plastik	Plastik	Plastik	1	1	1
18	Plastik	Plastik	Plastik	Plastik	1	1	1
19	Plastik	Plastik	Plastik	Plastik	1	1	1
20	Plastik	Plastik	Plastik	Plastik	1	1	1
21	Plastik	Plastik	Plastik	Plastik	1	1	1
22	Plastik	Plastik	Pembungkus Makanan	Plastik	1	0	1
23	Plastik	Plastik	Plastik	Plastik	1	1	1
24	Plastik	Plastik	Plastik	Plastik	1	1	1
25	Plastik	Plastik	Plastik	Plastik	1	1	1
26	Plastik	Plastik	Plastik	Plastik	1	1	1
27	Plastik	Plastik	Plastik	Plastik	1	1	1
28	Plastik	Plastik	Plastik	Plastik	1	1	1
29	Plastik	Plastik	Plastik	Plastik	1	1	1
30	Plastik	Kaca	Plastik	Kaca	0	1	0
31	Plastik	Plastik	Plastik	Plastik	1	1	1
32	Plastik	Plastik	Plastik	Pembungkus Makanan	1	1	0
33	Plastik	Plastik	Plastik	Plastik	1	1	1
34	Plastik	Plastik	Plastik	Plastik	1	1	1
35	Plastik	Plastik	Kaca	Kaca	1	0	0

36	Plastik	Plastik	Plastik	Plastik	1	1	1
37	Plastik	Plastik	Plastik	Plastik	1	1	1
38	Plastik	Plastik	Plastik	Plastik	1	1	1
39	Plastik	Plastik	Plastik	Plastik	1	1	1
40	Plastik	Kaca	Plastik	Plastik	0	1	1
41	Plastik	Plastik	Plastik	Plastik	1	1	1
42	Plastik	Plastik	Plastik	Plastik	1	1	1
43	Plastik	Plastik	Plastik	Plastik	1	1	1
44	Plastik	Plastik	Kaca	Kaca	1	0	0
45	Plastik	Plastik	Plastik	Plastik	1	1	1
46	Plastik	Plastik	Plastik	Plastik	1	1	1
47	Plastik	Kaca	Plastik	Plastik	0	1	1
48	Plastik	Plastik	Plastik	Plastik	1	1	1
49	Plastik	Plastik	Plastik	Plastik	1	1	1
50	Plastik	Plastik	Plastik	Plastik	1	1	1
51	Plastik	Plastik	Plastik	Plastik	1	1	1
52	Plastik	Plastik	Plastik	Plastik	1	1	1
Jumlah objek yang dikesan betul					46	46	45
Ketepatan (<i>Accuracy</i>)					88.46	88.46	86.54

Ketepatan terendah = 86.5384615

Identifikasi Sampah Dengan Seni Bina CNN Pra-Latihan

VGG16

Jenis Sampah :

10. Getah

No	Label Sebenarnya	Keputusan Ujian			Sampah Dikesan Betul		
		Label Ramalan Ke-1	Label Ramalan Ke-2	Label Ramalan Ke-3	Ke-1	Ke-2	Ke-3
1	Getah	Getah	Getah	Getah	1	1	1
2	Getah	Getah	Getah	Getah	1	1	1
3	Getah	Fabrik	Getah	Fabrik	0	1	0
4	Getah	Getah	Getah	Getah	1	1	1
5	Getah	Getah	Getah	Getah	1	1	1
6	Getah	Getah	Getah	Getah	1	1	1
7	Getah	Getah	Getah	Getah	1	1	1
8	Getah	Getah	Getah	Fabrik	1	1	0
9	Getah	Getah	Getah	Getah	1	1	1
10	Getah	Fabrik	Getah	Getah	0	1	1
11	Getah	Getah	Getah	Getah	1	1	1
12	Getah	Getah	Fabrik	Getah	1	0	1
13	Getah	Getah	Getah	Getah	1	1	1
14	Getah	Fabrik	Getah	Getah	0	1	1
Jumlah objek yang dikesan betul					11	13	12
Ketepatan (Accuracy)					78.57	92.86	85.71

Ketepatan terendah = 78.5714286

Identifikasi Sampah Dengan Seni Bina CNN Pra-Latihan VGG16

Jenis Sampah : 11.Kayu

No	Label Sebenarnya	Keputusan Ujian			Sampah Dikesan Betul		
		Label Ramalan Ke-1	Label Ramalan Ke-2	Label Ramalan Ke-3	Ke-1	Ke-2	Ke-3
1	Kayu	Kayu	Kayu	Kayu	1	1	1
2	Kayu	Kayu	Pembungkus Makanan	Kayu	1	0	1
3	Kayu	Kayu	Kayu	Kayu	1	1	1
4	Kayu	Kayu	Kayu	Kayu	1	1	1
5	Kayu	Kayu	Kayu	Kayu	1	1	1
6	Kayu	Daun	Kayu	Kayu	0	1	1
7	Kayu	Kayu	Kayu	Kadbod	1	1	0
8	Kayu	Kayu	Kayu	Kertas	1	1	0
9	Kayu	Kayu	Kayu	Kayu	1	1	1
10	Kayu	Kayu	Kayu	Kayu	1	1	1
11	Kayu	Kertas	Kertas	Kayu	0	0	1
12	Kayu	Kayu	Kayu	Kayu	1	1	1
13	Kayu	Kayu	Kayu	Kayu	1	1	1
14	Kayu	Kayu	Kayu	Kayu	1	1	1
15	Kayu	Kayu	Kayu	Kayu	1	1	1
16	Kayu	Kayu	Kayu	Kayu	1	1	1
17	Kayu	Kayu	Kayu	Kayu	1	1	1
18	Kayu	Kayu	Kayu	Kertas	1	1	0
19	Kayu	Daun	Kayu	Kayu	0	1	1
20	Kayu	Kayu	Kayu	Kayu	1	1	1
21	Kayu	Kayu	Kertas	Kayu	1	0	1
22	Kayu	Kayu	Kayu	Kayu	1	1	1
23	Kayu	Kayu	Kayu	Kayu	1	1	1
24	Kayu	Kertas	Kayu	Kayu	0	1	1
25	Kayu	Kayu	Kayu	Kayu	1	1	1
26	Kayu	Kayu	Kayu	Kayu	1	1	1
27	Kayu	Kayu	Kayu	Kayu	1	1	1
Jumlah objek yang dikesan betul					23	24	24
Ketepatan (Accuracy)					85.19	88.89	88.89

Ketepatan terendah = 85.1851852

LAMPIRAN D

KEPUTUSAN UJIAN SENI BINA GOOGLNET

Identifikasi Sampah Dengan Seni Bina CNN Pra-Latihan GoogleNet

Jenis Sampah :

1.Kadbod

No	Label Sebenarnya	Keputusan Ujian			Sampah Dikesan Betul		
		Label Ramalan Ke-1	Label Ramalan Ke-2	Label Ramalan Ke-3	Ke-1	Ke-2	Ke-3
1	Kadbod	Kadbod	Kadbod	Kadbod	1	1	1
2	Kadbod	Kadbod	Kadbod	Kadbod	1	1	1
3	Kadbod	Kadbod	Kadbod	Kadbod	1	1	1
4	Kadbod	Kadbod	Kadbod	Kadbod	1	1	1
5	Kadbod	Kadbod	Kadbod	Kadbod	1	1	1
6	Kadbod	Kadbod	Kadbod	Kadbod	1	1	1
7	Kadbod	Kadbod	Kadbod	Kadbod	1	1	1
8	Kadbod	Kadbod	Kadbod	Kadbod	1	1	1
9	Kadbod	Kadbod	Pembungkus makanan	Kadbod	1	0	1
10	Kadbod	Kadbod	Kadbod	Kadbod	1	1	1
11	Kadbod	Kadbod	Kadbod	Kadbod	1	1	1
12	Kadbod	Kertas	Kadbod	Kadbod	0	1	1
13	Kadbod	Kadbod	Kadbod	Kadbod	1	1	1
14	Kadbod	Kadbod	Kadbod	Kadbod	1	1	1
15	Kadbod	Kadbod	Kadbod	Kadbod	1	1	1
16	Kadbod	Kadbod	Kadbod	Kadbod	1	1	1
17	Kadbod	Kadbod	Kadbod	Kadbod	1	1	1
18	Kadbod	Kadbod	Kadbod	Kadbod	1	1	1
19	Kadbod	Kadbod	Kadbod	Kadbod	1	1	1
20	Kadbod	Kadbod	Kadbod	Kadbod	1	1	1
21	Kadbod	Kadbod	Kadbod	Kadbod	1	1	1
22	Kadbod	Kadbod	Kadbod	Kadbod	1	1	1
23	Kadbod	Kadbod	Kadbod	Kadbod	1	1	1
24	Kadbod	Kadbod	Kadbod	Kadbod	1	1	1
25	Kadbod	Kadbod	Kadbod	Kadbod	1	1	1
26	Kadbod	Kadbod	Kadbod	Kadbod	1	1	1
27	Kadbod	Kadbod	Kadbod	Kadbod	1	1	1
28	Kadbod	Kadbod	Kadbod	Kadbod	1	1	1
29	Kadbod	Kadbod	Kadbod	Kadbod	1	1	1
30	Kadbod	Kadbod	Kadbod	Kadbod	1	1	1
31	Kadbod	Kadbod	Kadbod	Kadbod	1	1	1
32	Kadbod	Kadbod	Kadbod	Kadbod	1	1	1
33	Kadbod	Kadbod	Kadbod	Kadbod	1	1	1
34	Kadbod	Kadbod	Kadbod	Kadbod	1	1	1
35	Kadbod	Kadbod	Kadbod	Kadbod	1	1	1
36	Kadbod	Kadbod	Kadbod	Kadbod	1	1	1

