

**PROTOTYPE SISTEM KONTROL LAMPU LALU LINTAS
BERBASIS ARDUINO UNO**

TUGAS AKHIR

**Karya tulis sebagai salah satu syarat
untuk memperoleh gelar Sarjana dari
Universitas Fajar**

Oleh :

HERMAN HARUN

1520221041



**PROGRAM STUDI TEKNIK ELEKTRO
FAKULTAS TEKNIK
UNIVERSITAS FAJAR
MAKASSAR
2022**

HALAMAN PENGESAHAN

**Prototype Sistem Kontrol Lampu Lalu Lintas
Berbasis Arduino Uno**

Disusun Oleh :

**Herman Harun
1520221041**

Telah diperiksa dan disetujui oleh Dosen Pembimbing

Makassar, 27 Januari 2023

<p>Pembimbing I</p>  <p><u>Safaruddin, S.Si., M.T</u> NIDN: 0909106901</p>	<p>Pembimbing II</p>  <p><u>Indah Purwitasari Ihsan, S.T., M.T</u> NIDN: 1221089001</p>
--	---

Mengetahui,

<p>Dekan Fakultas Teknik</p>  <p><u>Prof. Dr. Ir. Ernati, S.T., M.T</u> NIDN: 0906107701</p>	<p>Ketua Program Studi</p>  <p><u>Safaruddin, S.Si., M.T</u> NIDN: 0909106901</p>
--	--

PERNYATAAN ORISINAL

Penulis dengan ini menyatakan bahwa Tugas Akhir :
"Prototype Sistem Kontrol Lampu Lalu Lintas Berbasis Arduino Uno" adalah karya orisinal saya dan setiap serta seluruh sumber acuan telah ditulis sesuai dengan panduan penulisan ilmiah yang berlaku di Fakultas Teknik Universitas Fajar Makassar.

Makassar, 27 Januari 2023

Yang menyatakan



Herman Harun

ABSTRAK

Prototype Sistem Kontrol Lampu Lalu Lintas Berbasis Arduino Uno, Teknik Elektro, Herman Haru. Setiap tahun peningkatan jumlah kendaraan semakin bertambah, hal ini berakibat munculnya masalah lalu lintas kendaraan, misalnya kemacetan lalu lintas, kecelakaan lalu lintas, polusi udara, dan sebagainya. Kemacetan lalu lintas telah menjadi masalah signifikan. Salah satu solusi untuk mengatasi permasalahan tersebut dengan pengaturan lalu lintas menggunakan lampu lalu lintas. Sistem kendali atau sistem kontrol (*control system*) adalah suatu alat (kumpulan alat) untuk mengendalikan, memerintah, dan mengatur keadaan dari suatu sistem. Sistem kendali ini bisa menjadi salah satu solusi untuk mengatasi kelemahan dari penggunaan *traffic light*. Berdasarkan hasil penelitian membuat suatu sistem kontrol menggunakan arduino diperoleh waktu nyala pada kondisi sepi adalah hijau 10 detik dan merah 10 detik, kondisi cukup padat adalah hijau 20 detik dan merah 20 detik, kondisi padat adalah hijau 30 detik merah 30 detik.

ABSTRACT

Traffic Light Control System Prototype Based On Arduino UNO, Electrical Engineering, Herman Harun. Every year the increase in the number of vehicles is increasing, this results in the emergence of vehicular traffic problems, such as traffic jams, traffic accidents, air pollution, and so on. Traffic jams have become a significant problem. One solution to overcome this problem is to regulate traffic using traffic lights. A control system or control system is a tool (a collection of tools) to control, command, and regulate the state of a system. This control system can be a solution to overcome the weakness of using traffic light. Based on the results of the study making a control system using Arduino, it was found that the flame time in quiet conditions was green 10 seconds and red 10 seconds, moderately dense conditions were green 20 seconds and red 20 seconds, solid conditions were green 30 seconds and red 30 seconds.

KATA PENGANTAR

Syukur Alhamdulillah saya panjatkan kepada ALLAH SWT karena hanya dengan rahmat dan hidayah-Nya sehingga saya dapat menyelesaikan laporan ini yang berjudul “Prototype Sistem Kontrol Lampu Lalu Lintas Berbasis Arduino Uno”. Laporan skripsi ini disusun sebagai salah satu syarat bagi mahasiswa Program Studi Teknik Elektro untuk dapat menyelesaikan studi Program Strata Satu (S1) pada Universitas Fajar Makassar. Pada saat penyusunan tugas akhir ini sangat banyak mendapat bantuan dari berbagai pihak, oleh karna itu dalam kesempatan ini pula ijinkan saya untuk mengucapkan terima kasih serta penghargaan setinggi-tingginya kepada:

1. Allah SWT atas berkat dan rahmatnya yang tidak terputus kepada kami.
2. Ayah dan ibu serta keluarga besar yang senantiasa mendoakan agar kami selalu diberi kekuatan dan kesabaran.
3. Ibu Prof. Dr. Erniati ST.,MT selaku Dekan Fakultas Teknik Universitas Fajar Makassar.
4. Ibu Asmawaty Azis S.T., MT selaku Ketua Prodi Teknik Elektro Universitas Fajar Makassar
5. Bapak Safaruddin S.Si.,MT selaku Dosen Pembimbing I.
6. Ibu Indah Purwitasari Ihsan, S.T., M.T. selaku Dosen Pembimbing II
7. Dosen-dosen Prodi Teknik Elektro Universitas Fajar Makassar
8. Rekan-rekan Mahasiswa Universitas Fajar, yang selalu memberikan bantuan dan masukan, serta semua pihak yang telah membantu sehingga laporan Skripsi ini dapat terselesaikan

Akhirnya, kepada Allah SWT jualah penulis kembalikan semua permasalahan yang berada diluar kemampuan. Semoga Allah SWT senantiasa melimpahkan rahmat dan hidayahnya kepada kami atas usaha, perjuangan dan pengorbanan yang dilakukan.

Makassar, 27 Januari 2023

Penulis

DAFTAR ISI

HALAMAN JUDUL	i
LEMBAR PERSETUJUAN	ii
PERNYATAAN ORISINAL	iii
ABSTRAK	iv
KATA PENGANTAR	vi
DAFTAR ISI	vii
DAFTAR TABEL	x
DAFTAR GAMBAR	xi
BAB I PENDAHULUAN	1
I.1 Latar Belakang	1
I.2 Rumusan Masalah	4
I.3 Tujuan Penelitian	4
I.4 Batasan Masalah	4
BAB II LANDASAN TEORI	5
II.1 Kerangka Teori	5
II.1.1 Prototype	5
II.1.2 Sistem Kendali	6
II.1.3 Lampu Lalu Lintas	7
II.1.4 Arduino Software IDE	8
II.1.5 Bahasa Pemrograman C++	10
II.1.6 Pengujian Sistem	11
II.1.7 Flowchart.....	13
II.1.8 Arduino UNO	14
II.1.9 Sensor Ultrasonik HC-SR04	15
II.1.10 LED (Light Emitting Diode)	17
II.1.11 Kabel Jumper	18
II.1.12 Papan PCB	20
II.2. Penelitian Terdahulu (State of The Art)	23

II.3. Kerangka pikir	31
BAB III METODE PENELITIAN	32
III.1 Waktu dan Tempat Penelitian	32
III.2 Alat dan Bahan Penelitian	32
III.1.1 Alat	32
III.2.2 Bahan	32
III.3 Tahapan Penelitian	33
III.4 Metode Pengumpulan Data	34
III.4.1 Data Primer	34
III.4.2 Data Sekunder	34
III.5 Analisis Data/Analisis Sistem	34
III.6 Bagan Alur Penelitian	35
III.6.1 Sistem Berjalan	35
III.6.2 Rancangan Penelitian Yang Diusulkan	35
III.6.3 Perancangan Penelitian	35
BAB IV HASIL DAN PEMBAHASAN	40
4.1 Hasil	40
4.1.1 Tampilan Prototype Jalan Raya	40
4.1.2 Tampilan Alat	41
4.1.3 Tampilan Kondisi 1 Jalur A	42
4.1.4 Tampilan Kondisi 1 Jalur B	46
4.1.5 Tampilan Kondisi 2 Jalur A	50
4.1.6 Tampilan Kondisi 2 Jalur B	53
4.1.7 Tampilan Kondisi 3 Jalur A	57
4.1.8 Tampilan Kondisi 3 Jalur B	60
4.1.9 Tampilan Kondisi 4 Jalur A	63
4.1.10 Tampilan Kondisi 4 Jalur B	67
4.1.11 Tampilan Kondisi 5 Jalur A	70
4.1.12 Tampilan Kondisi 5 Jalur B	73
4.1.13 Tampilan Kondisi 6 Jalur A	77
4.1.14 Tampilan Kondisi 6 Jalur B	80
4.1.15 Tampilan Kondisi 7 Jalur A	83

4.1.16 Tampilan Kondisi 7 JalurB	87
4.1.17 Tampilan Kondisi 8 Jalur A	90
4.1.18 Tampilan Kondisi 8 Jalur B	93
4.1.19 Tampilan Kondisi 9 Jalur A	97
4.1.20 Tampilan Kondisi 9 Jalur B	100
4.1.21 Penujian Black Box	103
4.1.22 Pengujian White Box	107
4.2 Pembahasan	125
BAB V PENUTUP	133
5.1 Kesimpulan	133
5.2 Saran	133
DAFTAR PUSTAKA	134

