



**RANCANG BANGUN *MOBILE ROBOT VISION* PENGANTAR
MAKANAN PADA SEBUAH RESTORAN**

TUGAS AKHIR

Program Studi

S1 Teknik Komputer

**INSTITUT BISNIS
DAN INFORMATIKA**

stikom
SURABAYA

Oleh:

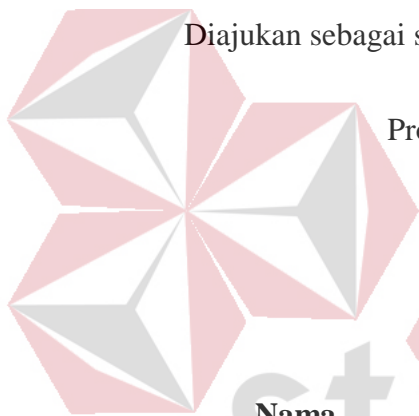
MUHAMMAD FARIS AKBAR

15410200070

**FAKULTAS TEKNOLOGI DAN INFORMATIKA
INSTITUT BISNIS DAN INFORMATIKA STIKOM SURABAYA
2019**

**RANCANG BANGUN *MOBILE ROBOT VISION* PENGANTAR
MAKANAN PADA SEBUAH RESTORAN**

TUGAS AKHIR



Diajukan sebagai salah satu syarat untuk menyelesaikan

Program Sarjana Teknik

Oleh:

Nama : MUHAMMAD FARIS AKBAR

NIM : 15.41020.0070

Program : S1 (Strata Satu)

Jurusan : Teknik Komputer

FAKULTAS TEKNOLOGI DAN INFORMATIKA

INSTITUT BISNIS DAN INFORMATIKA STIKOM SURABAYA

2019



INSTITUT BISNIS
DAN INFORMATIKA

***“Melestarikan hal-hal lama yang baik dan mengembangkan hal-hal yang baru
yang lebih baik dan bermanfaat”***

stikom
SURABAYA

Kupersembahkan Kepada ALLAH SWT

Ibu, Bapak, Kakak, Adik dan semua keluarga tercinta,

**Yang selalu mendukung, memotivasi dan menyisipkan nama saya dalam
doa-doa terbaiknya.**

**Beserta semua teman yang selalu membantu, mendukung dan memotivasi
agar tetap berusaha menjadi lebih baik.**

**Serta seseorang yang selalu mendukung dalam keadaan sehat ataupun
dalam keadaan tak sehat sekalipun.**

RANCANG BANGUN *MOBILE ROBOT VISION* PENGANTAR MAKANAN PADA SEBUAH RESTORAN

Dipersiapkan dan disusun oleh
MUHAMMAD FARIS AKBAR
NIM : 15.41020.0070

Telah diperiksa, diuji, dan disetujui oleh Dewan Pembahas
Pada : Agustus 2019

Susunan Dewan Pembimbing dan Penguji

Pembimbing

- I. Dr. Susijanto Tri Rasmana, S.Kom., M.T.**
NIDN. 0727097302
- II. Harianto, S.Kom., M.Eng.**
NIDN. 0722087701



Penguji

- I. Weny Indah Kusumawati, S.Kom., M.MT.**
NIDN. 0721047201



Tugas Akhir ini telah diterima sebagai salah satu persyaratan

Untuk memperoleh gelar Sarjana



FAKULTAS TEKNOLOGI
DAN INFORMATIKA
stikom
SURABAYA

Dr. Jusak

7/8/19

Dekan Fakultas Teknik dan Informatika

FAKULTAS TEKNOLOGI DAN INFORMATIKA
INSTITUT BISNIS DAN INFORMATIKA STIKOM SURABAYA

SURAT PERNYATAAN

PERSETUJUAN PUBLIKASI DAN KEASLIAN KARYA ILMIAH

Sebagai mahasiswa Institut Bisnis dan Informatika Stikom Surabaya, saya :

Nama : Muhammad Faris Akbar

NIM : 15.41020.0070

Program Studi : S1 Teknik Komputer

Fakultas : Fakultas Teknologi dan Informatika

Jenis Karya : Tugas Akhir

Judul Karya : **RANCANG BANGUN MOBILE ROBOT VISION
PENGANTAR MAKANAN PADA SEBUAH
RESTORAN**

Menyatakan dengan sesungguhnya bahwa:

1. Demi pengembangan Ilmu Pengetahuan, Teknologi dan Seni, saya menyetujui memberikan kepada Institut Bisnis dan Informatika Stikom Surabaya Hak Bebas Royalti Non-Eksklusif (*Non-Exclusive Royalti Free Right*) atas seluruh isi/ sebagian karya ilmiah saya tersebut di atas untuk disimpan, dialihmediakan dan dikelola dalam bentuk pangkalan data (*database*) untuk selanjutnya didistribusikan atau dipublikasikan demi kepentingan akademis dengan tetap mencantumkan nama saya sebagai penulis atau pencipta dan sebagai pemilik Hak Cipta
2. Karya tersebut di atas adalah karya asli saya, bukan plagiat baik sebagian maupun keseluruhan. Kutipan, karya atau pendapat orang lain yang ada dalam karya ilmiah ini adalah semata hanya rujukan yang dicantumkan dalam Daftar Pustaka saya
3. Apabila dikemudian hari ditemukan dan terbukti terdapat tindakan plagiat pada karya ilmiah ini, maka saya bersedia untuk menerima pencabutan terhadap gelar kesarjanaan yang telah diberikan kepada saya.

Demikian surat pernyataan ini saya buat dengan sebenarnya.

Surabaya, Agustus 2019

Yang menandatangani



Muhammad Faris Akbar

NIM: 15.41020.0070

ABSTRAK

Di kafe klasik, restoran dan hotel, pelanggan menghadapi banyak masalah karena banyaknya pekerjaan pelayan pada pelanggan, tidak tersedianya pelayan dan pemesanan makanan masih secara manual. Kekurangan ini dapat ditangani dengan menggunakan sebuah sistem otomatisasi restoran seperti mengganti pelayan dengan menggunakan *Mobile Robot*.

Pada riset sebelumnya *Mobile Robot* yang digunakan atau yang diteliti rata-rata masih menggunakan sensor Photodiode untuk membaca garis yang dibuat sebagai rute. Oleh karena itu, pada Tugas Akhir ini telah mengembangkan pembacaan garis rute menggunakan kamera Webcam sebagai komponen utama dalam pengolahan citra serta menggunakan logika *maze mapping* untuk penyelesaian masalah dalam menentukan rute dalam setiap destinasinya, sehingga pengembangan ini menciptakan istilah robot baru yaitu *Mobile Robot Vision*.

Hasil pengujian yang didapatkan pada Tugas Akhir ini berupa robot dapat melaksanakan setiap tujuan melayani satu meja atau dua meja baik dengan beban atau tanpa beban, dengan waktu tempuh rata-rata dan kecepatan rata-rata dipengaruhi beban yang dibawa, semakin berat beban semakin lama waktu robot berjalan dan semakin berkurang kecepatan robot dan keberhasilan rata-rata didapatkan hingga 100%.

Kata Kunci: *Mobile Robot, Maze Mapping, Robot, Robot Pengantar Makanan, Pengolahan Citra, Restoran.*

KATA PENGANTAR

Segala Puja dan Puji syukur atas kehadiran Allah SWT, atas segala rahmat yang telah diberikan-Nya, sehingga penulis dapat menyelesaikan Laporan Tugas Akhir yang berjudul “RANCANG BANGUN *MOBILE ROBOT VISION* PENGANTAR MAKANAN PADA SEBUAH RESTORAN”. Penulisan Laporan Tugas Akhir ini disusun dalam rangka penulisan laporan untuk memperoleh gelar Sarjana Komputer pada Program Studi S1 Teknik Komputer Institut Bisnis dan Informatika Stikom Surabaya.

Dalam usaha menyelesaikan penulisan Laporan Tugas Akhir ini penulis banyak mendapat bantuan dari berbagai pihak baik moral maupun materi. Oleh karena itu penulis mengucapkan terima kasih dan penghargaan setinggi-tingginya kepada:

1. Allah SWT, karena dengan rahmatnya dan hidayahnya penulis dapat menyelesaikan laporan Tugas Akhir ini.
2. Orang Tua yang telah memberikan dorongan dan bantuan baik moral maupun materi sehingga penulis dapat menempuh dan menyelesaikan Tugas Akhir maupun laporan ini.
3. Kepada Bapak Dr. Susijanto Tri Rasmana, S.Kom., M.T., dan Bapak Harianto, S.Kom., M.Eng., selaku Dosen Pembimbing yang selalu memberikan arahan dan bimbingan dalam menyelesaikan Tugas Akhir beserta laporan ini.
4. Kepada Bapak Pauladie Susanto, S.Kom., M.T., selaku Ketua Program Studi S1 Teknik Komputer yang selalu memberikan dukungannya dalam menyelesaikan Tugas Akhir ini.

5. Bapak Dr. Jusak selaku Dekan Fakultas Teknologi dan Informatika (FTI) Institut Bisnis dan Informatika Stikom Surabaya telah membantu proses penyelesaian Tugas Akhir yang dibuat oleh penulis dengan baik.
6. Teman- teman komunitas STIKOM Robotik yang selalu memberikan warna dan dukungan dalam menyelesaikan laporan Tugas Akhir ini.
7. Teman-teman seperjuangan TK angkatan '15 dan semua pihak yang terlibat namun tidak dapat penulis sebutkan satu persatu atas bantuan dan dukungannya.

Penulis berharap semoga Laporan Tugas Akhir ini dapat berguna dan bermanfaat untuk menambah wawasan bagi pembacanya. Penulis juga menyadari dalam penulisan laporan Tugas Akhir ini banyak terdapat kekurangan. Oleh karena itu penulis sangat mengharapkan saran dan kritik untuk memperbaiki kekurangan dan berusaha untuk lebih baik lagi.

Surabaya, Agustus 2019

Penulis

DAFTAR ISI

	Halaman
ABSTRAK	vii
KATA PENGANTAR	viii
DAFTAR ISI	x
DAFTAR GAMBAR	xiv
DAFTAR TABEL	xviii
DAFTAR LAMPIRAN	xxii
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Perumusan Masalah.....	2
1.3 Batasan Masalah.....	3
1.4 Tujuan	3
1.5 Sistematika Penulisan.....	4
BAB II LANDASAN TEORI	5
2.1 Mobile Robot.....	5
2.1.1 Robot Differential Drive	5
2.2 OpenCV.....	6
2.2.1 RGB to HSV	6
2.2.2 <i>Gaussian-blur</i>	8
2.2.3 <i>Edge Detection : Canny</i>	8
2.2.4 <i>Hough Circle Transform</i>	10

2.3	<i>Maze Mapping</i>	10
2.4	Arduino IDE	11
2.5	Remote Desktop Connection	14
2.6	Logitech C270 HD Webcam	16
2.7	Raspberry Pi 3 Model B	17
2.8	Arduino DUE	19
2.9	Keypad Arduino	21
2.10	LCD Display 20x4	23
2.11	Sensor IR Proximity	24
2.12	Baterai	25
2.13	Motor DC	26
2.14	<i>Motor Driver</i> EMS 30A H-Bridge	27
2.15	UBEC	31
2.16	Router ZTE	32
BAB III PERANCANGAN SISTEM		34
3.1	Perancangan Perangkat Keras	34
3.1.1	Blok Diagram	34
3.1.2	Desain Rancang Bangun Robot	42
3.1.3	Hasil Jadi Rancang Bangun Robot	44
3.2	Perancangan Perangkat Lunak (Program)	46
3.2.1	Pengolahan Citra pada Raspberry Pi 3	46
3.2.2	Pengolahan Data pada Arduino DUE	56
3.2.3	Denah Meja	65
BAB IV HASIL DAN PEMBAHASAN		68

4.1	Uji Kamera Deteksi Garis	68
4.1.1	Tujuan Uji Kamera Deteksi Garis	68
4.1.2	Alat yang Digunakan Pada Uji Kamera Deteksi Garis	68
4.1.3	Prosedur Pengujian Pada Uji Kamera Deteksi Garis	69
4.1.4	Hasil Pengujian Pada Uji Kamera Deteksi Garis	69
4.2	Uji Halangan Dengan Sensor Jarak.....	79
4.2.1.	Tujuan Uji Halangan Dengan Sensor Jarak	79
4.2.2.	Alat yang Digunakan Pada Uji Halangan Sensor Jarak	79
4.2.3.	Prosedur Pengujian Pada Uji Halangan Sensor Jarak	79
4.2.4.	Hasil Pengujian Pada Uji Halangan Sensor Jarak	80
4.3	Uji Destinasi 1 Meja.....	82
4.3.1	Tujuan Uji Destinasi 1 Meja	82
4.3.2	Alat yang Digunakan Pada Uji Destinasi 1 Meja.....	82
4.3.3	Prosedur Pengujian Pada Uji Destinasi 1 Meja.....	83
4.3.4	Hasil Pengujian Waktu Tempuh dan Kecepatan Pada Uji Destinasi 1 Meja.....	83
4.3.5.	Hasil Pengujian Keberhasilan Pada Destinasi 1 Meja.....	94
4.4	Uji Destinasi 2 Meja.....	99
4.4.1	Tujuan Uji Destinasi 2 Meja	99
4.4.2	Alat yang Digunakan Pada Uji Destinasi 2 Meja.....	100
4.4.3	Prosedur Pengujian Pada Uji Destinasi 2 Meja.....	100
4.4.4	Hasil Pengujian Waktu Tempuh dan Kecepatan Pada Uji Destinasi 2 Meja.....	101
4.4.5	Hasil Pengujian Keberhasilan Pada Uji Destinasi 2 Meja ...	119
4.5	Analisis Keseluruhan Pengujian yang Telah Dilakukan	126

BAB V PENUTUP.....	130
5.1 Kesimpulan.....	130
5.2 Saran	131
DAFTAR PUSTAKA	132
BIODATA PENULIS	159



DAFTAR GAMBAR

	Halaman
Gambar 2.1 Sistem Gerak Diffrential Drive	5
Gambar 2.2 OpenCV (<i>Open Source Computer Vision Library</i>).....	6
Gambar 2.3 Struktur Warna HSV	7
Gambar 2.4 Hasil Canny Edge Detection	9
Gambar 2.5 Hasil Deteksi Lingkaran Hough Circle Transform	10
Gambar 2.6 Strategi Maze Mapping	11
Gambar 2.7 Arduino IDE <i>Startup</i>	12
Gambar 2.8 Tampilan Lembar Kerja Arduino IDE	13
Gambar 2.9 Aplikasi Remote Desktop Connection	15
Gambar 2.10 Logitech C270 HD Webcam.....	16
Gambar 2.11 Raspberry Pi 3 Board	17
Gambar 2.12 IO Raspberry Pi 3.....	18
Gambar 2.13 Board Arduino DUE.....	19
Gambar 2.14 Penekanan pada Tombol Nomor 5	22
Gambar 2.15 Keypad Arduino	22
Gambar 2.16 LCD 20x4.....	23
Gambar 2.17 Sensor IR Proximity	25
Gambar 2.18 Baterai Li-Po	25

Gambar 2.19 Motor DC PG-45	27
Gambar 2.20 <i>Motor Driver</i> EMS 30A H-Bridge	27
Gambar 2.21 Tata Letak Komponen	29
Gambar 2.22 UBEC (Step Down).....	32
Gambar 2.23 Router ZTE F609	33
Gambar 3.1 Blok Diagram Keseluruhan	35
Gambar 3.2 SSID <i>Access Point</i>	37
Gambar 3.3 Tampilan Remote Desktop Connection	38
Gambar 3.4 Tampilan Login ke Raspberry Pi 3	38
Gambar 3.5 Tampilan Aplikasi Terminal bagian Desktop	39
Gambar 3.6 Tampilan Aplikasi Terminal bagian Folder syna	39
Gambar 3.7 Desain Robot Tampak Depan	42
Gambar 3.8 Desain Robot dari Samping.....	43
Gambar 3.9 Robot Tampak dari Depan	44
Gambar 3.10 Robot Tampak dari Samping.....	44
Gambar 3.11 Tombol Interface Robot	45
Gambar 3.12 Tampilan Display LCD dan Keypad Arduino.....	45
Gambar 3.13 Flowchart Keseluruhan Raspberry Pi 3.....	46
Gambar 3.14 Flowchart Pengolahan Citra Bagian 1	47
Gambar 3.15 Citra RGB Garis Warna Hitam	48
Gambar 3.16 Citra RGB Garis Warna Kuning	48

Gambar 3.17 Citra RGB Garis Warna Merah.....	49
Gambar 3.18 Citra HSV Garis Warna Hitam.....	50
Gambar 3.19 Citra HSV Garis Warna Kuning.....	50
Gambar 3.20 Citra HSV Garis Warna Merah	51
Gambar 3.21 Flowchart Pengolahan Citra Bagian 2.....	51
Gambar 3.22 Citra <i>Canny Edge Detection</i> Garis Warna Hitam	52
Gambar 3.23 Citra <i>Canny Edge Detection</i> Garis Warna Kuning	52
Gambar 3.24 Citra <i>Canny Edge Detection</i> Garis Warna Merah.....	53
Gambar 3.25 Flowchart Pengolahan Citra Bagian 3.....	54
Gambar 3.26 Nilai Koordinat X,Y, dan Jari-Jari Warna Hitam.....	55
Gambar 3.27 Nilai Koordinat X,Y, dan Jari-Jari Warna Kuning.....	55
Gambar 3.28 Nilai Koordinat X,Y, dan Jari-Jari Warna Merah.....	56
Gambar 3.29 Flowchart Program Utama Arduino DUE.....	57
Gambar 3.30 Flowchart Program Destinasi	59
Gambar 3.31 Flowchart Program <i>Maze Mapping</i>	60
Gambar 3.32 Flowchart Program Gerak	62
Gambar 3.33 Flowchart Program Sub-Cabang (Tracking, Next Trip, Homing) ..	64
Gambar 3.34 Denah Meja	65
Gambar 4.1 Citra RGB (a) Deteksi Warna Hitam, (b) Deteksi Warna Kuning, (c) Deteksi Warna Merah	70
Gambar 4.2 Citra HSV (a) Deteksi Warna Hitam, (b) Deteksi Warna Kuning, (c) Deteksi Warna Merah	73

Gambar 4.3 Citra <i>Edge</i> (a) Deteksi Garis Hitam, (b) Deteksi Garis Kuning, (c) Deteksi Garis Merah	74
Gambar 4.4 Nilai Koordinat X, Y, dan Jari-Jari (r) dari Warna Hitam	76
Gambar 4.5 Nilai Koordinat X, Y, dan Jari-Jari (r) dari Warna Kuning	77
Gambar 4.6 Nilai Koordinat X, Y, dan Jari-Jari (r) dari Warna Merah.....	78
Gambar 4.7 Titik Jalur yang Dihalang	80



DAFTAR TABEL

	Halaman
Tabel 2.1 Spesifikasi Raspberry Pi 3	17
Tabel 2.2 Konektor Raspberry Pi 3	18
Tabel 2.3 Datasheet EMS 30A H-Bridge	29
Tabel 2.4 Keterangan Power dan Motor	30
Tabel 3.1 Ukuran Desain Rancang Bangun	42
Tabel 3.2 Nama-nama Komponen	43
Tabel 3.3 Nilai HSV Tiap Warna	49
Tabel 4.1 Nilai Skala HSV Warna Hitam	71
Tabel 4.2 Nilai Skala HSV Warna Kuning	72
Tabel 4.3 Nilai Skala HSV Warna Merah	72
Tabel 4.4 Pengujian Halangan Meja 1	81
Tabel 4.5 Pengujian Halangan Meja 2	81
Tabel 4.6 Pengujian Halangan Meja 3	81
Tabel 4.7 Pengujian Halangan Meja 4	82
Tabel 4.8 Pengujian Meja Nomor 1 Tanpa Beban	84
Tabel 4.9 Pengujian Meja Nomor 2 Tanpa Beban	84
Tabel 4.10 Pengujian Meja Nomor 3 Tanpa Beban	85
Tabel 4.11 Pengujian Meja Nomor 4 Tanpa Beban	85

Tabel 4.12 Pengujian Meja Nomor 1 Beban 1 Kg	87
Tabel 4.13 Pengujian Meja Nomor 2 Beban 1 Kg	87
Tabel 4.14 Pengujian Meja Nomor 3 Beban 1 Kg	87
Tabel 4.15 Pengujian Meja Nomor 4 Beban 1 Kg	88
Tabel 4.16 Pengujian Meja Nomor 1 Beban 2 Kg	89
Tabel 4.17 Pengujian Meja Nomor 2 Beban 2 Kg	89
Tabel 4.18 Pengujian Meja Nomor 3 Beban 2 Kg	89
Tabel 4.19 Pengujian Meja Nomor 4 Beban 2 Kg	90
Tabel 4.20 Pengujian Meja Nomor 1 Beban 3 Kg	91
Tabel 4.21 Pengujian Meja Nomor 2 Beban 3 Kg	91
Tabel 4.22 Pengujian Meja Nomor 3 Beban 3 Kg	91
Tabel 4.23 Pengujian Meja Nomor 4 Beban 3 Kg	92
Tabel 4.24 Pengujian Meja Nomor 1 Beban 4 Kg	93
Tabel 4.25 Pengujian Meja Nomor 2 Beban 4 Kg	93
Tabel 4.26 Pengujian Meja Nomor 3 Beban 4 Kg	93
Tabel 4.27 Pengujian Meja Nomor 4 Beban 4 Kg	94
Tabel 4.28 <i>Rule</i> Keberhasilan Destinasi 1 Meja	94
Tabel 4.29 Pengujian Keberhasilan Tanpa Beban	95
Tabel 4.30 Pengujian Keberhasilan Beban 1 Kg	97
Tabel 4.31 Pengujian Keberhasilan Beban 2 Kg	98
Tabel 4.32 Pengujian Keberhasilan Beban 3 Kg	98

Tabel 4.33 Pengujian Keberhasilan Beban 4 Kg	99
Tabel 4.34 Pengujian Urutan 1 dan 2, 2 dan 1 Tanpa Beban.....	102
Tabel 4.35 Pengujian Urutan 1 dan 3, 3 dan 1 Tanpa Beban.....	102
Tabel 4.36 Pengujian Urutan 1 dan 4, 4 dan 1 Tanpa Beban.....	103
Tabel 4.37 Pengujian Urutan 2 dan 3, 3 dan 2 Tanpa Beban.....	104
Tabel 4.38 Pengujian Urutan 2 dan 4, 4 dan 2 Tanpa Beban.....	104
Tabel 4.39 Pengujian Urutan 3 dan 4, 4 dan 3 Tanpa Beban.....	105
Tabel 4.39 Pengujian Urutan 1 dan 2, 2 dan 1 Beban 1 Kg.....	108
Tabel 4.40 Pengujian Urutan 2 dan 3, 3 dan 2 Beban 1 Kg.....	109
Tabel 4.41 Pengujian Urutan 3 dan 4, 4 dan 3 Beban 1 Kg.....	109
Tabel 4.42 Pengujian Urutan 1 dan 2, 2 dan 1 Beban 2 Kg.....	111
Tabel 4.43 Pengujian Urutan 2 dan 3, 3 dan 2 Beban 2 Kg.....	112
Tabel 4.44 Pengujian Urutan 3 dan 4, 4 dan 3 Beban 2 Kg.....	112
Tabel 4.45 Pengujian Urutan 1 dan 2, 2 dan 1 Beban 3 Kg.....	114
Tabel 4.46 Pengujian Urutan 2 dan 3, 3 dan 2 Beban 3 Kg.....	115
Tabel 4.47 Pengujian Urutan 3 dan 4, 4 dan 3 Beban 3 Kg.....	115
Tabel 4.48 Pengujian Urutan 1 dan 2, 2 dan 1 Beban 4 Kg.....	117
Tabel 4.49 Pengujian Urutan 2 dan 3, 3 dan 2 Beban 4 Kg.....	117
Tabel 4.50 Pengujian Urutan 3 dan 4, 4 dan 3 Beban 4 Kg.....	118
Tabel 4.51 <i>Rule</i> Keberhasilan Destinasi 2 Meja	119
Tabel 4.52 Pengujian Keberhasilan Tanpa Beban	120

Tabel 4.53 Pengujian Keberhasilan Beban 1 Kg	124
Tabel 4.54 Pengujian Keberhasilan Beban 2 Kg	125
Tabel 4.55 Pengujian Keberhasilan Beban 3 Kg	125
Tabel 4.56 Pengujian Keberhasilan Beban 4 Kg	126
Tabel 4.57 Nilai Skala <i>Hue</i> , <i>Saturation</i> , dan <i>Value</i>	127
Tabel 4.57 Nilai Keseluruhan Destinasi 1 Meja	127
Tabel 4.58 Nilai Keseluruhan Destinasi 2 Meja	128



DAFTAR LAMPIRAN

	Halaman
Lampiran 1. Program OpenCV 2.1.0	134
Lampiran 2. Program <i>Mobile Robot Vision</i>	139



BAB I

PENDAHULUAN

1.1 Latar Belakang

Pelanggan adalah nyawa dari setiap bisnis yang dibangun. Bisnis tidak akan bisa tumbuh dengan besar tanpa adanya dukungan dan dorongan dari pelanggan yang mempercayakan kebutuhan mereka kepada bisnis tersebut. Salah satu cara mendapatkan kepercayaan dari pelanggan yakni pelayanan pengantaran hidangan atau makanan dengan cepat dan tepat oleh pelayan.

Di kafe klasik, restoran dan hotel, pelanggan menghadapi banyak masalah karena banyaknya pekerjaan pelayan pada pelanggan, tidak tersedianya pelayan dan pemesanan makanan secara manual. Kekurangan ini dapat ditangani dengan menggunakan sebuah sistem otomatisasi restoran seperti menggunakan “Pelayan Robot” yang membantu untuk mengantarkan makanan atau minuman. (Asif, 2015)

Robot yang biasanya digunakan dalam aktivitas kehidupan sehari-hari adalah *Mobile Robot*. *Mobile Robot* adalah robot yang memiliki aktuator roda atau kaki untuk memindahkan robot ke setpoint. Salah satunya adalah robot pembantu. Ini memiliki tujuan untuk membuat pelanggan memesan menu dengan mudah dan mempermudah pekerjaan bagi staf dan pemilik restoran (Safii, 2017)

Pada penelitian sebelumnya yang dilakukan oleh (Safii, 2017) dalam jurnal yang berjudul “Mobile Robot Pembawa Peralatan Makan Kotor Otomatis Pada Sebuah Restoran” dan (Asif, 2015) dalam jurnalnya yang berjudul “Waiter Robot - Solution to Restaurant Automation”. Pada kedua jurnal tersebut menjelaskan robot yang digunakan adalah robot *line tracer* atau *line follower* yang

memakai Photodiode sebagai sensor pendeteksi garis yang digunakan untuk rute robot, sedangkan kelemahan dari sensor garis ini yaitu sensor hanya dapat mengerti jarak antara robot dengan garis, itupun dengan ketelitian yang terbatas, tergantung dari jumlah sensor yang dipakai dan peletakkannya. Hal ini menyebabkan robot *line follower* tidak mengetahui besar simpangan antara robot dengan garis seperti ketika membaca garis perempatan ataupun persimpangan, robot akan mendeteksi garis berdasarkan garis yang terlebih dahulu terdeteksi oleh sensor dan juga kurangnya ketelitian dalam membaca posisi garis, sehingga terkadang robot akan bingung dalam pemilihan garis yang akan dilewati saat di perempatan ataupun di persimpangan. Salah satu solusi untuk mengatasi masalah tersebut yaitu dengan mengganti sensor pendeteksi garis yang memakai Photodiode dengan menggunakan kamera Webcam.

Kamera Webcam sebagai *vision* (penglihatan) dapat membaca garis dengan memanfaatkan warna dari garis, sehingga robot dapat berjalan mengikuti warna jalur yang digunakan sebagai rute serta robot dapat mengantarkan makanan pada meja makan pelanggan dengan rute yang sesuai menggunakan algoritma *Maze Mapping* pada tujuan yang telah dipilih sebelumnya dengan keypad, tombol *push button* untuk melanjutkan destinasi dan *switch mode* untuk menentukan robot maju menuju destinasi atau mundur kembali ke posisi awal (*home*).

1.2 Perumusan Masalah

Berdasarkan latar belakang yang telah diuraikan di atas, diperoleh perumusan masalah sebagai berikut:

1. Bagaimana cara mendeteksi warna garis menggunakan kamera pada rute yang digunakan?
2. Bagaimana cara robot mengetahui posisi meja yang akan dituju dan kembali ke posisi awal (*home*)?

1.3 Batasan Masalah

Dalam sistem ini, agar tidak menyimpang dari tujuan yang nantinya akan dicapai, maka pembahasan masalah dibatasi pada hal-hal sebagai berikut:

1. Tidak membawa minuman atau makanan yang berkuah dan beban maksimal yang dapat dibawa oleh robot yakni 3 Kg.
2. Kamera yang digunakan adalah kamera Webcam.
3. Mendeteksi garis tebal yang berukuran 2,5 cm pada banner.
4. Rute yang dibuat telah ditentukan dan berjalan dipermukaan datar dan tidak bergelombang.
5. Untuk deteksi warna menggunakan algoritma Pengolahan Citra yang sudah disediakan pada OpenCV 2.1.0.
6. Algoritma *Maze Mapping* yang digunakan menggunakan metode *return*.

1.4 Tujuan

Tujuan dari Tugas Akhir ini sebagai berikut:

1. Kamera Webcam dapat mendeteksi warna garis.
2. Robot mengetahui posisi meja yang akan dituju dengan algoritma *Maze Mapping* dan kembali posisi awal (*home*).

1.5 Sistematika Penulisan

Untuk memudahkan pembaca dalam memahami persoalan dan pembahasannya, maka penulisan Laporan Tugas Akhir ini dibuat dengan sistematika sebagai berikut:

BAB I PENDAHULUAN

Pada bab ini membahas tentang latar belakang masalah dan penjelasan permasalahan secara umum, perumusan masalah serta batasan masalah yang dibuat, serta tujuan dari pembuatan Tugas Akhir dan sistematika penulisan buku ini.

BAB II LANDASAN TEORI

Pada bab ini membahas teori-teori yang berhubungan dan mendukung dalam pembuatan Tugas Akhir seperti *Mobile Robot*, *Hough Line Transform*, *Maze Mapping*, Arduino dan literatur yang menunjang dalam pembuatan Tugas Akhir ini.

BAB III PERANCANGAN SISTEM

Pada bab ini membahas tentang perancangan sistem baik pada bagian perangkat keras, maupun perangkat lunak pada penerapan robot nantinya.

BAB IV HASIL DAN PEMBAHASAN

Pada bab ini menjelaskan tentang hasil pengujian robot.

BAB V PENUTUP

Pada bab ini menjelaskan tentang kesimpulan dan saran. Kesimpulan akan dijelaskan berdasarkan dari hasil pengujian alat Tugas Akhir ini, serta saran-saran untuk pengembangan.

BAB II

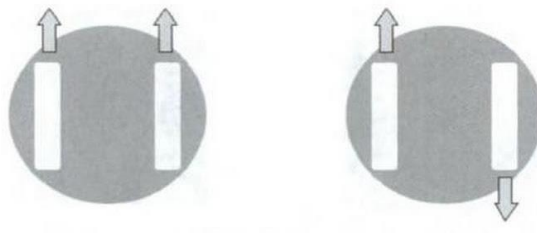
LANDASAN TEORI

2.1 Mobile Robot

Robot adalah peralatan elektro-mekanik atau bio-mekanik, atau gabungan peralatan yang menghasilkan gerakan otonomi maupun berdasarkan gerakan yang diperintahkan. *Mobile robot* adalah robot yang memiliki ciri khas mempunyai roda sebagai aktuatornya, yang memungkinkan robot untuk melakukan pergerakan. Contoh aplikasi robot dalam kehidupan sehari-hari adalah pesawat terbang, roket, dan komputer. *Mobile robot* merupakan robot yang dirancang untuk dapat membawa objek dari satu titik menuju ke titik lain sesuai yang diperintahkan. (Halim, 2007)

2.1.1 Robot Differential Drive

Robot yang menggunakan sistem gerak *differential drive* terdiri dari dua buah roda yang terpasang pada kiri dan kanan robot, perhatikan Gambar 2.1. Sistem ini memungkinkan robot berputar di tempat dengan cara memutar motor dengan arah berlawanan. Contoh sistem gerak ini pada kehidupan sehari-hari adalah pada gardan belakang mobil dan mainan mobil *radio control* (RC). (Halim, 2007)



Gambar 2.1 Sistem Gerak Diffrential Drive (Halim, 2007)

2.2 OpenCV

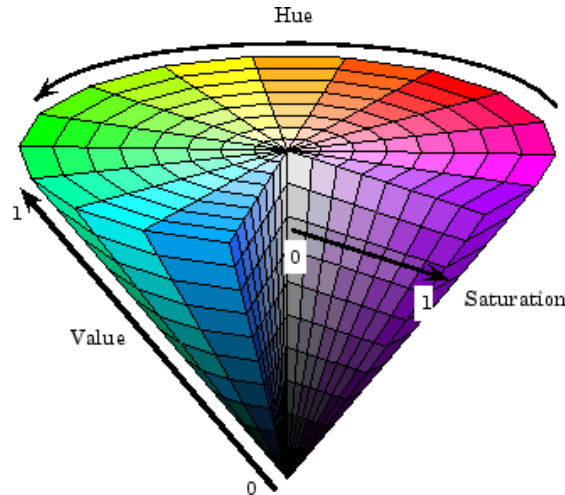


Gambar 2.2 OpenCV (*Open Source Computer Vision Library*) (OpenCV, 2017)

OpenCV adalah program open source berbasis C++ yang saat ini banyak digunakan sebagai program *computer vision*. Salah satu penerapannya adalah pada robotika. Dengan OpenCV, dapat membuat interaksi antara manusia atau objek lain dan robot. Kesemuanya itu membutuhkan OpenCV sebagai program utama antara webcam dan pengolahannya yaitu komputer. (Nalwan, 2010). Pada Tugas Akhir ini digunakan beberapa fitur dari OpenCV yakni RGB to HSV, *GAUSSIAN Blur*, *Edge Detection : Canny*. Berikut beberapa penjelasannya:

2.2.1 RGB to HSV

Model HSV, pertama kali diperkenalkan oleh A.R Smith pada tahun 1978, yang ditunjukkan pada Gambar 2.3. melalui model gambar tersebut, HSV memiliki 3 karakteristik yaitu *Hue*, *Saturation*, dan *Value*. (Fauzan, 2015)



Gambar 2.3 Struktur Warna HSV (Fauzan, 2015)

- a. *Hue* : Menyatakan warna sebenarnya, seperti merah, violet dan lain sebagainya dan digunakan menentukan kemerahan (*redness*), kehijauan (*greenness*), dan sebagainya.
- b. *Saturation* : atau biasa disebut chroma, adalah kemurnian atau kekuatan warna.
- c. *Value* : Kecerahan dari warna, nilaiya berkisar antara 0% - 100% atau 0 – 255 bit. Apabila nilainya 0 maka warnanya akan menjadi hitam, semakin besar nilai maka semakin cerah dan muncul variasi-variasi baru dari warna tersebut.

Rumus HSV:

$$H = \tan\left(\frac{3(G-B)}{(R-G)+(R-B)}\right)$$

$$S = 1 - \frac{\min(R,G,B)}{V}$$

$$V = \frac{R+G+B}{3}$$

Dalam OpenCV cukup menggunakan syntax: `cvCvtColor(gambar_asli, gambar_hasil_hsv, CV_BGR2HSV)`.

2.2.2 *Gaussian-blur*

Gaussian-blur adalah metode yang menggunakan fungsi *Gaussian* dan berfungsi untuk memperhalus citra atau menghilangkan *noise* pada citra. Bobot pada mask penghalusan mengikuti distribusi normal sebagaimana yang dinyatakan dalam persamaan dibawah ini: (Wibisono, 2011)

$$f_0(x,y) = \frac{f_r(x,y)+f_g(x,y)+f_b(x,y)}{3}$$

Dalam OpenCV cukup menggunakan syntax: `cvSmooth(gambar_asli, gambar_hasil_blur, CV_GAUSSIAN,9,9)`.

2.2.3 *Edge Detection : Canny*

Tepi (*edge*) adalah perubahan nilai intensitas derajat keabuan yang cepat atau tiba-tiba (besar) dalam jarak yang singkat. Tujuan mendeteksi tepi sendiri adalah untuk mengelompokkan objek-objek dalam citra, dan juga digunakan untuk menganalisis citra lebih lanjut. Ada banyak algoritma yang digunakan untuk mendeteksi tepi, salah satunya diantaranya adalah deteksi tepi Canny (*Canny Edge Detection*) yang ditunjukkan pada Gambar 2.4.