37	Kadbod	Kadbod	Kadbod	Kadbod	1	1	1
38	Kadbod	Kadbod	Kadbod	Kadbod	1	1	1
39	Kadbod	Kadbod	Kadbod	Kadbod	1	1	1
40	Kadbod	Kadbod	Kadbod	Kadbod	1	1	1
41	Kadbod	Kadbod	Kadbod	Kadbod	1	1	1
42	Kadbod	Kadbod	Kadbod	Kadbod	1	1	1
43	Kadbod	Kadbod	Kadbod	Kadbod	1	1	1
44	Kadbod	Kadbod	Kadbod	Kadbod	1	1	1
45	Kadbod	Kadbod	Kadbod	Kadbod	1	1	1
Jumlah objek yang dikesan betul					44	44	45
Ketepatan (<i>Accuracy</i>)					97.78	97.78	100

Ketepatan terendah = 97.777778



Identifikasi Sampah Dengan Seni Bina CNN Pra-Latihan GoogleNet

Jenis Sampah : 2.Fabrik

No	Label Sebenar	Keputusan Ujian			Sampah Dikesan Betul		
		Label Ramalan Ke-1	Label Ramalan Ke-2	Label Ramalan Ke-3	Ke-1	Ke-2	Ke-3
1	Fabrik	Fabrik	Fabrik	Fabrik	1	1	1
2	Fabrik	Fabrik	Fabrik	Fabrik	1	1	1
3	Fabrik	Fabrik	Fabrik	Fabrik	1	1	1
4	Fabrik	Fabrik	Fabrik	Fabrik	1	1	1
5	Fabrik	Fabrik	Fabrik	Fabrik	1	1	1
6	Fabrik	Fabrik	Fabrik	Fabrik	1	1	1
7	Fabrik	Kadbod	Fabrik	Fabrik	0	1	1
8	Fabrik	Fabrik	Fabrik	Fabrik	1	1	1
9	Fabrik	Fabrik	Fabrik	Fabrik	1	1	1
10	Fabrik	Fabrik	Fabrik	Kadbod	1	1	0
11	Fabrik	Fabrik	Fabrik	Fabrik	1	1	1
12	Fabrik	Fabrik	Fabrik	Fabrik	1	1	1
13	Fabrik	Fabrik	Fabrik	Fabrik	1	1	1
14	Fabrik	Fabrik	Fabrik	Fabrik	1	1	1
15	Fabrik	Fabrik	Fabrik	Fabrik	1	1	1
16	Fabrik	Fabrik	Fabrik	Fabrik	1	1	1
17	Fabrik	Fabrik	Fabrik	Fabrik	1	1	1
Jumlah objek yang dikesan betul					16	17	16
Ketepatan (Accuracy)					94.12	100	94.12

Ketepatan terendah = 94.1176471

Identifikasi Sampah Dengan Seni Bina CNN Pra-Latihan GoogleNet
Jenis Sampah : 3. Pembungkus Makanan

No	Label Sebenarnya	Keputusan Ujian			Sampah Dikesan Betul		
		Label Ramalan Ke-1	Label Ramalan Ke-2	Label Ramalan Ke-3	Ke-1	Ke-2	Ke-3
1	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	1	1	1
2	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	1	1	1
3	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	1	1	1
4	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	1	1	1
5	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	1	1	1
6	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	1	1	1
7	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	1	1	1
8	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	1	1	1
9	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	1	1	1
10	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	1	1	1
11	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	1	1	1
12	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	1	1	1
13	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	1	1	1
14	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	1	1	1
15	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	1	1	1
16	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	1	1	1
17	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	1	1	1
18	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	1	1	1
19	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	1	1	1
20	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	1	1	1
21	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	1	1	1

47	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	1	1	1
48	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	1	1	1
49	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	1	1	1
50	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	1	1	1
51	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	1	1	1
52	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	1	1	1
53	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	1	1	1
54	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	1	1	1
55	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	1	1	1
56	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	1	1	1
57	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	1	1	1
58	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	1	1	1
59	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	1	1	1
60	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	1	1	1
61	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	1	1	1
62	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	1	1	1
63	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	1	1	1
Jumlah objek yang dikesan betul					62	63	62
Ketepatan (<i>Accuracy</i>)					98.41	100	98.41

Ketepatan terendah = 98.4126984

Identifikasi Sampah Dengan Seni Bina CNN Pra-Latihan GoogleNet

Jenis Sampah : 4. Buah

No	Label Sebenarnya	Keputusan Ujian			Sampah Dikesan Betul		
		Label Ramalan Ke-1	Label Ramalan Ke-2	Label Ramalan Ke-3	Ke-1	Ke-2	Ke-3
1	Buah	Buah	Buah	Buah	1	1	1
2	Buah	Buah	Buah	Buah	1	1	1
3	Buah	Buah	Buah	Buah	1	1	1
4	Buah	Buah	Buah	Buah	1	1	1
5	Buah	Buah	Buah	Buah	1	1	1
6	Buah	Buah	Buah	Buah	1	1	1
7	Buah	Buah	Buah	Buah	1	1	1
8	Buah	Buah	Daun	Daun	1	0	0
9	Buah	Buah	Buah	Buah	1	1	1
10	Buah	Buah	Buah	Buah	1	1	1
11	Buah	Buah	Buah	Buah	1	1	1
12	Buah	Buah	Buah	Buah	1	1	1
13	Buah	Buah	Buah	Buah	1	1	1
14	Buah	Buah	Buah	Buah	1	1	1
15	Buah	Buah	Buah	Buah	1	1	1
16	Buah	Buah	Buah	Buah	1	1	1
17	Buah	Buah	Buah	Buah	1	1	1
18	Buah	Buah	Buah	Buah	1	1	1
19	Buah	Buah	Buah	Buah	1	1	1
20	Buah	Buah	Buah	Buah	1	1	1
21	Buah	Buah	Buah	Buah	1	1	1
Jumlah objek yang dikesan betul					21	20	20
Ketepatan (Accuracy)					100	95.24	95.24

Ketepatan terendah = 95.2380952

Identifikasi Sampah Dengan Seni Bina CNN Pra-Latihan GoogleNet

Jenis Sampah : 5. Kaca

No	Label Sebenarnya	Keputusan Ujian			Sampah Dikesan Betul		
		Label Ramalan Ke-1	Label Ramalan Ke-2	Label Ramalan Ke-3	Ke-1	Ke-2	Ke-3
1	Kaca	Kaca	Kaca	Kaca	1	1	1
2	Kaca	Kaca	Kaca	Kaca	1	1	1
3	Kaca	Kaca	Kaca	Kaca	1	1	1
4	Kaca	Kaca	Kaca	Kaca	1	1	1
5	Kaca	Kaca	Kaca	Kaca	1	1	1
6	Kaca	Kaca	Kaca	Kaca	1	1	1
7	Kaca	Kaca	Kaca	Kaca	1	1	1
8	Kaca	Kaca	Kaca	Kaca	1	1	1
9	Kaca	Kaca	Kaca	Kaca	1	1	1
10	Kaca	Kaca	Kaca	Kaca	1	1	1
11	Kaca	Kaca	Kaca	Kaca	1	1	1
12	Kaca	Kaca	Kaca	Kaca	1	1	1
13	Kaca	Kaca	Kaca	Kaca	1	1	1
14	Kaca	Kaca	Kaca	Kaca	1	1	1
15	Kaca	Plastik	Plastik	Kaca	0	0	1
16	Kaca	Kaca	Kaca	Kaca	1	1	1
17	Kaca	Kaca	Kaca	Kaca	1	1	1
18	Kaca	Kaca	Kaca	Kaca	1	1	1
19	Kaca	Kaca	Kaca	Kaca	1	1	1
20	Kaca	Kaca	Kaca	Kaca	1	1	1
21	Kaca	Kaca	Kaca	Kaca	1	1	1
22	Kaca	Kaca	Kaca	Kaca	1	1	1
23	Kaca	Kaca	Kaca	Kaca	1	1	1
24	Kaca	Kaca	Kaca	Kaca	1	1	1
25	Kaca	Kaca	Kaca	Kaca	1	1	1
26	Kaca	Kaca	Kaca	Kaca	1	1	1
27	Kaca	Kaca	Kaca	Kaca	1	1	1
28	Kaca	Kaca	Kaca	Kaca	1	1	1
29	Kaca	Kaca	Kaca	Kaca	1	1	1
30	Kaca	Kaca	Kaca	Kaca	1	1	1
31	Kaca	Kaca	Kaca	Kaca	1	1	1
32	Kaca	Kaca	Kaca	Kaca	1	1	1
33	Kaca	Kaca	Kaca	Kaca	1	1	1
34	Kaca	Kaca	Kaca	Kaca	1	1	1
Jumlah objek yang dikesan betul					33	33	34
Ketepatan (Accuracy)					97.06	97.06	100