DAFTAR TABEL

Tabel II.1 Penelitian Terdahulu (State Of The Art)	23
Tabel IV.1 Data Kendaraan	103
Tabel IV.2 Kondisi Kepadatan Kendaraan	104
Tabel IV.3 Data Prototype	104
Tabel IV.4 Pengujian Sistem KerjaPrototype	105
Taba IV.5 Rangkuman Ilustrasi	130

DAFTAR GAMBAR

Gambar 2. 1 Sistem Kendali	7
Gambar 2. 2 Tampilan <i>Software Arduino IDE</i>	9
Gambar 2. 3 Logo C++	11
Gambar 2. 4 Sistem Pengujian	11
Gambar 2. 5 Simbol Flowchart	14
Gambar 2. 6 Arduino UNO	15
Gambar 2. 7 <i>Sensor HC-SR04</i>	16
Gambar 2. 8 <i>LED (Light Emitting Diode)</i>	17
Gambar 2. 9 Kabel Jumper	18
Gambar 2. 10 Kabel Jumper Male To Male	19
Gambar 2. 11 Kabel Jumper Male To Female.....	19
Gambar 2. 12 Kabel Jumper Female To Female	20
Gambar 2. 13 Papan PCB	21
Gambar 2. 14 Struktur Dan Komposisi PCB	21
Gambar 2. 15 Kerangka Pikir	31
Gambar 3. 1 Bagan Prosedur Penelitian	33
Gambar 3. 2 Desain Alat	35
Gambar 3. 3 Rancangan Prototype Sistem Control Lampu Lalu Lintas Berbasis Arduino UNO	37
Gambar 3. 4 Flow Chart Sistem.....	38
Gambar 4.1 Prototype Jalan Raya	40

Gambar 4.2 Tampilan Alat (Arduino UNO)	41
Gambar 4.3 Tampilan Alat (Sensor)	41
Gambar 4.4 Tampilan Alat (LED)	42
Gambar 4.5 Kondisi 1 jalur A Lampu Merah	42
Gambar 4.6 Script Kondisi 1 jalur A	43
Gambar 4.7 Monitoring Jarak Objek Kondisi 1 jalur A	44
Gambar 4.8 Kondisi 1 jalur A Lampu Hijau.....	44
Gambar 4.9 Script Kondisi 1 jalur A	45
Gambar 4.10 Monitoring Jarak Objek Kondisi 1 jalur A.....	46
Gambar 4.11 Kondisi 1 jalur B	46
Gambar 4.12 Script Kondisi 1 jalur B.....	47
Gambar 4.13 Monitoring Jarak Objek Kondisi 1 jalur B.....	48
Gambar 4.14 Kondisi 1 jalur B	48
Gambar 4.15 Script Kondisi 1 jalur B.....	49
Gambar 4.16 Monitoring Jarak Objek Kondisi 1 jalur B.....	50
Gambar 4.17 Kondisi 2 Jalur A	50
Gambar 4.18 Script Kondisi 2 Jalur A.....	51
Gambar 4.19 Monitoring Jarak Objek Kondisi 2 jalur A.....	51
Gambar 4.20 Kondisi 2 Jalur A	52
Gambar 4.21 Script Kondisi 2 Jalur A.....	53
Gambar 4.22 Monitoring Jarak Objek Kondisi 2 jalur A.....	53
Gambar 4.23 Kondisi 2 Jalur B	53

Gambar 4.24 Script Kondisi 2 Jalur B	54
Gambar 4.25 Monitoring Jarak Objek Kondisi 2 jalur B.....	55
Gambar 4.26 Kondisi 2 Jalur B	55
Gambar 4.27 Script Kondisi 2 Jalur B	56
Gambar 4.28 Monitoring Jarak Objek Kondisi 2 jalur B.....	56
Gambar 4.29 Kondisi 3 Jalur A	57
Gambar 4.30 Script Kondisi 3 Jalur A.....	58
Gambar 4.31 Monitoring Jarak Objek Kondisi 3 jalur A.....	58
Gambar 4.32 Kondisi 3 Jalur A	58
Gambar 4.33 Script Kondisi 3 Jalur A.....	59
Gambar 4.34 Monitoring Jarak Objek Kondisi 3 jalur A.....	60
Gambar 4.35 Kondisi 3 Jalur B	60
Gambar 4.36 Script Kondisi 3 Jalur B	61
Gambar 4.37 Monitoring Jarak Objek Kondisi 3 jalur B.....	61
Gambar 4.38 Kondisi 3 Jalur B	62
Gambar 4.39 Script Kondisi 3 Jalur B	63
Gambar 4.40 Monitoring Jarak Objek Kondisi 3 jalur B.....	63
Gambar 4.41 Kondisi 4 Jalur A	63
Gambar 4.42 Script Kondisi 4 Jalur A.....	64
Gambar 4.43 Monitoring Jarak Objek Kondisi 4 jalur A.....	65
Gambar 4.44 Kondisi 4 Jalur A	65
Gambar 4.45 Script Kondisi 4 Jalur A.....	66