Canny Edge Detection dikembangkan oleh John F. Canny pada tahun 1986 dan menggunakan algoritma multi-tahap untuk mendeteksi berbagai tepi dalam gambar. Walaupun metode tersebut telah berumur lama sekali, namun metode tersebut telah menjadi metode deteksi tepi standard dan masih pakai dalam penelitian. (Danil, 2001)

Adapun kategori algoritma yang dikembangkan oleh John F. Canny adalah sebagai berikut:

Deteksi : Kemungkinan mendeteksi titik tepi yang benar harus dimaksimalkan sementara kemungkinan salah mendeteksi titik tepi harus diminimalkan. Hal ini dimaksudkan untuk memaksimalkan rasio *signal-to-noise*.

Lokalisasi : Tepi terdeteksi harus sedekat mungkin dengan tepi yang nyata.

Jumlah tanggapan : Satu tepi nyata tidak harus menghasilkan lebih dari satu ujung yang terdeteksi.

Dengan rumusan John F. Canny tentang kriteria ini, maka *Canny Edge Detection* optimal untuk kelas tepian tertentu (dikenal sebagai *step edge*). (Danil, 2001)



Gambar 2.4 Hasil Canny Edge Detection

Adapun syntax di OpenCV untuk *Canny Edge Detection*: `cvCanny(gambar_asli, gambar_hasil_canny, 50, 150, 3)`.

2.2.4 *Hough Circle Transform*

Cara kerja *Hough Circle Transform* hampir sama dengan cara kerja *Hough Line Transform*. Hanya saja untuk *Hough Line Transform* mendeteksi garis didefinisikan menggunakan dua paracentimeter (r, θ) sedangkan untuk *Hough Circle Transform* mendeteksi lingkaran menggunakan tiga paracentimeter yakni:

$$C : (X_{center}, Y_{center}, r)$$

Dimana (X_{center}, Y_{center}) mendefinisikan posisi tengah (*gree point*) dan r adalah jari-jari, yang memungkinkan untuk mendefinisikan lingkaran utuh, seperti yang dapat dilihat pada Gambar 2.2. yang dilingkari menggunakan garis merah. (OpenCV, 2017)



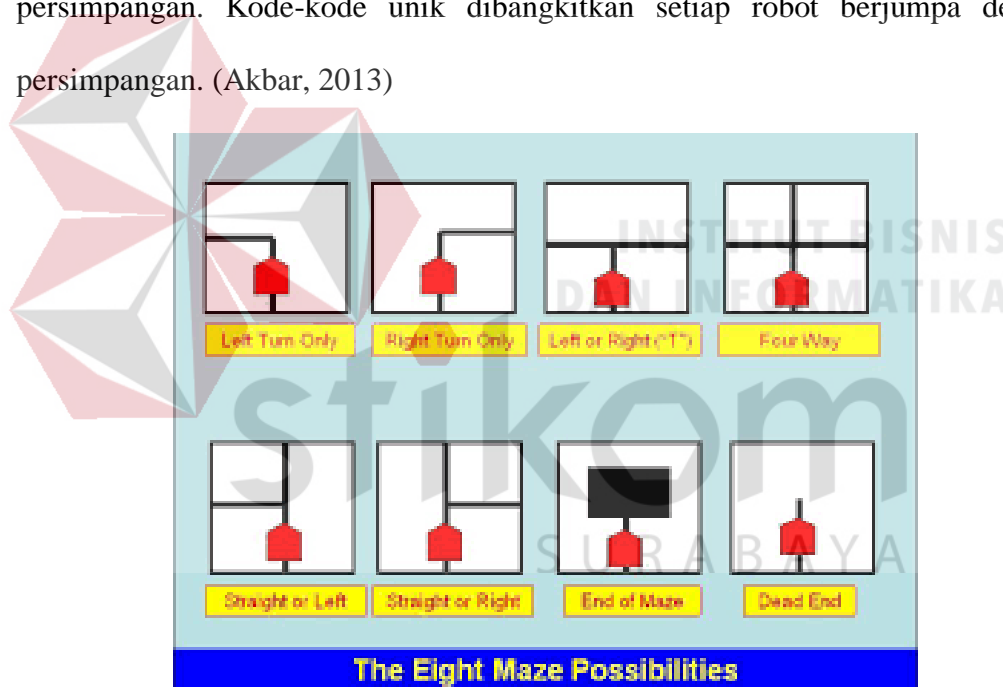
Gambar 2.5 Hasil Deteksi Lingkaran Hough Circle Transform (OpenCV, 2017)

2.3 *Maze Mapping*

Maze adalah suatu jaringan jalan yang rumit. Pada bidang robotika ada dua jenis maze yang umum digunakan, yaitu *wall maze* dan *line maze*. *Wall maze* pada umumnya dikenal dengan istilah labirin, yakni suatu jaringan jalan yang terbentuk atas lorong-lorong dengan dinding tanpa atap. Pada *line maze*, jaringan jalan yang terbentuk dibuat dengan menggunakan garis. Jika garis berwarna putih maka

background berwarna hitam atau sebaliknya. Permasalahan yang timbul pada line maze adalah cara untuk mendapatkan jalur terpendek dari *line maze*.

Permasalahan pada *line maze* diselesaikan dengan algoritma *maze mapping*. Algoritma ini terdiri dari dua mode, yaitu mode *return* dan mode *search*. Pada mode *return*, robot sudah berjalan dari start menuju finish dengan jalur terpendeknya. Jalur terpendek diperoleh dari kode-kode unik yang telah dikonversi. Pada mode *search*, robot melakukan perjalanan dari posisi start menuju finish dengan aturan bahwa robot akan mengutamakan belok kiri bila menjumpai persimpangan. Kode-kode unik dibangkitkan setiap robot berjumpa dengan persimpangan. (Akbar, 2013)



Gambar 2.6 Strategi Maze Mapping (Akbar, 2013)

2.4 Arduino IDE

Arduino IDE (*Integrated Development Environment*) adalah software yang digunakan untuk memprogram di arduino, dengan kata lain Arduino IDE sebagai media untuk memprogram *board* arduino. Arduino IDE ini berguna sebagai

text editor untuk membuat, mengedit, dan juga mevalidasi kode program. Bisa juga digunakan untuk meng-upload ke board Arduino. Kode program yang digunakan pada Arduino disebut dengan istilah Arduino “sketch” atau disebut juga source code arduino, dengan ekstensi file source code .ino. Gambar 2.7 menampilkan tampilan awal Arduino IDE saat pertama kali dibuka.



Gambar 2.7 Arduino IDE *Startup* (Sinaryuda, 2017)

Editor Programming pada umumnya memiliki fitur untuk cut / paste dan untuk find / replace teks, demikian juga pada Arduino IDE. Gambar 2.8 menampilkan tampilan lembar kerja pada Arduino IDE. Pada bagian keterangan aplikasi memberikan pesan balik saat menyimpan dan mengeksport serta sebagai tempat menampilkan kesalahan. **Konsol log** menampilkan teks log dari aktifitas Arduino IDE, termasuk pesan kesalahan yang lengkap dan informasi lainnya. Pojok kanan bawah menampilkan port serial yang digunakan. Tombol toolbar terdapat ikon tombol pintas untuk memverifikasi dan meng-upload program, membuat, membuka, dan menyimpan sketch, dan membuka monitor serial. (Sinaryuda, 2017)



Gambar 2.8 Tampilan Lembar Kerja Arduino IDE (Sinaryuda, 2017)

Verify pada versi sebelumnya dikenal dengan istilah *Compile*. Sebelum aplikasi di-upload ke board Arduino, membiasakan untuk memverifikasi terlebih dahulu sketch yang dibuat. Jika ada kesalahan pada sketch, nanti akan muncul error. Proses *Verify / Compile* mengubah sketch ke binary code untuk di-upload ke mikrokontroler.

Upload tombol ini berfungsi untuk mengupload sketch ke board Arduino. Walaupun tidak mengeklik tombol verify, maka sketch akan di-*compile*, kemudian langsung diupload ke board. Berbeda dengan tombol verify yang hanya berfungsi

untuk memverifikasi source code saja.

- a. **New Sketch** Membuka window dan membuat sketch baru.
- b. **Open Sketch** Membuka sketch yang sudah pernah dibuat. Sketch yang dibuat dengan IDE Arduino akan disimpan dengan ekstensi file .ino.
- c. **Save Sketch** menyimpan sketch, tapi tidak disertai dengan meng-*compile*.
- d. **Serial Monitor** Membuka interface untuk komunikasi serial.
- e. **Keterangan Aplikasi** pesan-pesan yang dilakukan aplikasi akan muncul di sini, misal *Compiling* dan *Done Uploading* ketika meng-*compile* dan mengupload sketch ke board Arduino.
- f. **Konsol log** Pesan-pesan yang dikerjakan aplikasi dan pesan-pesan tentang sketch akan muncul pada bagian ini. Misal, ketika aplikasi meng-*compile* atau ketika ada kesalahan pada sketch yang telah dibuat, maka informasi error dan baris akan diinformasikan di bagian ini.
- g. **Baris Sketch** bagian ini akan menunjukkan posisi baris kursor yang sedang aktif pada sketch.

2.5 Remote Desktop Connection

Remote Desktop Connection adalah fitur Windows yang memungkinkan untuk bisa mengakses desktop komputer secara remote dari komputer lain, bisa melakukan input keyboard, klik mouse, menjalankan aplikasi, reboot dan sebagainya dari komputer lain walaupun berbeda lokasi dan jaringan. Fitur ini tersedia di Windows XP atau versi di atasnya. Tampilan aplikasi Remote Desktop Connection seperti pada Gambar 2.9.



Gambar 2.9 Aplikasi Remote Desktop Connection

Dengan Remote Desktop Connection, user bisa meremote komputer tanpa harus menginstal aplikasi lain. Remote Desktop Connection menggunakan protokol Remote Desktop Protocol (RDP) yang berkomunikasi lewat TCP port 3389. (Ilham, 2013)

Kelebihan:

- a. Mudah digunakan Karena sudah tersedia secara default di Sistem Operasi Windows.
- b. Dapat mengendalikan secara jarak jauh.
- c. Mampu memanipulasi data.
- d. Dapat berbagi file.
- e. Menghidupkan dan mematikan PC secara remote.
- f. Keamanan yang terjamin, karena client tidak bisa memanipulasi seluruh data server kecuali yang di sharing atau diijinkan oleh server.
- g. Adanya peran Client dan Server.

Kekurangan: PC akan bekerja lebih berat.

2.6 Logitech C270 HD Webcam

Webcam adalah sebuah kamera video digital kecil yang dihubungkan ke komputer biasanya melalui colokan USB atau pun colokan PORTCOM. Pada umumnya *webcam* tidak membutuhkan kaset atau tempat penyimpanan data, data hasil perekaman yang didapat langsung ditransfer ke komputer. (Susanto, Effendy, & Alfian Noor , 2014)



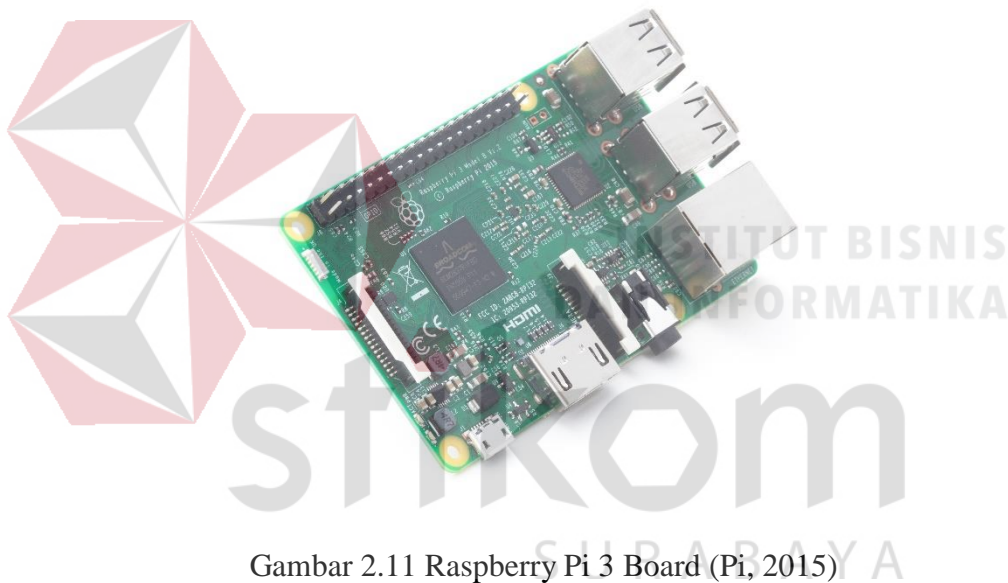
Gambar 2.10 Logitech C270 HD Webcam (Logitech, 2007)

Spesifikasi kamera pada Gambar 2.10:

- a. *Full HD 720p video capture (up to 1080 pixels)*
- b. *HD video calling (1080 x 720 pixels) with recommended system*
- c. *Logitech Fluid Crystal™ Technology**
- d. *Autofocus*
- e. *Photos: Up to 8 megapixels (software enhanced)*
- f. *Built-in mics with automatic noise reduction*
- g. *Hi-Speed USB 2.0 certified (recommended) and Universal clip fits laptops, LCD or CRT monitors*

2.7 Raspberry Pi 3 Model B

Raspberry Pi 3 Model B adalah generasi ketiga Raspberry Pi. Raspberry Pi 3 ini berukuran tunggal board komputer dapat digunakan untuk banyak aplikasi dan menggantikan asli Raspberry Pi Model B+ dan Raspberry Pi 2 Model B. Sementara mempertahankan bentuk mini PC Raspberry Pi 3 B membawa prosesor yang lebih cepat, 10 kali lebih cepat dari generasi pertama Raspberry Pi. Selain itu menambah nirkabel konektivitas Wifi, LAN dan Bluetooth menjadikannya solusi ideal untuk desain komunikasi yang bagus. (Pi, 2015)



Gambar 2.11 Raspberry Pi 3 Board (Pi, 2015)

Spesifikasi:

Tabel 2.1 Spesifikasi Raspberry Pi 3

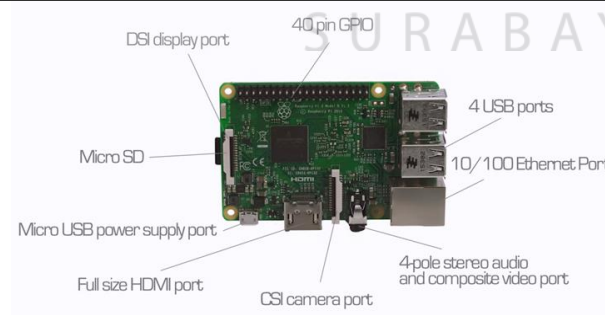
Nama Komponen	Keterangan
<i>Processor</i>	Broadcom BCM2387 chipset. 1.2GHz <i>Quad-Core</i> ARM Cortex-A53 802.11 b/g/n <i>Wireless LAN and Bluetooth 4.1 (Bluetooth Classic and LE)</i>
<i>GPU</i>	<i>Dual Core VideoCore IV® Multimedia Co-Processor. Provides Open GL ES 2.0, hardware-accelerated OpenVG, and 1080p30</i>

Nama Komponen	Keterangan
	H.264 <i>high-profile</i> decode. Capable of 1Gpixel/s, 1.5Gtexel/s or 24GFLOPs with texture filtering and DMA infrastructure
<i>Memory</i>	1GB LPDDR2
<i>Operating System</i>	Boots from Micro SD card, running a version of the Linux operating system or Windows 10 IoT
<i>Dimensions</i>	85 x 56 x 17mm
<i>Power</i>	Micro USB socket 5V1, 2.5A

Konektor (Gambar 2.12):

Tabel 2.2 Konektor Raspberry Pi 3

Nama Konektor	Keterangan
<i>Ethernet</i>	10/100 BaseT Ethernet socket
<i>Video Output</i>	HDMI (rev 1.3 & 1.4) Composite RCA (PAL and NTSC)
<i>Audio Output</i>	Audio Output 3.5mm jack, HDMI USB 4 x USB 2.0 Connector
<i>GPIO Connector</i>	40-pin 2.54 mm (100 mil) expansion header: 2x20 strip Providing 27 GPIO pins as well as +3.3 V, +5 V and GND supply lines
<i>Camera Connector</i>	15-pin MIPI Camera Serial Interface (CSI-2)
<i>Display Connector</i>	Display Serial Interface (DSI) 15 way flat flex cable connector with two data lanes and a clock lane
<i>Memory Card Slot</i>	Push/pull Micro SDIO



Gambar 2.12 IO Raspberry Pi 3 (Pi, 2015)

Key Benefits:

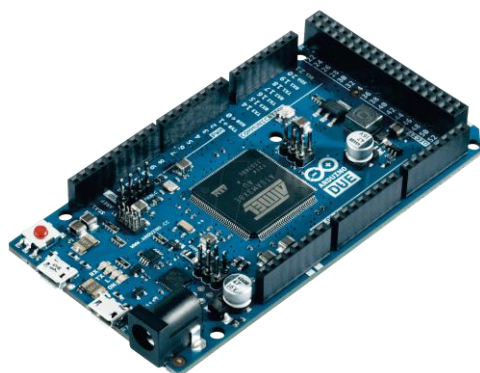
a. *Low cost*

- b. 10x faster processing*
- c. Consistent board format*
- d. Added connectivity*

Key Applications:

- a. Low cost PC/tablet/laptop*
- b. Media center*
- c. Industrial/Home automation*
- d. Print server*
- e. Web camera*
- f. Wireless access point*
- g. IoT applications*
- h. Robotics*
- i. Server/cloud server*
- j. Security monitoring*
- k. Gaming*
- l. Environmental sensing/monitoring(e.g. weather station)*

2.8 Arduino DUE



Gambar 2.13 Board Arduino DUE (Irawan, 2016)

Arduino DUE pada Gambar 2.13 adalah board mikrokontroler yang berbasis pada CPU Atmel SAM3X8E ARM Cortex-M3. Ini adalah board Arduino pertama yang berbasis pada mikrokontroler ARM inti 32-bit. Ini memiliki 54 pin input/output digital (12 di antaranya dapat digunakan sebagai output PWM), 12 input analog, 4 UART (port serial perangkat keras), jam 84 MHz, koneksi USB OTG yang mampu, 2 DAC (digital to analog) , 2 TWI, colokan listrik, header SPI, header JTAG, tombol reset dan tombol hapus.

Peringatan: Tidak seperti kebanyakan board Arduino, board Arduino Due berjalan pada 3.3V. Tegangan maksimum yang dapat ditoleransi I / O pin adalah 3.3V. Menerapkan voltase yang lebih tinggi dari 3.3V ke pin I / O manapun dapat merusak board.

Board berisi segala sesuatu yang dibutuhkan untuk mendukung mikrokontroler cukup menghubungkan ke komputer dengan kabel micro-USB atau menyalakan dengan adaptor AC-ke-DC atau baterai untuk memulai. The Due kompatibel dengan semua perisai Arduino yang bekerja di 3.3V dan sesuai dengan pinout Arduino 1.0. (Irawan, 2016)

Due mengikuti 1.0 pinout:

1. TWI: pin SDA dan SCL yang berada di dekat pin AREF.
2. IOREF: memungkinkan perisai terlampir dengan konfigurasi yang tepat untuk menyesuaikan voltase yang disediakan oleh board. Hal ini memungkinkan kompatibilitas perisai dengan board 3.3V seperti board berbasis Due dan AVR yang beroperasi pada 5V.
3. Pin yang tidak terhubung, disediakan untuk hal tertentu.

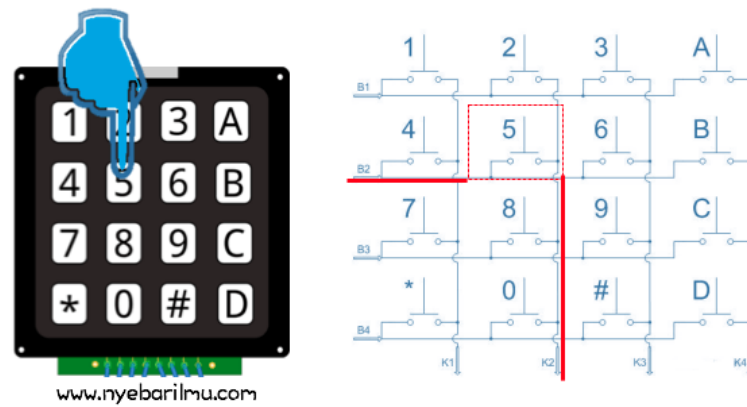
Spesifikasi:

1. Microcontroller AT91SAM3X8E

2. Operating Voltage 3.3V
3. Input Voltage (recommended) 7-12V
4. Input Voltage (limits) 6-16V
5. Digital I/O Pins 54 (of which 12 provide PWM output)
6. Analog Input Pins 12
7. Analog Output Pins 2 (DAC)
8. Total DC Output Current on all I/O lines 130 mA
9. DC Current for 3.3V Pin 800 mA
10. DC Current for 5V Pin 800 mA
11. Flash Memory 512 KB all available for the user applications
12. SRAM 96 KB (two banks: 64KB and 32KB)
13. Clock Speed 84 MHz
14. Length 101.52 mm
15. Width 53.3 mm
16. Weight 36 g

2.9 Keypad Arduino

Keypad seperti pada Gambar 2.15 merupakan antarmuka antara komunikasi perangkat elektronik dengan manusia yang disebut dengan istilah HMI (*Human Machine Interface*). Keypad tersusun atas 16 buah push button yang dirangkai dengan konfigurasi dalam bentuk matrix, sehingga memiliki index baris dan kolom sehingga pin input ke Arduino dapat dikurangi.



Gambar 2.14 Penekanan pada Tombol Nomor 5 (Rezsa, 2017)

Proses pembacaan pada gambar 2.14 dilakukan secara matrix yaitu dengan menggunakan teknik *scanning*, dan pada proses tersebut hal yang dilakukan dengan memberikan umpan data pada satu bagian dan memantau akan adanya *feedback* /*umpan balik* pada bagian lainnya. Umpan data dilakukan di bagian baris dan feedback yang ada dilakukan pengecekan pada bagian kolom. Kondisi saat baris diberikan umpan data, baris lainnya dalam kondisi inversi. (Rezsa, 2017)



Gambar 2.15 Keypad Arduino (Rezsa, 2017)

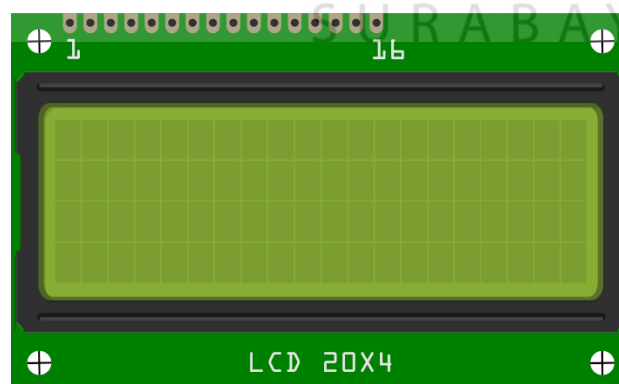
2.10 LCD Display 20x4

LCD kepanjangan dari *Liquid Crystal Display* merupakan jenis penampil yang mempergunakan kristal cair sebagai bahan untuk menampilkan data yang berupa tulisan maupun gambar. Pengaplikasian pada kehidupan sehari-hari yang mudah dijumpai antara lain pada kalkulator, gamebot, televisi, atau pun layar komputer. (Munandar, 2012)

Jenis dari perangkat ini ada yang dan pada postingan ini akan dibahas tentang Tutorial Arduino mengakses LCD 20x4 dengan mudah. Adapun fitur-fitur yang tersedia antara lain:

- a. Terdiri dari 20 kolom dan 4 baris.
- b. Dilengkapi dengan back light.
- c. Mempunyai 192 karakter tersimpan.
- d. Dapat dialMengamati dengan mode 4-bit dan 8-bit.
- e. Terdapat karakter generator terprogram.

Pin – pin LCD 20x4 dan keterangannya



Gambar 2.16 LCD 20x4 (Munandar, 2012)

Keterangan Gambar 2.16:

- a. **GND** : catu daya 0Vdc.
- b. **VCC** : catu daya positif.
- c. **Constrate** : untuk kontras tulisan pada LCD.
- d. **RS** atau **Register Select**.
- e. **High** : untuk mengirim data.
- f. **Low** : untuk mengirim instruksi.
- g. **R/W** atau **Read/Write**.
- h. **High** : mengirim data.
- i. **Low** : mengirim instruksi.
- j. Disambungkan dengan **LOW** untuk pengiriman data ke layar.
- k. **E (enable)** : untuk mengontrol ke LCD ketika bernilai **LOW**, LCD tidak dapat diakses.
- l. **D0 – D7** = Data Bus 0 – 7.
- m. **Backlight +** : disambungkan ke **VCC** untuk menyalakan lampu latar.
- n. **Backlight –** : disambungkan ke **GND** untuk menyalakan lampu latar.

2.11 Sensor IR Proximity

Sensor Jarak Ultrasonik atau Ultrasonic Proximity Sensor adalah sensor jarak yang menggunakan prinsip operasi yang mirip dengan radar atau sonar yaitu dengan menghasilkan gelombang frekuensi tinggi untuk menganalisis gema yang diterima setelah terpantul dari objek yang mendekatnya. Bentuk fisik dari Sensor IR Proximity dapat dilihat di Gambar 2.17. Sensor Proximity ini akan menghitung waktu antara pengiriman sinyal dengan penerimaan sinyal untuk menentukan jarak

objek yang bersangkutan, sering digunakan untuk mendeteksi keberadaan objek dan mengukur jarak objek di proses otomasi pabrik. (Kho, Pengertian Proximity Sensor (Sensor Jarak) dan Jenis-jenisnya, 2016)



Gambar 2.17 Sensor IR Proximity (Kho, Pengertian Proximity Sensor (Sensor Jarak) dan Jenis-jenisnya, 2016)

2.12 Baterai

Baterai adalah sebuah alat yang dapat mengubah energi kimia yang disimpannya menjadi energi Listrik yang dapat digunakan oleh suatu perangkat Elektronik. Hampir semua perangkat elektronik yang portabel seperti Handphone, Laptop, Senter, ataupun Remote Control menggunakan Baterai sebagai sumber listriknya. Dengan adanya Baterai tidak perlu menyambungkan kabel listrik untuk dapat mengaktifkan perangkat elektronik sehingga dapat dengan mudah dibawa kemana-mana. (Kho, Pengertian Baterai dan Jenis-jenis Baterai, 2015)



Gambar 2.18 Baterai Li-Po (Kho, Pengertian Baterai dan Jenis-jenis Baterai, 2015)

Baterai Li-Po yang digambarkan pada Gambar 2.18 adalah singkatan *Lithium Polymer*, baterai ini bersifat cair (*Liquid*), menggunakan elektrolit polimer yang padat, dan mampu menghantarkan daya lebih cepat dan jenis baterai ini adalah hasil pengembangan dari *Lithium Ion*. Baterai Li-Po ini disebut sebagai baterai ramah lingkungan.

1. Kelebihan Baterai Li-Po

- a. Ramah Lingkungan.
- b. Fleksibel bisa dibuat berdasarkan kebutuhan.
- c. Lebih Aman.
- d. Lebih ringan.

2. Kekurangan Baterai Li-Po

- a. Biaya manufaktur mahal.
- b. Harga baterai Juga mahal karena cost untuk energi ini juga mahal.
- c. Butuh perawatan khusus untuk isi ulang, seperti jangan sampai baterai habis baru di isi ulang.
- d. Usia Baterai lebih pendek.

2.13 Motor DC

Motor DC yang digunakan adalah Planetary Gear 45 dapat dilihat pada Gambar 2.19, dengan spesifikasi torsi 25 kgfcm, kecepatan 870 RPm, memiliki rotary internal dengan pulse 7ppr, dan disuplai dengan tegangan 12-24Vdc.



Gambar 2.19 Motor DC PG-45 (MRI, 2012)

Rotary internal yang sudah termasuk pada motor sebenarnya tergolong rotari dengan pulse rendah namun rotari tersebut terpasang langsung pada putaran inti dinamo dan dengan adanya gearbox untuk output putaran sehingga rotari dapat lebih detil dalam melakukan pembacaan putaran. (MRI, 2012)

2.14 Motor Driver EMS 30A H-Bridge



Gambar 2.20 Motor Driver EMS 30A H-Bridge (Electronics, 2010)

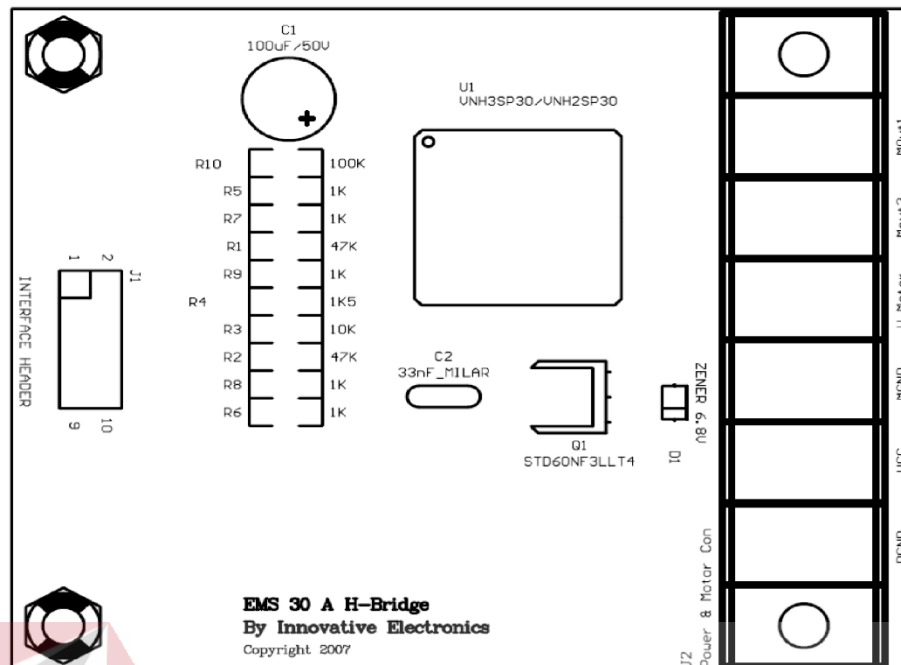
Embedded Module Series (EMS) 30 A H-Bridge merupakan *driver* H-Bridge yang didisain untuk menghasilkan *drive* 2 arah dengan arus kontinyu sampai dengan 30A pada tegangan 5,5 Volt sampai 16 Volt. Modul ini dilengkapi dengan rangkaian sensor arus beban yang dapat digunakan sebagai umpan balik ke

pengendali. Bentuk fisik dari motor driver ini terdapat pada gambar 2.20. Modul ini mampu men-*drive* beban-beban induktif seperti misalnya Relay, Solenoida, Motor DC, Motor Stepper, dan berbagai macam beban lainnya. (Electronics, 2010)

Spesifikasi:

- a. Terdiri dari 1 *driver full* H-Bridge beserta rangkaian *current sense*.
- b. Mampu melewatkan arus kontinyu 30 A.
- c. *Range* tegangan output untuk beban: 5,5 V sampai 16 V.
- d. Input kompatibel dengan level tegangan TTL dan CMOS.
- e. Jalur catu daya input (VCC) terpisah dari jalur catu daya untuk beban (V Mot).
- f. Output tri-state.
- g. Dilengkapi dengan dioda eksternal untuk pengaman beban induktif.
- h. Frekuensi PWM sampai dengan 20 KHz.
- i. *Fault Detection*.
- j. Proteksi hubungan singkat.
- k. Proteksi *overttemperature*.
- l. *Undervoltage* dan *Overvoltage Shutdown*.
- m. *Reverse Battery Protection*.

Tata Letak Komponen dapat dilihat pada Gambar 2.21 berikut ini:



Gambar 2.21 Tata Letak Komponen (Electronics, 2010)

Modul H-Bridge memiliki 1 set *header* (J1) dan 1 set terminal konektor (J2). Pada bagian ini akan dijelaskan deskripsi dan fungsi dari masing-masing *header* dan konektor tersebut.

Interface Header (J1) berfungsi sebagai input untuk antarmuka dengan input-output digital serta output analog dari modul H-Bridge. Berikut deskripsi dari masing-masing pin pada **Interface Header**:

Tabel 2.3 Datasheet EMS 30A H-Bridge

No. Pin	Nama	I/O	Fungsi
1	MIN1	I	Pin input untuk menentukan output MOUT1
2	MIN2	I	Pin input untuk menentukan output MOUT2

No. Pin	Nama	I/O	Fungsi
3	MEN1	I/O	Pin <i>enable</i> untuk output MOUT1 Diberi logika High untuk mengaktifkan <i>half</i> HBridge 1, diberi logika Low secara eksternal untuk menonaktifkan <i>half</i> H-Bridge 1 Jika terjadi kondisi <i>Fault</i> (<i>thermal shutdown</i> , <i>undervoltage</i> , <i>overvoltage</i> , dsb.), maka pin ini akan ditarik Low secara internal oleh modul HBridge untuk melaporkan adanya kondisi <i>Fault</i>
4	MEN2	I/O	Pin <i>enable</i> untuk output MOUT2 Diberi logika High untuk mengaktifkan <i>half</i> HBridge 2, diberi logika Low secara eksternal untuk menonaktifkan <i>half</i> H-Bridge 2 Jika terjadi kondisi <i>Fault</i> (<i>thermal shutdown</i> , <i>undervoltage</i> , <i>overvoltage</i> , dsb.), maka pin ini akan ditarik Low secara internal oleh modul HBridge untuk melaporkan adanya kondisi <i>Fault</i>
5	MCS	O	Output tegangan analog yang berbanding lurus dengan arus beban (<i>Range</i> output 0 – 5 Volt)
6	MPWM	I	Pin input untuk mengatur kerja modul H-Bridge secara PWM
7,9	VCC	-	Terhubung ke catu daya untuk input (5 Volt)
8,10	PGND	-	Titik referensi untuk catu daya input

Arus (dalam Ampere) yang dilewatkan oleh H-Bridge dapat dihitung dengan rumus:

$$I = \frac{\text{Tegangan output pada pin MCS} \times 11370}{1500}$$

Power & Motor Con (J2) berfungsi sebagai konektor untuk catu daya dan beban. Berikut deskripsi dari masing-masing terminal pada *Power & Motor Con*:

Tabel 2.4 Keterangan Power dan Motor

Nama	Fungsi
PGND	Titik referensi untuk catu daya input
VCC	Terhubung ke catu daya untuk input (5 Volt)
MGND	Titik referensi untuk catu daya output ke beban

Nama	Fungsi
V MOTOR (V MOT)	Terhubung ke catu daya untuk output ke beban
MOUT2	Output ke beban dari <i>half</i> H-Bridge kedua
MOUT1	Output ke beban dari <i>half</i> H-Bridge pertama

Sebuah modul H-Bridge 30A dapat digunakan untuk mengatur kerja 1 buah motor DC secara 2 arah. (Electronics, 2010)

2.15 UBEC

Mengubah tegangan, tinggi ke rendah atau sebaliknya, memerlukan rangkaian yang tepat, agar daya dapat di-*deliver* dengan tingkat efisiensi setinggi mungkin. Menurunkan tegangan dengan menggunakan IC regulator seperti 7805, sangat umum digunakan. Regulator ini memiliki kemampuan menangani arus hingga 1A, dengan V_{in} minimal sama dengan 7V, untuk menghasilkan *output* 5V. Dengan perhitungan sederhana, bila $V_{in} = 9V$, maka disipasi daya ~ 4 Watt, satu nilai yang cukup besar (panas), atau menggunakan regulator linier tipe LDO, seperti 2940, yang juga memiliki kemampuan menangani arus hingga 1A, dengan V_{in} minimal sama dengan 5.5V, untuk menghasilkan *output* 5V.

Pilihan lain adalah regulator switching. Untuk kebutuhan mencatu motor servo atau rangkaian lain yang bekerja pada tingkat tegangan 5V – 6V, dapat menggunakan UBEC. UBEC – *Universal Battery Elimination Circuit* adalah rangkaian elektronik yang mengambil daya dari battery pack atau sumber DC lainnya, dan menurunkannya ke level tegangan 5V atau 6V. Tegangan *input* maksimum tergantung pada spesifikasi UBEC.