Ketepatan terendah = 97.0588235

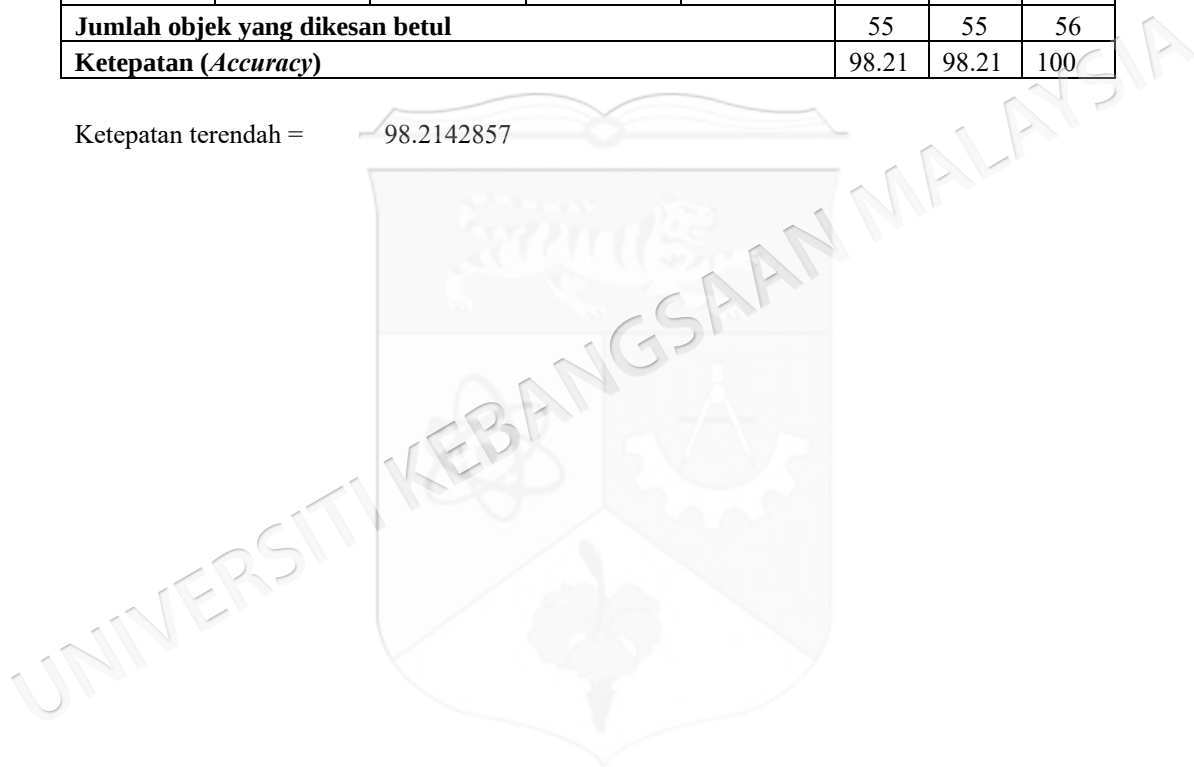
Identifikasi Sampah Dengan Seni Bina CNN Pra-Latihan GoogleNet

Jenis Sampah : 6. Daun

No	Label Sebenar	Keputusan Ujian			Sampah Dikesan Betul		
		Label Ramalan Ke-1	Label Ramalan Ke-2	Label Ramalan Ke-3	Ke-1	Ke-2	Ke-3
1	Daun	Daun	Daun	Daun	1	1	1
2	Daun	Daun	Daun	Daun	1	1	1
3	Daun	Daun	Daun	Daun	1	1	1
4	Daun	Daun	Daun	Daun	1	1	1
5	Daun	Daun	Daun	Daun	1	1	1
6	Daun	Daun	Daun	Daun	1	1	1
7	Daun	Daun	Daun	Daun	1	1	1
8	Daun	Daun	Daun	Daun	1	1	1
9	Daun	Daun	Daun	Daun	1	1	1
10	Daun	Daun	Daun	Daun	1	1	1
11	Daun	Daun	Daun	Daun	1	1	1
12	Daun	Daun	Daun	Daun	1	1	1
13	Daun	Daun	Daun	Daun	1	1	1
14	Daun	Daun	Daun	Daun	1	1	1
15	Daun	Daun	Daun	Daun	1	1	1
16	Daun	Daun	Daun	Daun	1	1	1
17	Daun	Daun	Daun	Daun	1	1	1
18	Daun	Daun	Daun	Daun	1	1	1
19	Daun	Daun	Daun	Daun	1	1	1
20	Daun	Daun	Daun	Daun	1	1	1
21	Daun	Daun	Buah	Daun	1	0	1
22	Daun	Daun	Daun	Daun	1	1	1
23	Daun	Daun	Daun	Daun	1	1	1
24	Daun	Daun	Daun	Daun	1	1	1
25	Daun	Daun	Daun	Daun	1	1	1
26	Daun	Daun	Daun	Daun	1	1	1
27	Daun	Daun	Daun	Daun	1	1	1
28	Daun	Daun	Daun	Daun	1	1	1
29	Daun	Daun	Daun	Daun	1	1	1
30	Daun	Daun	Daun	Daun	1	1	1
31	Daun	Daun	Daun	Daun	1	1	1
32	Daun	Daun	Daun	Daun	1	1	1
33	Daun	Daun	Daun	Daun	1	1	1
34	Daun	Buah	Daun	Daun	0	1	1
35	Daun	Daun	Daun	Daun	1	1	1
36	Daun	Daun	Daun	Daun	1	1	1
37	Daun	Daun	Daun	Daun	1	1	1
38	Daun	Daun	Daun	Daun	1	1	1
39	Daun	Daun	Daun	Daun	1	1	1
40	Daun	Daun	Daun	Daun	1	1	1
41	Daun	Daun	Daun	Daun	1	1	1
42	Daun	Daun	Daun	Daun	1	1	1

43	Daun	Daun	Daun	Daun	1	1	1
44	Daun	Daun	Daun	Daun	1	1	1
45	Daun	Daun	Daun	Daun	1	1	1
46	Daun	Daun	Daun	Daun	1	1	1
47	Daun	Daun	Daun	Daun	1	1	1
48	Daun	Daun	Daun	Daun	1	1	1
49	Daun	Daun	Daun	Daun	1	1	1
50	Daun	Daun	Daun	Daun	1	1	1
51	Daun	Daun	Daun	Daun	1	1	1
52	Daun	Daun	Daun	Daun	1	1	1
53	Daun	Daun	Daun	Daun	1	1	1
54	Daun	Daun	Daun	Daun	1	1	1
55	Daun	Daun	Daun	Daun	1	1	1
56	Daun	Daun	Daun	Daun	1	1	1
Jumlah objek yang dikesan betul					55	55	56
Ketepatan (<i>Accuracy</i>)					98.21	98.21	100

Ketepatan terendah = 98.2142857



Identifikasi Sampah Dengan Seni Bina CNN Pra-Latihan GoogleNet
Jenis Sampah : 7.Logam

No	Label Sebenar	Keputusan Ujian			Sampah Dikesan Betul		
		Label Ramalan Ke-1	Label Ramalan Ke-2	Label Ramalan Ke-3	Ke-1	Ke-2	Ke-3
1	Logam	Logam	Logam	Logam	1	1	1
2	Logam	Logam	Logam	Logam	1	1	1
3	Logam	Logam	Logam	Logam	1	1	1
4	Logam	Logam	Logam	Logam	1	1	1
5	Logam	Logam	Logam	Logam	1	1	1
6	Logam	Logam	Logam	Logam	1	1	1
7	Logam	Logam	Logam	Logam	1	1	1
8	Logam	Logam	Logam	Logam	1	1	1
9	Logam	Logam	Logam	Logam	1	1	1
10	Logam	Logam	Logam	Logam	1	1	1
11	Logam	Logam	Logam	Logam	1	1	1
12	Logam	Logam	Logam	Logam	1	1	1
13	Logam	Logam	Logam	Logam	1	1	1
14	Logam	Logam	Logam	Logam	1	1	1
15	Logam	Logam	Logam	Logam	1	1	1
16	Logam	Logam	Logam	Logam	1	1	1
17	Logam	Logam	Logam	Logam	1	1	1
18	Logam	Logam	Logam	Logam	1	1	1
19	Logam	Logam	Logam	Logam	1	1	1
20	Logam	Pembungkus Makanan	Logam	Pembungkus Makanan	0	1	0
21	Logam	Logam	Logam	Logam	1	1	1
Jumlah objek yang dikesan betul					20	21	20
Ketepatan (Accuracy)					95.24	100	95.24

Ketepatan terendah = 95.2380952

Identifikasi Sampah Dengan Seni Bina CNN Pra-Latihan GoogleNet

Jenis Sampah : 8.Kertas

No	Label Sebenar	Keputusan Ujian			Sampah Dikesan Betul		
		Label Ramalan Ke-1	Label Ramalan Ke-2	Label Ramalan Ke-3	Ke-1	Ke-2	Ke-3
1	Kertas	Kertas	Kertas	Kertas	1	1	1
2	Kertas	Kertas	Kertas	Kertas	1	1	1
3	Kertas	Kertas	Kertas	Kertas	1	1	1
4	Kertas	Kertas	Kertas	Kertas	1	1	1
5	Kertas	Kertas	Kertas	Kertas	1	1	1
6	Kertas	Kertas	Kertas	Kertas	1	1	1
7	Kertas	Kertas	Kertas	Kertas	1	1	1
8	Kertas	Kertas	Kertas	Kertas	1	1	1
9	Kertas	Kertas	Kertas	Kertas	1	1	1
10	Kertas	Kertas	Kadbod	Kertas	1	0	1
11	Kertas	Kertas	Kertas	Kertas	1	1	1
12	Kertas	Kertas	Kertas	Kertas	1	1	1
13	Kertas	Kertas	Kertas	Kertas	1	1	1
14	Kertas	Kertas	Kertas	Kertas	1	1	1
15	Kertas	Kertas	Kertas	Kertas	1	1	1
16	Kertas	Kertas	Kertas	Kertas	1	1	1
17	Kertas	Kertas	Kertas	Kertas	1	1	1
18	Kertas	Kertas	Kertas	Kertas	1	1	1
19	Kertas	Kertas	Kertas	Kertas	1	1	1
20	Kertas	Kertas	Kertas	Kertas	1	1	1
21	Kertas	Kertas	Kertas	Kertas	1	1	1
22	Kertas	Kertas	Kertas	Kertas	1	1	1
23	Kertas	Kertas	Kertas	Kertas	1	1	1
24	Kertas	Kertas	Kertas	Kertas	1	1	1
25	Kertas	Kertas	Kertas	Kertas	1	1	1
26	Kertas	Kertas	Kertas	Kertas	1	1	1
27	Kertas	Kertas	Kertas	Kertas	1	1	1
28	Kertas	Kertas	Kertas	Kertas	1	1	1
29	Kertas	Kertas	Kertas	Kertas	1	1	1
30	Kertas	Kertas	Kertas	Kertas	1	1	1
31	Kertas	Kertas	Kertas	Kertas	1	1	1
32	Kertas	Kertas	Kertas	Kertas	1	1	1
33	Kertas	Kertas	Kertas	Kertas	1	1	1
34	Kertas	Kertas	Kertas	Kertas	1	1	1
35	Kertas	Kertas	Kertas	Kertas	1	1	1
36	Kertas	Kertas	Kertas	Kertas	1	1	1
37	Kertas	Kertas	Kertas	Kertas	1	1	1
38	Kertas	Kertas	Kertas	Kertas	1	1	1
39	Kertas	Kertas	Kertas	Kertas	1	1	1
40	Kertas	Kertas	Kertas	Kertas	1	1	1
41	Kertas	Kertas	Kertas	Kertas	1	1	1
42	Kertas	Kertas	Kertas	Kertas	1	1	1

43	Kertas	Kertas	Kertas	Kertas	1	1	1
44	Kertas	Kertas	Kertas	Kertas	1	1	1
45	Kertas	Kertas	Kertas	Kertas	1	1	1
46	Kertas	Kertas	Kertas	Kertas	1	1	1
47	Kertas	Kertas	Kertas	Kertas	1	1	1
48	Kertas	Kertas	Kertas	Kertas	1	1	1
49	Kertas	Kertas	Kertas	Kertas	1	1	1
50	Kertas	Kertas	Kertas	Kertas	1	1	1
51	Kertas	Kertas	Kertas	Kertas	1	1	1
52	Kertas	Kertas	Kertas	Kertas	1	1	1
53	Kertas	Kertas	Kertas	Kertas	1	1	1
54	Kertas	Kertas	Kertas	Kertas	1	1	1
55	Kertas	Kertas	Kertas	Kertas	1	1	1
56	Kertas	Kertas	Kertas	Kertas	1	1	1
57	Kertas	Kertas	Kertas	Kertas	1	1	1
58	Kertas	Kertas	Kertas	Kertas	1	1	1
59	Kertas	Kertas	Kertas	Kertas	1	1	1
60	Kertas	Kertas	Kertas	Kertas	1	1	1
61	Kertas	Kertas	Kertas	Kertas	1	1	1
Jumlah objek yang dikesan betul					61	60	61
Ketepatan (Accuracy)					100	98.36	100

Ketepatan terendah = 98.3606557

Identifikasi Sampah Dengan Seni Bina CNN Pra-Latihan GoogleNet

Jenis Sampah : 9.Plastik

No	Label Sebenarnya	Keputusan Ujian			Sampah Diklasifikasi Betul		
		Label Ramalan Ke-1	Label Ramalan Ke-2	Label Ramalan Ke-3	Ke-1	Ke-2	Ke-3
1	Plastik	Plastik	Plastik	Plastik	1	1	1
2	Plastik	Plastik	Plastik	Plastik	1	1	1
3	Plastik	Plastik	Plastik	Plastik	1	1	1
4	Plastik	Plastik	Plastik	Plastik	1	1	1
5	Plastik	Plastik	Plastik	Plastik	1	1	1
6	Plastik	Plastik	Plastik	Plastik	1	1	1
7	Plastik	Plastik	Plastik	Plastik	1	1	1
8	Plastik	Plastik	Plastik	Plastik	1	1	1
9	Plastik	Plastik	Plastik	Plastik	1	1	1
10	Plastik	Plastik	Plastik	Plastik	1	1	1
11	Plastik	Plastik	Plastik	Plastik	1	1	1
12	Plastik	Plastik	Plastik	Plastik	1	1	1
13	Plastik	Plastik	Plastik	Plastik	1	1	1
14	Plastik	Plastik	Plastik	Plastik	1	1	1
15	Plastik	Plastik	Plastik	Plastik	1	1	1
16	Plastik	Plastik	Plastik	Plastik	1	1	1
17	Plastik	Plastik	Plastik	Plastik	1	1	1
18	Plastik	Plastik	Plastik	Plastik	1	1	1
19	Plastik	Plastik	Plastik	Plastik	1	1	1
20	Plastik	Plastik	Plastik	Plastik	1	1	1
21	Plastik	Plastik	Plastik	Plastik	1	1	1
22	Plastik	Plastik	Plastik	Plastik	1	1	1
23	Plastik	Plastik	Plastik	Plastik	1	1	1
24	Plastik	Plastik	Plastik	Plastik	1	1	1
25	Plastik	Plastik	Plastik	Plastik	1	1	1
26	Plastik	Plastik	Plastik	Plastik	1	1	1
27	Plastik	Kaca	Kaca	Plastik	0	0	1
28	Plastik	Plastik	Plastik	Plastik	1	1	1
29	Plastik	Plastik	Plastik	Plastik	1	1	1
30	Plastik	Plastik	Plastik	Plastik	1	1	1
31	Plastik	Plastik	Plastik	Plastik	1	1	1
32	Plastik	Plastik	Plastik	Plastik	1	1	1
33	Plastik	Plastik	Plastik	Plastik	1	1	1
34	Plastik	Plastik	Plastik	Plastik	1	1	1
35	Plastik	Plastik	Plastik	Plastik	1	1	1
36	Plastik	Plastik	Plastik	Plastik	1	1	1
37	Plastik	Plastik	Plastik	Plastik	1	1	1
38	Plastik	Plastik	Plastik	Plastik	1	1	1
39	Plastik	Plastik	Plastik	Plastik	1	1	1
40	Plastik	Plastik	Plastik	Plastik	1	1	1
41	Plastik	Plastik	Plastik	Plastik	1	1	1
42	Plastik	Plastik	Plastik	Plastik	1	1	1