Gambar 2.22 UBEC (Step Down) (Tjahyadi, 2011)

UBEC biasanya digunakan pada aplikasi yang memerlukan arus lebih tinggi, dan perangkat ini mampu mengantarkan daya dengan efisiensi hingga 92%. Ketika memilih UBEC, pastikan model UBEC yang dipilih memiliki rating arus yang sesuai dengan kebutuhan (beban). Bentuk fisik dari ubec ditampilkan pada gambar 2.22. Rangkaian lain yang juga sering dibutuhkan adalah DC-DC *Booster*. Sebagai contoh, satu produk DC-DC, mampu menghasilkan *output* 3.7V – 34V dengan *input* 3.7V – 34V. Artinya, dengan tegangan *input* minimum 3.7V dapat dihasilkan *output* maksimum 34V, dengan arus *input* maksimum 3A, serta mampu men-deliver daya dengan tingkat efisiensi hingga 90%.

Tegangan *input* tidak boleh lebih besar dari *output* yang dihasilkan. Dalam banyak aplikasi, khususnya aplikasi robotik, seringkali dibutuhkan kombinasi keduanya, agar dapat menggunakan satu catu battery pack. (Tjahyadi, 2011)

2.16 Router ZTE

Router memiliki komponen-komponen yang sama dengan PC dekstop, router mempunyai CPU, memori, sistem bus, dan banyak interface input/output sehingga banyak yang mengatakan bahwa router adalah sebuah komputer khusus.

Tetapi router didesain untuk melakukan tugas khusus yang tidak dimiliki oleh PC desktop. Contoh, router menghubungkan dan mengizinkan komunikasi antara dua jaringan dan menentukan jalur data yang melalui koneksi jaringan. (Yuliandoko, 2018)



Gambar 2.23 Router ZTE F609 (Yuliandoko, 2018)

Spesifikasi Router ZTE F609 seperti pada Gambar 2.23:

- a. Frekuensi : 2.4 GHz, IEEE 802.11 b/g/n.
- b. 128 bits WEP *data encryption*.
- c. WPA/WPA2 Security, WPS.
- d. WMM, Max 4 Broadcast/hidden SSIDs dan Max 128 users one SSID.

BAB III

PERANCANGAN SISTEM

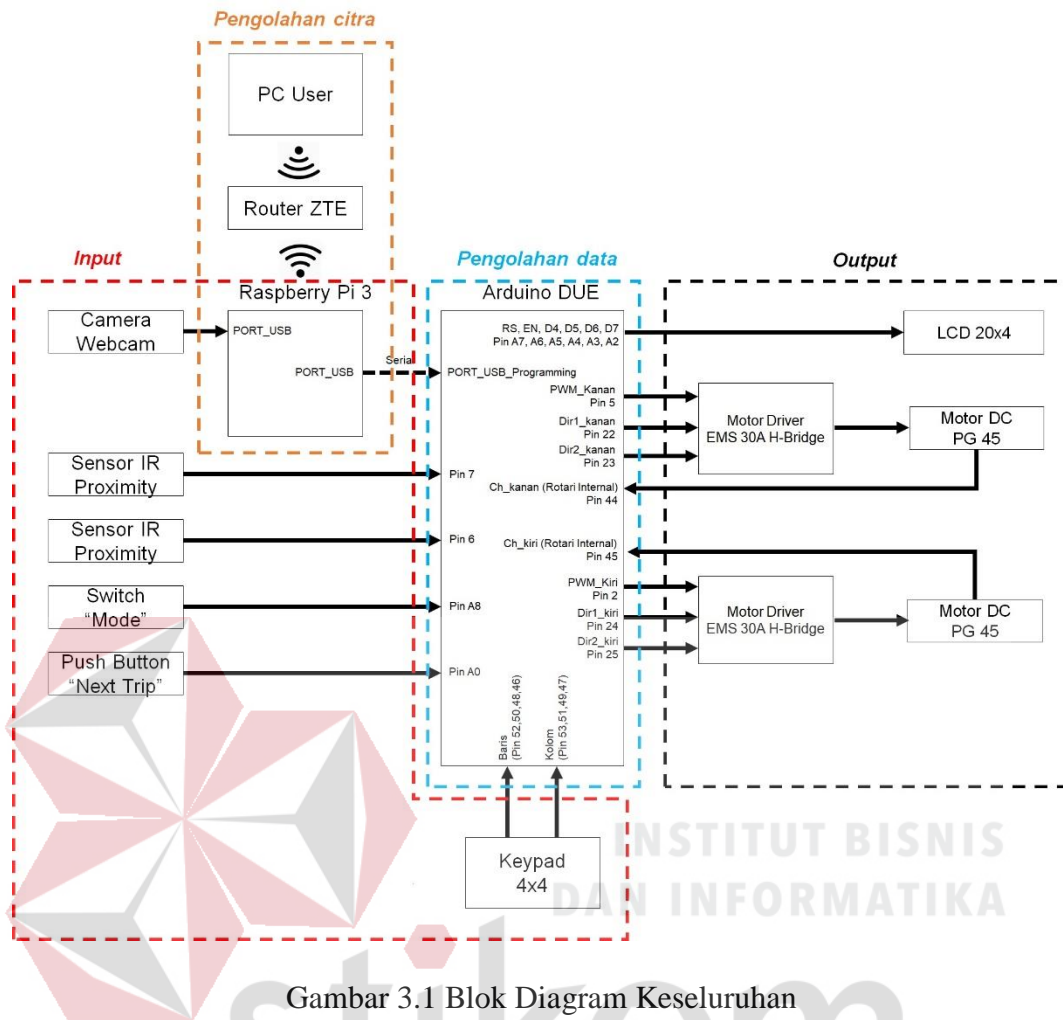
Pada bagian bab ini akan menjelaskan beberapa hal mengenai perancangan keseluruhan sistem. Mulai dari perancangan perangkat keras dan perangkat lunak (Program). Mulai dari perancangan perangkat keras meliputi Blok Diagram keseluruhan, penggunaan komponen atau modul, perancangan desain mekanik robot, dan hasil jadi rancang bangun robot, sedangkan perancangan perangkat lunak meliputi pengolahan citra pada Raspberry Pi 3, pengolahan data pada Arduino DUE dan Denah Meja. Berikut penjelasan dari perancangan perangkat keras Tugas Akhir ini:

3.1 Perancangan Perangkat Keras

Pada bagian ini menjelaskan perincian perangkat robot seperti Blok Diagram secara keseluruhan, Desain Rancang Bangun Robot, bentuk hasil dari Rancang Bangun. Berikut penjelasan masing-masing bagian:

3.1.1 Blok Diagram

Pada bagian Blok Diagram, disini pembahasan akan dibagi menjadi empat 4 materi yakni Input yang ditandai dengan kotak Garis Warna Merah, Pengolahan Citra yang ditandai dengan kotak garis warna orange, Pengolahan Data yang ditandai dengan kotak garis warna biru, dan Output yang ditandai dengan kotak Garis Warna Hitam. Untuk lebih jelas bagaimana bentuk Blok Diagram secara keseluruhan bisa dilihat pada Gambar 3.1 dan penjelasan dibawahnya.



Gambar 3.1 Blok Diagram Keseluruhan

Tiap-tiap bagian Blok Diagram pada gambar diatas dapat dijelaskan sebagai berikut:

a. Input

1. Kamera Webcam:

- Mengambil citra yang dilihat, kemudian menjadi data output dari kamera.
- Data output yang keluar dari kamera masuk ke Raspberry pi 3 melalui PORT_USB Raspberry Pi 3, untuk kemudian diolah dari hasil gambar citra yang didapatkan oleh kamera Webcam.

2. Sensor IR Proximity:

- a. Mendeteksi objek yang berada di depan kanan dan kiri dengan jarak tertentu yakni dengan jarak 30 – 50 cm dari robot.
- b. Ketika objek tidak terdeteksi maka sensor akan memberikan sinyal digital bernilai 1 atau aktif high.
- c. Ketika objek terdeteksi maka sensor akan mengeluarkan sinyal digital bernilai 0 atau aktif low.
- d. Sinyal digital yang dikirim masuk lewat Pin 7 Arduino DUE untuk sensor IR Proximity di bagian kanan dan Pin 6 untuk sensor IR Proximity di bagian kiri.

3. Switch “Mode”:

- a. Sebagai inputan untuk menentukan arah robot dalam 2 mode yakni “Maju” dan “Balik”.
- b. Inputan akan memberikan mode “Maju” ketika switch dalam keadaan On sehingga mengirim sinyal 0 atau aktif low ke Arduino DUE.
- c. Inputan akan memberikan mode “Balik” ketika switch dalam keadaan Off sehingga mengirim sinyal 1 atau aktif high ke Arduino DUE.
- d. Sinyal digital yang dikirim Switch ”Mode” masuk lewat Pin A8 Arduino DUE.

4. Push Button “Next Trip”:

- a. Untuk memberikan aksi *next trip* pada robot jika mempunyai tujuan 2 meja dan telah sampai pada destinasi meja pertama, dengan memberikan sinyal 0 atau aktif low ketika terjadi penekanan tombol.

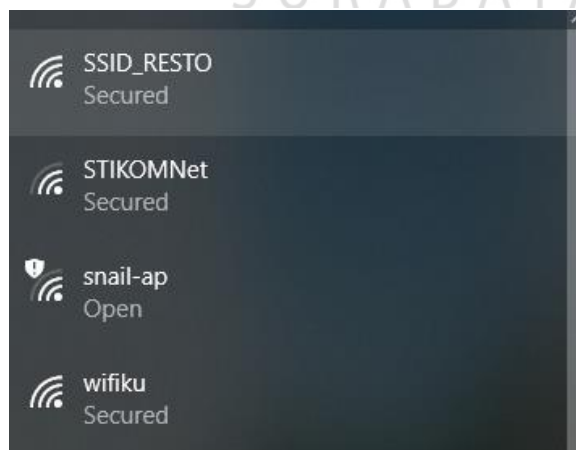
- b. Dan memberikan sinyal 1 atau aktif high ketika tidak terjadi penekanan tombol.
- c. Sinyal digital yang dikirim masuk lewat Pin A0 Arduino DUE.

5. Keypad Arduino:

- a. Sebagai inputan pada Mikrokontroller untuk menunjukkan angka meja yang akan dituju, dengan batasan meja yang disediakan yakni nomor 1, 2, 3, dan 4.
- b. Sinyal digital dari Keypad Arduino masuk lewat Pin menyesuaikan *datasheet* baris dan kolomnya. Untuk baris masuk pada Pin 52, 50, 48, 46, sedangkan untuk kolom masuk pada Pin 53, 51, 49, 47.

b. Pengolahan Citra (Raspberry Pi 3)

1. Pertama menyalakan Raspberry Pi 3 sebelumnya, kemudian pada PC user hubungkan Wifi pada *Access Point* seperti pada Gambar 3.2 yang digunakan untuk koneksi dengan Raspberry Pi 3. Dalam hal ini nama SSID yang digunakan pada *access point* adalah “SSID_RESTO” dengan IP Address 192.168.20.1.



Gambar 3.2 SSID Access Point

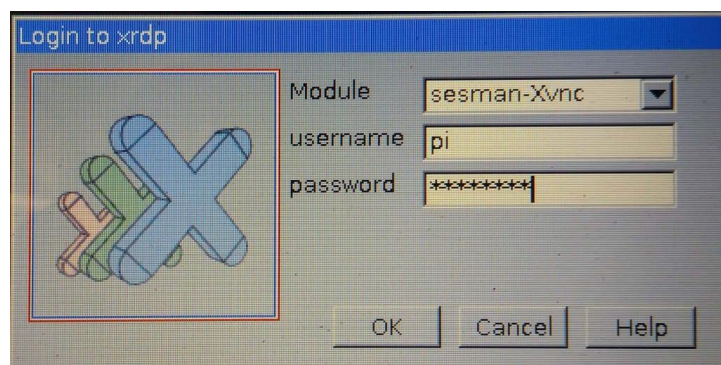
2. Kedua, setelah terkoneksi dengan SSID yang telah ditentukan maka langkah selanjutnya meremote Raspberry Pi 3 dengan menggunakan aplikasi Remote Desktop Connection. Pada textbar Computer tuliskan alamat IP dari Raspberry Pi 3 yang akan diremote (sebelumnya pada Raspberry Pi 3 dibuatkan IP *static* untuk memudahkan akses). IP Address yang digunakan Raspberry Pi 3 yakni 192.168.20.20.



Gambar 3.3 Tampilan Remote Desktop Connection

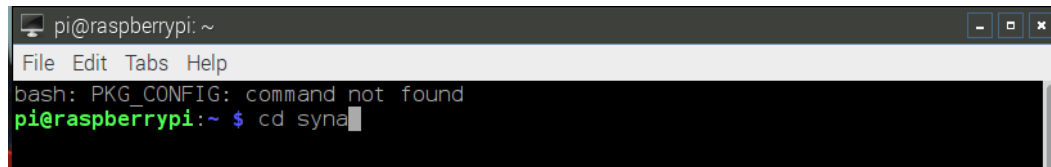
Lalu mengeklik Connect untuk masuk pada layar Raspberry Pi 3.

3. Ketiga, sebelum akses Raspberry Pi 3 akan dimintakan username dan password seperti pada gambar 3.4. Untuk Raspberry Pi 3 yang digunakan mempunyai username: “pi” dan password: “4Ndaanda”.



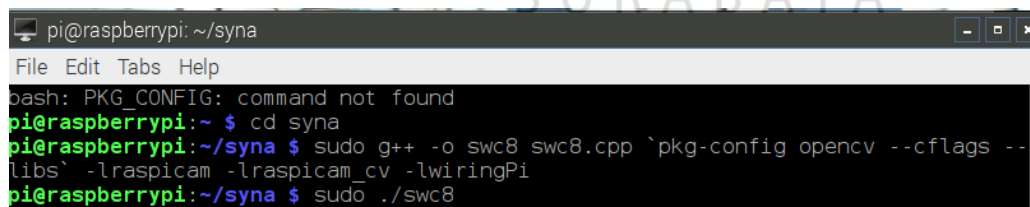
Gambar 3.4 Tampilan Login ke Raspberry Pi 3

4. Keempat, setelah masuk pada tampilan Raspberry Pi 3. Membuka aplikasi Terminal dengan cara mengeklik dua kali pada icon aplikasi Terminal atau klik kanan kemudian pilih “Open Terminal”. Lalu user mengetikkan ‘`cd syna`’ untuk masuk pada folder yang ada di penyimpanan Raspberry Pi 3 sebagaimana pada Gambar 3.5. Folder yang dipakai bernama “syna”.



Gambar 3.5 Tampilan Aplikasi Terminal bagian Desktop

5. Kelima setelah masuk pada folder syna, user mengetikkan “`sudo g++ -o swc8 swc8.cpp `pkg-config opencv --cflags --libs` -lraspicam -lraspicam_cv -lwiringPi`” untuk meng-*compile* file `swc8.cpp`, setelah proses *compile* selesai user mengetikkan “`sudo ./swc8`” untuk menjalankan program OpenCV yang telah dibuat di file `swc8.cpp` yang digambarkan pada Gambar 3.6.



Gambar 3.6 Tampilan Aplikasi Terminal bagian Folder syna

Kemudian setelah menjalankan program OpenCV, data output dari kamera berupa citra asli yang dilihat kemudian diolah dengan beberapa metode untuk mendapatkan nilai koordinat garis yang ditangkap kamera. Nilai

koordinat garis yang didapatkan dan karakter abjad untuk warna yang berbeda dikirim melalui komunikasi serial ke Arduino DUE melalui PORT_USB_Programming.

c. Pengolahan Data (Arduino DUE)

- 1 Arduino DUE menerima data output dari Raspberry Pi 3 berupa nilai koordinat objek dan nilai dari karakter abjad yang dikirim melalui komunikasi serial.
- 2 Untuk sensor IR Proximity mengirimkan sinyal digital sebagai inputan Mikrokontroller.
- 3 Jika output sensor IR Proximity baik sensor yang diletakkan di kanan atau di kiri bernilai 0 atau aktif low maka mikrokontroller akan menghentikan pergerakan Motor DC dengan memberikan nilai 0 pada PWM di Motor Driver.
- 4 Jika bernilai 1 atau aktif high maka mikrokontroller akan menggerakkan Motor DC dengan memberikan nilai > 0 pada PWM di Motor Driver sebagaimana sesuai pada program *Maze Mapping* yang dibuat.
- 5 Menerima data inputan dari Keypad Arduino untuk menentukan tujuan robot berjalan ke posisi meja yang telah ditentukan.
- 6 Arduino DUE juga menerima data inputan dari Switch “Mode” untuk menentukan mode “Maju” atau “Balik” dan Push Button “Next Trip” untuk melakukan aksi *next trip* yang difungsikan pada program *Maze Mapping*.
- 7 Setelah menerima nilai koordinat dari Raspberry Pi 3 dan nomor meja yang telah diinputkan serta mode yang telah ditentukan, maka robot akan

menjalankan program *Maze Mapping* yang telah dibuat.

- 8 Menerima Input RESET dari Push Button kuning jika terjadi error (*emergency*) atau akan menginputkan kembali meja yang akan dituju.

d. Output

1. Motor Driver EMS 30A H-Bridge

- a. Menerima sinyal PWM yang diberikan oleh Arduino DUE.
- b. Ketika Motor Driver menerima sinyal PWM maka aktuator bergerak dengan kecepatan yang telah ditentukan pada program *Maze Mapping*.
- c. Ketika Motor Driver tidak menerima sinyal PWM atau sinyal PWM=0, maka Aktuator berhenti bergerak.
- d. Pin yang digunakan untuk akses Motor Driver yakni Pin 5 untuk PWM Motor Driver kanan, Pin 2 untuk PWM Motor Driver kiri, Pin 22 dan 23 untuk Direksi Motor Driver kanan, dan Pin 24 dan 25 untuk Direksi Motor Driver kiri.

2. Motor DC PG-45 (Aktuator)

- a. Bergerak atau berputar jika Motor Driver menerima sinyal PWM dan menyesuaikan kecepatan yang diberikan.
- b. Dan tidak bergerak jika Motor Driver tidak menerima sinyal PWM atau sinyal PWM=0.
- c. Rotari internal (*rotary encoder*) pada Motor DC memberikan nilai pulse pada Arduino DUE untuk menghitung banyaknya putaran pada masing-masing *rotary encoder*.
- d. Pin yang digunakan untuk membaca nilai pulse dari *rotary encoder*

yakni Pin 44 untuk Channel Kanan Motor DC dan pin 45 untuk Channel Kiri Motor DC.

3. LCD 20x4

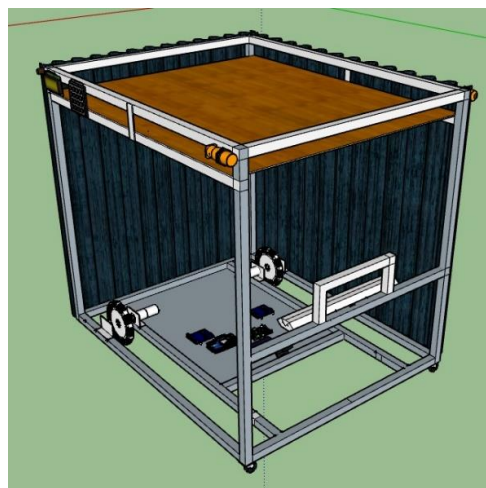
- a. Menampilkan nomor Meja 1, nomor Meja 2, status Camera dan Mode yang digunakan pada robot.
- b. Pin yang digunakan untuk konfigurasi LCD 20x4: RS=Pin A7, EN=Pin A6, D4=Pin A5, D5=Pin A4, D6=Pin A3, D7=Pin A2.

3.1.2 Desain Rancang Bangun Robot

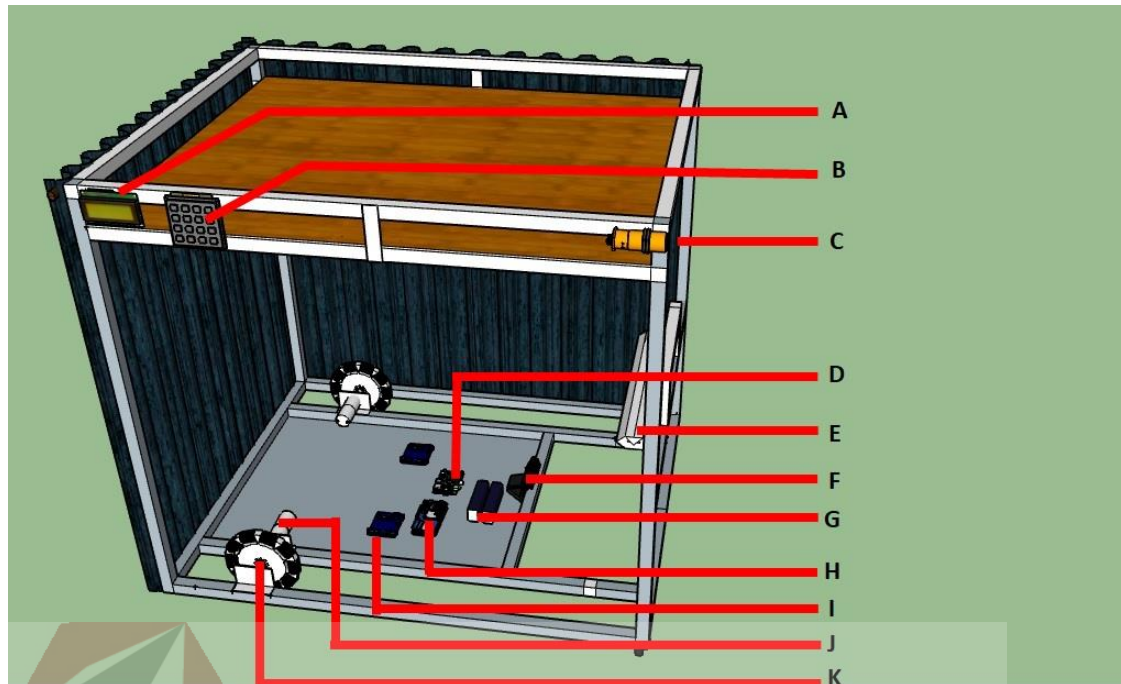
Berikut desain rancang bangun robot yang digunakan pada Tugas Akhir, pada Gambar 3.7 memperlihatkan robot dari depan, dan Gambar 3.8 memperlihatkan robot dari samping serta tata letak komponen dan keterangan komponen yang digunakan.

Tabel 3.1 Ukuran Desain Rancang Bangun

Panjang Robot	:	80 cm
Lebar Robot	:	60 cm
Tinggi Robot	:	70 cm



Gambar 3.7 Desain Robot Tampak Depan



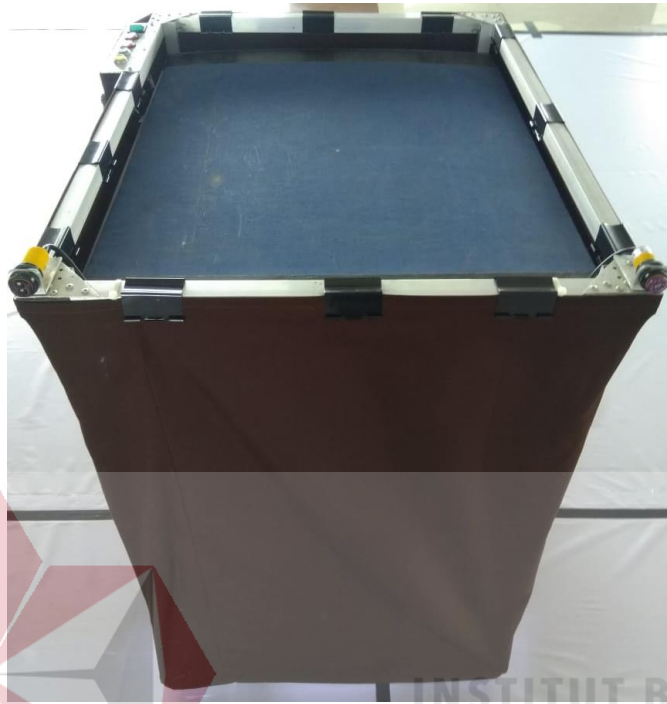
Gambar 3.8 Desain Robot dari Samping

Tabel 3.2 Nama-nama Komponen

Objek yang Ditunjuk	Keterangan
A	LCD 20x4
B	Keypad Arduino
C	Sensor IR Proximity
D	Raspberry Pi 3
E	Lampu
F	Kamera Webcam
G	Baterai 12V dan 24V
H	Arduino DUE
I	EMS 30A H-Bridge
J	Motor DC 24V
K	Roda

3.1.3 Hasil Jadi Rancang Bangun Robot

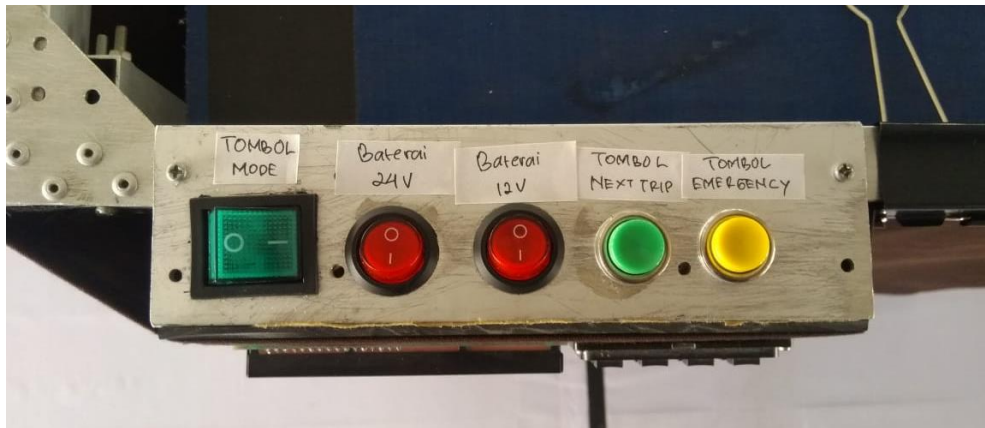
Berikut gambar hasil jadi Rancang Bangun Robot:



Gambar 3.9 Robot Tampak dari Depan



Gambar 3.10 Robot Tampak dari Samping



Gambar 3.11 Tombol Interface Robot

Keterangan:

1. Switch hijau (Tombol Mode) digunakan untuk menentukan robot bergerak menggunakan mode “Maju” atau “Balik”.
2. Switch merah (Tombol Baterai 12/24 V) digunakan untuk menyalakan robot.
3. Push Button hijau (Tombol Next Trip) digunakan untuk memberikan robot aksi melanjutkan destinasi meja berikutnya sesuai dengan input.
4. Push Button kuning (Tombol Emergency/RESET) digunakan untuk mereset robot ketika dalam keadaan error atau memulai kembali proses pengantaran makanan.



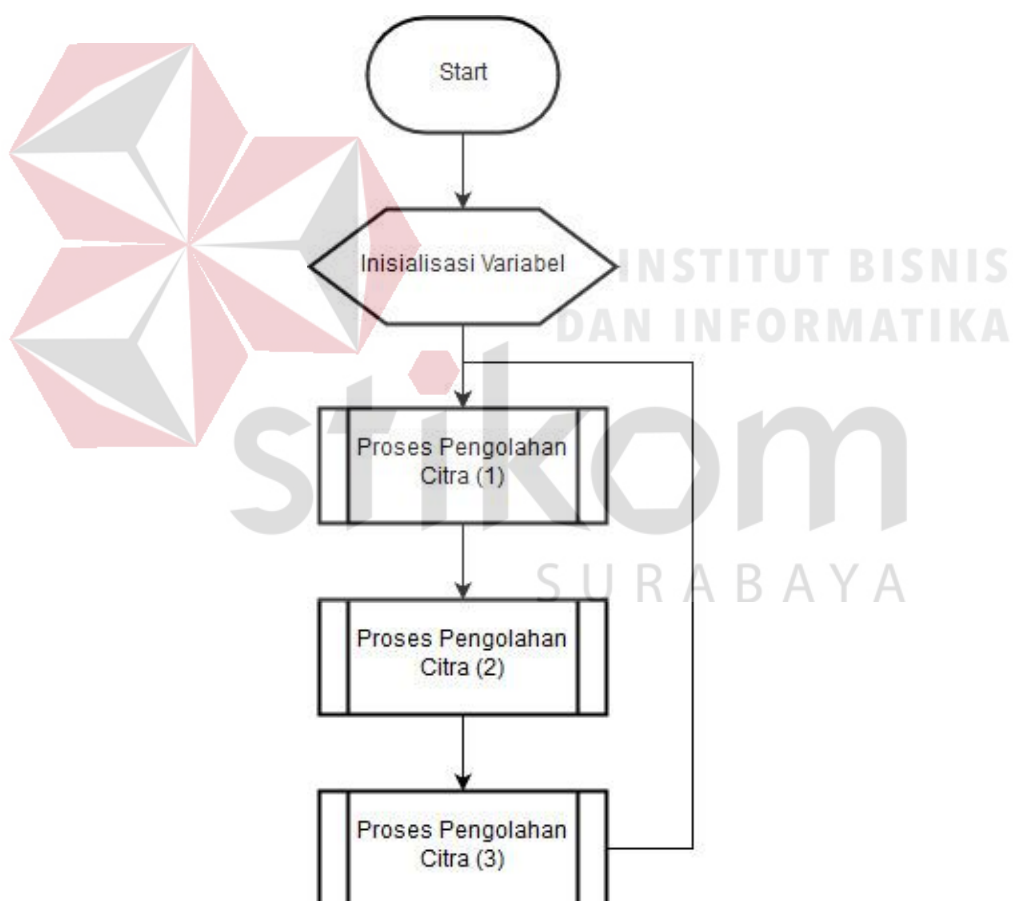
Gambar 3.12 Tampilan Display LCD dan Keypad Arduino

3.2 Perancangan Perangkat Lunak (Program)

Pada bagian ini menjelaskan jalan program seperti flowchart dan penjelasan masing-masing list program yang dibuat, berikut penjelasannya:

3.2.1 Pengolahan Citra pada Raspberry Pi 3

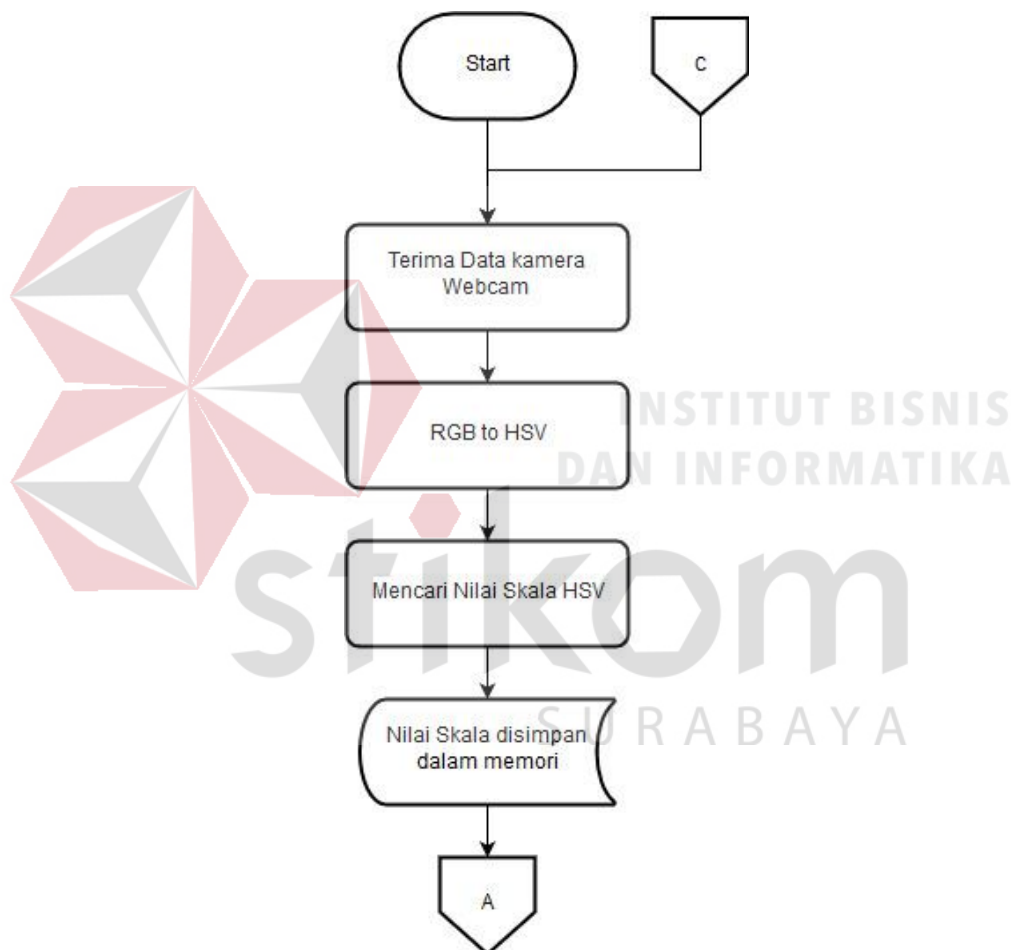
Pada pembahasan materi disini akan menjelaskan jalannya program untuk pengolahan citra pada Raspberry Pi 3.



Gambar 3.13 Flowchart Keseluruhan Raspberry Pi 3

Pada Gambar 3.13, flowchart yang digunakan sangatlah detail dan rinci sehingga perlu penjabaran pada tiap simbol-simbol flowchart yang digunakan.

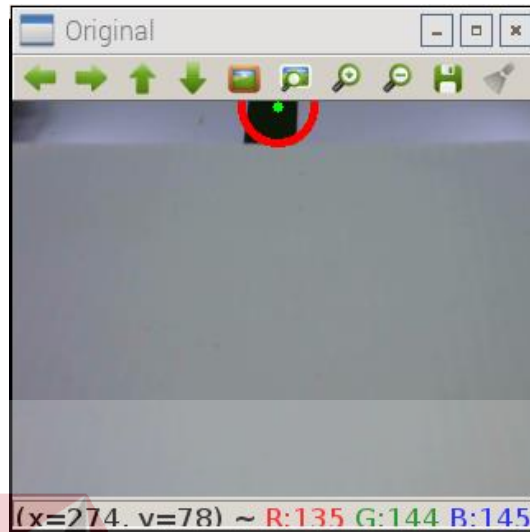
Berikut penjelasan masing-masing simbol flowchart sesuai dengan urutannya, pada awal mula ketika Raspberry Pi 3 menyala maka akan memulai inisialisasi variabel lalu kemudian dilanjutkan ke program Proses Pengolahan Citra (1) pada Gambar 3.14, lalu setelah itu dilanjutkan ke program Proses Pengolahan Citra (2) pada Gambar 3.17, dan kemudian ke program Proses Pengolahan Citra (3) pada Gambar 3.25.



Gambar 3.14 Flowchart Pengolahan Citra Bagian 1

Pada Gambar 3.14 setelah melakukan inisialisasi variabel, Raspberry Pi 3 menerima data kamera berupa citra dalam format RGB seperti pada Gambar 3.15, Gambar 3.16, dan Gambar 3.17. Ukuran resolusi yang ditangkap yakni 320x240

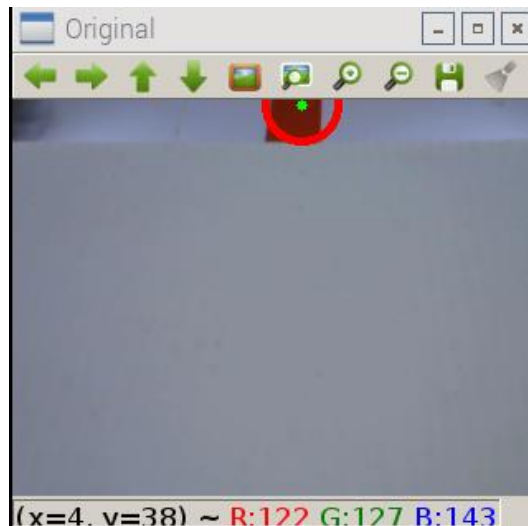
dengan FPS (*frame per second*) sebesar 100 untuk mengurangi beban kinerja Raspberry Pi 3 sehingga gambar yang didapatkan rinci, detail, dan halus serta pengiriman data melalui komunikasi serial ke Arduino lebih cepat.



Gambar 3.15 Citra RGB Garis Warna Hitam



Gambar 3.16 Citra RGB Garis Warna Kuning





Gambar 3.17 Citra RGB Garis Warna Merah

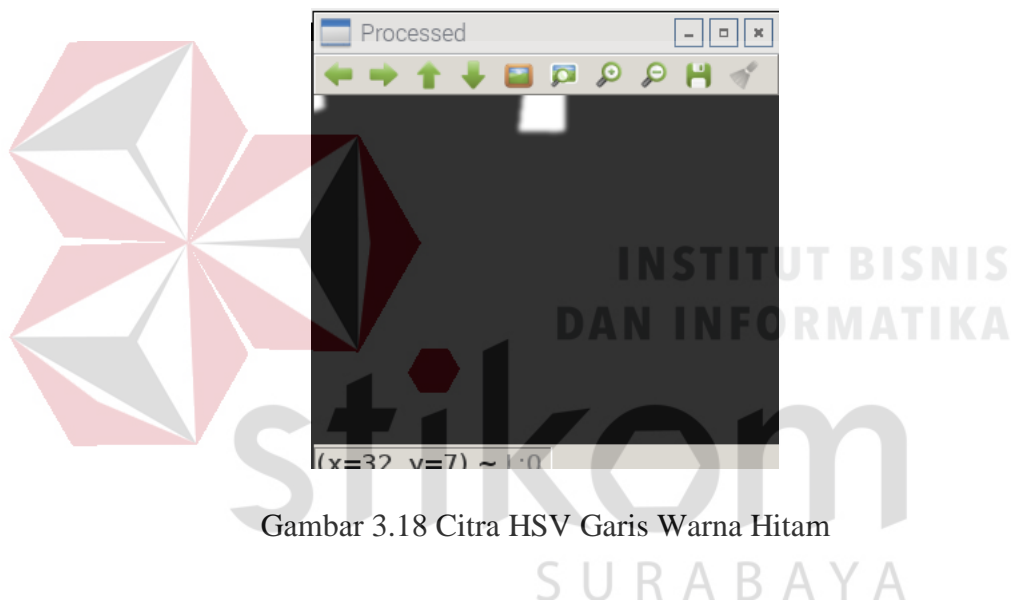
Citra yang telah ditangkap dikonversi dari format citra RGB ke HSV. Setelah dikonversi ke dalam format citra HSV, dicari nilai HSV-nya dalam bentuk skala sesuai dengan panduan warna yang telah disediakan. Berikut panduan nilai HSV pada masing-masing warna pada Tabel 3.3 diantaranya:

Tabel 3.3 Nilai HSV Tiap Warna

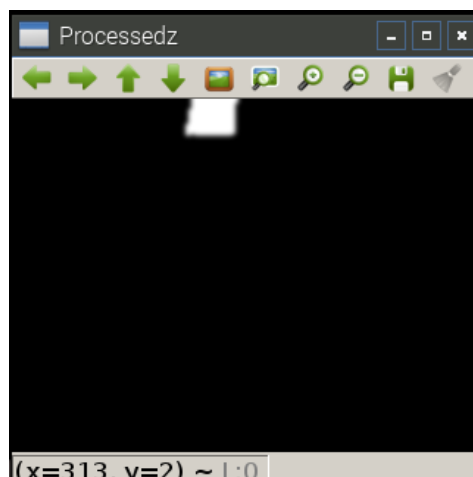
Nama	Contoh	Kode Warna	RGB			HSV		
			R	G	B	H	S	V
Abu-abu		#808080	128	128	128	0	0	50
Biru		#0000FF	0	0	255	240	100	100
Biru Laut		#000080	0	0	128	240	100	50
Coklat		#964B00	150	75	0	30	100	59
Emas		#FFD700	255	215	0	51	100	100
Hijau		#00FF00	0	255	0	120	100	100
Hitam		#000000	0	0	0	0	0	0
Kuning		#FFFF00	255	255	0	60	100	100
Magenta		#FF00FF	255	0	255	300	100	100
Mawar		#FF007F	255	0	127	330	100	100
Merah		#FF0000	255	0	0	0	100	100

Nama	Contoh	Kode Warna	RGB			HSV		
			R	G	B	H	S	V
Merah Marun		#800000	128	0	0	0	100	50
Merah Jambu		#FFC0CB	255	192	203	350	25	100

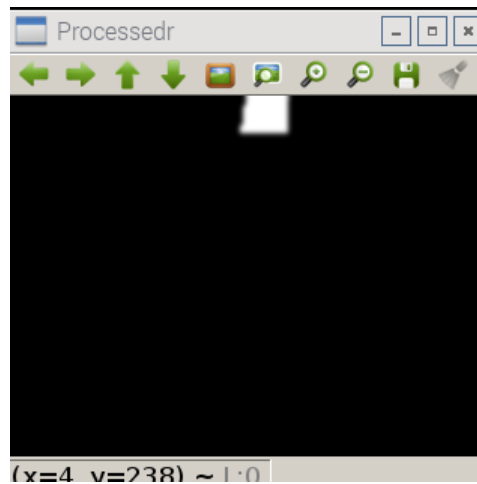
Setelah mendapatkan hasil dari skala HSV nya maka secara otomatis warna yang masuk di nilai skala akan menjadi warna putih, selain dari warna yang masuk dalam skala akan menjadi warna hitam sebagaimana pada Gambar 3.18, Gambar 3.19 dan Gambar 3.20.



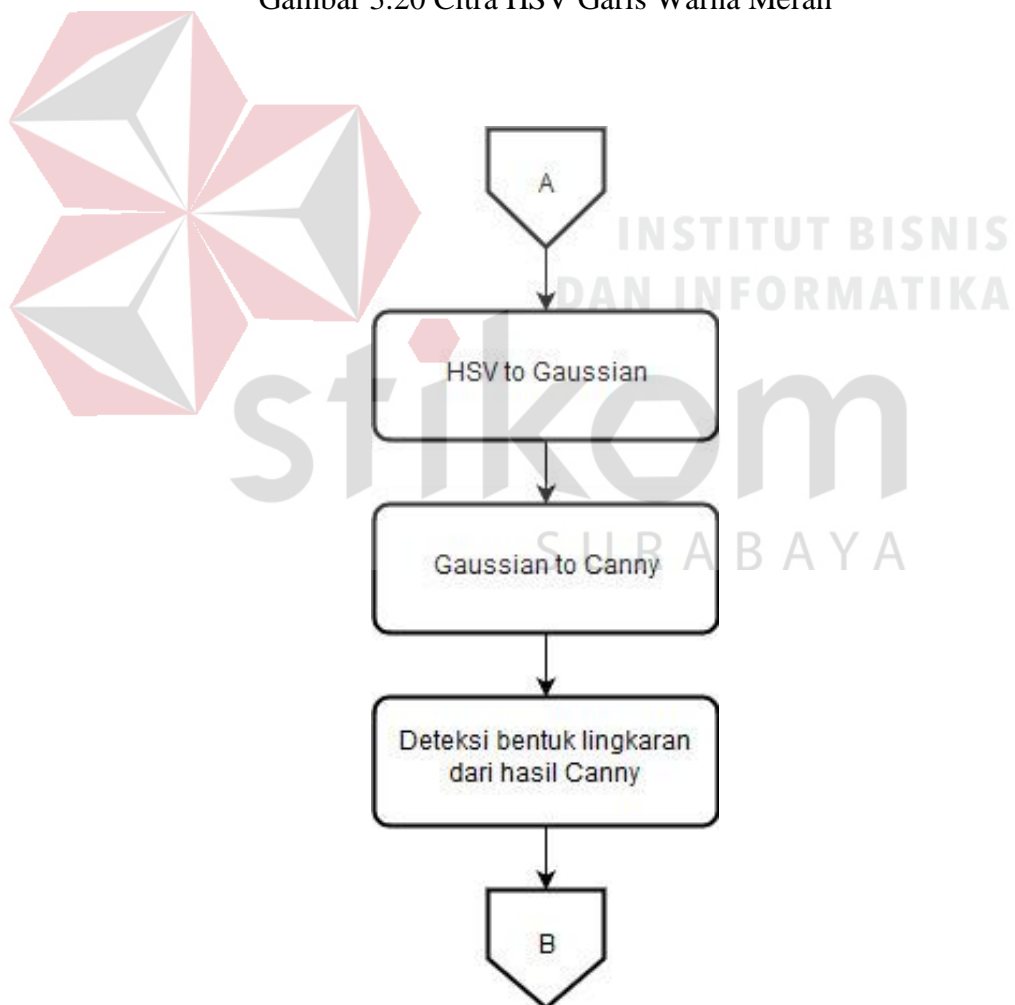
Gambar 3.18 Citra HSV Garis Warna Hitam



Gambar 3.19 Citra HSV Garis Warna Kuning

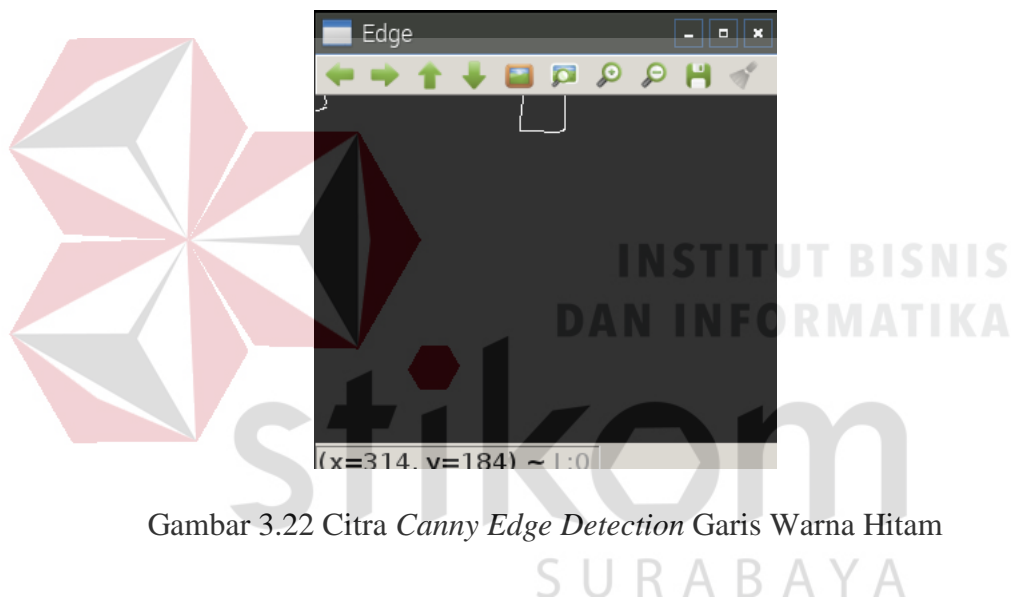


Gambar 3.20 Citra HSV Garis Warna Merah

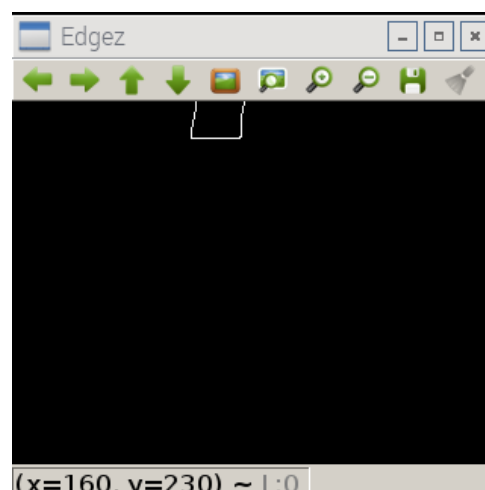


Gambar 3.21 Flowchart Pengolahan Citra Bagian 2

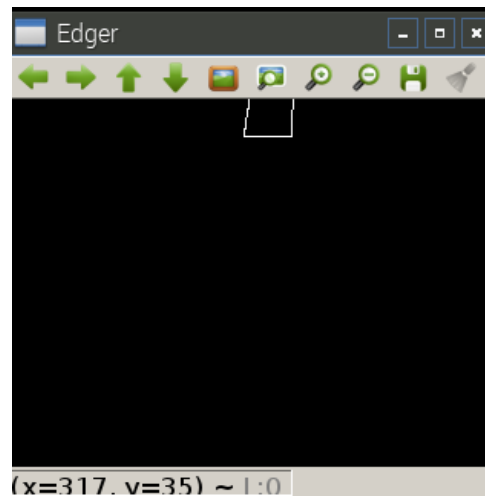
Pada Gambar 3.21 citra yang telah disimpan dikonversi lagi untuk menghilangkan *noise* dengan cara meng-blurkannya menggunakan metode *Gaussian-blur* untuk mendapatkan hasil citra yang bersih dengan *noise* yang sedikit. Hasil citra yang di-blurkan selanjutnya dideteksi tepi garisnya dengan metode *Canny Edge Detection*, penggunaan *Canny* untuk mendeteksi tepi garis sangatlah detail dan rinci sehingga citra sesuai dengan apa yang ditangkap oleh kamera, hasil *Canny* dapat dilihat seperti pada Gambar 3.22, Gambar 3.23, dan Gambar 3.24.



Gambar 3.22 Citra *Canny Edge Detection* Garis Warna Hitam

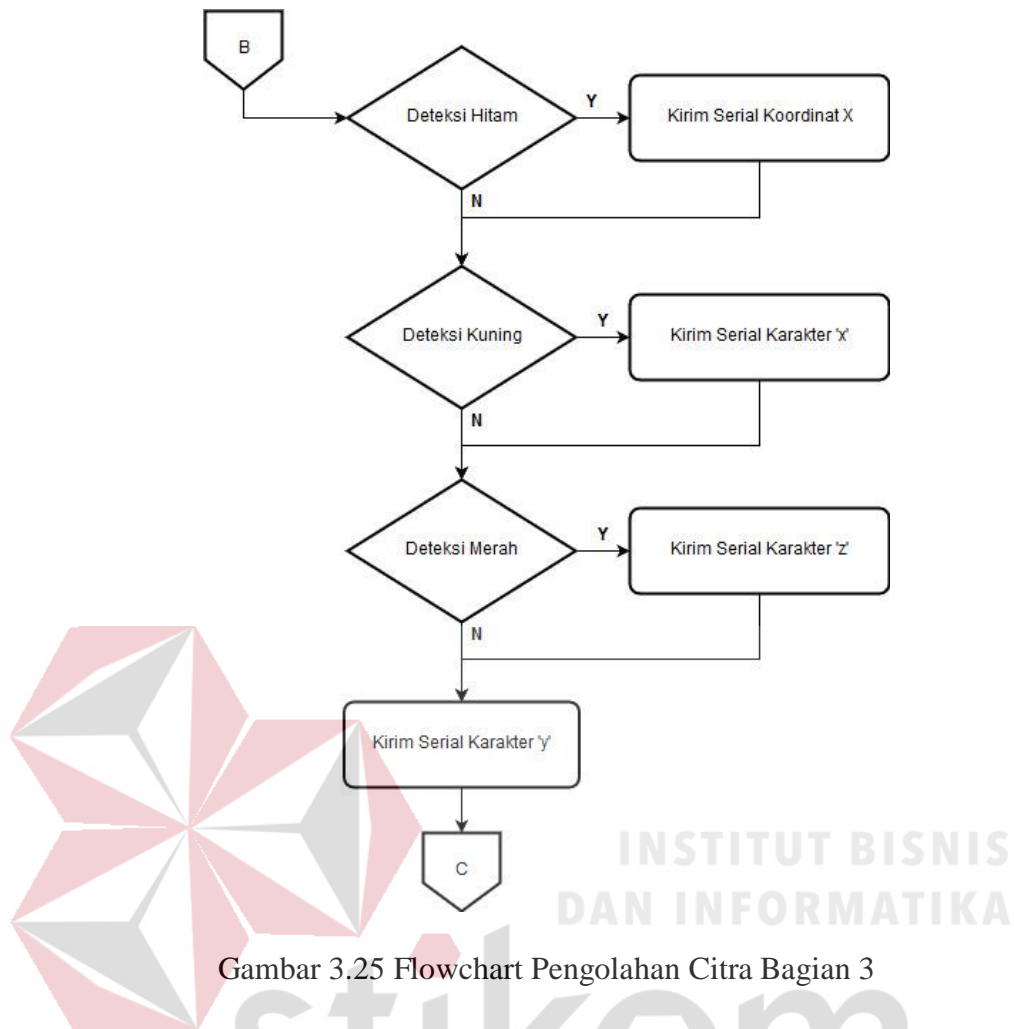


Gambar 3.23 Citra *Canny Edge Detection* Garis Warna Kuning



Gambar 3.24 Citra *Canny Edge Detection* Garis Warna Merah

Setelah mendapatkan tepi garisnya, citra selanjutnya dicari bentuk lingkarannya untuk menandakan objek berdasarkan warna HSV yang telah ditentukan sebelumnya dan dibatasi pembacaannya dengan hasil citra *Edge Detection*, dengan metode *Hough Circle Transform* didapatkan bentuk lingkaran dari citra yang ditangkap, hasil bentuk lingkaran dapat dilihat pada Gambar 3.15, Gambar 3.16, dan Gambar 3.17.



Gambar 3.25 Flowchart Pengolahan Citra Bagian 3

Pada Gambar 3.25, setelah objek didapatkan maka akan dipilah sesuai dengan warna HSV objek yang dideteksi, pada percabangan pertama kondisi jika warna yang dideteksi adalah warna hitam. Jika IYA mendeteksi Warna Hitam maka objek tersebut diambil nilai koordinatnya X-nya kemudian dikirim seperti yang dicontohkan pada Gambar 3.26 melalui komunikasi serial ke Arduino DUE, sedangkan jika TIDAK mendeteksi Warna Hitam maka akan dilanjutkan ke proses percabangan selanjutnya yakni mendeteksi Warna Kuning.


```

pi@raspberrypi: ~/syna
File Edit Tabs Help
black x = 161.000000, y = 11.000000, r = 13.038404
black x = 163.000000, y = 7.000000, r = 19.697716
black x = 161.000000, y = 7.000000, r = 19.416489
black x = 163.000000, y = 11.000000, r = 13.416408
black x = 163.000000, y = 11.000000, r = 13.416408
black x = 161.000000, y = 3.000000, r = 21.540659
black x = 163.000000, y = 7.000000, r = 19.697716
black x = 161.000000, y = 7.000000, r = 19.416489
black x = 163.000000, y = 11.000000, r = 13.416408
black x = 163.000000, y = 11.000000, r = 13.416408
black x = 159.000000, y = 11.000000, r = 14.866069
black x = 159.000000, y = 11.000000, r = 15.033297
black x = 161.000000, y = 11.000000, r = 13.038404
black x = 163.000000, y = 11.000000, r = 13.416408
black x = 163.000000, y = 11.000000, r = 13.416408
black x = 161.000000, y = 11.000000, r = 13.038404

```

Gambar 3.26 Nilai Koordinat X,Y, dan Jari-Jari Warna Hitam

Jika IYA mendeteksi Warna Kuning maka objek tersebut diambil nilai koordinatnya seperti pada Gambar 3.27, kemudian kirim karakter 'x' melalui komunikasi serial ke Arduino DUE, sedangkan jika TIDAK mendeteksi Warna Kuning maka akan dilanjutkan ke proses percabangan selanjutnya yakni mendeteksi Warna Merah.

```

pi@raspberrypi: ~/syna
File Edit Tabs Help
yellow x = 143.000000, y = 15.000000, r = 10.000000
yellow x = 127.000000, y = 15.000000, r = 10.049875
yellow x = 131.000000, y = 5.000000, r = 20.591261
yellow x = 127.000000, y = 15.000000, r = 10.049875
yellow x = 143.000000, y = 15.000000, r = 10.000000
yellow x = 143.000000, y = 15.000000, r = 10.000000
yellow x = 143.000000, y = 15.000000, r = 10.000000
yellow x = 139.000000, y = 3.000000, r = 21.931713
yellow x = 141.000000, y = 13.000000, r = 12.083046
yellow x = 143.000000, y = 15.000000, r = 10.000000
yellow x = 143.000000, y = 15.000000, r = 10.000000
yellow x = 127.000000, y = 15.000000, r = 10.049875
yellow x = 127.000000, y = 15.000000, r = 10.049875
yellow x = 127.000000, y = 15.000000, r = 10.049875
yellow x = 139.000000, y = 3.000000, r = 21.931713

```

Gambar 3.27 Nilai Koordinat X,Y, dan Jari-Jari Warna Kuning

Jika IYA mendeteksi Warna Merah maka objek tersebut diambil nilai koordinatnya seperti pada Gambar 3.28, kemudian kirim karakter 'z' melalui komunikasi serial ke Arduino DUE, sedangkan jika TIDAK mendeteksi Warna Merah maka kirim karakter 'y' ke Arduino DUE. Lalu kembali menjalankan program dari awal.

```

pi@raspberrypi: ~/syna
File Edit Tabs Help
red x = 175.000000, y = 17.000000, r = 9.219544
red x = 163.000000, y = 15.000000, r = 10.000000
red x = 173.000000, y = 3.000000, r = 21.931713
red x = 169.000000, y = 9.000000, r = 14.866069
red x = 163.000000, y = 15.000000, r = 10.000000
red x = 169.000000, y = 9.000000, r = 14.764823
red x = 163.000000, y = 15.000000, r = 10.000000
red x = 169.000000, y = 17.000000, r = 15.297058
red x = 167.000000, y = 5.000000, r = 20.396078
red x = 163.000000, y = 15.000000, r = 10.000000
red x = 169.000000, y = 9.000000, r = 14.866069
red x = 165.000000, y = 3.000000, r = 20.615528
red x = 175.000000, y = 17.000000, r = 9.219544
red x = 163.000000, y = 15.000000, r = 10.000000
red x = 163.000000, y = 15.000000, r = 10.000000
red x = 163.000000, y = 15.000000, r = 10.000000
red x = 163.000000, y = 15.000000, r = 10.000000
red x = 173.000000, y = 3.000000, r = 21.931713
red x = 163.000000, y = 15.000000, r = 10.000000
red x = 169.000000, y = 9.000000, r = 14.764823
red x = 169.000000, y = 9.000000, r = 14.764823
red x = 175.000000, y = 3.000000, r = 21.540659
red x = 167.000000, y = 5.000000, r = 20.396078
red x = 163.000000, y = 15.000000, r = 10.000000
red x = 175.000000, y = 17.000000, r = 9.219544
red x = 173.000000, y = 3.000000, r = 21.931713
red x = 175.000000, y = 17.000000, r = 9.219544
red x = 169.000000, y = 9.000000, r = 16.763054
red x = 175.000000, y = 3.000000, r = 21.540659

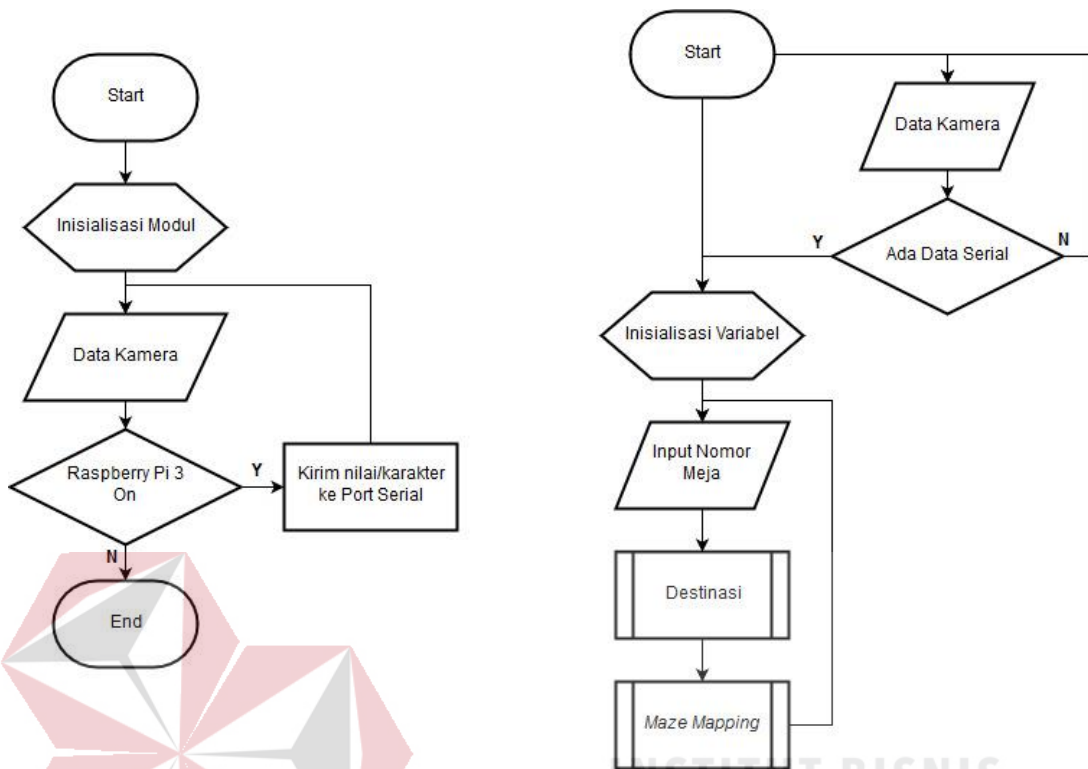
```

Gambar 3.28 Nilai Koordinat X,Y, dan Jari-Jari Warna Merah

3.2.2 Pengolahan Data pada Arduino DUE

Pada pembahasan materi disini akan menjelaskan program untuk pengolahan data Arduino DUE yang akan dibagi menjadi tiga bagian yakni flowchart program utama, flowchart program cabang, dan flowchart sub-cabang.

1. Flowchart Program Utama



Gambar 3.29 Flowchart Program Utama Arduino DUE

Pada Gambar 3.29 digambarkan terdapat dua program yang berjalan yakni Program untuk Raspberry Pi 3 yang dijelaskan secara umum (penjelasan rinci telah dijelaskan sebelumnya di bagian Pengolahan Citra) dan Program utama Arduino DUE yang juga dijelaskan secara umum. Berikut penjelasan dari program diatas:

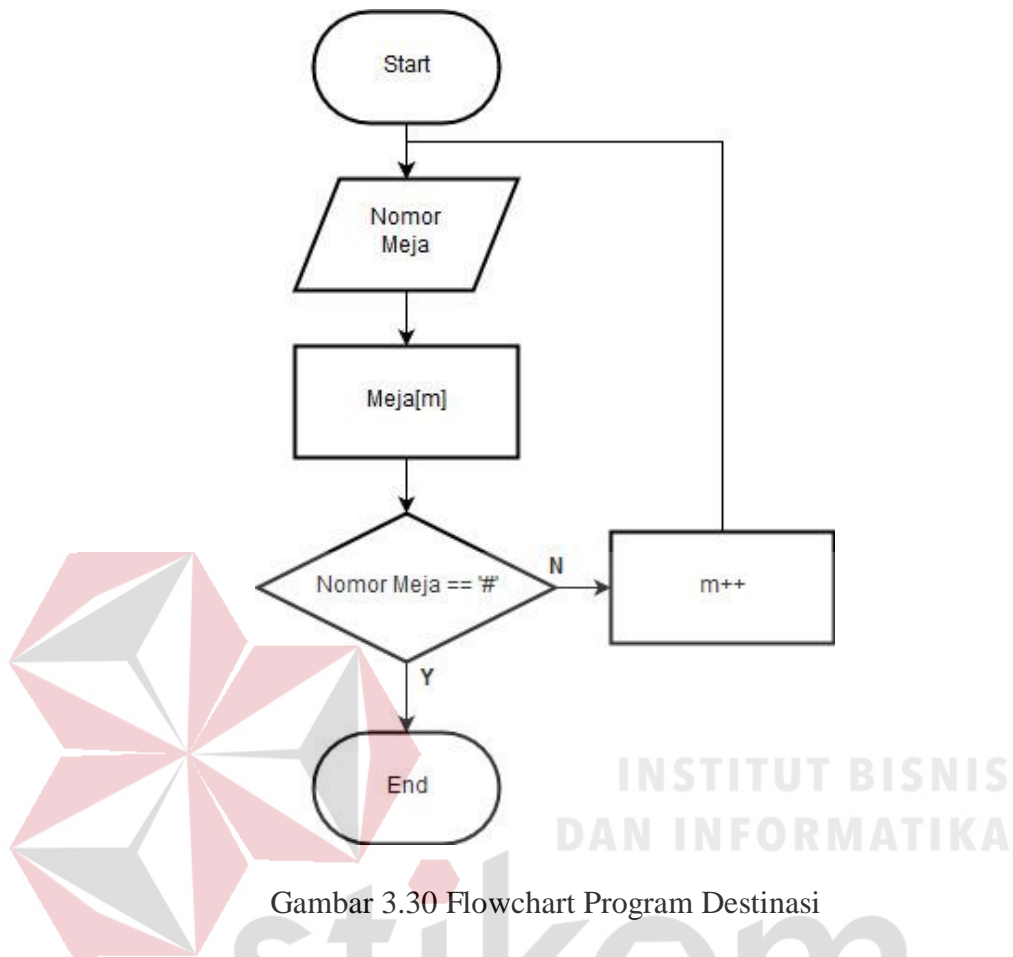
- Pertama pada Raspberry Pi 3 memulai inisialisasi modul, lalu kamera menangkap objek.
- Disisi lain pada program utama Arduino DUE, ketika Arduino DUE menyala maka akan memulai inisialisasi variabel dan menunggu terima data berupa nilai/karakter dari Raspberry Pi 3.
- Hasil tangkapan kamera dikirimkan di Raspberry Pi 3.

- d. Ketika Raspberry Pi 3 dalam keadaan menyala maka akan mengirimkan nilai koordinat garis atau karakter, sedangkan
- e. Ketika tidak dalam keadaan menyala maka Raspberry Pi 3 langsung mengakhiri programnya.
- f. Selanjutnya Arduino DUE menerima nilai koordinat atau karakter yang dikirimkan secara serial oleh Raspberry Pi 3.
- g. Lalu menginputkan nomor meja yang akan dituju dan masuk pada program cabang Destinasi.
- h. Selesai menjalankan program Destinasi, dilanjutkan menjalankan program cabang *Maze Mapping* dan apabila telah selesai menyelesaikan programnya, maka akan kembali melanjutkan atau meneruskan dari awal dan kembali menginputkan nomor meja.

2. Flowchart Program Cabang

Program cabang merupakan program yang dijalankan di program utama Arduino DUE. Terdiri dari dua program, yakni program Destinasi dan *Maze Mapping*. Program Destinasi merupakan program yang digunakan untuk menentukan nomor meja yang akan dituju, sedangkan program *Maze Mapping* adalah program yang digunakan untuk menentukan jalur atau rute berdasarkan *return* atau jalur yang telah ditentukan dan disepakati bersama.

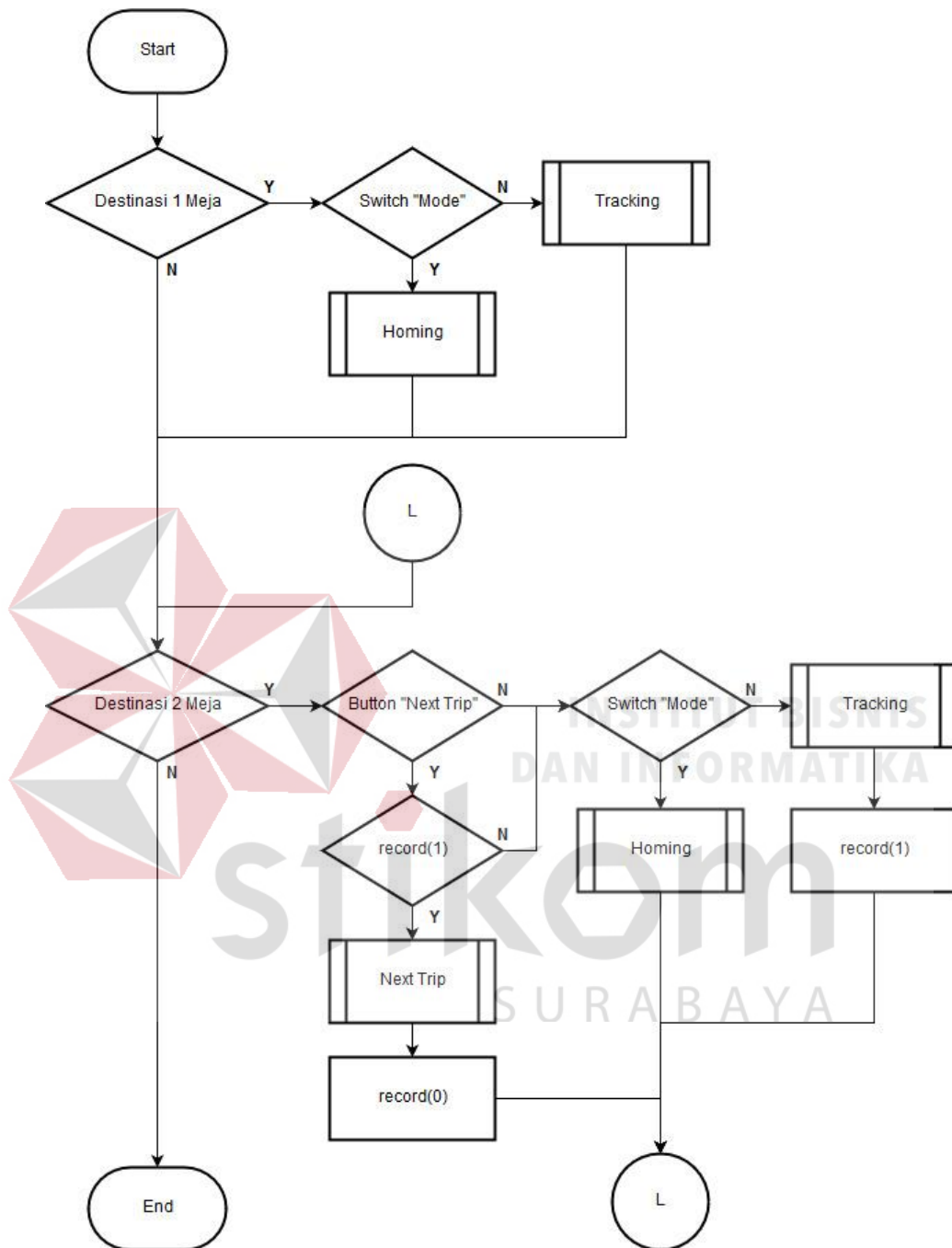
a. Program Destinasi



Gambar 3.30 Flowchart Program Destinasi

Berikut penjelasan dari Gambar 3.30, program dimulai dengan menginputkan nomor meja dari Keypad Arduino. Kemudian, meja yang diinputkan masuk pada variabel array yang kemudian di cek apakah input yang dimasukkan sama dengan karakter “#”, jika TIDAK maka variabel array bertambah satu, untuk input nomor selanjutnya. Jika IYA maka program akan langsung diakhiri dan menuju ke program selanjutnya.

b. Program *Maze Mapping*



Gambar 3.31 Flowchart Program *Maze Mapping*

Pada Gambar 3.31 *Maze Mapping* terdapat 2 kondisi yang dijalankan yakni, ketika menjalankan kondisi destinasi 1 meja dan

destinasi 2 meja. Jika kondisi pertama yakni destinasi 1 meja terpenuhi maka program akan menjalankan program sub-cabang Tracking bila tidak ada penekanan Switch “Mode”, jika terdapat penekanan maka akan menjalankan program sub-cabang Homing karena tujuan destinasi meja hanya satu. Jika yang terpenuhi yakni destinasi 2 meja maka program akan menjalankan program sub-cabang Tracking bila tidak ada penekanan Push Button “Next Trip” dan Switch “Mode”. Jika kondisi Tracking terpenuhi maka akan mengaktifkan nilai 1 atau HIGH sebuah variabel untuk menjalankan syarat program sub-cabang Next Trip untuk melanjutkan perjalanan robot dari meja pertama ke meja kedua, lalu setelah menjalankan program Next Trip maka variabel sebelumnya diberi nilai 0 atau LOW untuk mematikan fungsi Next Trip. Penekanan Switch “Mode” untuk menjalankan program Homing setelah mencapai destinasi meja kedua.