43	Plastik	Plastik	Plastik	Plastik	1	1	1
44	Plastik	Plastik	Plastik	Plastik	1	1	1
45	Plastik	Plastik	Plastik	Plastik	1	1	1
46	Plastik	Plastik	Plastik	Plastik	1	1	1
47	Plastik	Plastik	Plastik	Plastik	1	1	1
48	Plastik	Plastik	Plastik	Plastik	1	1	1
49	Plastik	Plastik	Plastik	Plastik	1	1	1
50	Plastik	Plastik	Plastik	Plastik	1	1	1
51	Plastik	Plastik	Plastik	Plastik	1	1	1
52	Plastik	Plastik	Plastik	Plastik	1	1	1
Jumlah objek yang dikesan betul					51	51	52
Ketepatan (<i>Accuracy</i>)					98.08	98.08	100

Ketepatan terendah = 98.0769231



Identifikasi Sampah Dengan Seni Bina CNN Pra-Latihan GoogleNet

Jenis Sampah :

10. Getah

No	Label Sebenar	Keputusan Ujian			Sampah Dikesan Betul		
		Label Ramalan Ke-1	Label Ramalan Ke-2	Label Ramalan Ke-3	Ke-1	Ke-2	Ke-3
1	Getah	Getah	Getah	Getah	1	1	1
2	Getah	Getah	Getah	Getah	1	1	1
3	Getah	Getah	Getah	Getah	1	1	1
4	Getah	Getah	Getah	Getah	1	1	1
5	Getah	Getah	Getah	Getah	1	1	1
6	Getah	Getah	Getah	Getah	1	1	1
7	Getah	Getah	Getah	Getah	1	1	1
8	Getah	Getah	Getah	Getah	1	1	1
9	Getah	Getah	Getah	Getah	1	1	1
10	Getah	Getah	Getah	Getah	1	1	1
11	Getah	Getah	Getah	Getah	1	1	1
12	Getah	Getah	Pembungkus Makanan	Pembungkus Makanan	1	0	0
13	Getah	Getah	Getah	Getah	1	1	1
14	Getah	Getah	Getah	Getah	1	1	1
Jumlah objek yang dikesan betul					14	13	13
Ketepatan (Accuracy)					100	92.86	92.86

Ketepatan terendah = 92.8571429

Identifikasi Sampah Dengan Seni Bina CNN Pra-Latihan GoogleNet
Jenis Sampah : 11.Kayu

No	Label Sebenarnya	Keputusan Ujian			Sampah Dikesan Betul		
		Label Ramalan Ke-1	Label Ramalan Ke-2	Label Ramalan Ke-3	Ke-1	Ke-2	Ke-3
1	Kayu	Kayu	Kayu	Kayu	1	1	1
2	Kayu	Kayu	Kayu	Kayu	1	1	1
3	Kayu	Kayu	Kayu	Kayu	1	1	1
4	Kayu	Kayu	Kayu	Kayu	1	1	1
5	Kayu	Kayu	Kayu	Kayu	1	1	1
6	Kayu	Kayu	Kayu	Kayu	1	1	1
7	Kayu	Kayu	Kayu	Kayu	1	1	1
8	Kayu	Kayu	Kayu	Kayu	1	1	1
9	Kayu	Kayu	Kayu	Kayu	1	1	1
10	Kayu	Kayu	Kayu	Kayu	1	1	1
11	Kayu	Kayu	Kayu	Kayu	1	1	1
12	Kayu	Kayu	Kayu	Kayu	1	1	1
13	Kayu	Kayu	Kayu	Kayu	1	1	1
14	Kayu	Kayu	Kayu	Kayu	1	1	1
15	Kayu	Kayu	Kayu	Kayu	1	1	1
16	Kayu	Kayu	Kayu	Kayu	1	1	1
17	Kayu	Kayu	Kayu	Kayu	1	1	1
18	Kayu	Kayu	Kayu	Kayu	1	1	1
19	Kayu	Kayu	Kayu	Kayu	1	1	1
20	Kayu	Kayu	Kayu	Kayu	1	1	1
21	Kayu	Kayu	Kayu	Kayu	1	1	1
22	Kayu	Kayu	Kayu	Kayu	1	1	1
23	Kayu	Daun	Daun	Kayu	0	0	1
24	Kayu	Kayu	Kayu	Kayu	1	1	1
25	Kayu	Kayu	Kayu	Kayu	1	1	1
26	Kayu	Kayu	Kayu	Kayu	1	1	1
27	Kayu	Kayu	Kayu	Kayu	1	1	1
Jumlah objek yang dikesan betul					26	26	27
Ketepatan (Accuracy)					96.3	96.3	100

Ketepatan terendah = 96.2962963

LAMPIRAN E

KEPUTUSAN UJIAN SENI BINA RESNET18

Identifikasi Sampah Dengan Seni Bina CNN Pra-Latihan ResNet18

Jenis Sampah :

1.Kadbod

No	Label Sebenar	Keputusan Ujian			Sampah Dikesan Betul		
		Label Ramalan Ke-1	Label Ramalan Ke-2	Label Ramalan Ke-3	Ke-1	Ke-2	Ke-3
1	Kadbod	Kadbod	Kadbod	Kadbod	1	1	1
2	Kadbod	Kadbod	Kadbod	Kadbod	1	1	1
3	Kadbod	Kadbod	Kadbod	Kadbod	1	1	1
4	Kadbod	Kadbod	Kadbod	Kadbod	1	1	1
5	Kadbod	Kadbod	Kadbod	Kadbod	1	1	1
6	Kadbod	Kadbod	Kadbod	Kadbod	1	1	1
7	Kadbod	Kadbod	Kertas	Kertas	1	0	0
8	Kadbod	Kadbod	Kadbod	Kadbod	1	1	1
9	Kadbod	Kadbod	Kadbod	Kadbod	1	1	1
10	Kadbod	Kadbod	Kadbod	Kadbod	1	1	1
11	Kadbod	Kadbod	Kadbod	Kadbod	1	1	1
12	Kadbod	Kadbod	Kadbod	Kadbod	1	1	1
13	Kadbod	Kadbod	Kadbod	Kadbod	1	1	1
14	Kadbod	Kadbod	Kadbod	Kadbod	1	1	1
15	Kadbod	Kadbod	Kadbod	Kadbod	1	1	1
16	Kadbod	Kadbod	Kadbod	Kadbod	1	1	1
17	Kadbod	Kadbod	Kadbod	Kadbod	1	1	1
18	Kadbod	Kadbod	Kadbod	Kadbod	1	1	1
19	Kadbod	Kadbod	Kadbod	Kadbod	1	1	1
20	Kadbod	Kadbod	Kadbod	Kadbod	1	1	1
21	Kadbod	Kadbod	Kadbod	Kadbod	1	1	1
22	Kadbod	Kadbod	Kadbod	Kadbod	1	1	1
23	Kadbod	Kadbod	Kadbod	Kadbod	1	1	1
24	Kadbod	Kadbod	Kadbod	Kadbod	1	1	1
25	Kadbod	Kadbod	Kadbod	Kadbod	1	1	1
26	Kadbod	Kadbod	Kadbod	Kadbod	1	1	1
27	Kadbod	Kadbod	Kadbod	Kadbod	1	1	1
28	Kadbod	Kadbod	Kadbod	Kadbod	1	1	1
29	Kadbod	Kadbod	Kadbod	Kadbod	1	1	1
30	Kadbod	Kadbod	Kadbod	Kertas	1	1	0
31	Kadbod	Kadbod	Kadbod	Kadbod	1	1	1
32	Kadbod	Kadbod	Kadbod	Kadbod	1	1	1
33	Kadbod	Kadbod	Kadbod	Kadbod	1	1	1
34	Kadbod	Kadbod	Kadbod	Kadbod	1	1	1
35	Kadbod	Kadbod	Kadbod	Kadbod	1	1	1
36	Kadbod	Kadbod	Kadbod	Kadbod	1	1	1
37	Kadbod	Kertas	Kadbod	Kadbod	0	1	1

38	Kadbod	Kadbod	Kadbod	Kadbod	1	1	1
39	Kadbod	Kadbod	Kadbod	Kadbod	1	1	1
40	Kadbod	Kadbod	Kadbod	Kadbod	1	1	1
41	Kadbod	Kadbod	Kadbod	Kadbod	1	1	1
42	Kadbod	Kadbod	Kadbod	Kadbod	1	1	1
43	Kadbod	Kadbod	Kadbod	Kadbod	1	1	1
44	Kadbod	Kadbod	Kertas	Kadbod	1	0	1
45	Kadbod	Kadbod	Kadbod	Kadbod	1	1	1
Jumlah objek yang dikesan betul					44	43	43
Ketepatan (<i>Accuracy</i>)					97.78	95.56	95.56

Ketepatan terendah = 95.555556



Identifikasi Sampah Dengan Seni Bina CNN Pra-Latihan ResNet18

Jenis Sampah : 2.Fabrik

No	Label Sebenar	Keputusan Ujian			Sampah Dikesan Betul		
		Label Ramalan Ke-1	Label Ramalan Ke-2	Label Ramalan Ke-3	Ke-1	Ke-2	Ke-3
1	Fabrik	Fabrik	Fabrik	Fabrik	1	1	1
2	Fabrik	Fabrik	Fabrik	Fabrik	1	1	1
3	Fabrik	Fabrik	Fabrik	Fabrik	1	1	1
4	Fabrik	Fabrik	Fabrik	Fabrik	1	1	1
5	Fabrik	Fabrik	Fabrik	Fabrik	1	1	1
6	Fabrik	Fabrik	Fabrik	Fabrik	1	1	1
7	Fabrik	Fabrik	Getah	Kayu	1	0	0
8	Fabrik	Fabrik	Fabrik	Fabrik	1	1	1
9	Fabrik	Kayu	Kayu	Fabrik	0	0	1
10	Fabrik	Fabrik	Fabrik	Fabrik	1	1	1
11	Fabrik	Fabrik	Fabrik	Fabrik	1	1	1
12	Fabrik	Fabrik	Fabrik	Fabrik	1	1	1
13	Fabrik	Fabrik	Fabrik	Fabrik	1	1	1
14	Fabrik	Fabrik	Fabrik	Fabrik	1	1	1
15	Fabrik	Fabrik	Fabrik	Fabrik	1	1	1
16	Fabrik	Fabrik	Fabrik	Fabrik	1	1	1
17	Fabrik	Fabrik	Fabrik	Fabrik	1	1	1
Jumlah objek yang dikesan betul					16	15	16
Ketepatan (Accuracy)					94.12	88.24	94.12

Ketepatan terendah = 88.2352941

Identifikasi Sampah Dengan Seni Bina CNN Pra-Latihan ResNet18
Jenis Sampah : 3. Pembungkus Makanan

No	Label Sebenarnya	Keputusan Ujian			Sampah Dikesan Betul		
		Label Ramalan Ke-1	Label Ramalan Ke-2	Label Ramalan Ke-3	Ke-1	Ke-2	Ke-3
1	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	1	1	1
2	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	1	1	1
3	Pembungkus makanan	Kadbod	Pembungkus makanan	Pembungkus makanan	0	1	1
4	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	1	1	1
5	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	1	1	1
6	Pembungkus makanan	Kadbod	Pembungkus makanan	Kadbod	0	1	0
7	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	1	1	1
8	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	1	1	1
9	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	1	1	1
10	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	1	1	1
11	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	1	1	1
12	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	1	1	1
13	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	1	1	1
14	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	1	1	1
15	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	1	1	1
16	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	1	1	1
17	Pembungkus makanan	Pembungkus makanan	Kertas	Pembungkus makanan	1	0	1
18	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	1	1	1
19	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	1	1	1
20	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	1	1	1
21	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	1	1	1

[illegible]

47	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	1	1	1
48	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	1	1	1
49	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	1	1	1
50	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	1	1	1
51	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	1	1	1
52	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	1	1	1
53	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	1	1	1
54	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	1	1	1
55	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	1	1	1
56	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	1	1	1
57	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	1	1	1
58	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	1	1	1
59	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	1	1	1
60	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	1	1	1
61	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	1	1	1
62	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	1	1	1
63	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	Pembungkus makanan	1	1	1
Jumlah objek yang dikesan betul					61	62	62
Ketepatan (Accuracy)					96.83	98.41	98.41

Ketepatan terendah = 96.8253968

Identifikasi Sampah Dengan Seni Bina CNN Pra-Latihan ResNet18

Jenis Sampah : 4. Buah

No	Label Sebenarnya	Keputusan Ujian			Sampah Dikesan Betul		
		Label Ramalan Ke-1	Label Ramalan Ke-2	Label Ramalan Ke-3	Ke-1	Ke-2	Ke-3
1	Buah	Buah	Buah	Buah	1	1	1
2	Buah	Buah	Buah	Buah	1	1	1
3	Buah	Buah	Buah	Buah	1	1	1
4	Buah	Buah	Buah	Buah	1	1	1
5	Buah	Buah	Buah	Buah	1	1	1
6	Buah	Buah	Buah	Buah	1	1	1
7	Buah	Buah	Buah	Buah	1	1	1
8	Buah	Buah	Buah	Buah	1	1	1
9	Buah	Buah	Buah	Daun	1	1	0
10	Buah	Buah	Buah	Buah	1	1	1
11	Buah	Buah	Buah	Buah	1	1	1
12	Buah	Buah	Buah	Buah	1	1	1
13	Buah	Buah	Buah	Buah	1	1	1
14	Buah	Buah	Buah	Buah	1	1	1
15	Buah	Buah	Buah	Buah	1	1	1
16	Buah	Buah	Buah	Buah	1	1	1
17	Buah	Buah	Buah	Buah	1	1	1
18	Buah	Buah	Buah	Buah	1	1	1
19	Buah	Buah	Buah	Buah	1	1	1
20	Buah	Buah	Buah	Buah	1	1	1
21	Buah	Buah	Buah	Buah	1	1	1
Jumlah objek yang dikesan betul					21	21	20
Ketepatan (Accuracy)					100	100	95.24

Ketepatan terendah = 95.2380952

Identifikasi Sampah Dengan Seni Bina CNN Pra-Latihan ResNet18

Jenis Sampah : 5. Kaca

No	Label Sebenarnya	Keputusan Ujian			Sampah Dikesan Betul		
		Label Ramalan Ke-1	Label Ramalan Ke-2	Label Ramalan Ke-3	Ke-1	Ke-2	Ke-3
1	Kaca	Kaca	Kaca	Kaca	1	1	1
2	Kaca	Kaca	Kaca	Kaca	1	1	1
3	Kaca	Kaca	Kaca	Kaca	1	1	1
4	Kaca	Kaca	Kaca	Kaca	1	1	1
5	Kaca	Kaca	Kaca	Kaca	1	1	1
6	Kaca	Kaca	Kaca	Kaca	1	1	1
7	Kaca	Kaca	Kaca	Kaca	1	1	1
8	Kaca	Kaca	Kaca	Kaca	1	1	1
9	Kaca	Kaca	Kaca	Kaca	1	1	1
10	Kaca	Kaca	Kaca	Kaca	1	1	1
11	Kaca	Kaca	Kaca	Kaca	1	1	1
12	Kaca	Kaca	Kaca	Kaca	1	1	1
13	Kaca	Plastik	Plastik	Kaca	0	0	1
14	Kaca	Kaca	Kaca	Kaca	1	1	1
15	Kaca	Kaca	Kaca	Kaca	1	1	1
16	Kaca	Kaca	Kaca	Kaca	1	1	1
17	Kaca	Kaca	Kaca	Kaca	1	1	1
18	Kaca	Kaca	Kaca	Kaca	1	1	1
19	Kaca	Kaca	Kaca	Kaca	1	1	1
20	Kaca	Kaca	Kaca	Kaca	1	1	1
21	Kaca	Kaca	Kaca	Kaca	1	1	1
22	Kaca	Kaca	Kaca	Kaca	1	1	1
23	Kaca	Kaca	Kaca	Kaca	1	1	1
24	Kaca	Kaca	Kaca	Kaca	1	1	1
25	Kaca	Kaca	Kaca	Kaca	1	1	1
26	Kaca	Kaca	Kaca	Kaca	1	1	1
27	Kaca	Kaca	Kaca	Kaca	1	1	1
28	Kaca	Kaca	Kaca	Kaca	1	1	1
29	Kaca	Kaca	Kaca	Kaca	1	1	1
30	Kaca	Kaca	Kaca	Kaca	1	1	1
31	Kaca	Kaca	Kaca	Kaca	1	1	1
32	Kaca	Kaca	Kaca	Kaca	1	1	1
33	Kaca	Kaca	Kaca	Kaca	1	1	1
34	Kaca	Kaca	Kaca	Kaca	1	1	1
Jumlah objek yang dikesan betul					33	33	34
Ketepatan (<i>Accuracy</i>)					97.06	97.06	100