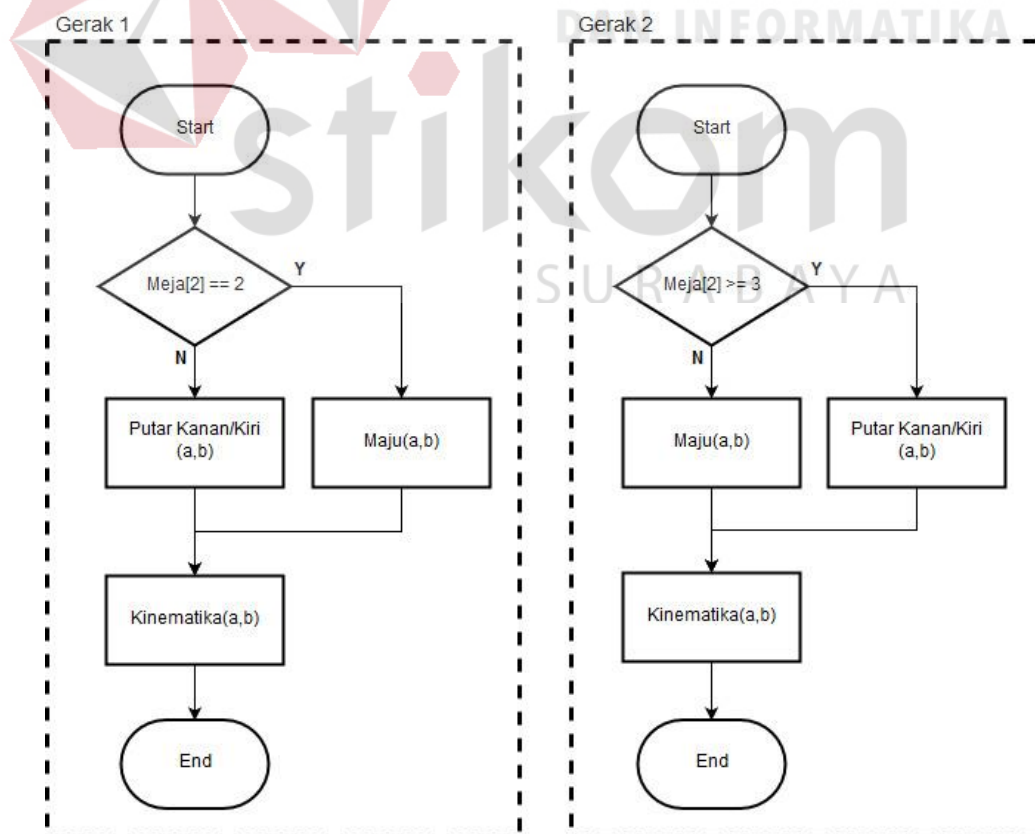
3. Flowchart Program Sub-Cabang

Program sub-cabang merupakan program yang dijalankan di program cabang, terutama pada program cabang *Maze Mapping* yang terdiri dari tiga program sub-cabang yakni Tracking, Next Trip, dan Homing. Masing-masing program sub-cabang mempunyai Flowchart yang sama, bisa dilihat pada Gambar 3.33. dalam flowchart yang dicantumkan terdapat empat kondisi, yaitu:

- (a.) kondisi pertama menerima nilai koordinat x atau bisa diartikan jika kamera mendeteksi adanya garis berwarna hitam maka program menjalankan perintah maju dan bergerak dengan kinematika menyesuaikan

paracentimeter yang telah dibuat.

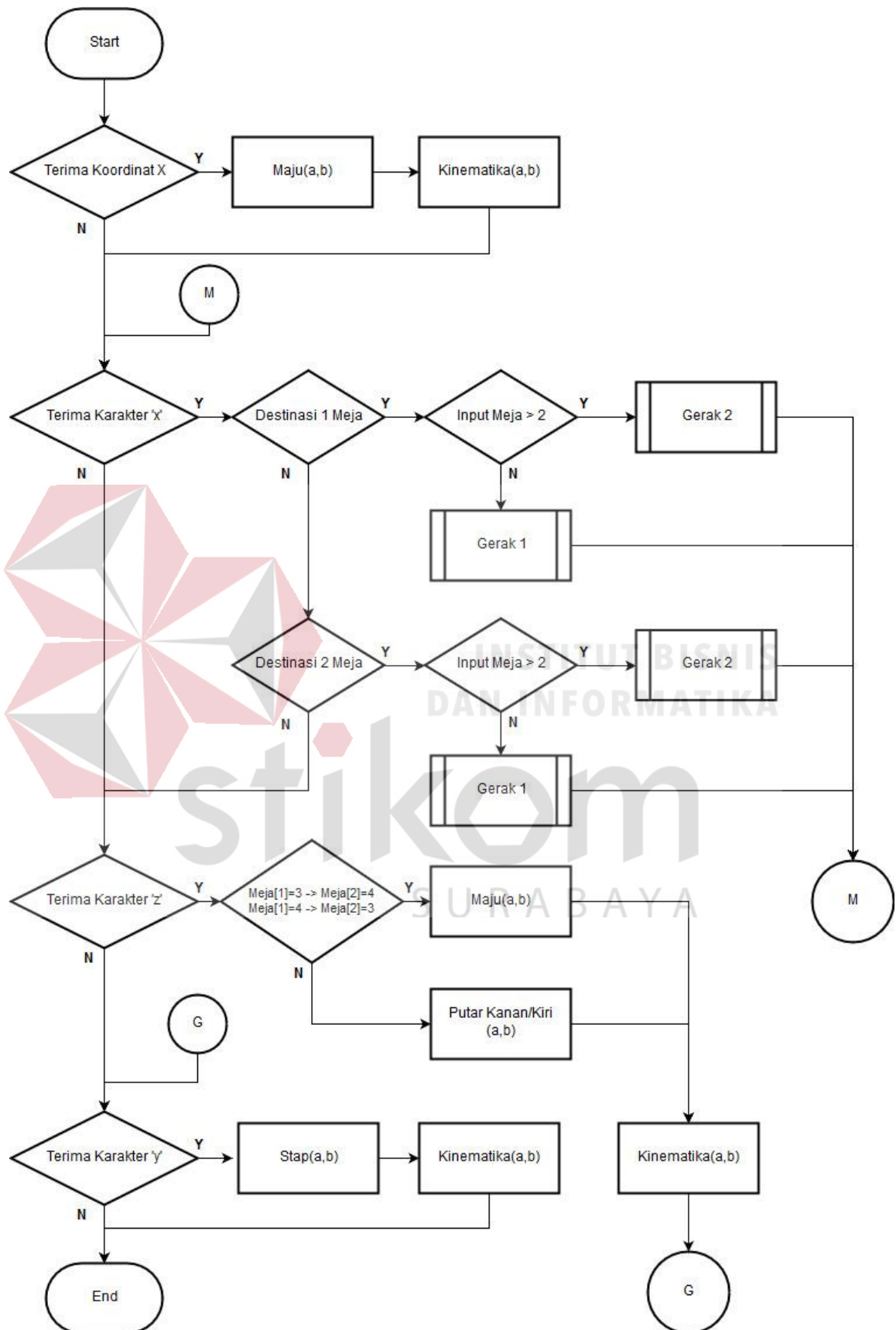
- (b.) kondisi kedua ketika menerima karakter 'x' atau kamera sedang mendeteksi garis berwarna kuning maka program akan menyesuaikan destinasi meja yang dipilih. Jika destinasi hanya satu meja saja maka akan menjalankan perintah sesuai input meja yang dimasukkan yakni: Jika meja yang diinputkan nomor 1 dan 2 maka akan menjalankan program Gerak 1 dan jika meja yang diinputkan diatas nomor 2 yakni nomor 3 atau 4, maka akan menjalankan program Gerak 2. Program Gerak 1 dan Gerak 2 menjalankan perintah sesuai kondisi (karena kondisi destinasi tidak mendapati variabel 'Meja[2]' diinputkan nilai maka akan menjalankan kondisi TIDAK pada masing-masing kondisi pada tiap program) dan parameter yang diinputkan. Penjelasan dapat dilihat pada Gambar 3.32.



Gambar 3.32 Flowchart Program Gerak

Sedangkan, jika destinasi meja yang dipilih dua maka akan menjalankan perintah sesuai dengan nomor inputan yang diberikan, Jika meja yang diinputkan nomor 1 atau 2 maka akan menjalankan program Gerak 1, dan jika meja yang diinputkan diatas nomor 2 yakni nomor 3 atau 4, maka akan menjalankan program Gerak 2. Program Gerak 1 dan Gerak 2 menjalankan perintah sesuai kondisi (karena kondisi destinasi mendapati variabel 'Meja[2]' diinputkan nilai maka akan menjalankan kondisi YA pada masing-masing kondisi pada tiap program) dan parameter yang diinputkan. Penjelasan dapat dilihat pada Gambar 3.32.

- (c.) Kondisi ketiga ketika menerima karakter 'z' atau kamera mendeteksi Warna Merah maka program akan menjalankan perintah sesuai kondisi, jika kondisi meja yang diinputkan TIDAK dalam keadaan dari meja 3 ke meja 4 atau sebaliknya, maka akan menjalankan Putar kanan/kiri sesuai parameter yang diinputkan. Jika kondisi meja yang diinputkan YA dalam keadaan dari meja 3 ke meja 4 atau sebaliknya maka akan menjalankan perintah Maju.
- (d.) kondisi keempat ketika menerima karakter 'y' atau kamera mendeteksi warna selain hitam, kuning dan merah maka program akan menjalankan perintah stap atau stop, lalu program selesai kembali ke program cabang *Maze Mapping* di Arduino DUE.

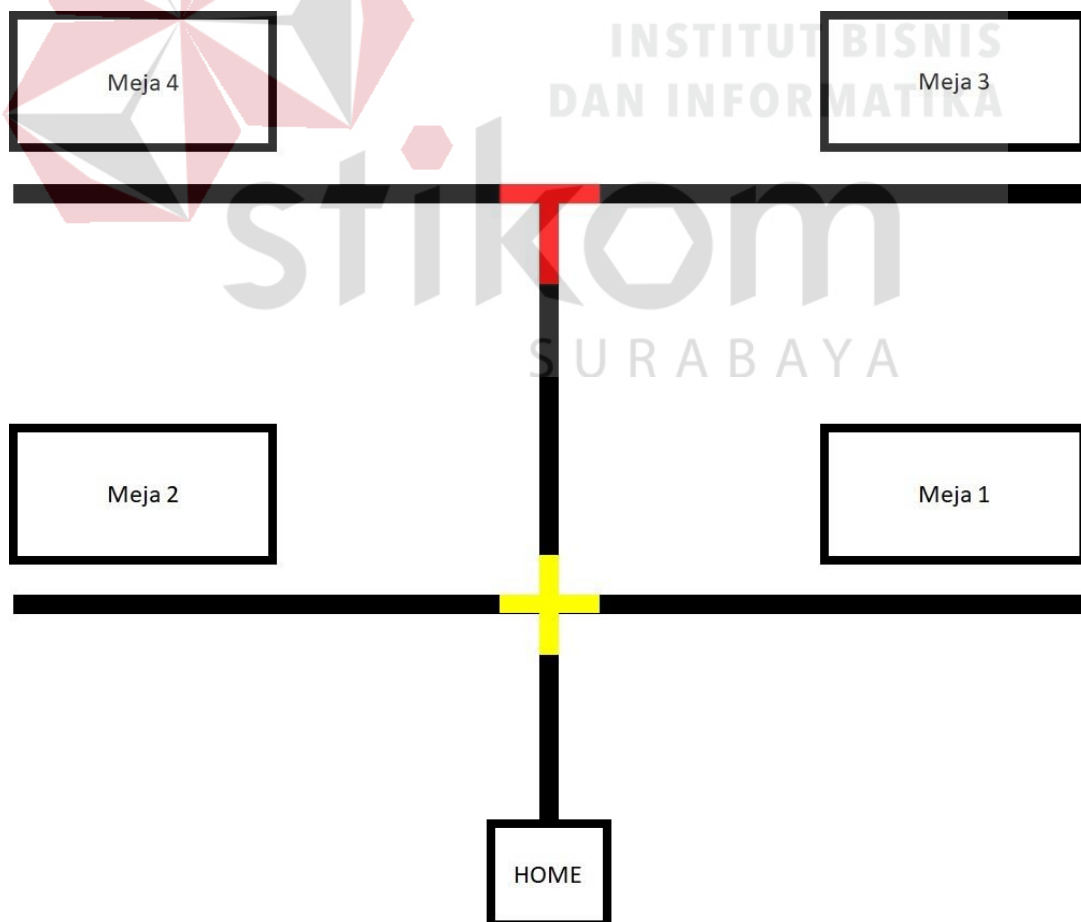


Gambar 3.33 Flowchart Program Sub-Cabang (Tracking, Next Trip, Homing)

Untuk lebih memahami dari perjalanan program yang dijelaskan, pada materi selanjutnya akan membahas denah meja dan alur rutanya.

3.2.3 Denah Meja

Pada Gambar 3.34, terdiri dari empat meja, yakni nomor “1”, “2”, “3”, dan “4”. Untuk nomor ganjil (1 dan 3) berada disisi kanan dari robot menghadap, sedangkan untuk nomor genap (2 dan 4) berada disisi kiri dari robot menghadap. Untuk jalur menggunakan lakban tiga warna yakni hitam untuk jalur utama, kuning untuk jalur perempatan yang menghubungkan meja nomor 1 dan 2 serta meneruskan jalur ke nomor 3 dan 4, dan merah untuk jalur pertigaan yang menghubungkan meja nomor 3 dan 4.



Gambar 3.34 Denah Meja

Keterangan rute:

1. Dari HOME ke Meja 1 robot melewati Perempatan kemudian belok kanan, kemudian bertemu persimpangan arah kanan robot berhenti.
 - a. Dari Meja 1 ke Meja 2 robot putar balik lalu jalan melewati Perempatan lurus kemudian bertemu persimpangan arah kiri robot berhenti.
 - b. Dari Meja 1 ke Meja 3 robot putar balik lalu jalan melewati Perempatan kemudian belok kanan sampai pada pertigaan belok kanan, kemudian bertemu persimpangan arah kanan robot berhenti.
 - c. Dari Meja 1 ke Meja 4 robot putar balik lalu jalan melewati Perempatan kemudian belok kanan sampai pada pertigaan belok kiri, kemudian bertemu persimpangan arah kiri robot berhenti.
2. Dari HOME ke Meja 2 robot melewati Perempatan kemudian belok kiri, kemudian bertemu persimpangan arah kiri robot berhenti.
 - a. Dari Meja 2 ke Meja 1 robot putar balik lalu jalan melewati Perempatan lurus kemudian bertemu persimpangan arah kanan robot berhenti.
 - b. Dari Meja 2 ke Meja 3 robot putar balik lalu jalan melewati Perempatan kemudian belok kiri sampai pada pertigaan belok kanan, kemudian bertemu persimpangan arah kanan robot berhenti.
 - c. Dari Meja 2 ke Meja 4 robot putar balik lalu jalan melewati Perempatan kemudian belok kiri sampai pada pertigaan belok kiri, kemudian bertemu persimpangan arah kiri robot berhenti.
3. Dari HOME ke Meja 3 robot melewati Perempatan kemudian lurus sampai pada pertigaan belok kanan, kemudian bertemu persimpangan arah kanan robot berhenti.

- a. Dari Meja 3 ke Meja 1 robot putar balik lalu jalan melewati pertigaan kemudian belok kiri sampai pada perempatan belok kiri, kemudian bertemu persimpangan arah kanan robot berhenti.
 - b. Dari Meja 3 ke Meja 2 robot putar balik lalu jalan melewati pertigaan kemudian belok kiri sampai pada perempatan belok kanan, kemudian bertemu persimpangan arah kiri robot berhenti.
 - c. Dari Meja 3 ke Meja 4 robot putar balik lalu jalan melewati pertigaan lurus kemudian bertemu persimpangan arah kiri robot berhenti.
4. Dari HOME ke Meja 4 robot melewati Perempatan kemudian lurus sampai pada pertigaan belok kiri, kemudian bertemu persimpangan arah kiri robot berhenti.
- a. Dari Meja 4 ke Meja 1 robot putar balik lalu jalan melewati pertigaan kemudian belok kanan sampai pada perempatan belok kiri, kemudian bertemu persimpangan arah kanan robot berhenti.
 - b. Dari Meja 4 ke Meja 2 robot putar balik lalu jalan melewati pertigaan kemudian belok kanan sampai pada perempatan belok kanan, kemudian bertemu persimpangan arah kiri robot berhenti.
 - c. Dari Meja 4 ke Meja 3 robot putar balik lalu jalan melewati pertigaan lurus kemudian bertemu persimpangan arah kanan robot berhenti.

BAB IV

HASIL DAN PEMBAHASAN

Dalam bab ini akan membahas beberapa pengujian untuk sistem yang telah dirancang pada Tugas Akhir ini. Tujuan dari bab ini adalah untuk mengetahui tingkat keberhasilan terhadap perancangan sistem yang telah diajukan dan dikerjakan. Pada bab ini terdapat beberapa pengujian yakni pengujian kamera dalam mendeteksi garis, waktu tempuh dari posisi start (*home*) ke meja pertama, waktu tempuh dari meja pertama ke meja kedua, waktu tempuh meja kedua atau terakhir kembali ke posisi start (*home*) dan kecepatan robot yang dirangkum dalam dua bagian yakni uji destinasi 1 meja dan uji destinasi 2 meja.

4.1 Uji Kamera Deteksi Garis

4.1.1 Tujuan Uji Kamera Deteksi Garis

Pengujian ini digunakan untuk mengetahui bentuk deteksi garis yang dibaca oleh kamera, pengolahan citra yang digunakan untuk mendeteksi garis, dan nilai koordinat dari x , y , dan jari-jari (r) citra garis yang diperoleh. Dengan diketahuinya kesemua itu, dapat dilihat keberhasilan kamera dalam menangkap gambar dan mendeteksi garis.

4.1.2 Alat yang Digunakan Pada Uji Kamera Deteksi Garis

Peralatan yang dibutuhkan untuk pengujian ini adalah sebagai berikut:

1. *Mobile Robot Vision* (terutama Raspberry Pi 3).
2. Kamera Webcam Logitech C270.

3. Laptop.
4. Program OpenCV 2.1.0.
5. Jalur garis atau lintasan.

4.1.3 Prosedur Pengujian Pada Uji Kamera Deteksi Garis

Langkah-langkah yang dilakukan dalam melakukan pengujian tahap ini sebagai berikut:

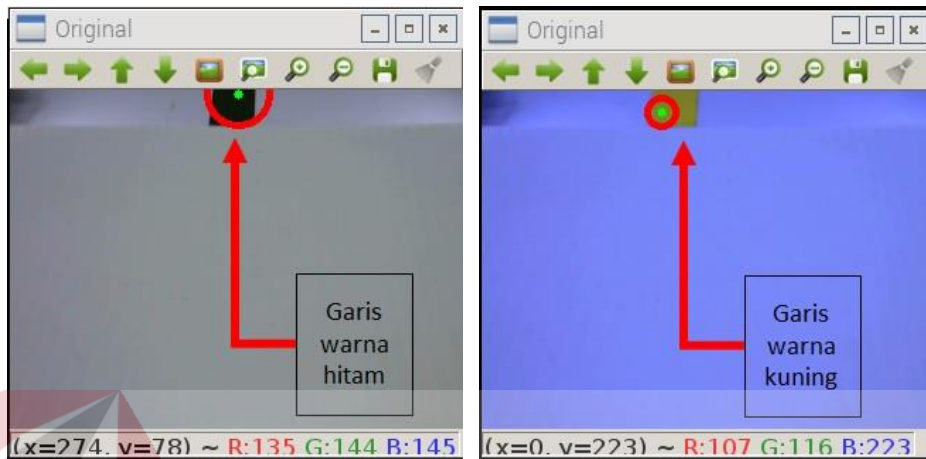
1. Mempersiapkan *Mobile Robot Vision* pada lintasan (garis hitam, garis kuning atau perempatan dan garis merah atau pertigaan) serta mengaktifkan OpenCV pada Raspberry Pi 3.
2. Memperhatikan hasil citra yang ditangkap oleh kamera Webcam Logitech C270 berupa gambar asli (RGB), gambar HSV dan Gambar *edge* serta nilai koordinat x, y, dan jari-jari (r).
3. Mengamati bentuk pengolahan citra yang ditangkap oleh kamera pada layar laptop.

4.1.4 Hasil Pengujian Pada Uji Kamera Deteksi Garis

Pada pengujian ini komponen yang dipakai yakni kamera Webcam Logitech C270 sebagai *vision* untuk menangkap citra garis dan Raspberry Pi 3 sebagai Mini PC untuk pengolahan citra yang ditangkap oleh kamera. Pengujian ini juga dilakukan tiga bentuk deteksi yaitu Deteksi Garis Hitam, Deteksi Garis Kuning dan Deteksi Garis Merah. Masing-masing deteksi memiliki beberapa bentuk pengolahan citra. Berikut tahapan-tahapan pengujian pada kamera menggunakan fitur OpenCV untuk mendeteksi objek garis:

1. Menangkap citra dengan format RGB

```
CvCapture* img = cvCaptureFromCAM(0);
p_imgOriginal = cvQueryFrame(img);
cvNamedWindow("Original", CV_WINDOW_AUTOSIZE);
```



(a)

(b)



(c)

Gambar 4.1 Citra RGB (a) Deteksi Warna Hitam, (b) Deteksi Warna Kuning, (c) Deteksi Warna Merah

Penjelasan diatas variabel `img` menyimpan hasil citra yang ditangkap oleh kamera, lalu hasil tangkapan dibentuk dalam frame dengan format RGB dan disimpan dalam variabel `p_imgOriginal`. Kemudian ditampilkan dengan nama “Original” seperti pada Gambar 4.1.

2. Konversi dari bentuk RGB to HSV

```
cvCvtColor(p_imgOriginal, p_imgHSV, CV_BGR2HSV);

cvInRangeS(p_imgHSV, cvScalar(H_MIN, S_MIN, V_MIN), cvScalar(H_MAX,
S_MAX, V_MAX), p_imgProcessed);
cvInRangeS(p_imgHSV, cvScalar(H_MINz, S_MINz, V_MINz), cvScalar(H_MAXz,
S_MAXz, V_MAXz), p_imgProcessedz);
cvInRangeS(p_imgHSV, cvScalar(H_MINr, S_MINr, V_MINr), cvScalar(H_MAXr,
S_MAXr, V_MAXr), p_imgProcessedr);

p_strStorage = cvCreateMemStorage(0);
```

Penjelasan sintaks diatas citra RGB yang disimpan dalam variabel `p_imgOriginal` dikonversi kedalam bentuk citra format HSV yang lalu hasil konversi tersebut dimasukkan ke dalam variabel `p_imgHSV`. Setelah dimasukkan ke dalam variabel `p_imgHSV`, dicarilah warna HSV-nya dalam bentuk skala. Didapatkan nilai skala HSV dari warna hitam seperti pada Tabel 4.1, nilai skala HSV dari warna kuning pada Tabel 4.2, dan nilai skala HSV dari warna merah pada Tabel 4.3. Setelah didapatkan nilai minimal dan maksimal dari masing-masing warna HSV maka langkah selanjutnya yakni nilai-nilai yang telah didapatkan disimpan dalam sebuah penyimpanan memori di Raspberry Pi 3.

Tabel 4.1 Nilai Skala HSV Warna Hitam

Hitam	
Nama Variabel	Nilai Skala
Nilai Minimal <i>Hue</i> (H_MIN)	0
Nilai Maksimal <i>Hue</i> (H_MAX)	255
Nilai Minimal <i>Saturation</i> (S_MIN)	0
Nilai Maksimal <i>Saturation</i> (S_MAX)	255
Nilai Minimal <i>Value</i> (V_MIN)	0
Nilai Maksimal <i>Value</i> (V_MAX)	68

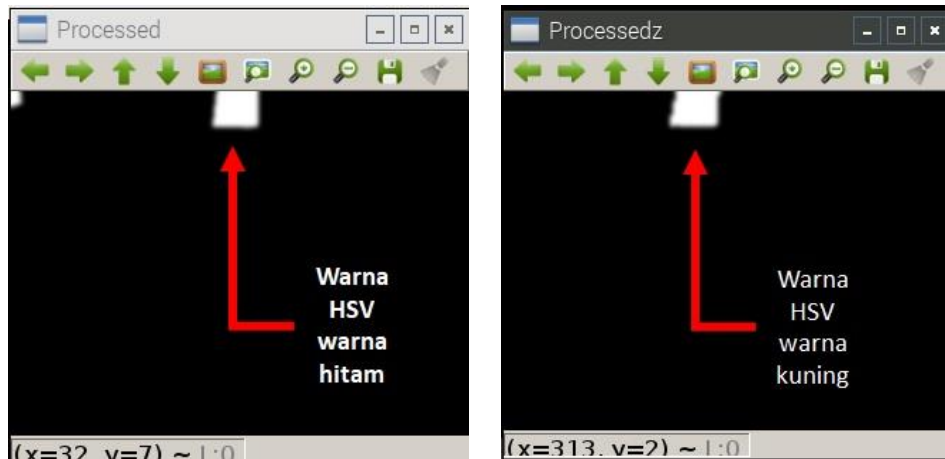
Tabel 4.2 Nilai Skala HSV Warna Kuning

Kuning	
Nama Variabel	Nilai Skala
Nilai Minimal <i>Hue</i> (H_MINz)	20
Nilai Maksimal <i>Hue</i> (H_MAXz)	57
Nilai Minimal <i>Saturation</i> (S_MINz)	50
Nilai Maksimal <i>Saturation</i> (S_MAXz)	255
Nilai Minimal <i>Value</i> (V_MINz)	80
Nilai Maksimal <i>Value</i> (V_MAXz)	195

Tabel 4.3 Nilai Skala HSV Warna Merah

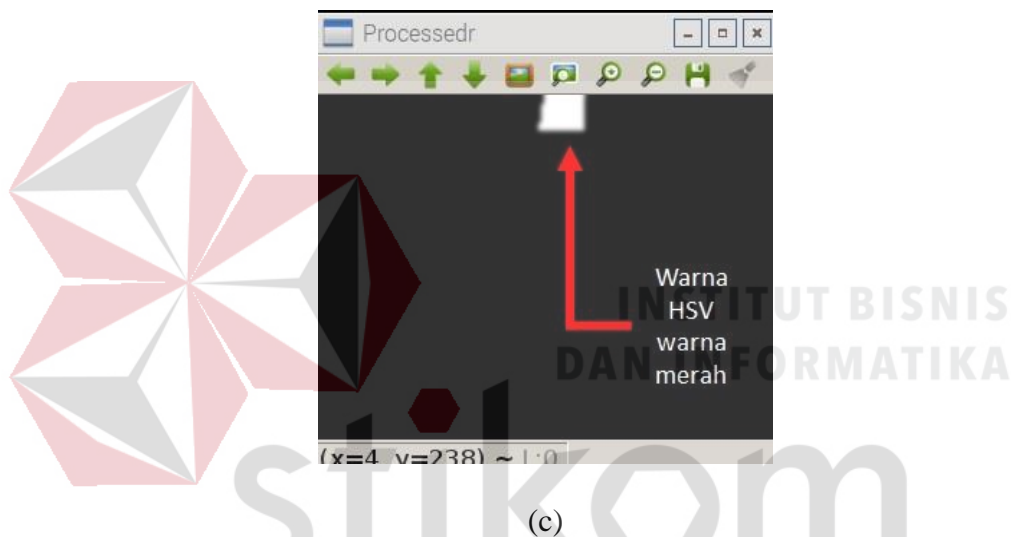
Merah	
Nama Variabel	Nilai Skala
Nilai Minimal <i>Hue</i> (H_MINr)	0
Nilai Maksimal <i>Hue</i> (H_MAXr)	51
Nilai Minimal <i>Saturation</i> (S_MINr)	181
Nilai Maksimal <i>Saturation</i> (S_MAXr)	255
Nilai Minimal <i>Value</i> (V_MINr)	73
Nilai Maksimal <i>Value</i> (V_MAXr)	113

Lalu setelah didapatkan nilai HSV nya, disimpan dalam variabel sesuai dengan nilai skala untuk masing-masing warna yaitu variabel `p_imgProcessed` untuk menyimpan citra HSV Garis Warna Hitam, `p_imgProcessedz` untuk menyimpan citra HSV Garis Warna Kuning, dan `p_imgProcessedr` untuk menyimpan citra HSV Garis Warna Merah. Hasil citra HSV bisa dilihat pada Gambar 4.2 lalu disimpan dalam memori Raspberry Pi 3.



(a)

(b)

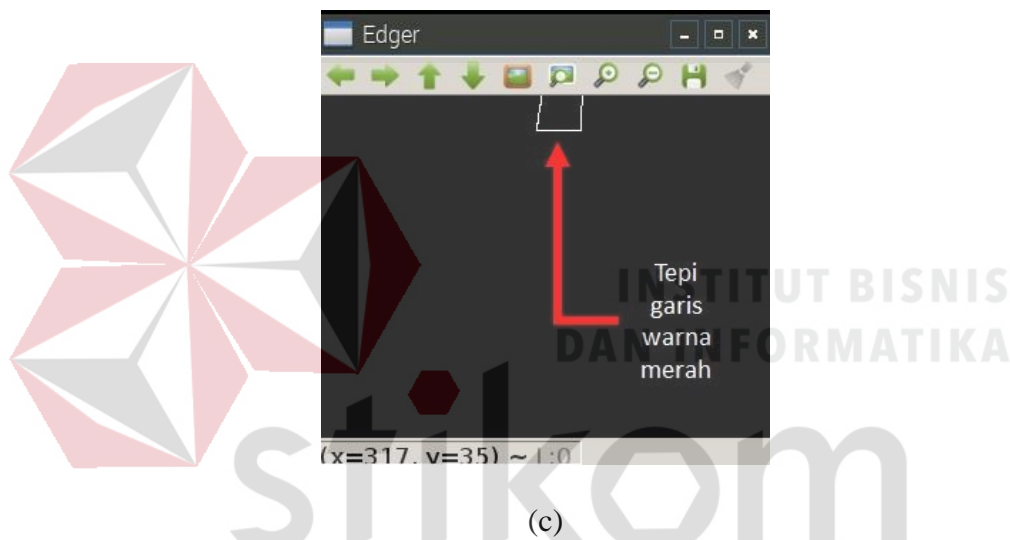
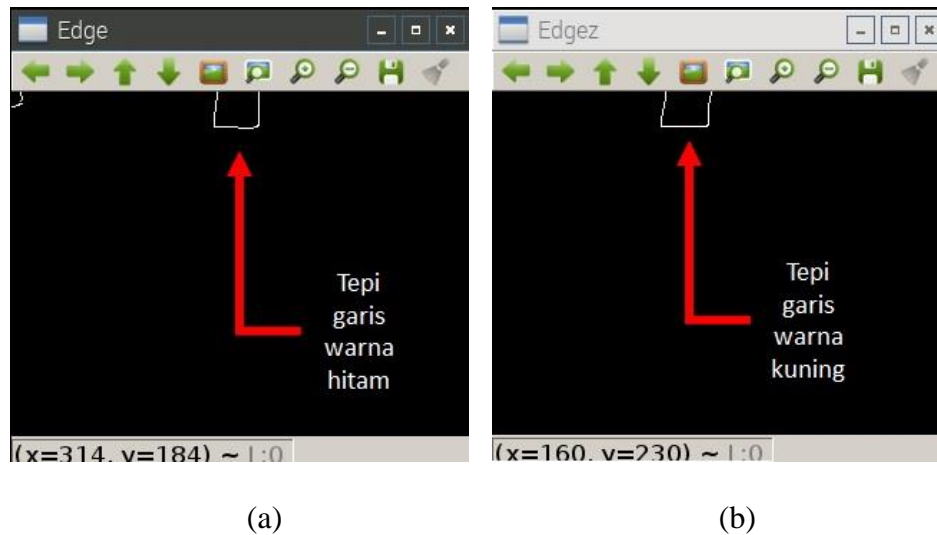


(c)

Gambar 4.2 Citra HSV (a) Deteksi Warna Hitam, (b) Deteksi Warna Kuning
(c) Deteksi Warna Merah

3. Menghilangkan noise dan deteksi tepi garis

```
cvSmooth(p_imgProcessed, p_imgProcessed, CV_GAUSSIAN, 9, 9);
cvSmooth(p_imgProcessedz, p_imgProcessedz, CV_GAUSSIAN, 9, 9);
cvSmooth(p_imgProcessedr, p_imgProcessedr, CV_GAUSSIAN, 9, 9);
cvCanny(p_imgProcessed, p_imgEdge, 50, 150, 3);
cvCanny(p_imgProcessedz, p_imgEdgez, 50, 150, 3);
cvCanny(p_imgProcessedr, p_imgEdger, 50, 150, 3);
p_seqCircles = cvHoughCircles(p_imgEdge, p_strStorage,
CV_HOUGH_GRADIENT, 2, p_imgEdge->height / 4, 100, 25, 0, 50);
p_seqCirclesz = cvHoughCircles(p_imgEdgez, p_strStorage,
CV_HOUGH_GRADIENT, 2, p_imgEdgez->height / 4, 100, 25, 0, 50);
p_seqCirclesr = cvHoughCircles(p_imgEdger, p_strStorage,
CV_HOUGH_GRADIENT, 2, p_imgEdger->height / 4, 100, 25, 0, 50);
```



Gambar 4.3 Citra *Edge* (a) Deteksi Garis Hitam, (b) Deteksi Garis Kuning
(c) Deteksi Garis Merah

Pada penjelasan tahap ini, citra HSV yang telah disimpan di dalam memori dihilangkan *noise*-nya (*smoothing*) dengan metode *Gaussian-blur* untuk memperhalus dan memperjelas gambar sehingga citra yang diperoleh bisa lebih rinci. Setelah di *smoothing*, citra di deteksi tepi garisnya dengan metode *Canny Edge Detection*. Setelah mendapatkan citra tepi garisnya, citra tersebut dimasukkan ke dalam variabel *p_imgEdge* untuk deteksi tepi Garis Warna Hitam, *p_imgEdgez* untuk deteksi tepi Garis Warna Kuning dan *p_imgEdger* untuk deteksi tepi Garis Warna Merah. Hasil citra *Edge* dapat

dilihat pada Gambar 4.3. Kemudian setelah mendapatkan tepi garisnya, citra selanjutnya dicari bentuk lingkarannya untuk menandakan objek berdasarkan warna HSV yang telah ditentukan sebelumnya dan dibatasi pembacaannya dengan hasil citra *Edge Detection*, dengan metode *Hough Circle Transform* didapatkan bentuk lingkaran dari citra yang ditangkap, hasil bentuk lingkaran dapat dilihat pada Gambar 4.1., dan hasil citra dari *Hough Circle Transform* disimpan dalam variabel `p_seqCircles` untuk Deteksi Warna Hitam, `p_seqCirclesz` untuk Deteksi Warna Kuning dan `p_seqCirclesr` untuk Deteksi Warna Merah.

4. Deteksi warna garis dan koordinat garis

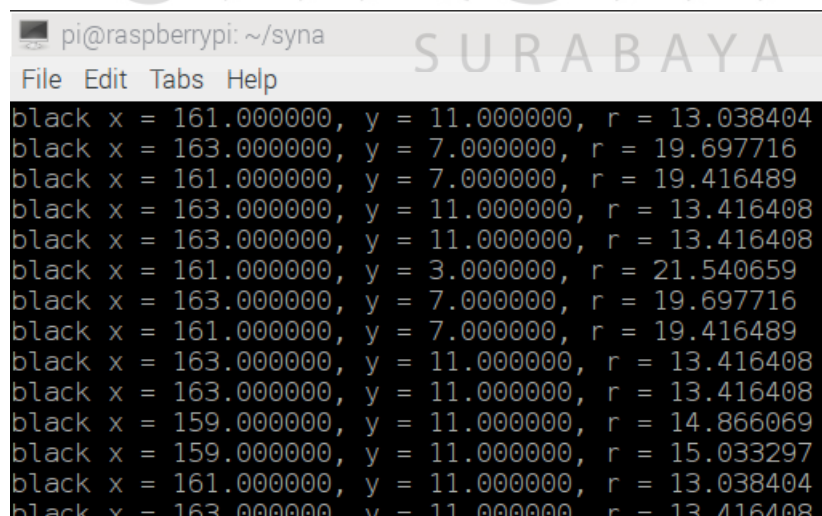
```
//deteksi hitam
if (p_seqCircles->total >= 1)
{
    p_fltXYRadius = (float*)cvGetSeqElem(p_seqCircles, 1);
    printf("black x = %f, y = %f, r = %f \n", p_fltXYRadius[0],
        p_fltXYRadius[1], p_fltXYRadius[2]);
    cvCircle(p_imgOriginal, cvPoint(cvRound(p_fltXYRadius[0]),
        cvRound(p_fltXYRadius[1])), 3, CV_RGB(0, 255, 0), CV_FILLED);
    cvCircle(p_imgOriginal, cvPoint(cvRound(p_fltXYRadius[0]),
        cvRound(p_fltXYRadius[1])), cvRound(p_fltXYRadius[2]), CV_RGB(255,
        0, 0), 3);
    nilaix = p_fltXYRadius[0] / 3.2;
    serialPutchar(fd, nilaix);
}
//deteksi kuning
else if (p_seqCirclesz->total >= 1)
{
    p_fltXYRadiusz = (float*)cvGetSeqElem(p_seqCirclesz, 1);
    printf("yellow x = %f, y = %f, r = %f \n", p_fltXYRadiusz[0],
        p_fltXYRadiusz[1], p_fltXYRadiusz[2]);
    cvCircle(p_imgOriginal, cvPoint(cvRound(p_fltXYRadiusz[0]),
        cvRound(p_fltXYRadiusz[1])), 3, CV_RGB(0, 255, 0), CV_FILLED);
    cvCircle(p_imgOriginal, cvPoint(cvRound(p_fltXYRadiusz[0]),
        cvRound(p_fltXYRadiusz[1])), cvRound(p_fltXYRadiusz[2]),
        CV_RGB(255, 0, 0), 3);
    serialPutchar(fd, 'x');
}
//deteksi merah
else if (p_seqCirclesr->total >= 1)
{
    p_fltXYRadiusr = (float*)cvGetSeqElem(p_seqCirclesr, 1);
    printf("red x = %f, y = %f, r = %f \n", p_fltXYRadiusr[0],
        p_fltXYRadiusr[1], p_fltXYRadiusr[2]);
```

```

    cvCircle(p_imgOriginal, cvPoint(cvRound(p_fltXYRadiusr[0]),
    cvRound(p_fltXYRadiusr[1])), 3, CV_RGB(0, 255, 0), CV_FILLED);
    cvCircle(p_imgOriginal, cvPoint(cvRound(p_fltXYRadiusr[0]),
    cvRound(p_fltXYRadiusr[1])), cvRound(p_fltXYRadiusr[2]),
    CV_RGB(255, 0, 0), 3);
    serialPutchar(fd, 'z');
}
//deteksi putih
else
{
    printf("kosong x = %f, y = %f, r = %f \n");
    serialPutchar(fd, 'y');
}
}

```

Penjelasan program diatas setelah citra disimpan dalam variabel `p_seqCircles`, `p_seqCirclesz`, dan `p_seqCirclesr`. Variabel yang mewakili masing-masing warna dicari nilai koordinat `x`, `y` dan jari-jari (`r`), untuk kondisi ketika mendeteksi Warna Hitam maka variabel `p_seqCircles` dipilah dengan sintaks `cvGetSeqElem` lalu dimasukkan pada variabel array `p_fltXYRadius`. Kemudian ditampilkan pada layar terminal Raspberry Pi 3, nilai koordinat `x` digunakan sebagai acuan membaca nilai garis yang dikirim oleh Raspberry Pi 3 ke Arduino DUE. Nilai koordinat `x`, `y` dan jari-jari (`r`) dapat dilihat pada Gambar 4.4.