Ketepatan terendah = 97.0588235

Identifikasi Sampah Dengan Seni Bina CNN Pra-Latihan ResNet18

Jenis Sampah : 6. Daun

No	Label Sebenarnya	Keputusan Ujian			Sampah Dikesan Betul		
		Label Ramalan Ke-1	Label Ramalan Ke-2	Label Ramalan Ke-3	Ke-1	Ke-2	Ke-3
1	Daun	Daun	Daun	Daun	1	1	1
2	Daun	Daun	Daun	Daun	1	1	1
3	Daun	Daun	Daun	Daun	1	1	1
4	Daun	Daun	Daun	Daun	1	1	1
5	Daun	Daun	Daun	Daun	1	1	1
6	Daun	Daun	Daun	Daun	1	1	1
7	Daun	Daun	Daun	Daun	1	1	1
8	Daun	Daun	Daun	Daun	1	1	1
9	Daun	Daun	Daun	Daun	1	1	1
10	Daun	Daun	Daun	Daun	1	1	1
11	Daun	Daun	Daun	Daun	1	1	1
12	Daun	Daun	Daun	Daun	1	1	1
13	Daun	Daun	Daun	Daun	1	1	1
14	Daun	Daun	Daun	Daun	1	1	1
15	Daun	Daun	Daun	Daun	1	1	1
16	Daun	Daun	Daun	Daun	1	1	1
17	Daun	Daun	Daun	Daun	1	1	1
18	Daun	Buah	Buah	Buah	0	0	0
19	Daun	Daun	Daun	Daun	1	1	1
20	Daun	Daun	Daun	Daun	1	1	1
21	Daun	Buah	Daun	Daun	0	1	1
22	Daun	Daun	Daun	Daun	1	1	1
23	Daun	Daun	Daun	Daun	1	1	1
24	Daun	Daun	Daun	Daun	1	1	1
25	Daun	Daun	Daun	Daun	1	1	1
26	Daun	Daun	Daun	Daun	1	1	1
27	Daun	Daun	Daun	Daun	1	1	1
28	Daun	Daun	Daun	Daun	1	1	1
29	Daun	Daun	Daun	Daun	1	1	1
30	Daun	Daun	Daun	Daun	1	1	1
31	Daun	Daun	Daun	Daun	1	1	1
32	Daun	Daun	Daun	Daun	1	1	1
33	Daun	Daun	Daun	Daun	1	1	1
34	Daun	Daun	Daun	Daun	1	1	1
35	Daun	Daun	Daun	Daun	1	1	1
36	Daun	Daun	Daun	Daun	1	1	1
37	Daun	Daun	Daun	Daun	1	1	1
38	Daun	Daun	Daun	Daun	1	1	1
39	Daun	Daun	Daun	Daun	1	1	1
40	Daun	Daun	Daun	Daun	1	1	1
41	Daun	Daun	Daun	Daun	1	1	1
42	Daun	Daun	Daun	Daun	1	1	1

43	Daun	Daun	Daun	Daun	1	1	1
44	Daun	Daun	Daun	Daun	1	1	1
45	Daun	Daun	Daun	Daun	1	1	1
46	Daun	Daun	Daun	Daun	1	1	1
47	Daun	Daun	Daun	Daun	1	1	1
48	Daun	Daun	Daun	Daun	1	1	1
49	Daun	Daun	Daun	Daun	1	1	1
50	Daun	Daun	Daun	Daun	1	1	1
51	Daun	Daun	Pembungkus Makanan	Daun	1	0	1
52	Daun	Daun	Daun	Daun	1	1	1
53	Daun	Daun	Daun	Daun	1	1	1
54	Daun	Daun	Daun	Daun	1	1	1
55	Daun	Daun	Daun	Daun	1	1	1
56	Daun	Daun	Daun	Daun	1	1	1
Jumlah objek yang dikesan betul					54	54	55
Ketepatan (Accuracy)					96.43	96.43	98.21

Ketepatan terendah = 96.4285714

Identifikasi Sampah Dengan Seni Bina CNN Pra-Latihan ResNet18
Jenis Sampah : 7.Logam

No	Label Sebenar	Keputusan Ujian			Sampah Dikesan Betul		
		Label Ramalan Ke-1	Label Ramalan Ke-2	Label Ramalan Ke-3	Ke-1	Ke-2	Ke-3
1	Logam	Logam	Logam	Logam	1	1	1
2	Logam	Logam	Logam	Logam	1	1	1
3	Logam	Logam	Logam	Logam	1	1	1
4	Logam	Logam	Logam	Logam	1	1	1
5	Logam	Logam	Logam	Logam	1	1	1
6	Logam	Logam	Logam	Logam	1	1	1
7	Logam	Logam	Logam	Logam	1	1	1
8	Logam	Logam	Logam	Logam	1	1	1
9	Logam	Logam	Pembungkus Makanan	Pembungkus Makanan	1	0	0
10	Logam	Logam	Logam	Logam	1	1	1
11	Logam	Logam	Logam	Logam	1	1	1
12	Logam	Logam	Logam	Logam	1	1	1
13	Logam	Logam	Logam	Logam	1	1	1
14	Logam	Logam	Logam	Logam	1	1	1
15	Logam	Logam	Logam	Logam	1	1	1
16	Logam	Logam	Logam	Logam	1	1	1
17	Logam	Logam	Logam	Logam	1	1	1
18	Logam	Logam	Logam	Logam	1	1	1
19	Logam	Logam	Logam	Logam	1	1	1
20	Logam	Logam	Logam	Logam	1	1	1
21	Logam	Logam	Logam	Logam	1	1	1
Jumlah objek yang dikesan betul					21	20	20
Ketepatan (Accuracy)					100	95.24	95.24

Ketepatan terendah = 95.2380952

Identifikasi Sampah Dengan Seni Bina CNN Pra-Latihan ResNet18
Jenis Sampah : 8.Kertas

No	Label Sebenar	Keputusan Ujian			Sampah Dikesan Betul		
		Label Ramalan Ke-1	Label Ramalan Ke-2	Label Ramalan Ke-3	Ke-1	Ke-2	Ke-3
1	Kertas	Kertas	Kertas	Kertas	1	1	1
2	Kertas	Kertas	Kertas	Kertas	1	1	1
3	Kertas	Kertas	Kertas	Kertas	1	1	1
4	Kertas	Kertas	Kertas	Kertas	1	1	1
5	Kertas	Kertas	Kertas	Kertas	1	1	1
6	Kertas	Kertas	Kertas	Kertas	1	1	1
7	Kertas	Kertas	Kertas	Kertas	1	1	1
8	Kertas	Kertas	Kertas	Kertas	1	1	1
9	Kertas	Kertas	Kertas	Kertas	1	1	1
10	Kertas	Kertas	Kertas	Kertas	1	1	1
11	Kertas	Kertas	Kertas	Kertas	1	1	1
12	Kertas	Kertas	Kertas	Kertas	1	1	1
13	Kertas	Kertas	Kertas	Kertas	1	1	1
14	Kertas	Kertas	Kertas	Kertas	1	1	1
15	Kertas	Kertas	Kertas	Kertas	1	1	1
16	Kertas	Kertas	Kertas	Kertas	1	1	1
17	Kertas	Kertas	Kertas	Kertas	1	1	1
18	Kertas	Kertas	Kertas	Kertas	1	1	1
19	Kertas	Kertas	Kertas	Kertas	1	1	1
20	Kertas	Kertas	Kertas	Kertas	1	1	1
21	Kertas	Kertas	Kertas	Kertas	1	1	1
22	Kertas	Kertas	Kertas	Kertas	1	1	1
23	Kertas	Kertas	Kertas	Kertas	1	1	1
24	Kertas	Kadbod	Kadbod	Kadbod	0	0	0
25	Kertas	Kadbod	Kadbod	Kertas	0	0	1
26	Kertas	Kertas	Kertas	Kertas	1	1	1
27	Kertas	Kertas	Kertas	Kertas	1	1	1
28	Kertas	Kertas	Kertas	Kertas	1	1	1
29	Kertas	Kertas	Kertas	Kertas	1	1	1
30	Kertas	Kertas	Kertas	Kertas	1	1	1
31	Kertas	Kertas	Kertas	Kertas	1	1	1
32	Kertas	Kertas	Kertas	Kertas	1	1	1
33	Kertas	Kertas	Kertas	Kertas	1	1	1
34	Kertas	Kertas	Kertas	Kertas	1	1	1
35	Kertas	Kertas	Kertas	Kertas	1	1	1
36	Kertas	Kertas	Kertas	Kertas	1	1	1
37	Kertas	Kertas	Kertas	Kertas	1	1	1
38	Kertas	Kertas	Kertas	Kertas	1	1	1
39	Kertas	Kertas	Kertas	Kertas	1	1	1
40	Kertas	Kertas	Kertas	Kertas	1	1	1
41	Kertas	Kertas	Kertas	Kertas	1	1	1
42	Kertas	Kertas	Kertas	Kertas	1	1	1