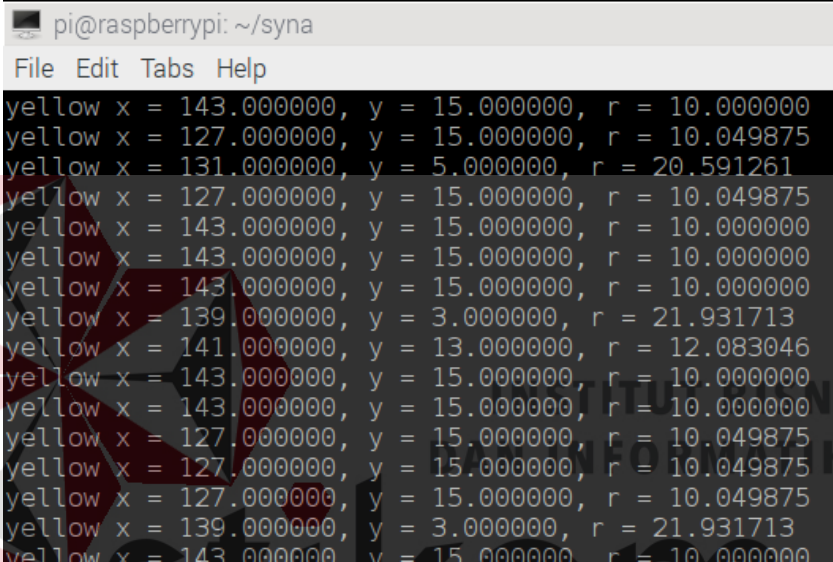
```

pi@raspberrypi: ~/syna
File Edit Tabs Help
black x = 161.000000, y = 11.000000, r = 13.038404
black x = 163.000000, y = 7.000000, r = 19.697716
black x = 161.000000, y = 7.000000, r = 19.416489
black x = 163.000000, y = 11.000000, r = 13.416408
black x = 163.000000, y = 11.000000, r = 13.416408
black x = 161.000000, y = 3.000000, r = 21.540659
black x = 163.000000, y = 7.000000, r = 19.697716
black x = 161.000000, y = 7.000000, r = 19.416489
black x = 163.000000, y = 11.000000, r = 13.416408
black x = 163.000000, y = 11.000000, r = 13.416408
black x = 159.000000, y = 11.000000, r = 14.866069
black x = 159.000000, y = 11.000000, r = 15.033297
black x = 161.000000, y = 11.000000, r = 13.038404
black x = 163.000000, y = 11.000000, r = 13.416408

```

Gambar 4.4 Nilai Koordinat X, Y, dan Jari-Jari (r) dari Warna Hitam

Untuk kondisi ketika mendeteksi Warna Kuning maka variabel `p_seqCirclesz` dipilah dengan sintaks `cvGetSeqElem` lalu dimasukkan pada variabel array `p_fltXYRadiusz`. Kemudian ditampilkan pada layar terminal Raspberry Pi 3, nilai koordinat x, y dan jari-jari (r) dapat dilihat pada Gambar 4.5.



```

pi@raspberrypi: ~/syna
File Edit Tabs Help
yellow x = 143.000000, y = 15.000000, r = 10.000000
yellow x = 127.000000, y = 15.000000, r = 10.049875
yellow x = 131.000000, y = 5.000000, r = 20.591261
yellow x = 127.000000, y = 15.000000, r = 10.049875
yellow x = 143.000000, y = 15.000000, r = 10.000000
yellow x = 143.000000, y = 15.000000, r = 10.000000
yellow x = 143.000000, y = 15.000000, r = 10.000000
yellow x = 139.000000, y = 3.000000, r = 21.931713
yellow x = 141.000000, y = 13.000000, r = 12.083046
yellow x = 143.000000, y = 15.000000, r = 10.000000
yellow x = 143.000000, y = 15.000000, r = 10.000000
yellow x = 127.000000, y = 15.000000, r = 10.049875
yellow x = 127.000000, y = 15.000000, r = 10.049875
yellow x = 127.000000, y = 15.000000, r = 10.049875
yellow x = 139.000000, y = 3.000000, r = 21.931713
yellow x = 143.000000, y = 15.000000, r = 10.000000

```

Gambar 4.5 Nilai Koordinat X, Y, dan Jari-Jari (r) dari Warna Kuning

Untuk kondisi ketika mendeteksi Warna Merah maka variabel `p_seqCirclesr` dipilah dengan sintaks `cvGetSeqElem` lalu dimasukkan pada variabel array `p_fltXYRadiusr`. Kemudian ditampilkan pada layar terminal Raspberry Pi 3, nilai koordinat x, y dan jari-jari (r) dapat dilihat pada Gambar 4.6.


```

pi@raspberrypi: ~/syna
File Edit Tabs Help
red x = 175.000000, y = 17.000000, r = 9.219544
red x = 163.000000, y = 15.000000, r = 10.000000
red x = 173.000000, y = 3.000000, r = 21.931713
red x = 169.000000, y = 9.000000, r = 14.866069
red x = 163.000000, y = 15.000000, r = 10.000000
red x = 169.000000, y = 9.000000, r = 14.764823
red x = 163.000000, y = 15.000000, r = 10.000000
red x = 169.000000, y = 17.000000, r = 15.297058
red x = 167.000000, y = 5.000000, r = 20.396078
red x = 163.000000, y = 15.000000, r = 10.000000
red x = 169.000000, y = 9.000000, r = 14.866069
red x = 165.000000, y = 3.000000, r = 20.615528
red x = 175.000000, y = 17.000000, r = 9.219544
red x = 163.000000, y = 15.000000, r = 10.000000
red x = 163.000000, y = 15.000000, r = 10.000000
red x = 163.000000, y = 15.000000, r = 10.000000

```

Gambar 4.6 Nilai Koordinat X, Y, dan Jari-Jari (r) dari Warna Merah

Untuk masing-masing warna mempunyai perintah yang berbeda-beda, ketika kamera menangkap citra warna hitam maka akan melakukan perintah kirim nilai koordinat x ke Arduino DUE. Jika menangkap citra warna kuning, maka akan melakukan perintah kirim karakter 'x' ke Arduino DUE. Jika menangkap citra warna merah, maka akan melakukan perintah kirim karakter 'z' ke Arduino DUE. Jika tidak mendeteksi Warna Hitam, kuning, ataupun merah maka akan melakukan perintah kirim karakter 'y' ke Arduino DUE.

4.2 Uji Halangan Dengan Sensor Jarak

4.2.1 Tujuan Uji Halangan Dengan Sensor Jarak

Pengujian ini digunakan untuk mengetahui apakah robot dapat mendeteksi halangan yang berada di depannya dengan menggunakan sensor jarak yakni Sensor IR Proximity dan juga menguji seberapa ampuh Sensor IR Proximity dalam membaca jarak yang telah ditentukan.

4.2.2 Alat yang Digunakan Pada Uji Halangan Sensor Jarak

Peralatan yang dibutuhkan untuk pengujian ini adalah sebagai berikut:

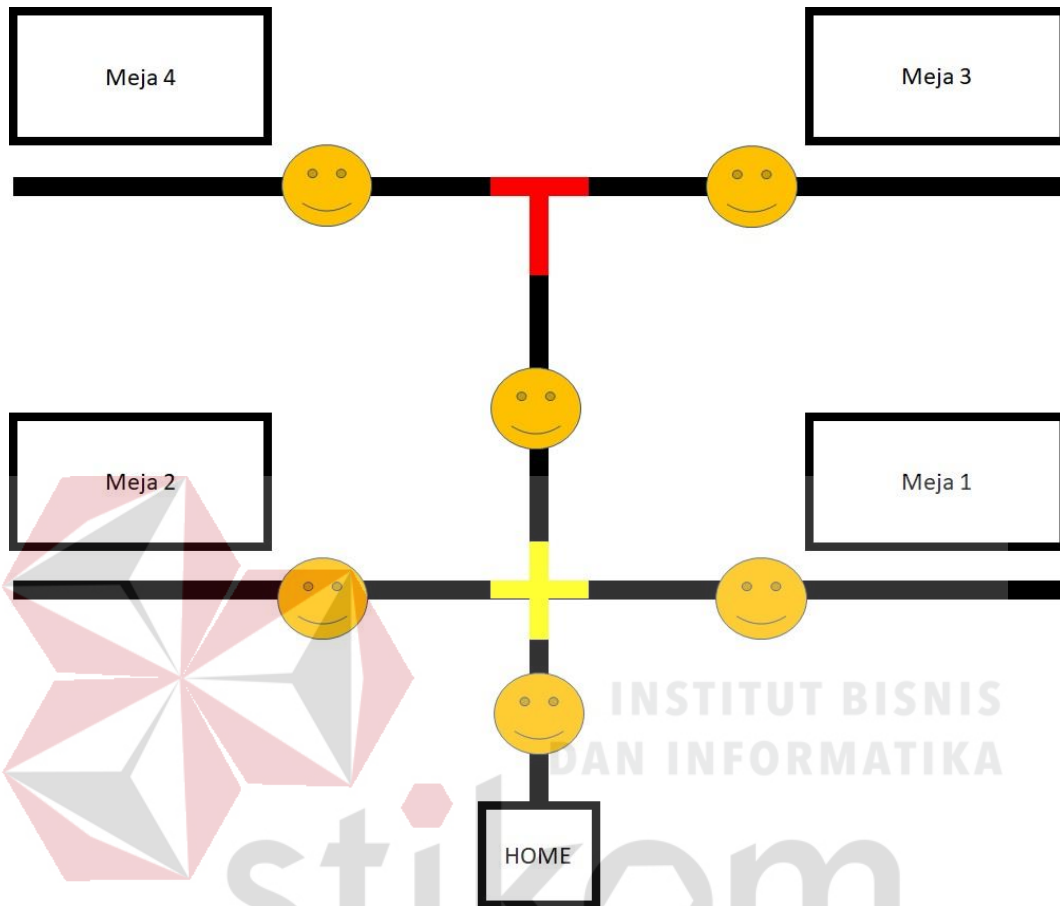
1. *Mobile Robot Vision*.
2. Laptop.
3. Program Arduino IDE.
4. Sensor IR Proximity (yang telah terpasang di sisi kanan dan sisi kiri depan robot).
5. Benda untuk halangan (orang).

4.2.3 Prosedur Pengujian Pada Uji Halangan Sensor Jarak

Langkah-langkah yang dilakukan dalam melakukan pengujian tahap ini sebagai berikut:

1. Mempersiapkan *Mobile Robot Vision* di posisi start (*home*) serta mengaktifkan OpenCV pada Raspberry Pi 3.
2. Melakukan transfer program dari laptop ke robot (lebih baik sudah dilakukan dari awal).

3. Pengujian pada tahap ini dilakukan dengan benda yang digunakan untuk menghalang berdiri di titik-titik jalur pada Gambar 4.7.



Gambar 4.7 Titik Jalur yang Dihalang

4. Robot melakukan Destinasi 1 Meja pada tiap masing-masing meja yang dituju dalam sekali percobaan.
5. Mengamati pergerakan robot dan perhatikan apakah robot berhasil berhenti mendeteksi adanya halangan atau tidak.

4.2.4 Hasil Pengujian Pada Uji Halangan Sensor Jarak

Pada pengujian pada tahap ini, pengujian berupa robot dapat berhenti bila terdapat adanya halangan yang berada didepannya dengan mendata robot ketika

mengantarkan makanan dan robot kembali ke *home*. Berikut tabel pengujian uji halangan sensor jarak yang ada pada Tabel 4.4, Tabel 4.5, Tabel 4.6, dan Tabel 4.7.

Tabel 4.4 Pengujian Halangan Meja 1

Meja 1

Posisi Benda	Keterangan	
	Berhasil	Tidak
Robot Mengantarkan Makanan		
Antara Home dan Perempatan	√	
Antara Perempatan dan Meja 1	√	
Robot Kembali ke Home		
Antara Meja 1 dan Perempatan	√	
Antara Perempatan dan Home	√	

Tabel 4.5 Pengujian Halangan Meja 2

Meja 2

Posisi Benda	Keterangan	
	Berhasil	Tidak
Robot Mengantarkan Makanan		
Antara Home dan Perempatan	√	
Antara Perempatan dan Meja 2	√	
Robot Kembali ke Home		
Antara Meja 2 dan Perempatan	√	
Antara Perempatan dan Home	√	

Tabel 4.6 Pengujian Halangan Meja 3

Meja 3

Posisi Benda	Keterangan	
	Berhasil	Tidak
Robot Mengantarkan Makanan		
Antara Home dan Perempatan	√	
Antara Perempatan dan Pertigaan	√	
Antara Pertigaan dan Meja 3	√	
Robot Kembali ke Home		
Antara Meja 3 dan Pertigaan	√	
Antara Pertigaan dan Perempatan	√	
Antara Perempatan dan Home	√	

Tabel 4.7 Pengujian Halangan Meja 4

Meja 4

Posisi Benda	Keterangan	
	Berhasil	Tidak
Robot Mengantarkan Makanan		
Antara Home dan Perempatan	√	
Antara Perempatan dan Pertigaan	√	
Antara Pertigaan dan Meja 4	√	
Robot Kembali ke Home		
Antara Meja 4 dan Pertigaan	√	
Antara Pertigaan dan Perempatan	√	
Antara Perempatan dan Home	√	

Dari hasil pengujian yang ditampilkan pada Tabel 4.4 – 4.7 dapat disimpulkan Sensor IR Proximity akurat dalam mendeteksi objek atau halangan yang berada di depannya.

4.3 Uji Destinasi 1 Meja

4.3.1 Tujuan Uji Destinasi 1 Meja

Pengujian ini digunakan untuk mengetahui waktu tempuh robot dari posisi start (*home*) ke meja dan kembali lagi ke posisi semula, kecepatan robot dalam melakukan perjalanan berangkat hingga kembali serta persentase keberhasilan robot dalam menuntaskan destinasinya. Dengan diketahuinya kesemua itu, dapat dilihat rata-rata waktu tempuh, kecepatan, dan keberhasilan robot.

4.3.2 Alat yang Digunakan Pada Uji Destinasi 1 Meja

Peralatan yang dibutuhkan untuk pengujian ini adalah sebagai berikut:

1. *Mobile Robot Vision*.
2. Laptop.

3. Program Arduino IDE.
4. 4 Kantong Beras (setiap kantong berat 1 Kg).
5. Stopwatch (menggunakan Smartphone).

4.3.3 Prosedur Pengujian Pada Uji Destinasi 1 Meja

Langkah-langkah yang dilakukan dalam melakukan pengujian tahap ini sebagai berikut:

1. Mempersiapkan *Mobile Robot Vision* di posisi start (*home*) serta mengaktifkan OpenCV pada Raspberry Pi 3
2. Melakukan transfer program dari laptop ke robot (lebih baik sudah dilakukan dari awal)
3. Pengujian pada tahap ini dilakukan dengan dua cara yakni robot diuji Tanpa Beban dan dengan beban. Pada pengujian robot Tanpa Beban masing-masing meja yang dituju dilakukan sebanyak 12 kali percobaan beserta keberhasilannya, sedangkan, dengan beban pengujian robot dilakukan pada masing-masing meja sebanyak 3 kali dengan beban yang dibawa berurutan mulai dari 1 Kg, 2 Kg, 3 Kg dan 4 Kg beserta keberhasilannya.
4. Mengamati pergerakan robot dan hitung waktu tempuh dengan menggunakan stopwatch.

4.3.4 Hasil Pengujian Waktu Tempuh dan Kecepatan Pada Uji Destinasi 1 Meja

1. Hasil Pengujian Waktu Tempuh dan Kecepatan Tanpa Beban

Pada pengujian pada tahap ini, pengujian berupa menghitung waktu tempuh robot pada destinasi 1 meja untuk mendapatkan kecepatan robot dan

dilakukan tanpa adanya beban pada robot. Berikut adalah hasil pengujian untuk masing-masing meja pada Tabel 4.8, Tabel 4.9, Tabel 4.10, dan Tabel 4.11.

Tabel 4.8 Pengujian Meja Nomor 1 Tanpa Beban

Jarak keseluruhan : 842 cm

Percobaan	Waktu Tempuh 1 (s)	Waktu Tempuh 2 (s)	Total Waktu (s)	Kecepatan (cm/s)
1	16,23	25,51	41,74	20,17
2	16,17	25,35	41,52	20,28
3	16,30	25,36	41,66	20,21
4	16,24	25,60	41,84	20,12
5	16,34	25,22	41,56	20,26
6	16,32	7,82	24,14	34,88
7	15,90	24,57	40,47	20,81
8	16,26	25,03	41,29	20,39
9	17,28	25,46	42,74	19,70
10	16,43	25,15	41,58	20,25
11	16,66	25,35	42,01	20,04
12	16,25	25,34	41,59	20,25
Rata-rata	16,37	23,81	40,18	21,45

Tabel 4.9 Pengujian Meja Nomor 2 Tanpa Beban

Jarak keseluruhan : 844 cm

Percobaan	Waktu Tempuh 1 (s)	Waktu Tempuh 2 (s)	Total Waktu (s)	Kecepatan (cm/s)
1	16,39	9,64	26,03	32,42
2	17,08	23,79	40,87	20,65
3	16,07	20,14	36,21	23,31
4	16,59	26,10	42,69	19,77
5	16,33	25,44	41,77	20,21
6	16,24	24,42	40,66	20,76
7	16,27	24,60	40,87	20,65
8	16,55	24,55	41,10	20,54
9	14,87	0,00	14,87	56,76
10	16,32	24,36	40,68	20,75
11	17,06	14,42	31,48	26,81

Percobaan	Waktu Tempuh 1 (s)	Waktu Tempuh 2 (s)	Total Waktu (s)	Kecepatan (cm/s)
12	16,44	24,47	40,91	20,63
Rata-rata	16,35	20,16	36,51	25,27

Tabel 4.10 Pengujian Meja Nomor 3 Tanpa Beban

Jarak keseluruhan : 1214 cm

Percobaan	Waktu Tempuh 1 (s)	Waktu Tempuh 2 (s)	Total Waktu (s)	Kecepatan (cm/s)
1	22,32	11,65	33,97	35,74
2	23,75	31,47	55,22	21,98
3	22,65	32,57	55,22	21,98
4	22,82	31,15	53,97	22,49
5	23,14	31,97	55,11	22,03
6	22,97	32,29	55,26	21,97
7	22,82	32,61	55,43	21,90
8	22,96	11,73	34,69	35,00
9	22,31	31,33	53,64	22,63
10	22,27	31,43	53,70	22,61
11	22,10	31,59	53,69	22,61
12	22,33	30,94	53,27	22,79
Rata-rata	22,70	28,39	51,10	24,48

Tabel 4.11 Pengujian Meja Nomor 4 Tanpa Beban

Jarak keseluruhan : 1216 cm

Percobaan	Waktu Tempuh 1 (s)	Waktu Tempuh 2 (s)	Total Waktu (s)	Kecepatan (cm/s)
1	22,32	14,99	37,31	32,59
2	22,39	31,63	54,02	22,51
3	23,29	32,29	55,58	21,88
4	23,00	32,19	55,19	22,03
5	23,00	31,88	54,88	22,16
6	22,65	32,08	54,73	22,22
7	22,92	30,38	53,30	22,81
8	22,37	30,95	53,32	22,81
9	22,50	30,92	53,42	22,76

Percobaan	Waktu Tempuh 1 (s)	Waktu Tempuh 2 (s)	Total Waktu (s)	Kecepatan (cm/s)
10	22,52	31,23	53,75	22,62
11	22,28	30,40	52,68	23,08
12	22,66	13,92	36,58	33,24
Rata-rata	22,66	28,57	51,23	24,23

Pada tabel-tabel diatas dapat dicari rata-rata waktu yang ditempuh, dan rata-rata kecepatan robot dalam menuntaskan destinasi.

- a. Pada Tabel 4.4 rata-rata waktu robot menuju ke meja nomor 1 yakni **16,37 seconds**, rata-rata waktu robot kembali ke posisi start (*home*) yakni **23,81 seconds**, dan rata-rata waktu secara keseluruhan yakni **40,18 seconds**, dan rata-rata kecepatan yang ditempuh yakni **21,45 cm/s**.
- b. Pada Tabel 4.5 rata-rata waktu robot menuju ke meja nomor 2 yakni **16,35 seconds**, rata-rata waktu robot kembali ke posisi start (*home*) yakni **20,16 seconds**, dan rata-rata waktu secara keseluruhan yakni **36,51 seconds**, dan rata-rata kecepatan yang ditempuh yakni **25,27 cm/s**.
- c. Pada Tabel 4.6 rata-rata waktu robot menuju ke meja nomor 3 yakni **22,70 seconds**, rata-rata waktu robot kembali ke posisi start (*home*) yakni **28,39 seconds**, dan rata-rata waktu secara keseluruhan yakni **51,10 seconds**, dan rata-rata kecepatan yang ditempuh yakni **24,48 cm/s**.
- d. Pada Tabel 4.7 rata-rata waktu robot menuju ke meja nomor 4 yakni **22,66 seconds**, rata-rata waktu robot kembali ke posisi start (*home*) yakni **28,57 seconds**, dan rata-rata waktu secara keseluruhan yakni **51,23 seconds**, dan rata-rata kecepatan yang ditempuh yakni **24,23 cm/s**.

2. Hasil Pengujian Waktu Tempuh dan Kecepatan Dengan Beban

Pada pengujian pada tahap ini, pengujian berupa menghitung waktu tempuh robot pada destinasi 1 meja untuk mendapatkan kecepatan robot dan dilakukan dengan menambah beban seberat 1 Kg atau 1 kantong beras pada robot. Berikut adalah hasil pengujian untuk masing-masing meja pada Tabel 4.12, Tabel 4.13, Tabel 4.14, dan Tabel 4.15 dengan beban 1 Kg.

Tabel 4.12 Pengujian Meja Nomor 1 Beban 1 Kg

Meja 1		Jarak Keseluruhan			842 Cm
Percobaan	Waktu Tempuh 1 (s)	Waktu Tempuh 2 (s)	Total Waktu (s)	Kecepatan (cm/s)	
1	17,34	26,22	43,56	19,33	
2	17,32	25,98	43,30	19,45	
3	16,90	25,57	42,47	19,83	
Rata-rata	17,19	25,92	43,11	19,53	

Tabel 4.13 Pengujian Meja Nomor 2 Beban 1 Kg

Meja 2		Jarak Keseluruhan			844 Cm
Percobaan	Waktu Tempuh 1 (s)	Waktu Tempuh 2 (s)	Total Waktu (s)	Kecepatan (cm/s)	
1	17,33	26,44	43,77	19,28	
2	17,24	25,42	42,66	19,78	
3	17,27	25,60	42,87	19,69	
Rata-rata	17,28	25,82	43,10	19,58	

Tabel 4.14 Pengujian Meja Nomor 3 Beban 1 Kg

Meja 3		Jarak Keseluruhan			1214 cm
Percobaan	Waktu Tempuh 1 (s)	Waktu Tempuh 2 (s)	Total Waktu (s)	Kecepatan (cm/s)	
1	24,14	32,97	57,11	21,26	
2	23,97	33,29	57,26	21,20	
3	23,82	33,61	57,43	21,14	
Rata-rata	23,98	33,29	57,27	21,20	

Tabel 4.15 Pengujian Meja Nomor 4 Beban 1 Kg

Meja 4	Jarak Keseluruhan		1216 cm	
Percobaan	Waktu Tempuh 1 (s)	Waktu Tempuh 2 (s)	Total Waktu (s)	Kecepatan (cm/s)
1	24,00	32,88	56,88	21,38
2	23,65	33,08	56,73	21,43
3	23,92	31,38	55,30	21,99
Rata-rata	23,86	32,45	56,30	21,60

Pada tabel-tabel diatas dapat dicari rata-rata waktu yang ditempuh, dan rata-rata kecepatan robot dalam menuntaskan destinasinya:

- a. Pada Tabel 4.12 rata-rata waktu robot menuju ke meja nomor 1 yakni **17,19 seconds**, rata-rata waktu robot kembali ke posisi start (*home*) yakni **25,92 seconds**, dan rata-rata waktu secara keseluruhan yakni **43,11 seconds**, dan rata-rata kecepatan yang ditempuh yakni **19,33 cm/s**.
- b. Pada Tabel 4.13 rata-rata waktu robot menuju ke meja nomor 2 yakni **17,28 seconds**, rata-rata waktu robot kembali ke posisi start (*home*) yakni **25,82 seconds**, dan rata-rata waktu secara keseluruhan yakni **43,10 seconds**, dan rata-rata kecepatan yang ditempuh yakni **19,58 cm/s**.
- c. Pada Tabel 4.14 rata-rata waktu robot menuju ke meja nomor 3 yakni **23,98 seconds**, rata-rata waktu robot kembali ke posisi start (*home*) yakni **33,29 seconds**, dan rata-rata waktu secara keseluruhan yakni **57,27 seconds**, dan rata-rata kecepatan yang ditempuh yakni **21,20 cm/s**.
- d. Pada Tabel 4.15 rata-rata waktu robot menuju ke meja nomor 4 yakni **23,86 seconds**, rata-rata waktu robot kembali ke posisi start (*home*) yakni

32,45 seconds, dan rata-rata waktu secara keseluruhan yakni **56,30**

seconds, dan rata-rata kecepatan yang ditempuh yakni **21,60 cm/s**.

Selanjutnya pengujian destinasi 1 meja dengan menggunakan beban 2 Kg atau 2 kantong beras pada Robot. Berikut hasil pengujian pada masing-masing meja pada Tabel 4.16, Tabel 4.17, Tabel 4.18, dan Tabel 4.19 dengan beban 2 Kg.

Tabel 4.16 Pengujian Meja Nomor 1 Beban 2 Kg

Meja 1	Jarak Keseluruhan 842 cm			
Percobaan	Waktu Tempuh 1 (s)	Waktu Tempuh 2 (s)	Total Waktu (s)	Kecepatan (cm/s)
1	18,32	27,20	45,52	18,50
2	18,30	26,96	45,26	18,60
3	17,88	26,55	44,43	18,95
Rata-rata	18,17	26,90	45,07	18,68

Tabel 4.17 Pengujian Meja Nomor 2 Beban 2 Kg

Meja 2	Jarak Keseluruhan 844 cm			
Percobaan	Waktu Tempuh 1 (s)	Waktu Tempuh 2 (s)	Total Waktu (s)	Kecepatan (cm/s)
1	18,31	27,42	45,73	18,46
2	18,22	26,40	44,62	18,92
3	18,25	26,58	44,83	18,83
Rata-rata	18,26	26,80	45,06	18,73

Tabel 4.18 Pengujian Meja Nomor 3 Beban 2 Kg

Meja 3	Jarak Keseluruhan 1214 cm			
Percobaan	Waktu Tempuh 1 (s)	Waktu Tempuh 2 (s)	Total Waktu (s)	Kecepatan (cm/s)
1	25,12	33,95	59,07	20,55
2	24,95	34,27	59,22	20,50
3	25,14	34,59	59,73	20,32
Rata-rata	25,07	34,27	59,34	20,46

Tabel 4.19 Pengujian Meja Nomor 4 Beban 2 Kg

Meja 4	Jarak Keseluruhan		1216 cm	
Percobaan	Waktu Tempuh 1 (s)	Waktu Tempuh 2 (s)	Total Waktu (s)	Kecepatan (cm/s)
1	24,98	33,86	58,84	20,67
2	24,63	34,06	58,69	20,72
3	24,90	32,36	57,26	21,24
Rata-rata	24,84	33,43	58,26	20,87

Pada tabel-tabel diatas dapat dicari rata-rata waktu yang ditempuh, dan rata-rata kecepatan robot dalam menuntaskan destinasinya.

- a. Pada Tabel 4.16 rata-rata waktu robot menuju ke meja nomor 1 yakni **18,17 seconds**, rata-rata waktu robot kembali ke posisi start (*home*) yakni **26,90 seconds**, dan rata-rata waktu secara keseluruhan yakni **45,07 seconds**, dan rata-rata kecepatan yang ditempuh yakni **18,68 cm/s**.
- b. Pada Tabel 4.17 rata-rata waktu robot menuju ke meja nomor 2 yakni **18,26 seconds**, rata-rata waktu robot kembali ke posisi start (*home*) yakni **26,80 seconds**, dan rata-rata waktu secara keseluruhan yakni **45,06 seconds**, dan rata-rata kecepatan yang ditempuh yakni **18,73 cm/s**.
- c. Pada Tabel 4.18 rata-rata waktu robot menuju ke meja nomor 3 yakni **25,07 seconds**, rata-rata waktu robot kembali ke posisi start (*home*) yakni **34,27 seconds**, dan rata-rata waktu secara keseluruhan yakni **59,34 seconds**, dan rata-rata kecepatan yang ditempuh yakni **20,46 cm/s**.
- d. Pada Tabel 4.19 rata-rata waktu robot menuju ke meja nomor 4 yakni **24,84 seconds**, rata-rata waktu robot kembali ke posisi start (*home*) yakni **33,43 seconds**, dan rata-rata waktu secara keseluruhan yakni **58,26 seconds**, dan rata-rata kecepatan yang ditempuh yakni **20,87 cm/s**.

Selanjutnya pengujian destinasi 1 meja dengan menggunakan beban 3 Kg atau 3 kantong beras pada Robot. Berikut hasil pengujian pada masing-masing meja pada Tabel 4.20, Tabel 4.21, Tabel 4.22, dan Tabel 4.23 dengan beban 3 Kg.

Tabel 4.20 Pengujian Meja Nomor 1 Beban 3 Kg

Meja 1	Jarak Keseluruhan				842 cm
Percobaan	Waktu Tempuh 1 (s)	Waktu Tempuh 2 (s)	Total Waktu (s)	Kecepatan (cm/s)	
1	20,75	29,63	50,38	16,71	
2	20,73	29,39	50,12	16,80	
3	20,31	28,98	49,29	17,08	
Rata-rata	20,60	29,33	49,93	16,87	

Tabel 4.21 Pengujian Meja Nomor 2 Beban 3 Kg

Meja 2	Jarak Keseluruhan				844 cm
Percobaan	Waktu Tempuh 1 (s)	Waktu Tempuh 2 (s)	Total Waktu (s)	Kecepatan (cm/s)	
1	20,74	29,85	50,59	16,68	
2	20,65	28,83	49,48	17,06	
3	20,68	29,01	49,69	16,99	
Rata-rata	20,69	29,23	49,92	16,91	

Tabel 4.22 Pengujian Meja Nomor 3 Beban 3 Kg

Meja 3	Jarak Keseluruhan				1214 cm
Percobaan	Waktu Tempuh 1 (s)	Waktu Tempuh 2 (s)	Total Waktu (s)	Kecepatan (cm/s)	
1	27,55	36,38	63,93	18,99	
2	27,38	36,70	64,08	18,95	
3	27,57	37,02	64,59	18,80	
Rata-rata	27,50	36,70	64,20	18,91	

Tabel 4.23 Pengujian Meja Nomor 4 Beban 3 Kg

Meja 4	Jarak Keseluruhan		1216 cm	
Percobaan	Waktu Tempuh 1 (s)	Waktu Tempuh 2 (s)	Total Waktu (s)	Kecepatan (cm/s)
1	27,41	36,29	63,70	19,09
2	27,06	36,49	63,55	19,13
3	27,33	34,79	62,12	19,58
Rata-rata	27,27	35,86	63,12	19,27

Pada tabel-tabel diatas dapat dicari rata-rata waktu yang ditempuh, dan rata-rata kecepatan robot dalam menuntaskan destinasinya.

- a. Pada Tabel 4.20 rata-rata waktu robot menuju ke meja nomor 1 yakni **20,60 seconds**, rata-rata waktu robot kembali ke posisi start (*home*) yakni **29,33 seconds**, dan rata-rata waktu secara keseluruhan yakni **49,93 seconds**, dan rata-rata kecepatan yang ditempuh yakni **16,87 cm/s**.
- b. Pada Tabel 4.21 rata-rata waktu robot menuju ke meja nomor 2 yakni **20,69 seconds**, rata-rata waktu robot kembali ke posisi start (*home*) yakni **29,23 seconds**, dan rata-rata waktu secara keseluruhan yakni **49,92 seconds**, dan rata-rata kecepatan yang ditempuh yakni **16,91 cm/s**.
- c. Pada Tabel 4.22 rata-rata waktu robot menuju ke meja nomor 3 yakni **27,50 seconds**, rata-rata waktu robot kembali ke posisi start (*home*) yakni **36,70 seconds**, dan rata-rata waktu secara keseluruhan yakni **64,20 seconds**, dan rata-rata kecepatan yang ditempuh yakni **18,91 cm/s**.
- d. Pada Tabel 4.23 rata-rata waktu robot menuju ke meja nomor 4 yakni **27,27 seconds**, rata-rata waktu robot kembali ke posisi start (*home*) yakni **35,86 seconds**, dan rata-rata waktu secara keseluruhan yakni **63,12 seconds**, dan rata-rata kecepatan yang ditempuh yakni **19,27 cm/s**.

Selanjutnya pengujian destinasi 1 meja dengan menggunakan beban 4 Kg atau 4 kantong beras pada Robot. Berikut hasil pengujian pada masing-masing meja pada Tabel 4.24, Tabel 4.25, Tabel 4.26, dan Tabel 4.27 dengan beban 4 Kg.

Tabel 4.24 Pengujian Meja Nomor 1 Beban 4 Kg

Meja 1	Jarak Keseluruhan				842 Cm
Percobaan	Waktu Tempuh 1 (s)	Waktu Tempuh 2 (s)	Total Waktu (s)	Kecepatan (cm/s)	
1	0	0	0	0	
2	0	0	0	0	
3	0	0	0	0	
Rata-rata	0	0	0	0	

Tabel 4.25 Pengujian Meja Nomor 2 Beban 4 Kg

Meja 2	Jarak Keseluruhan				844 cm
Percobaan	Waktu Tempuh 1 (s)	Waktu Tempuh 2 (s)	Total Waktu (s)	Kecepatan (cm/s)	
1	0	0	0	0	
2	0	0	0	0	
3	0	0	0	0	
Rata-rata	0	0	0	0	

Tabel 4.26 Pengujian Meja Nomor 3 Beban 4 Kg

Meja 3	Jarak Keseluruhan				1214 cm
Percobaan	Waktu Tempuh 1 (s)	Waktu Tempuh 2 (s)	Total Waktu (s)	Kecepatan (cm/s)	
1	0	0	0	0	
2	0	0	0	0	
3	0	0	0	0	
Rata-rata	0	0	0	0	

Tabel 4.27 Pengujian Meja Nomor 4 Beban 4 Kg

Meja 4	Jarak Keseluruhan		1216 cm	
Percobaan	Waktu Tempuh 1 (s)	Waktu Tempuh 2 (s)	Total Waktu (s)	Kecepatan (cm/s)
1	0	0	0	0
2	0	0	0	0
3	0	0	0	0
Rata-rata	0	0	0	0

Pada tabel-tabel diatas pengujian destinasi 1 meja pada beban 4 Kg robot tidak mampu mengantarkan makanan ke meja yang dituju atau kembali ke posisi awal (*home*) dan posisi robot masih di posisi awal (*home*).

4.3.5 Hasil Pengujian Keberhasilan Pada Destinasi 1 Meja

Pada pengujian pada tahap ini terdiri dari dua fase yang digunakan untuk persentase keberhasilan yakni Fase 1 ketika robot mengantarkan makanan ke meja dan Fase 2 ketika robot kembali ke posisi awal (*home*). Berikut adalah tabel *Rule* Keberhasilan Destinasi 1 Meja seperti pada Tabel 4.28.

Tabel 4.28 *Rule* Keberhasilan Destinasi 1 Meja

Destinasi 1 Meja	
Untuk pilihan meja 1 atau 2	
Keterangan	Persentase(%)
Fase 1	
<i>Home</i> ke Perempatan	0 – 25
Perempatan ke Meja	26 – 50
Fase 2	
Meja ke Perempatan	51 – 75
Perempatan ke <i>Home</i>	76 - 100

Destinasi 1 Meja	
Untuk pilihan meja 3 atau 4	
Keterangan	Persentase(%)
Fase 1	
<i>Home</i> ke Perempatan	0 - 10
Perempatan ke Pertigaan	11 - 25
Pertigaan ke Meja	26 - 50
Fase 2	
Meja ke Pertigaan	51 - 60
Pertigaan ke Perempatan	61 - 75
Perempatan ke <i>Home</i>	76 - 100

1. Hasil Pengujian Keberhasilan Tanpa Beban

Pada pengujian pada tahap ini, pengujian berupa mengamati berapa persen robot dari posisi awal (*home*) mengantarkan makanan ke meja yang dituju lalu kembali ke posisi awal (*home*) tanpa membawa beban. Robot dikatakan berhasil secara keseluruhan jika robot dapat menyelesaikan tugas hingga persentase 100% sesuai *Rule* Keberhasilan Destinasi 1 Meja. Berikut tabel keberhasilan destinasi 1 meja Tanpa Beban pada Tabel 4.29.

Tabel 4.29 Pengujian Keberhasilan Tanpa Beban

Meja 1		Meja 2	
Percobaan	Keberhasilan (%)	Percobaan	Keberhasilan (%)
1	100	1	55
2	100	2	100
3	100	3	80
4	100	4	100
5	100	5	100
6	55	6	100
7	100	7	100
8	100	8	100
9	100	9	30
10	100	10	100
11	100	11	60
12	100	12	100
Rata-rata	96	Rata-rata	85

Meja 3		Meja 4	
Percobaan	Keberhasilan (%)	Percobaan	Keberhasilan (%)
1	60	1	65
2	100	2	100
3	100	3	100
4	100	4	100
5	100	5	100
6	100	6	100
7	100	7	100

Percobaan	Keberhasilan (%)
8	60
9	100
10	100
11	100
12	100
Rata-rata	93

Percobaan	Keberhasilan (%)
8	100
9	100
10	100
11	100
12	60
Rata-rata	94

Keberhasilan yang dicapai pada destinasi 1 meja Tanpa Beban terdapat beberapa yang kurang berhasil atau dibawah nilai 100% yakni dengan keterangan sebagai berikut:

- a. Pada keberhasilan 55% untuk meja 1 percobaan ke 5, robot ketika putar balik kurang bisa menyesuaikan dengan garis hitam sehingga pembacaan garis error.
- b. Pada keberhasilan 55% untuk meja 2 percobaan ke 2, robot ketika putar balik kurang bisa menyesuaikan dengan garis hitam sehingga pembacaan garis error.
- c. Pada keberhasilan 80% untuk meja 2 percobaan ke 3, robot berhenti membaca garis saat akan mendekati home.
- d. Pada keberhasilan 30% untuk meja 2 percobaan ke 9, robot berhenti membaca garis setelah belok kiri di perempatan.
- e. Pada keberhasilan 60% untuk meja 2 percobaan ke 11, robot berhenti membaca garis saat akan mendekati perempatan.
- f. Pada keberhasilan 60% untuk meja 3 percobaan ke 1, robot berhenti membaca garis saat akan mendekati perempatan.

- g. Pada keberhasilan 60% untuk meja 3 percobaan ke 8, robot berhenti membaca garis saat akan mendekati perempatan.
- h. Pada keberhasilan 65% untuk meja 3 percobaan ke 1, robot berhenti membaca garis setelah putar melewati pertigaan.
- i. Pada keberhasilan 60% untuk meja 4 percobaan ke 12, robot berhenti membaca garis saat akan mendekati perempatan.

2. Hasil Pengujian Keberhasilan Dengan Beban

Pada pengujian pada tahap ini, pengujian berupa mengamati berapa persen robot dari posisi awal (*home*) mengantarkan makanan ke meja yang dituju lalu kembali ke posisi awal (*home*) dengan membawa beban 1 Kg, 2 Kg, 3 Kg, dan 4 Kg. Robot dikatakan berhasil secara keseluruhan jika robot dapat menyelesaikan tugas hingga persentase 100% sesuai *Rule* Keberhasilan Destinasi 1 Meja. Berikut tabel keberhasilan destinasi 1 meja dengan beban pada Tabel 4.30, Tabel 4.31, Tabel 4.32, dan Tabel 4.33.

Tabel 4.30 Pengujian Keberhasilan Beban 1 Kg

Meja 1		Meja 2	
Percobaan	Keberhasilan (%)	Percobaan	Keberhasilan (%)
1	100	1	100
2	100	2	100
3	100	3	100
Rata-rata	100	Rata-rata	100

Meja 3		Meja 4	
Percobaan	Keberhasilan (%)	Percobaan	Keberhasilan (%)
1	100	1	100
2	100	2	100
3	100	3	100
Rata-rata	100	Rata-rata	100

Tabel 4.31 Pengujian Keberhasilan Beban 2 Kg

Meja 1		Meja 2	
Percobaan	Keberhasilan (%)	Percobaan	Keberhasilan (%)
1	100	1	100
2	100	2	100
3	100	3	100
Rata-rata	100	Rata-rata	100

Meja 3		Meja 4	
Percobaan	Keberhasilan (%)	Percobaan	Keberhasilan (%)
1	100	1	100
2	100	2	100
3	100	3	100
Rata-rata	100	Rata-rata	100

Tabel 4.32 Pengujian Keberhasilan Beban 3 Kg

Meja 1		Meja 2	
Percobaan	Keberhasilan (%)	Percobaan	Keberhasilan (%)
1	100	1	100
2	100	2	100
3	100	3	100
Rata-rata	100	Rata-rata	100

Meja 3		Meja 4	
Percobaan	Keberhasilan (%)	Percobaan	Keberhasilan (%)
1	100	1	100
2	100	2	100
3	100	3	100
Rata-rata	100	Rata-rata	100

Tabel 4.33 Pengujian Keberhasilan Beban 4 Kg

Meja 1

Percobaan	Keberhasilan (%)
1	0
2	0
3	0
Rata-rata	0

Meja 2

Percobaan	Keberhasilan (%)
1	0
2	0
3	0
Rata-rata	0

Meja 3

Percobaan	Keberhasilan (%)
1	0
2	0
3	0
Rata-rata	0

Meja 4

Percobaan	Keberhasilan (%)
1	0
2	0
3	0
Rata-rata	0

Keberhasilan yang dicapai pada destinasi 1 meja dengan beban 1 Kg, 2 Kg, 3 Kg robot berhasil secara keseluruhan 100% mengantarkan makanan dan kembali ke posisi awal, sedangkan pada beban 4 Kg robot tidak berhasil secara keseluruhan, karena robot tidak dapat bergerak dari posisi awal (*home*).

4.4 Uji Destinasi 2 Meja

4.4.1 Tujuan Uji Destinasi 2 Meja

Pengujian ini digunakan untuk mengetahui waktu tempuh dari posisi start (*home*) ke meja pertama, waktu tempuh meja pertama ke meja kedua dan kembali lagi dari meja kedua ke posisi semula, kecepatan robot dalam melakukan perjalanan berangkat, menuju ke destinasi selanjutnya hingga kembali serta persentase keberhasilan robot dalam menuntaskan destinasinya secara keseluruhan. Dengan

diketuinya kesemua itu, dapat dilihat rata-rata waktu tempuh, kecepatan, dan keberhasilan robot.

4.4.2 Alat yang Digunakan Pada Uji Destinasi 2 Meja

Peralatan yang dibutuhkan untuk pengujian ini adalah sebagai berikut:

1. *Mobile Robot Vision*.
2. Laptop.
3. Program Arduino IDE.
4. 4 Kantong Beras (setiap kantong berat 1 Kg).
5. Stopwatch (menggunakan Smartphone).

4.4.3 Prosedur Pengujian Pada Uji Destinasi 2 Meja

Langkah-langkah yang dilakukan dalam melakukan pengujian tahap ini sebagai berikut:

1. Mempersiapkan *Mobile Robot Vision* di posisi start (*home*) serta mengaktifkan OpenCV pada Raspberry Pi 3
2. Melakukan transfer program dari laptop ke robot (lebih baik sudah dilakukan dari awal)
3. Pengujian pada tahap ini dilakukan dengan dua cara yakni robot diuji Tanpa Beban dan dengan beban. Pada pengujian robot Tanpa Beban setiap destinasi dengan urutan sebagai berikut:
 - a. 1 dan 2, 2 dan 1 sebanyak 6 kali
 - b. 1 dan 3, 3 dan 1 sebanyak 6 kali
 - c. 1 dan 4, 4 dan 1 sebanyak 6 kali

- d. 2 dan 3, 3 dan 2 sebanyak 6 kali
- e. 2 dan 4, 4 dan 2 sebanyak 6 kali
- f. 3 dan 4, 4 dan 3 sebanyak 6 kali

Percobaan diatas juga disertakan keberhasilannya, sedangkan, dengan beban pengujian robot dilakukan berdasarkan jarak terdekat, menengah, dan terjauh.

Berikut destinasi yang dipilih berdasarkan jarak:

- a. 1 dan 2, 2 dan 1 sebanyak 3 kali (jarak terpendek)
- b. 2 dan 3, 3 dan 2 sebanyak 3 kali (jarak menengah)
- c. 3 dan 4, 4 dan 3 sebanyak 3 kali (jarak terjauh)

Pada percobaan diatas masing-masing destinasi membawa dengan beban yang dibawa berurutan mulai dari 1 Kg, 2 Kg, 3 Kg dan 4 Kg beserta keberhasilannya.

- 4. Mengamati pergerakan robot dan hitung waktu tempuh dengan menggunakan stopwatch.

4.4.4 Hasil Pengujian Waktu Tempuh dan Kecepatan Pada Uji Destinasi 2 Meja

1. Hasil Pengujian Waktu Tempuh dan Kecepatan Tanpa Beban

Pada pengujian pada tahap ini, pengujian berupa menghitung waktu tempuh robot pada destinasi 2 meja untuk mendapatkan kecepatan robot dan dilakukan tanpa adanya beban pada robot. Berikut adalah hasil pengujian untuk masing-masing meja pada Tabel 4.34, Tabel 4.35, Tabel 4.36, Tabel 4.37, Tabel 4.38, dan Tabel 4.39.

Tabel 4.34 Pengujian Urutan 1 dan 2, 2 dan 1 Tanpa Beban

Jarak keseluruhan : 1252 cm

Percobaan	Waktu Tempuh 1 (s)	Waktu Tempuh 1-2 (s)	Waktu Tempuh 2 (s)	Total Waktu (s)	Kecepatan (cm/s)
Destinasi Meja 1 - 2					
1	16,28	16,45	24,48	57,21	21,88
2	15,58	16,56	24,54	56,68	22,09
3	16,02	16,79	24,42	57,23	21,88
4	16,49	16,79	24,50	57,78	21,67
5	15,95	16,99	24,36	57,30	21,85
6	16,42	16,78	11,54	44,74	27,98
Rata-rata	16,12	16,73	22,31	55,16	22,89
Destinasi Meja 2 - 1					
1	16,16	16,91	19,72	52,79	23,72
2	16,42	16,86	25,86	59,14	21,17
3	16,52	17,04	25,52	59,08	21,19
4	16,57	17,47	25,72	59,76	20,95
5	16,93	17,52	25,02	59,47	21,05
6	16,82	17,25	25,84	59,91	20,90
Rata-rata	16,57	17,18	24,61	58,36	21,50

Tabel 4.35 Pengujian Urutan 1 dan 3, 3 dan 1 Tanpa Beban

Jarak keseluruhan : 1622 cm

Percobaan	Waktu Tempuh 1 (s)	Waktu Tempuh 1-2 (s)	Waktu Tempuh 2 (s)	Total Waktu (s)	Kecepatan (cm/s)
Destinasi Meja 1 - 3					
1	16,49	28,15	18,68	63,32	25,62
2	16,26	28,24	30,93	75,43	21,50
3	15,97	28,40	31,03	75,40	21,51
4	16,31	24,19	0,00	40,50	40,05
5	15,77	27,91	30,85	74,53	21,76
6	16,19	28,74	30,58	75,51	21,48
Rata-rata	16,17	27,61	23,68	67,45	25,32
Destinasi Meja 3 - 1					
1	22,64	29,40	25,66	77,70	20,88
2	21,97	29,05	25,43	76,45	21,22

Percobaan	Waktu Tempuh 1 (s)	Waktu Tempuh 1-2 (s)	Waktu Tempuh 2 (s)	Total Waktu (s)	Kecepatan (cm/s)
3	22,50	30,04	25,68	78,22	20,74
4	22,27	29,77	6,61	58,65	27,66
5	22,44	29,91	25,63	77,98	20,80
6	22,18	29,80	25,51	77,49	20,93
Rata-rata	22,33	29,66	22,42	74,42	22,04

Tabel 4.36 Pengujian Urutan 1 dan 4, 4 dan 1 Tanpa Beban

Jarak keseluruhan : 1624 cm

Percobaan	Waktu Tempuh 1 (s)	Waktu Tempuh 1-2 (s)	Waktu Tempuh 2 (s)	Total Waktu (s)	Kecepatan (cm/s)
Destinasi Meja 1 - 4					
1	16,02	28,56	18,42	63,00	25,78
2	16,02	10,32	0,00	26,34	61,66
3	16,17	28,45	30,36	74,98	21,66
4	16,44	28,75	30,81	76,00	21,37
5	16,10	28,75	30,59	75,44	21,53
6	16,26	29,12	31,15	76,53	21,22
Rata-rata	16,17	25,66	23,56	65,38	28,87
Destinasi Meja 4 - 1					
1	23,23	24,60	0,00	47,83	33,95
2	23,06	29,37	25,59	78,02	20,82
3	23,34	29,54	25,87	78,75	20,62
4	22,99	30,24	23,32	76,55	21,21
5	22,89	30,67	25,71	79,27	20,49
6	23,20	29,38	25,52	78,10	20,79
Rata-rata	23,12	28,97	21,00	73,09	22,98

Tabel 4.37 Pengujian Urutan 2 dan 3, 3 dan 2 Tanpa Beban

Jarak keseluruhan : 1624 cm

Percobaan	Waktu Tempuh 1 (s)	Waktu Tempuh 1-2 (s)	Waktu Tempuh 2 (s)	Total Waktu (s)	Kecepatan (cm/s)
Destinasi Meja 2 - 3					
1	16,40	29,51	31,60	77,51	20,95
2	16,00	29,45	14,61	60,06	27,04
3	16,62	29,00	31,41	77,03	21,08
4	16,41	14,40	0,00	30,81	52,71
5	16,48	6,59	0,00	23,07	70,39
6	16,54	29,01	31,96	77,51	20,95
Rata-rata	16,41	22,99	18,26	57,67	35,52
Destinasi Meja 3 - 2					
1	22,72	28,77	24,70	76,19	21,32
2	22,37	10,93	0,00	33,30	48,77
3	22,59	29,42	25,41	77,42	20,98
4	22,69	17,91	0,00	40,60	40,00
5	22,40	29,99	26,09	78,48	20,69
6	22,67	28,85	26,10	77,62	20,92
Rata-rata	22,57	24,31	17,05	63,94	28,78

Tabel 4.38 Pengujian Urutan 2 dan 4, 4 dan 2 Tanpa Beban

Jarak keseluruhan : 1626 cm

Percobaan	Waktu Tempuh 1 (s)	Waktu Tempuh 1-2 (s)	Waktu Tempuh 2 (s)	Total Waktu (s)	Kecepatan (cm/s)
Destinasi Meja 2 - 4					
1	16,51	29,52	30,81	76,84	21,16
2	16,44	29,24	30,84	76,52	21,25
3	16,46	29,60	31,21	77,27	21,04
4	16,74	6,77	0,00	23,51	69,16
5	17,36	29,69	28,54	75,59	21,51
6	16,24	29,31	31,21	76,76	21,18
Rata-rata	16,63	25,69	25,44	67,75	29,22
Destinasi Meja 4 - 2					
1	22,68	27,00	0,00	49,68	32,73
2	23,25	28,61	25,23	77,09	21,09

Percobaan	Waktu Tempuh 1 (s)	Waktu Tempuh 1-2 (s)	Waktu Tempuh 2 (s)	Total Waktu (s)	Kecepatan (cm/s)
3	22,98	29,00	10,68	62,66	25,95
4	23,11	29,01	25,08	77,20	21,06
5	23,17	28,97	25,14	77,28	21,04
6	23,09	29,00	25,16	77,25	21,05
Rata-rata	23,05	28,60	18,55	70,19	23,82

Tabel 4.39 Pengujian Urutan 3 dan 4, 4 dan 3 Tanpa Beban

Jarak keseluruhan : 1630 cm

Percobaan	Waktu Tempuh 1 (s)	Waktu Tempuh 1-2 (s)	Waktu Tempuh 2 (s)	Total Waktu (s)	Kecepatan (cm/s)
Destinasi Meja 3 - 4					
1	22,39	11,47	0,00	33,86	48,14
2	22,34	17,45	31,52	71,31	22,86
3	22,58	17,20	31,22	71,00	22,96
4	22,36	16,96	31,02	70,34	23,17
5	22,52	17,07	31,96	71,55	22,78
6	22,57	17,26	31,12	70,95	22,97
Rata-rata	22,46	16,24	26,14	64,84	27,15
Destinasi Meja 4 -3					
1	22,97	16,99	31,63	71,59	22,77
2	23,51	17,21	31,98	72,70	22,42
3	23,44	17,14	31,74	72,32	22,54
4	23,35	17,12	31,61	72,08	22,61
5	23,37	17,07	31,60	72,04	22,63
6	22,75	17,41	31,40	71,56	22,78
Rata-rata	23,23	17,16	31,66	72,05	22,62

Pada tabel-tabel diatas dapat dicari rata-rata waktu yang ditempuh, dan rata-rata kecepatan robot dalam menuntaskan destinasinya, bisa dilihat pada tabel sebagai berikut:

- a. Pada Tabel 4.34 rata-rata waktu robot menuju ke meja nomor 1 yakni **16,12 seconds**, rata-rata waktu robot menuju ke meja nomor 2 yakni **16,73 seconds**, rata-rata waktu robot kembali ke posisi start (*home*) yakni **22,31 seconds**, dan rata-rata waktu secara keseluruhan yakni **50,16 seconds**, dan rata-rata kecepatan yang ditempuh yakni **22,89 cm/s**, sedangkan, rata-rata waktu robot menuju ke meja nomor 2 yakni **16,57 seconds**, rata-rata waktu robot menuju ke meja nomor 1 yakni **17,18 seconds**, rata-rata waktu robot kembali ke posisi start (*home*) yakni **24,61 seconds**, dan rata-rata waktu secara keseluruhan yakni **58,36 seconds**, dan rata-rata kecepatan yang ditempuh yakni **21,50 cm/s**.
- b. Pada Tabel 4.35 rata-rata waktu robot menuju ke meja nomor 1 yakni **16,17 seconds**, rata-rata waktu robot menuju ke meja nomor 3 yakni **27,61 seconds**, rata-rata waktu robot kembali ke posisi start (*home*) yakni **23,68 seconds**, dan rata-rata waktu secara keseluruhan yakni **67,45 seconds**, dan rata-rata kecepatan yang ditempuh yakni **25,32 cm/s**, sedangkan, rata-rata waktu robot menuju ke meja nomor 3 yakni **22,33 seconds**, rata-rata waktu robot menuju ke meja nomor 1 yakni **29,66 seconds**, rata-rata waktu robot kembali ke posisi start (*home*) yakni **22,42 seconds**, dan rata-rata waktu secara keseluruhan yakni **74,42 seconds**, dan rata-rata kecepatan yang ditempuh yakni **22,04 cm/s**.
- c. Pada Tabel 4.36 rata-rata waktu robot menuju ke meja nomor 1 yakni **16,17 seconds**, rata-rata waktu robot menuju ke meja nomor 4 yakni **25,66 seconds**, rata-rata waktu robot kembali ke posisi start (*home*) yakni **23,56 seconds**, dan rata-rata waktu secara keseluruhan yakni **65,38**

seconds, dan rata-rata kecepatan yang ditempuh yakni **28,87 cm/s**, sedangkan, rata-rata waktu robot menuju ke meja nomor 4 yakni **23,12 seconds**, rata-rata waktu robot menuju ke meja nomor 1 yakni **28,97 seconds**, rata-rata waktu robot kembali ke posisi start (*home*) yakni **21,00 seconds**, dan rata-rata waktu secara keseluruhan yakni **73,09 seconds**, dan rata-rata kecepatan yang ditempuh yakni **22,98 cm/s**.

- d. Pada Tabel 4.37 rata-rata waktu robot menuju ke meja nomor 2 yakni **16,41 seconds**, rata-rata waktu robot menuju ke meja nomor 3 yakni **22,99 seconds**, rata-rata waktu robot kembali ke posisi start (*home*) yakni **18,26 seconds**, dan rata-rata waktu secara keseluruhan yakni **57,67 seconds**, dan rata-rata kecepatan yang ditempuh yakni **35,52 cm/s**, sedangkan, rata-rata waktu robot menuju ke meja nomor 3 yakni **22,57 seconds**, rata-rata waktu robot menuju ke meja nomor 2 yakni **24,31 seconds**, rata-rata waktu robot kembali ke posisi start (*home*) yakni **17,05 seconds**, dan rata-rata waktu secara keseluruhan yakni **63,94 seconds**, dan rata-rata kecepatan yang ditempuh yakni **28,78 cm/s**.

- e. Pada Tabel 4.38 rata-rata waktu robot menuju ke meja nomor 2 yakni **16,63 seconds**, rata-rata waktu robot menuju ke meja nomor 4 yakni **25,69 seconds**, rata-rata waktu robot kembali ke posisi start (*home*) yakni **25,44 seconds**, dan rata-rata waktu secara keseluruhan yakni **67,75 seconds**, dan rata-rata kecepatan yang ditempuh yakni **29,22 cm/s**, sedangkan, rata-rata waktu robot menuju ke meja nomor 4 yakni **23,05 seconds**, rata-rata waktu robot menuju ke meja nomor 2 yakni **28,60 seconds**, rata-rata waktu robot kembali ke posisi start (*home*) yakni **18,55**

seconds, dan rata-rata waktu secara keseluruhan yakni **70,19 seconds**, dan rata-rata kecepatan yang ditempuh yakni **23,82 cm/s**.

- f. Pada Tabel 4.39 rata-rata waktu robot menuju ke meja nomor 3 yakni **22,46 seconds**, rata-rata waktu robot menuju ke meja nomor 4 yakni **16,24 seconds**, rata-rata waktu robot kembali ke posisi start (*home*) yakni **26,14 seconds**, dan rata-rata waktu secara keseluruhan yakni **64,84 seconds**, dan rata-rata kecepatan yang ditempuh yakni **27,15 cm/s**, sedangkan, rata-rata waktu robot menuju ke meja nomor 4 yakni **23,23 seconds**, rata-rata waktu robot menuju ke meja nomor 3 yakni **17,16 seconds**, rata-rata waktu robot kembali ke posisi start (*home*) yakni **31,66 seconds**, dan rata-rata waktu secara keseluruhan yakni **72,05 seconds**, dan rata-rata kecepatan yang ditempuh yakni **22,62 cm/s**.

2. Hasil Pengujian Waktu Tempuh dan Kecepatan Dengan Beban

Pada pengujian pada tahap ini, pengujian berupa menghitung waktu tempuh robot pada destinasi 2 meja untuk mendapatkan kecepatan robot dan dilakukan dengan adanya beban pada robot. Berikut adalah hasil pengujian untuk masing-masing meja pada Tabel 4.39, Tabel 4.40, dan Tabel 4.41 dengan beban 1 Kg.

Tabel 4.39 Pengujian Urutan 1 dan 2, 2 dan 1 Beban 1 Kg

Jarak keseluruhan : 1252 cm

Percobaan	Waktu Tempuh 1 (s)	Waktu Tempuh 1-2 (s)	Waktu Tempuh 2 (s)	Total Waktu (s)	Kecepatan (cm/s)
Destinasi Meja 1 - 2					
1	16,58	17,56	25,54	59,68	20,98
2	17,02	17,79	25,42	60,23	20,79
3	17,49	17,79	25,50	60,78	20,60

Percobaan	Waktu Tempuh 1 (s)	Waktu Tempuh 1-2 (s)	Waktu Tempuh 2 (s)	Total Waktu (s)	Kecepatan (cm/s)
Rata-rata	17,03	17,71	25,49	60,23	20,79
Destinasi Meja 2 - 1					
1	17,52	18,04	26,52	62,08	20,17
2	17,57	18,47	26,72	62,76	19,95
3	17,93	18,52	26,02	62,47	20,04
Rata-rata	17,67	18,34	26,42	62,44	20,05

Tabel 4.40 Pengujian Urutan 2 dan 3, 3 dan 2 Beban 1 Kg

Jarak keseluruhan : 1624 cm

Percobaan	Waktu Tempuh 1 (s)	Waktu Tempuh 1-2 (s)	Waktu Tempuh 2 (s)	Total Waktu (s)	Kecepatan (cm/s)
Destinasi Meja 2 - 3					
1	17,00	30,45	33,06	80,51	20,17
2	17,62	30,00	32,41	80,03	20,29
3	17,41	30,12	32,45	79,98	20,31
Rata-rata	17,34	30,19	32,64	80,17	20,26
Destinasi Meja 3 - 2					
1	23,59	30,42	26,41	80,42	20,19
2	23,69	30,77	26,87	81,33	19,97
3	23,40	30,99	27,09	81,48	19,93
Rata-rata	23,56	30,73	26,79	81,08	20,03

Tabel 4.41 Pengujian Urutan 3 dan 4, 4 dan 3 Beban 1 Kg

Jarak keseluruhan : 1630 cm

Percobaan	Waktu Tempuh 1 (s)	Waktu Tempuh 1-2 (s)	Waktu Tempuh 2 (s)	Total Waktu (s)	Kecepatan (cm/s)
Destinasi Meja 3 - 4					
1	23,34	18,45	32,52	74,31	21,94
2	23,58	18,20	32,22	74,00	22,03
3	23,36	17,96	32,02	73,34	22,23
Rata-rata	23,43	18,20	32,25	73,88	22,06
Destinasi Meja 4 - 3					

Percobaan	Waktu Tempuh 1 (s)	Waktu Tempuh 1-2 (s)	Waktu Tempuh 2 (s)	Total Waktu (s)	Kecepatan (cm/s)
1	24,44	18,14	32,74	75,32	21,64
2	24,35	18,12	32,61	75,08	21,71
3	24,37	18,07	32,60	75,04	21,72
Rata-rata	24,39	18,11	32,65	75,15	21,69

Pada tabel-tabel diatas dapat dicari rata-rata waktu yang ditempuh, dan rata-rata kecepatan robot dalam menuntaskan destinasinya, bisa dilihat pada tabel sebagai berikut:

- a. Pada Tabel 4.39 rata-rata waktu robot menuju ke meja nomor 1 yakni **17,03 seconds**, rata-rata waktu robot menuju ke meja nomor 2 yakni **17,71 seconds**, rata-rata waktu robot kembali ke posisi start (*home*) yakni **25,49 seconds**, dan rata-rata waktu secara keseluruhan yakni **60,23 seconds**, dan rata-rata kecepatan yang ditempuh yakni **20,79 cm/s**, sedangkan, rata-rata waktu robot menuju ke meja nomor 2 yakni **17,67 seconds**, rata-rata waktu robot menuju ke meja nomor 1 yakni **18,34 seconds**, rata-rata waktu robot kembali ke posisi start (*home*) yakni **26,42 seconds**, dan rata-rata waktu secara keseluruhan yakni **62,44 seconds**, dan rata-rata kecepatan yang ditempuh yakni **20,05 cm/s**.
- b. Pada Tabel 4.40 rata-rata waktu robot menuju ke meja nomor 2 yakni **17,34 seconds**, rata-rata waktu robot menuju ke meja nomor 3 yakni **30,19 seconds**, rata-rata waktu robot kembali ke posisi start (*home*) yakni **32,64 seconds**, dan rata-rata waktu secara keseluruhan yakni **80,17 seconds**, dan rata-rata kecepatan yang ditempuh yakni **20,26 cm/s**, sedangkan, rata-rata waktu robot menuju ke meja nomor 3 yakni **23,56 seconds**, rata-rata waktu

robot menuju ke meja nomor 2 yakni **30,73 seconds**, rata-rata waktu robot kembali ke posisi start (*home*) yakni **26,79 seconds**, dan rata-rata waktu secara keseluruhan yakni **81,08 seconds**, dan rata-rata kecepatan yang ditempuh yakni **20,03 cm/s**.

- c. Pada Tabel 4.41 rata-rata waktu robot menuju ke meja nomor 3 yakni **24,43 seconds**, rata-rata waktu robot menuju ke meja nomor 4 yakni **18,20 seconds**, rata-rata waktu robot kembali ke posisi start (*home*) yakni **32,25 seconds**, dan rata-rata waktu secara keseluruhan yakni **73,88 seconds**, dan rata-rata kecepatan yang ditempuh yakni **22,06 cm/s**, sedangkan, rata-rata waktu robot menuju ke meja nomor 4 yakni **24,39 seconds**, rata-rata waktu robot menuju ke meja nomor 3 yakni **18,11 seconds**, rata-rata waktu robot kembali ke posisi start (*home*) yakni **32,65 seconds**, dan rata-rata waktu secara keseluruhan yakni **75,15 seconds**, dan rata-rata kecepatan yang ditempuh yakni **21,69 cm/s**.

Selanjutnya pengujian destinasi 2 meja dengan beban 2 Kg pada Tabel 4.42, Tabel 4.43, dan Tabel 4.44.

Tabel 4.42 Pengujian Urutan 1 dan 2, 2 dan 1 Beban 2 Kg

Jarak keseluruhan : 1252 cm

Percobaan	Waktu Tempuh 1 (s)	Waktu Tempuh 1-2 (s)	Waktu Tempuh 2 (s)	Total Waktu (s)	Kecepatan (cm/s)
Destinasi Meja 1 - 2					
1	17,56	18,54	26,52	62,62	19,99
2	18,00	18,77	26,40	63,17	19,82
3	18,47	18,77	26,48	63,72	19,65
Rata-rata	18,01	18,69	26,47	63,17	19,82
Destinasi Meja 2 - 1					
1	18,50	19,02	27,50	65,02	19,26
2	18,55	19,45	27,70	65,70	19,06

Percobaan	Waktu Tempuh 1 (s)	Waktu Tempuh 1-2 (s)	Waktu Tempuh 2 (s)	Total Waktu (s)	Kecepatan (cm/s)
3	18,91	19,50	27,00	65,41	19,14
Rata-rata	18,65	19,32	27,40	65,38	19,15

Tabel 4.43 Pengujian Urutan 2 dan 3, 3 dan 2 Beban 2 Kg

Jarak keseluruhan : 1624 cm

Percobaan	Waktu Tempuh 1 (s)	Waktu Tempuh 1-2 (s)	Waktu Tempuh 2 (s)	Total Waktu (s)	Kecepatan (cm/s)
Destinasi Meja 2 - 3					
1	17,98	31,43	34,04	83,45	19,46
2	18,60	30,98	33,39	82,97	19,57
3	18,39	31,10	33,43	82,92	19,59
Rata-rata	18,32	31,17	33,62	83,11	19,54
Destinasi Meja 3 - 2					
1	24,57	31,40	27,39	83,36	19,48
2	24,67	31,75	27,85	84,27	19,27
3	24,38	31,97	28,07	84,42	19,24
Rata-rata	24,54	31,71	27,77	84,02	19,33

Tabel 4.44 Pengujian Urutan 3 dan 4, 4 dan 3 Beban 2 Kg

Jarak keseluruhan : 1630 cm

Percobaan	Waktu Tempuh 1 (s)	Waktu Tempuh 1-2 (s)	Waktu Tempuh 2 (s)	Total Waktu (s)	Kecepatan (cm/s)
Destinasi Meja 3 - 4					
1	24,32	19,43	33,50	77,25	21,10
2	24,56	19,18	33,20	76,94	21,19
3	24,34	18,94	33,00	76,28	21,37
Rata-rata	24,41	19,18	33,23	76,82	21,22
Destinasi Meja 4 -3					
1	25,42	19,12	33,72	78,26	20,83
2	25,33	19,10	33,59	78,02	20,89
3	25,35	19,05	33,58	77,98	20,90
Rata-rata	25,37	19,09	33,63	78,09	20,87

Pada tabel-tabel diatas dapat dicari rata-rata waktu yang ditempuh, dan rata-rata kecepatan robot dalam menuntaskan destinasinya, bisa dilihat pada tabel sebagai berikut:

- a. Pada Tabel 4.42 rata-rata waktu robot menuju ke meja nomor 1 yakni **18,01 seconds**, rata-rata waktu robot menuju ke meja nomor 2 yakni **18,69 seconds**, rata-rata waktu robot kembali ke posisi start (*home*) yakni **26,47 seconds**, dan rata-rata waktu secara keseluruhan yakni **63,17 seconds**, dan rata-rata kecepatan yang ditempuh yakni **19,82 cm/s**, sedangkan, rata-rata waktu robot menuju ke meja nomor 2 yakni **18,65 seconds**, rata-rata waktu robot menuju ke meja nomor 1 yakni **19,32 seconds**, rata-rata waktu robot kembali ke posisi start (*home*) yakni **27,40 seconds**, dan rata-rata waktu secara keseluruhan yakni **65,38 seconds**, dan rata-rata kecepatan yang ditempuh yakni **19,15 cm/s**.
- b. Pada Tabel 4.43 rata-rata waktu robot menuju ke meja nomor 2 yakni **18,32 seconds**, rata-rata waktu robot menuju ke meja nomor 3 yakni **31,17 seconds**, rata-rata waktu robot kembali ke posisi start (*home*) yakni **33,62 seconds**, dan rata-rata waktu secara keseluruhan yakni **83,11 seconds**, dan rata-rata kecepatan yang ditempuh yakni **19,54 cm/s**, sedangkan, rata-rata waktu robot menuju ke meja nomor 3 yakni **24,54 seconds**, rata-rata waktu robot menuju ke meja nomor 2 yakni **31,71 seconds**, rata-rata waktu robot kembali ke posisi start (*home*) yakni **27,77 seconds**, dan rata-rata waktu secara keseluruhan yakni **84,02 seconds**, dan rata-rata kecepatan yang ditempuh yakni **19,33 cm/s**.

- c. Pada Tabel 4.44 rata-rata waktu robot menuju ke meja nomor 3 yakni **24,41 seconds**, rata-rata waktu robot menuju ke meja nomor 4 yakni **19,18 seconds**, rata-rata waktu robot kembali ke posisi start (*home*) yakni **33,23 seconds**, dan rata-rata waktu secara keseluruhan yakni **76,82 seconds**, dan rata-rata kecepatan yang ditempuh yakni **21,22 cm/s**, sedangkan, rata-rata waktu robot menuju ke meja nomor 4 yakni **23,57 seconds**, rata-rata waktu robot menuju ke meja nomor 3 yakni **19,09 seconds**, rata-rata waktu robot kembali ke posisi start (*home*) yakni **33,63 seconds**, dan rata-rata waktu secara keseluruhan yakni **78,09 seconds**, dan rata-rata kecepatan yang ditempuh yakni **20,87 cm/s**.