43	Kertas	Kertas	Kertas	Kertas	1	1	1
44	Kertas	Kertas	Kertas	Kertas	1	1	1
45	Kertas	Kertas	Kertas	Kertas	1	1	1
46	Kertas	Kertas	Kertas	Kertas	1	1	1
47	Kertas	Kertas	Kertas	Kertas	1	1	1
48	Kertas	Kertas	Kertas	Kertas	1	1	1
49	Kertas	Kertas	Kertas	Kertas	1	1	1
50	Kertas	Kertas	Kertas	Kertas	1	1	1
51	Kertas	Kertas	Kertas	Kertas	1	1	1
52	Kertas	Kertas	Kertas	Kertas	1	1	1
53	Kertas	Kertas	Kertas	Kertas	1	1	1
54	Kertas	Kertas	Kertas	Kertas	1	1	1
55	Kertas	Kertas	Kertas	Kertas	1	1	1
56	Kertas	Kertas	Kertas	Kertas	1	1	1
57	Kertas	Kertas	Kertas	Kertas	1	1	1
58	Kertas	Kertas	Kertas	Kertas	1	1	1
59	Kertas	Kertas	Kertas	Kertas	1	1	1
60	Kertas	Kertas	Kertas	Kertas	1	1	1
61	Kertas	Kertas	Kertas	Kertas	1	1	1
Jumlah objek yang dikesan betul					59	59	60
Ketepatan (Accuracy)					96.72	96.72	98.36

Ketepatan terendah = 96.7213115

Identifikasi Sampah Dengan Seni Bina CNN Pra-Latihan ResNet18

Jenis Sampah : 9.Plastik

No	Label Sebenar	Keputusan Ujian			Sampah Dikesan Betul		
		Label Ramalan Ke-1	Label Ramalan Ke-2	Label Ramalan Ke-3	Ke-1	Ke-2	Ke-3
1	Plastik	Plastik	Plastik	Plastik	1	1	1
2	Plastik	Plastik	Plastik	Plastik	1	1	1
3	Plastik	Plastik	Plastik	Plastik	1	1	1
4	Plastik	Plastik	Plastik	Plastik	1	1	1
5	Plastik	Plastik	Plastik	Plastik	1	1	1
6	Plastik	Plastik	Plastik	Plastik	1	1	1
7	Plastik	Plastik	Plastik	Plastik	1	1	1
8	Plastik	Plastik	Plastik	Plastik	1	1	1
9	Plastik	Plastik	Plastik	Plastik	1	1	1
10	Plastik	Plastik	Plastik	Plastik	1	1	1
11	Plastik	Plastik	Plastik	Plastik	1	1	1
12	Plastik	Plastik	Plastik	Plastik	1	1	1
13	Plastik	Plastik	Plastik	Plastik	1	1	1
14	Plastik	Plastik	Plastik	Plastik	1	1	1
15	Plastik	Plastik	Plastik	Plastik	1	1	1
16	Plastik	Plastik	Plastik	Plastik	1	1	1
17	Plastik	Plastik	Plastik	Plastik	1	1	1
18	Plastik	Plastik	Plastik	Plastik	1	1	1
19	Plastik	Plastik	Plastik	Plastik	1	1	1
20	Plastik	Plastik	Plastik	Plastik	1	1	1
21	Plastik	Plastik	Plastik	Plastik	1	1	1
22	Plastik	Plastik	Plastik	Plastik	1	1	1
23	Plastik	Plastik	Plastik	Plastik	1	1	1
24	Plastik	Plastik	Plastik	Plastik	1	1	1
25	Plastik	Plastik	Plastik	Plastik	1	1	1
26	Plastik	Plastik	Plastik	Plastik	1	1	1
27	Plastik	Plastik	Plastik	Plastik	1	1	1
28	Plastik	Plastik	Plastik	Plastik	1	1	1
29	Plastik	Plastik	Plastik	Plastik	1	1	1
30	Plastik	Kaca	Kaca	Plastik	0	0	1
31	Plastik	Plastik	Plastik	Plastik	1	1	1
32	Plastik	Plastik	Plastik	Kaca	1	1	0
33	Plastik	Plastik	Plastik	Plastik	1	1	1
34	Plastik	Plastik	Plastik	Plastik	1	1	1
35	Plastik	Plastik	Plastik	Plastik	1	1	1
36	Plastik	Kaca	Plastik	Plastik	0	1	1
37	Plastik	Plastik	Plastik	Plastik	1	1	1
38	Plastik	Plastik	Plastik	Plastik	1	1	1
39	Plastik	Plastik	Plastik	Plastik	1	1	1
40	Plastik	Plastik	Plastik	Plastik	1	1	1
41	Plastik	Plastik	Plastik	Plastik	1	1	1
42	Plastik	Plastik	Plastik	Plastik	1	1	1

43	Plastik	Plastik	Plastik	Plastik	1	1	1
44	Plastik	Plastik	Plastik	Plastik	1	1	1
45	Plastik	Plastik	Plastik	Plastik	1	1	1
46	Plastik	Plastik	Plastik	Plastik	1	1	1
47	Plastik	Plastik	Plastik	Plastik	1	1	1
48	Plastik	Plastik	Plastik	Plastik	1	1	1
49	Plastik	Plastik	Plastik	Plastik	1	1	1
50	Plastik	Plastik	Plastik	Plastik	1	1	1
51	Plastik	Plastik	Plastik	Plastik	1	1	1
52	Plastik	Plastik	Plastik	Plastik	1	1	1
Jumlah objek yang dikesan betul					50	51	51
Ketepatan (<i>Accuracy</i>)					96.15	98.08	98.08

Ketepatan terendah = 96.1538462



Identifikasi Sampah Dengan Seni Bina CNN Pra-Latihan ResNet18

Jenis Sampah :

10. Getah

No	Label Sebenar	Keputusan Ujian			Sampah Dikesan Betul		
		Label Ramalan Ke-1	Label Ramalan Ke-2	Label Ramalan Ke-3	Ke-1	Ke-2	Ke-3
1	Getah	Getah	Getah	Getah	1	1	1
2	Getah	Getah	Getah	Getah	1	1	1
3	Getah	Getah	Getah	Getah	1	1	1
4	Getah	Getah	Getah	Getah	1	1	1
5	Getah	Getah	Getah	Getah	1	1	1
6	Getah	Getah	Getah	Getah	1	1	1
7	Getah	Getah	Getah	Getah	1	1	1
8	Getah	Fabrik	Fabrik	Getah	0	0	1
9	Getah	Getah	Getah	Getah	1	1	1
10	Getah	Getah	Getah	Getah	1	1	1
11	Getah	Getah	Getah	Getah	1	1	1
12	Getah	Getah	Getah	Getah	1	1	1
13	Getah	Getah	Getah	Getah	1	1	1
14	Getah	Getah	Getah	Getah	1	1	1
Jumlah objek yang dikesan betul					13	13	14
Ketepatan (Accuracy)					92.86	92.86	100

Ketepatan terendah = 92.8571429

Identifikasi Sampah Dengan Seni Bina CNN Pra-Latihan ResNet18

Jenis Sampah : 11.Kayu

No	Label Sebenar	Keputusan Ujian			Sampah Dikesan Betul		
		Label Ramalan Ke-1	Label Ramalan Ke-2	Label Ramalan Ke-3	Ke-1	Ke-2	Ke-3
1	Kayu	Kayu	Kayu	Kayu	1	1	1
2	Kayu	Kayu	Kayu	Kayu	1	1	1
3	Kayu	Kayu	Kayu	Kayu	1	1	1
4	Kayu	Kayu	Kayu	Kayu	1	1	1
5	Kayu	Kayu	Kayu	Kayu	1	1	1
6	Kayu	Kayu	Kayu	Kayu	1	1	1
7	Kayu	Kayu	Kayu	Kayu	1	1	1
8	Kayu	Kayu	Kayu	Kayu	1	1	1
9	Kayu	Kayu	Kayu	Kayu	1	1	1
10	Kayu	Kayu	Kayu	Kayu	1	1	1
11	Kayu	Kayu	Kayu	Kayu	1	1	1
12	Kayu	Daun	Daun	Kayu	0	0	1
13	Kayu	Kayu	Kayu	Kayu	1	1	1
14	Kayu	Kayu	Kayu	Kayu	1	1	1
15	Kayu	Kayu	Kayu	Kayu	1	1	1
16	Kayu	Kayu	Kayu	Kayu	1	1	1
17	Kayu	Kayu	Kayu	Kayu	1	1	1
18	Kayu	Kayu	Kayu	Kayu	1	1	1
19	Kayu	Kayu	Kayu	Kayu	1	1	1
20	Kayu	Kayu	Kayu	Kayu	1	1	1
21	Kayu	Kayu	Kayu	Kayu	1	1	1
22	Kayu	Kayu	Kayu	Kayu	1	1	1
23	Kayu	Kayu	Kayu	Kayu	1	1	1
24	Kayu	Kayu	Kayu	Kayu	1	1	1
25	Kayu	Kayu	Kayu	Kayu	1	1	1
26	Kayu	Kayu	Kayu	Kayu	1	1	1
27	Kayu	Kayu	Kayu	Kayu	1	1	1
Jumlah objek yang dikesan betul					26	26	27
Ketepatan (Accuracy)					96.3	96.3	100

Ketepatan terendah = 96.2962963