Selanjutnya pengujian destinasi 2 meja dengan beban 3 Kg pada Tabel 4.45, Tabel 4.46, dan Tabel 4.47.

Tabel 4.45 Pengujian Urutan 1 dan 2, 2 dan 1 Beban 3 Kg

Jarak keseluruhan : 1252 cm

Percobaan	Waktu Tempuh 1 (s)	Waktu Tempuh 1-2 (s)	Waktu Tempuh 2 (s)	Total Waktu (s)	Kecepatan (cm/s)
Destinasi Meja 1 - 2					
1	19,99	20,97	28,95	69,91	17,91
2	20,43	21,20	28,83	70,46	17,77
3	20,90	21,20	28,91	71,01	17,63
Rata-rata	20,44	21,12	28,90	70,46	17,77
Destinasi Meja 2 -1					
1	20,93	21,45	29,93	72,31	17,31
2	20,98	21,88	30,13	72,99	17,15
3	21,34	21,93	29,43	72,70	17,22
Rata-rata	21,08	21,75	29,83	72,67	17,23

Tabel 4.46 Pengujian Urutan 2 dan 3, 3 dan 2 Beban 3 Kg

Jarak keseluruhan : 1624 cm

Percobaan	Waktu Tempuh 1 (s)	Waktu Tempuh 1-2 (s)	Waktu Tempuh 2 (s)	Total Waktu (s)	Kecepatan (cm/s)
Destinasi Meja 2 - 3					
1	20,41	33,86	36,47	90,74	17,90
2	21,03	33,41	35,82	90,26	17,99
3	20,82	33,53	35,86	90,21	18,00
Rata-rata	20,75	33,60	36,05	90,40	17,96
Destinasi Meja 3 - 2					
1	27,00	33,83	29,82	90,65	17,92
2	27,10	34,18	30,28	91,56	17,74
3	26,81	34,40	30,50	91,71	17,71
Rata-rata	26,97	34,14	30,20	91,31	17,79

Tabel 4.47 Pengujian Urutan 3 dan 4, 4 dan 3 Beban 3 Kg

Jarak keseluruhan : 1630 cm

Percobaan	Waktu Tempuh 1 (s)	Waktu Tempuh 1-2 (s)	Waktu Tempuh 2 (s)	Total Waktu (s)	Kecepatan (cm/s)
Destinasi Meja 3 - 4					
1	26,75	21,86	35,93	84,54	19,28
2	26,99	21,61	35,63	84,23	19,35
3	26,77	21,37	35,43	83,57	19,50
Rata-rata	26,84	21,61	35,66	84,11	19,38
Destinasi Meja 4 -3					
1	27,85	21,55	36,15	85,55	19,05
2	27,76	21,53	36,02	85,31	19,11
3	27,78	21,48	36,01	85,27	19,12
Rata-rata	27,80	21,52	36,06	85,38	19,09

Pada tabel-tabel diatas dapat dicari rata-rata waktu yang ditempuh, dan rata-rata kecepatan robot dalam menuntaskan destinasinya, bisa dilihat pada tabel sebagai berikut:

- a. Pada Tabel 4.45 rata-rata waktu robot menuju ke meja nomor 1 yakni **20,44 seconds**, rata-rata waktu robot menuju ke meja nomor 2 yakni **21,12 seconds**, rata-rata waktu robot kembali ke posisi start (*home*) yakni **28,90 seconds**, dan rata-rata waktu secara keseluruhan yakni **70,46 seconds**, dan rata-rata kecepatan yang ditempuh yakni **17,77 cm/s**, sedangkan, rata-rata waktu robot menuju ke meja nomor 2 yakni **21,08 seconds**, rata-rata waktu robot menuju ke meja nomor 1 yakni **21,75 seconds**, rata-rata waktu robot kembali ke posisi start (*home*) yakni **29,83 seconds**, dan rata-rata waktu secara keseluruhan yakni **72,67 seconds**, dan rata-rata kecepatan yang ditempuh yakni **17,23 cm/s**.
- b. Pada Tabel 4.46 rata-rata waktu robot menuju ke meja nomor 2 yakni **20,75 seconds**, rata-rata waktu robot menuju ke meja nomor 3 yakni **33,60 seconds**, rata-rata waktu robot kembali ke posisi start (*home*) yakni **36,05 seconds**, dan rata-rata waktu secara keseluruhan yakni **90,40 seconds**, dan rata-rata kecepatan yang ditempuh yakni **17,96 cm/s**, sedangkan, rata-rata waktu robot menuju ke meja nomor 3 yakni **26,97 seconds**, rata-rata waktu robot menuju ke meja nomor 2 yakni **34,14 seconds**, rata-rata waktu robot kembali ke posisi start (*home*) yakni **30,20 seconds**, dan rata-rata waktu secara keseluruhan yakni **91,31 seconds**, dan rata-rata kecepatan yang ditempuh yakni **17,79 cm/s**.
- c. Pada Tabel 4.47 rata-rata waktu robot menuju ke meja nomor 3 yakni **26,84 seconds**, rata-rata waktu robot menuju ke meja nomor 4 yakni **21,61 seconds**, rata-rata waktu robot kembali ke posisi start (*home*) yakni **35,66 seconds**, dan rata-rata waktu secara keseluruhan yakni **84,11 seconds**, dan

rata-rata kecepatan yang ditempuh yakni **19,38 cm/s**, sedangkan, rata-rata waktu robot menuju ke meja nomor 4 yakni **27,80 seconds**, rata-rata waktu robot menuju ke meja nomor 3 yakni **21,52 seconds**, rata-rata waktu robot kembali ke posisi start (*home*) yakni **36,06 seconds**, dan rata-rata waktu secara keseluruhan yakni **85,38 seconds**, dan rata-rata kecepatan yang ditempuh yakni **19,09 cm/s**.

Selanjutnya pengujian destinasi 2 meja dengan beban 4 Kg pada Tabel 4.48, Tabel 4.49, dan Tabel 4.50.

Tabel 4.48 Pengujian Urutan 1 dan 2, 2 dan 1 Beban 4 Kg

Jarak keseluruhan : 1252 cm

Percobaan	Waktu Tempuh 1 (s)	Waktu Tempuh 1-2 (s)	Waktu Tempuh 2 (s)	Total Waktu (s)	Kecepatan (cm/s)
Destinasi Meja 1 - 2					
1	0,00	0,00	0,00	0,00	0,00
2	0,00	0,00	0,00	0,00	0,00
3	0,00	0,00	0,00	0,00	0,00
Rata-rata	0,00	0,00	0,00	0,00	0,00
Destinasi Meja 2 - 1					
1	0,00	0,00	0,00	0,00	0,00
2	0,00	0,00	0,00	0,00	0,00
3	0,00	0,00	0,00	0,00	0,00
Rata-rata	0,00	0,00	0,00	0,00	0,00

Tabel 4.49 Pengujian Urutan 2 dan 3, 3 dan 2 Beban 4 Kg

Jarak keseluruhan : 1624 cm

Percobaan	Waktu Tempuh 1 (s)	Waktu Tempuh 1-2 (s)	Waktu Tempuh 2 (s)	Total Waktu (s)	Kecepatan (cm/s)
Destinasi Meja 1 - 2					
1	0,00	0,00	0,00	0,00	0,00
2	0,00	0,00	0,00	0,00	0,00

Percobaan	Waktu Tempuh 1 (s)	Waktu Tempuh 1-2 (s)	Waktu Tempuh 2 (s)	Total Waktu (s)	Kecepatan (cm/s)
3	0,00	0,00	0,00	0,00	0,00
Rata-rata	0,00	0,00	0,00	0,00	0,00
Destinasi Meja 2 -1					
1	0,00	0,00	0,00	0,00	0,00
2	0,00	0,00	0,00	0,00	0,00
3	0,00	0,00	0,00	0,00	0,00
Rata-rata	0,00	0,00	0,00	0,00	0,00

Tabel 4.50 Pengujian Urutan 3 dan 4, 4 dan 3 Beban 4 Kg

Jarak keseluruhan : 1630 cm

Percobaan	Waktu Tempuh 1 (s)	Waktu Tempuh 1-2 (s)	Waktu Tempuh 2 (s)	Total Waktu (s)	Kecepatan (cm/s)
Destinasi Meja 1 – 2					
1	0,00	0,00	0,00	0,00	0,00
2	0,00	0,00	0,00	0,00	0,00
3	0,00	0,00	0,00	0,00	0,00
Rata-rata	0,00	0,00	0,00	0,00	0,00
Destinasi Meja 2 -1					
1	0,00	0,00	0,00	0,00	0,00
2	0,00	0,00	0,00	0,00	0,00
3	0,00	0,00	0,00	0,00	0,00
Rata-rata	0,00	0,00	0,00	0,00	0,00

Pada tabel-tabel diatas pengujian destinasi 2 meja pada beban 4 Kg robot tidak mampu mengantarkan makanan ke meja pertama, ke meja kedua atau kembali ke posisi awal (*home*) dan posisi robot masih di posisi awal (*home*).

4.4.5 Hasil Pengujian Keberhasilan Pada Uji Destinasi 2 Meja

Pada pengujian pada tahap ini untuk destinasi 2 meja terdiri dari tiga fase yang digunakan untuk persentase keberhasilan yakni Fase 1 ketika robot mengantarkan makanan ke meja pertama, Fase 2 robot mengantarkan makanan ke meja kedua dan Fase 3 ketika robot kembali ke posisi awal (*home*). Berikut adalah tabel *Rule* Keberhasilan Destinasi 2 Meja seperti pada Tabel 4.51.

Tabel 4.51 *Rule* Keberhasilan Destinasi 2 Meja

Destinasi 2 Meja	
Untuk pilihan meja 1 dan/atau 2	
Keterangan	Persentase(%)
Fase 1	
Home ke Perempatan	0 – 15
Perempatan ke Meja 1	16 – 33
Fase 2	
Meja 1 ke Perempatan	34 – 50
Perempatan ke Meja 2	51 – 65
Fase 3	
Meja 2 ke Perempatan	66 – 83
Perempatan ke <i>Home</i>	84 – 100

Untuk pilihan meja 1/2 dan 3/4	
Keterangan	Persentase(%)
Fase 1	
Home ke Perempatan	0 - 15
Perempatan ke Meja 1	16 - 33
Fase 2	
Meja 1 ke Perempatan	34 - 44
Perempatan ke Pertigaan	45 - 54
Pertigaan ke Meja 2	55 – 65
Fase 3	
Meja 2 ke Pertigaan	66 – 78
Pertigaan ke Perempatan	79 – 89
Perempatan ke <i>Home</i>	90 - 100

Destinasi 2 Meja	
Untuk pilihan meja 3 dan/atau 4	
Keterangan	Persentase(%)
Fase 1	
Home ke Perempatan	0 - 13
Perempatan ke Pertigaan	14 - 23
Pertigaan ke Meja 1	24 - 33
Fase 2	
Meja 1 ke Pertigaan	34 - 50
Pertigaan ke Meja 2	51 - 65
Fase 3	
Meja 2 ke Pertigaan	66 - 83
Pertigaan ke Perempatan	66 - 83
Perempatan ke <i>Home</i>	84 - 100

Untuk pilihan meja 3/4 dan 1/2	
Keterangan	Persentase(%)
Fase 1	
Home ke Perempatan	0 - 13
Perempatan ke Pertigaan	14 - 23
Pertigaan ke Meja 1	24 - 33
Fase 2	
Meja 1 ke Pertigaan	34 - 44
Pertigaan ke Perempatan	45 - 54
Perempatan ke Meja 2	55 - 65
Fase 3	
Meja 2 ke Perempatan	66 - 83
Perempatan ke <i>Home</i>	84 - 100

1. Hasil Pengujian Keberhasilan Tanpa Beban

Pada pengujian pada tahap ini, pengujian berupa mengamati berapa persen robot dari posisi awal (*home*) mengantarkan makanan ke meja yang dituju lalu kembali ke posisi awal (*home*) tanpa membawa beban. Robot dikatakan berhasil secara keseluruhan jika robot dapat menyelesaikan tugas hingga persentase 100% sesuai *Rule* Keberhasilan Destinasi 1 Meja. Berikut tabel keberhasilan destinasi 1 meja Tanpa Beban pada Tabel 4.52.

Tabel 4.52 Pengujian Keberhasilan Tanpa Beban

Meja 1/2	
Percobaan	Keberhasilan (%)
Destinasi 1 - 2	
1	100
2	100
3	100
4	70
5	100
6	66
Rata-rata	89
Destinasi 2 - 1	
1	70
2	100
3	100
4	100
5	100
6	100
Rata-rata	95

Meja 1/3	
Percobaan	Keberhasilan (%)
Destinasi 1 - 3	
1	70
2	100
3	100
4	50
5	100
6	100
Rata-rata	87
Destinasi 3 - 1	
1	100
2	100
3	100
4	66
5	100
6	100
Rata-rata	94

Meja 1/4

Percobaan	Keberhasilan (%)
Destinasi 1 - 4	
1	70
2	50
3	100
4	100
5	100
6	100
Rata-rata	87
Destinasi 4 - 1	
1	45
2	100
3	100
4	85
5	100
6	100
Rata-rata	88

Meja 2/4

Percobaan	Keberhasilan (%)
Destinasi 2 - 4	
1	100
2	100
3	100
4	33
5	90
6	100
Rata-rata	87
Destinasi 4 - 2	
1	58
2	100
3	70
4	100
5	100
6	100
Rata-rata	88

Meja 2/3

Percobaan	Keberhasilan (%)
Destinasi 2 - 3	
1	100
2	70
3	100
4	50
5	33
6	100
Rata-rata	76
Destinasi 3 - 2	
1	100
2	50
3	100
4	50
5	100
6	100
Rata-rata	83

Meja 3/4

Percobaan	Keberhasilan (%)
Destinasi 3 - 4	
1	50
2	100
3	100
4	100
5	100
6	100
Rata-rata	92
Destinasi 4 - 3	
1	100
2	100
3	100
4	100
5	100
6	100
Rata-rata	100

Keberhasilan yang dicapai pada destinasi 2 meja Tanpa Beban terdapat beberapa yang kurang berhasil atau dibawah nilai 100% yakni dengan keterangan sebagai berikut:

- a. Pada keberhasilan 70% untuk meja 1 dan 2 percobaan ke 4, robot berhenti membaca garis setelah menjauh dari meja 2 dan mendekati perempatan.
- b. Pada keberhasilan 66% untuk meja 1 dan 2 percobaan ke 6, robot ketika putar balik kurang bisa menyesuaikan dengan garis hitam sehingga pembacaan garis error.
- c. Pada keberhasilan 70% untuk meja 2 dan 1 percobaan ke 1, robot berhenti membaca garis setelah menjauh dari meja 1 dan mendekati perempatan.
- d. Pada keberhasilan 70% untuk meja 1 dan 3 percobaan ke 1, robot berhenti membaca garis setelah menjauh dari meja 3 dan mendekati pertigaan.
- e. Pada keberhasilan 50% untuk meja 1 dan 3 percobaan ke 4, robot berhenti membaca garis diantara perempatan dan pertigaan.
- f. Pada keberhasilan 66% untuk meja 3&1 percobaan ke 4, robot ketika putar balik kurang bisa menyesuaikan dengan garis hitam sehingga pembacaan garis error.
- g. Pada keberhasilan 70% untuk meja 1 dan 4 percobaan ke 1, robot berhenti membaca garis setelah menjauh dari meja 4 dan mendekati pertigaan.
- h. Pada keberhasilan 50% untuk meja 1 dan 4 percobaan ke 2, robot berhenti membaca garis diantara perempatan dan pertigaan.
- i. Pada keberhasilan 45% untuk meja 4&1 percobaan ke 1, robot berhenti membaca garis setelah melewati perempatan.

- j. Pada keberhasilan 85% untuk meja 4&1 percobaan ke 4, robot berhenti membaca garis setelah melewati perempatan dan akan menuju *home*.
- k. Pada keberhasilan 70% untuk meja 2 dan 3 percobaan ke 2, robot berhenti membaca garis setelah menjauh dari meja 3 dan mendekati pertigaan.
- l. Pada keberhasilan 50% untuk meja 2 dan 3 percobaan ke 4, robot berhenti membaca garis diantara perempatan dan pertigaan.
- m. Pada keberhasilan 33% untuk meja 2 dan 3 percobaan ke 5, robot ketika putar balik kurang bisa menyesuaikan dengan garis hitam sehingga pembacaan garis error.
- n. Pada keberhasilan 50% untuk meja 3 dan 2 percobaan ke 2, robot berhenti membaca garis diantara perempatan dan pertigaan.
- o. Pada keberhasilan 50% untuk meja 3 dan 2 percobaan ke 4, robot berhenti membaca garis diantara perempatan dan pertigaan.
- p. Pada keberhasilan 33% untuk meja 2 dan 4 percobaan ke 4, robot ketika putar balik kurang bisa menyesuaikan dengan garis hitam sehingga pembacaan garis error.
- q. Pada keberhasilan 90% untuk meja 2 dan 4 percobaan ke 5, robot berhenti membaca garis setelah melewati perempatan dan akan menuju *home*.
- r. Pada keberhasilan 58% untuk meja 4&2 percobaan ke 1, robot berhenti membaca garis saat akan mendekati meja 2.
- s. Pada keberhasilan 70% untuk meja 4&2 percobaan ke 3, robot berhenti membaca garis setelah menjauh dari meja 2 dan mendekati perempatan.
- t. Pada keberhasilan 50% untuk meja 3 dan 4 percobaan ke 1, robot berhenti membaca garis saat akan melewati pertigaan.

2. Hasil Pengujian Keberhasilan Dengan Beban

Pada pengujian pada tahap ini, pengujian berupa mengamati berapa persen robot dari posisi awal (*home*) mengantarkan makanan ke meja pertama, lalu mengantarkan makanan ke meja kedua serta kembali ke posisi awal (*home*) dengan membawa beban 1 Kg, 2 Kg, 3 Kg, dan 4 Kg. Robot dikatakan berhasil secara keseluruhan jika robot dapat menyelesaikan tugas hingga persentase 100% sesuai *Rule* Keberhasilan Destinasi 2 Meja. Berikut tabel keberhasilan destinasi 1 meja dengan beban pada Tabel 4.53, Tabel 4.54, Tabel 4.55, dan Tabel 4.56.

Tabel 4.53 Pengujian Keberhasilan Beban 1 Kg

Meja 1/2		Meja 2/3		Meja 3/4	
Percobaan	Keberhasilan (%)	Percobaan	Keberhasilan (%)	Percobaan	Keberhasilan (%)
Destinasi 1 - 2		Destinasi 2 - 3		Destinasi 3 - 4	
1	100	1	100	1	100
2	100	2	100	2	100
3	100	3	100	3	100
Rata-rata	100	Rata-rata	100	Rata-rata	100
Destinasi 2 - 1		Destinasi 3 - 2		Destinasi 4 - 3	
1	100	1	100	1	100
2	100	2	100	2	100
3	100	3	100	3	100
Rata-rata	100	Rata-rata	100	Rata-rata	100

Tabel 4.54 Pengujian Keberhasilan Beban 2 Kg

Meja 1/2		Meja 2/3		Meja 3/4	
Percobaan	Keberhasilan (%)	Percobaan	Keberhasilan (%)	Percobaan	Keberhasilan (%)
Destinasi 1 - 2		Destinasi 2 - 3		Destinasi 3 - 4	
1	100	1	100	1	100
2	100	2	100	2	100
3	100	3	100	3	100
Rata-rata	100	Rata-rata	100	Rata-rata	100
Destinasi 2 - 1		Destinasi 3 - 2		Destinasi 4 - 3	
1	100	1	100	1	100
2	100	2	100	2	100
3	100	3	100	3	100
Rata-rata	100	Rata-rata	100	Rata-rata	100

Tabel 4.55 Pengujian Keberhasilan Beban 3 Kg

Meja 1/2		Meja 2/3		Meja 3/4	
Percobaan	Keberhasilan (%)	Percobaan	Keberhasilan (%)	Percobaan	Keberhasilan (%)
Destinasi 1 - 2		Destinasi 2 - 3		Destinasi 3 - 4	
1	100	1	100	1	100
2	100	2	100	2	100
3	100	3	100	3	100
Rata-rata	100	Rata-rata	100	Rata-rata	100
Destinasi 2 - 1		Destinasi 3 - 2		Destinasi 4 - 3	
1	100	1	100	1	100
2	100	2	100	2	100
3	100	3	100	3	100
Rata-rata	100	Rata-rata	100	Rata-rata	100

Tabel 4.56 Pengujian Keberhasilan Beban 4 Kg

Meja 1/2		Meja 2/3		Meja 3/4	
Percobaan	Keberhasilan (%)	Percobaan	Keberhasilan (%)	Percobaan	Keberhasilan (%)
Destinasi 1 - 2		Destinasi 2 - 3		Destinasi 3 - 4	
1	0	1	0	1	0
2	0	2	0	2	0
3	0	3	0	3	0
Rata-rata	0	Rata-rata	0	Rata-rata	0
Destinasi 2 - 1		Destinasi 3 - 2		Destinasi 4 - 3	
1	0	1	0	1	0
2	0	2	0	2	0
3	0	3	0	3	0
Rata-rata	0	Rata-rata	0	Rata-rata	0

Keberhasilan yang dicapai pada destinasi 1 meja dengan beban 1 Kg, 2 Kg, 3 Kg robot berhasil secara keseluruhan 100% mengantarkan makanan dan kembali ke posisi awal, sedangkan pada beban 4 Kg robot tidak berhasil secara keseluruhan, karena robot tidak dapat bergerak dari posisi awal (*home*).

4.5 Analisis Keseluruhan Pengujian yang Telah Dilakukan

Setelah dilakukan beberapa pengujian diatas yaitu uji kamera deteksi garis, uji destinasi 1 meja dan uji destinasi 2 meja. Analisis yang diperoleh dari hasil uji kamera deteksi garis dengan menggunakan kamera Webcam Logitech C270 menunjukkan bahwa kamera dapat mendeteksi Warna Hitam, kuning dan merah berdasarkan citra yang dikonversi ke dalam warna HSV dan dicari nilai skala *Hue*, *Saturation* dan *Value* dari masing-masing warna. Didapatkan nilai skala HSV sebagaimana pada tabel 4.57.

Tabel 4.57 Nilai Skala *Hue*, *Saturation*, dan *Value*

Hitam		Kuning		Merah	
Keterangan	Nilai Skala	Keterangan	Nilai Skala	Keterangan	Nilai Skala
<i>Hue</i>	0 - 255	<i>Hue</i>	20 - 57	<i>Hue</i>	0 – 51
<i>Saturation</i>	0 - 255	<i>Saturation</i>	50 - 255	<i>Saturation</i>	181 - 255
<i>Value</i>	0 - 68	<i>Value</i>	80 - 195	<i>Value</i>	73 - 113

Analisis yang diperoleh dari uji destinasi 1 meja dalam menguji waktu tempuh dan kecepatan sebagaimana yang ditunjukkan pada Tabel 4.57 menunjukkan bahwa semakin berat beban yang diberikan maka pengaruh terhadap waktu tempuh semakin lama dan kecepatan semakin berkurang hingga sampai pada batas maksimal robot ini yakni 3 Kg serta keberhasilan yang diperoleh terpengaruhi dengan adanya beban, sedangkan yang tidak adanya beban mempunyai rata-rata keberhasilan dibawah 100%.

Tabel 4.57 Nilai Keseluruhan Destinasi 1 Meja

Kalkulasi 1 Meja				
Tanpa Beban				
Meja	1	2	3	4
Total Waktu (s)	40,18	36,51	51,10	51,23
Kecepatan (cm/s)	21,45	25,27	24,48	24,23
Keberhasilan (%)	96	85	93	94
Beban 1 Kg				
Meja	1	2	3	4
Total Waktu (s)	43,11	43,10	57,27	56,30
Kecepatan (cm/s)	19,53	19,58	21,20	21,60
Keberhasilan (%)	100	100	100	100
Beban 2 Kg				
Meja	1	2	3	4
Total Waktu (s)	45,07	45,06	59,34	58,26
Kecepatan (cm/s)	18,68	18,73	20,46	20,87
Keberhasilan (%)	100	100	100	100

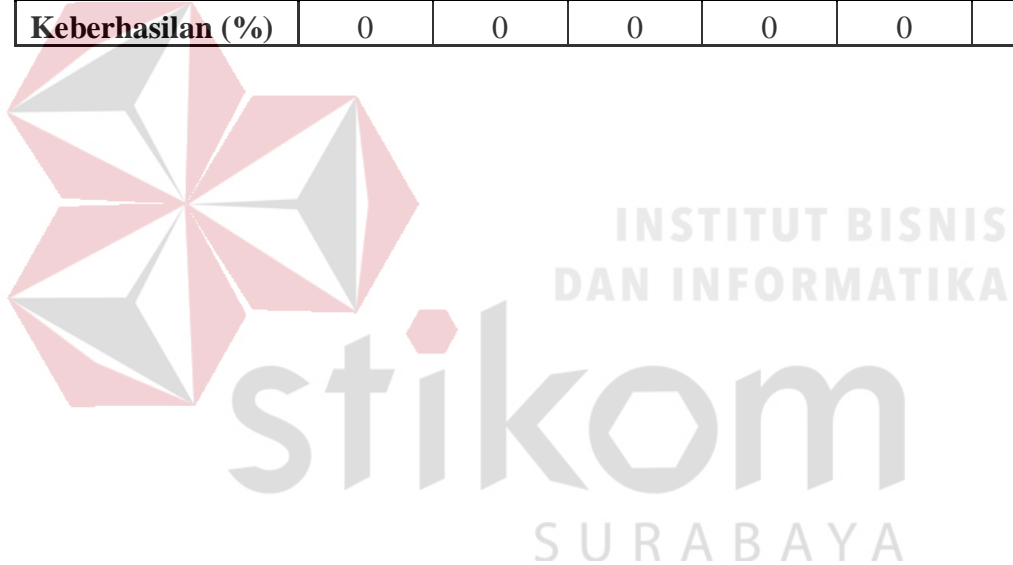
Beban 3 Kg				
Meja	1	2	3	4
Total Waktu (s)	49,93	49,92	64,20	63,12
Kecepatan (cm/s)	16,87	16,91	18,91	19,27
Keberhasilan (%)	100	100	100	100
Beban 4 Kg				
Meja	1	2	3	4
Total Waktu (s)	0,00	0,00	0,00	0,00
Kecepatan (cm/s)	0,00	0,00	0,00	0,00
Keberhasilan (%)	0	0	0	0

Analisis yang diperoleh dari uji destinasi 2 meja dalam waktu dan kecepatan seperti yang dilihat pada Tabel 4.58 menunjukkan bahwa semakin berat beban yang diberikan maka pengaruh terhadap waktu tempuh semakin lama dan kecepatan semakin berkurang hingga sampai pada batas maksimal robot ini yakni 3 Kg serta keberhasilan yang diperoleh terpengaruhi dengan adanya beban, sedangkan yang tidak adanya beban mempunyai rata-rata keberhasilan dibawah 100%.

Tabel 4.58 Nilai Keseluruhan Destinasi 2 Meja

Kalkulasi 2 Meja						
Tanpa Beban						
Meja	1 dan 2	1 dan 3	1 dan 4	2 dan 3	2 dan 4	3 dan 4
Total Waktu (s)	56,76	70,93	55,59	60,80	68,97	68,44
Kecepatan (cm/s)	22,19	23,68	48,26	32,15	26,52	24,89
Keberhasilan (%)	92	91	90	79	88	96
Beban 1 Kg						
Meja	1 dan 2	2 dan 1	2 dan 3	3 dan 2	3 dan 4	4 dan 3
Total Waktu (s)	60,23	62,44	80,17	81,08	73,88	75,15
Kecepatan (cm/s)	20,79	20,05	20,26	20,03	22,06	21,69
Keberhasilan (%)	100	100	100	100	100	100

Beban 2 Kg						
Meja	1 dan 2	2 dan 1	2 dan 3	3 dan 2	3 dan 4	4 dan 3
Total Waktu (s)	63,17	65,38	83,11	84,02	76,82	78,09
Kecepatan (cm/s)	19,82	19,15	19,54	19,33	21,22	20,87
Keberhasilan (%)	100	100	100	100	100	100
Beban 3 Kg						
Meja	1 dan 2	2 dan 1	2 dan 3	3 dan 2	3 dan 4	4 dan 3
Total Waktu (s)	70,46	72,67	90,40	91,31	84,11	85,38
Kecepatan (cm/s)	17,77	17,23	17,96	17,79	19,38	19,09
Keberhasilan (%)	100	100	100	100	100	100
Beban 4 Kg						
Meja	1 dan 2	2 dan 1	2 dan 3	3 dan 2	3 dan 4	4 dan 3
Total Waktu (s)	0,00	0,00	0,00	0,00	0,00	0,00
Kecepatan (cm/s)	0,00	0,00	0,00	0,00	0,00	0,00
Keberhasilan (%)	0	0	0	0	0	0



BAB V

PENUTUP

Dalam bab ini berisi tentang kesimpulan dan saran yang berdasarkan pada hasil dari pengujian yang telah dilakukan pada Tugas Akhir ini, maka dapat diperoleh beberapa kesimpulan dan saran untuk pengembangan sistem berikutnya.

5.1 Kesimpulan

Berdasarkan hasil pengujian pada Tugas Akhir ini dapat disimpulkan beberapa kesimpulan sebagai berikut:

1. Hasil dari proses pengujian kamera deteksi garis menunjukkan bahwa kamera berhasil membedakan deteksi Garis Warna Hitam, Garis Warna Kuning dan Garis Warna Merah. Dengan menggunakan fitur dari OpenCV 2.1.0 dengan urutan tahapan yaitu menangkap citra RGB, konversi RGB to HSV, mencari nilai HSV dengan skala, menghilangkan *noise* dengan *Gaussian-blur*, deteksi tepi garis dengan *Canny Edge Detection*, dan membentuk lingkaran berdasarkan warna HSV dan tepi garis dengan *Hough Circle Transform*.
2. Nilai HSV yang didapatkan untuk warna hitam yakni H : 0-255, S : 0-255, V : 0-68. Warna kuning H : 20-57, S : 50-255, V : 80-195, dan warna merah H : 0-51, S : 181-255, V : 73-113.
3. Hasil dari proses pengujian destinasi 1 meja dan destinasi 2 meja menunjukkan bahwa semakin berat beban robot yang dibawa, rata-rata waktu tempuh yang dibutuhkan semakin banyak dan rata-rata kecepatan semakin berkurang hingga robot mencapai batas maksimalnya mengantarkan makanan yakni 3 Kg serta

keberhasilan dengan adanya beban yang ada pada robot membuat berjalan stabil dan dapat menyelesaikan secara keseluruhan hingga 100%.

5.2 Saran

Saran yang diberikan oleh penulis pada pengembangan Tugas Akhir ini selanjutnya adalah sebagai berikut:

1. Untuk mekanik robot dapat diperbarui lebih efisiensi, agar pergerakan robot tidak menyeret dan menghindari selip.
2. Penggunaan Mini PC di upgrade agar pengolahan citra bisa lebih cepat, tepat dan rinci.
3. Penggunaan Dual-Kamera agar dapat memudahkan robot dalam membaca garis serta menghemat pergerakan robot, sehingga tidak berpacu pada satu kamera saja.
4. Jika menggunakan 2 robot lebih, maka perlu pengembangan komunikasi antar robot itu sendiri.
5. Penggunaan ban karet untuk mencegah ban selip ataupun menyeret ketika akan melakukan belokan atau putar balik.
6. Menambah sistem kontrol atau sistem pengaturan pada robot sehingga robot dapat berjalan dengan lebih stabil.
7. Untuk pengembangan selanjutnya penerapan robot dilakukan tanpa menggunakan garis sebagai jalur melainkan memahami letak meja yang akan dituju.

DAFTAR PUSTAKA

- Akbar, A. H. 2013. Penyelesaian Jalur Terpendek dengan menggunakan Maze Mapping Pada Line Maze. *Jurnalpa*.
- Asif, M. 2015. Waiter Robot - Solution to Restaurant Automation. *MDSRC*.
- Danil, C. 2001. Edge Detection dengan Algoritma Canny. *acamedia.edu*, 197-203.
- Electronics, I. (2010, Juli 12). *EMS 30 A H-Bridge*. Diambil kembali dari *innovativeelectronics: support@innovativeelectronics.com*
- Fauzan, A. (2015, Januari 03). *charisfauzan*. Diambil kembali dari <http://www.charisfauzan.net>: <http://www.charisfauzan.net/2015/01/ruang-warna-hue-saturation-value-hsv.html>
- Halim, S. 2007. *Merancang Mobile Robot Pembawa Objek Menggunakan OOPic-R*. Jakarta: PT. Elex Media Komputindo.
- Ilham. (2013, November 12). *vouchacare*. Diambil kembali dari voucha.co.id: <https://support.voucha.co.id/kb/cara-mengaktifkan-remote-desktop-connection-di-windows/>
- Irawan, B. (2016, Januari 23). *Arduino Due R3 SAM3X8E 32-bit ARM Cortex-M Compatible Board Module*. Diambil kembali dari Indo-Ware: <https://www.indo-ware.com/produk-3771-arduino-due-r3-sam3x8e-32bit-arm-cortexm-compatible-board-module.html>
- Kho, D. (2015, Mei 16). *Pengertian Baterai dan Jenis-jenis Baterai*. Diambil kembali dari teknikelektronika.com: <https://teknikelektronika.com/pengertian-baterai-jenis-jenis-baterai/>
- Kho, D. 2016. *Pengertian Proximity Sensor (Sensor Jarak) dan Jenis-jenisnya*. Diambil kembali dari Teknik Elektronika: <https://teknikelektronika.com/pengertian-proximity-sensor-sensor-jarak-jenis-jenis-sensor-proximity/>
- Logitech. 2007, Maret 22. *Logitech HD Webcam C270*. Diambil kembali dari [logitech](http://logitech.com): https://support.logitech.com/en_us/product/hd-webcam-c270/specs
- MRI. (2012, Maret 23). *PG 45 Series*. Diambil kembali dari [brontoseno](http://www.brontoseno.com): <http://www.brontoseno.com>

- Munandar, A. (2012, Juni 27). *Liquid Crystal Display (LCD) 16x2*. Diambil kembali dari Les Elektronika: <http://www.leselektronika.com/2012/06/liquid-crystal-display-lcd-16-x-2.html>
- Nalwan, W. B. 2010. *Membuat Sendiri Robot Humanoid*. Jakarta: PT. Elex Media Komputindo.
- OpenCV. (2017, Januari 12). *opencv.org*. Diambil kembali dari docs.opencv.org: https://docs.opencv.org/2.4.13.7/doc/tutorials/imgproc/imgtrans/hough_circle/hough_circle.html
- Pi, R. 2015. Diambil kembali dari Raspberry Pi 3 Model B: www.raspberrypi.com/components.com/raspberrypi
- Rezza. (2017, Agustus 14). *Cara Mengakses Keypad Menggunakan Arduino Uno*. Diambil kembali dari nyebarilmu.com: <https://www.nyebarilmu.com/cara-mengakses-keypad-menggunakan-arduino-uno/>
- Safii, C. A. 2017. Mobile Robot Pembawa Peralatan Makan Kotor Otomatis Pada Sebuah Restoran. *JCONES*, 160-167.
- Sinaryuda. (2017, Januari 22). *Mengenal Aplikasi Arduino IDE dan Arduino Sketch*. Diambil kembali dari Sinaryuda.web.id: <https://www.sinaryuda.web.id/microcontroller/mengenal-aplikasi-arduino-ide-dan-arduino-sketch.html>
- Susanto, A., Effendy, M., & Alfian Noor, Y. (2014, April 10). *PENGERTIAN WEBCAM BESERTA FUNGSINYA*. Diambil kembali dari KOM-SWB: <http://kom-swb.blogspot.co.id/2012/10/pengertian-webcam-beserta-fungsinya.html>
- Tjahyadi, C. (2011, September 4). *DC-DC Converter*. Diambil kembali dari christianto.tjahyadi.com: <http://christianto.tjahyadi.com/elektronika/ubec.html>
- Wibisono, D. C. 2011. Metode Gaussian Smoothing Untuk Peningkatan Kualitas Citra Medis Yang Blur. *publikasi.dinus.ac.id*, 1-6.
- Yuliandoko, H. 2018. *Jaringan Komputer Wire dan Wireless Beserta Penerapannya*. Yogyakarta: CV Budi Utama.