Gambar 4.46 Monitoring Jarak Objek Kondisi 4 jalur A.....	66
Gambar 4.47 Kondisi 4 Jalur B	67
Gambar 4.48 Script Kondisi 4 Jalur B	68
Gambar 4.49 Monitoring Jarak Objek Kondisi 4 jalur B.....	68
Gambar 4.50 Kondisi 4 Jalur B	68
Gambar 4.51 Script Kondisi 4 Jalur B	69
Gambar 4.52 Monitoring Jarak Objek Kondisi 4 jalur B.....	70
Gambar 4.53 Kondisi 5 Jalur A	70
Gambar 4.54 Script Kondisi 5 Jalur A	71
Gambar 4.55 Monitoring Jarak Objek Kondisi 5 jalur A.....	71
Gambar 4.56 Kondisi 5 Jalur A	72
Gambar 4.57 Script Kondisi 5 Jalur A	73
Gambar 4.58 Monitoring Jarak Objek Kondisi 5 jalur A.....	73
Gambar 4.59 Kondisi 5 Jalur B	73
Gambar 4.60 Script Kondisi 5 Jalur B	74
Gambar 4.61 Monitoring Jarak Objek Kondisi 5 jalur B.....	75
Gambar 4.62 Kondisi 5 Jalur B	75
Gambar 4.63 Script Kondisi 5 Jalur B	76
Gambar 4.64 Monitoring Jarak Objek Kondisi 5 jalur B.....	76
Gambar 4.65 Kondisi 6 Jalur A	77
Gambar 4.66 Script Kondisi 6 Jalur A	78
Gambar 4.67 Monitoring Jarak Objek Kondisi 6 jalur A.....	78

Gambar 4.68 Kondisi 6 Jalur A	78
Gambar 4.69 Script Kondisi 6 Jalur A	79
Gambar 4.70 Monitoring Jarak Objek Kondisi 6 jalur A.....	80
Gambar 4.71 Kondisi 6 Jalur B	80
Gambar 4.72 Script Kondisi 6 Jalur B	81
Gambar 4.73 Monitoring Jarak Objek Kondisi 6 jalur B.....	81
Gambar 4.74 Kondisi 6 Jalur B	82
Gambar 4.75 Script Kondisi 6 Jalur B	83
Gambar 4.76 Monitoring Jarak Objek Kondisi 6 jalur B.....	83
Gambar 4.77 Kondisi 7 Jalur A	83
Gambar 4.78 Script Kondisi 7 Jalur A	84
Gambar 4.79 Monitoring Jarak Objek Kondisi 7 jalur A.....	85
Gambar 4.80 Kondisi 7 Jalur A	85
Gambar 4.81 Script Kondisi 7 Jalur A	86
Gambar 4.782 Monitoring Jarak Objek Kondisi 7 jalur A.....	86
Gambar 4.83 Kondisi 7 Jalur B	87
Gambar 4.84 Script Kondisi 7 Jalur B	88
Gambar 4.85 Monitoring Jarak Objek Kondisi 7 jalur B.....	88
Gambar 4.86 Kondisi 7 Jalur B	88
Gambar 4.87 Script Kondisi 7 Jalur B	89
Gambar 4.88 Monitoring Jarak Objek Kondisi 7 jalur B.....	90
Gambar 4.89 Kondisi 8 Jalur A	90

Gambar 4.90 Script Kondisi 8 Jalur A	91
Gambar 4.91 Monitoring Jarak Objek Kondisi 8 jalur A.....	91
Gambar 4.92 Kondisi 8 Jalur A	92
Gambar 4.93 Script Kondisi 8 Jalur A	93
Gambar 4.94 Monitoring Jarak Objek Kondisi 8 jalur A.....	93
Gambar 4.95 Kondisi 8 Jalur B	93
Gambar 4.96 Script Kondisi 8 Jalur B	94
Gambar 4.97 Monitoring Jarak Objek Kondisi 8 jalur B.....	95
Gambar 4.98 Kondisi 8 Jalur B	95
Gambar 4.99 Script Kondisi 8 Jalur B	96
Gambar 4.100 Monitoring Jarak Objek Kondisi 8 jalur B.....	96
Gambar 4.101 Kondisi 9 Jalur A	97
Gambar 4.102 Script Kondisi 9 Jalur A	98
Gambar 4.103 Monitoring Jarak Objek Kondisi 9 jalur A.....	98
Gambar 4.104 Kondisi 9 Jalur A	98
Gambar 4.105 Script Kondisi 9 Jalur A	99
Gambar 4.106 Monitoring Jarak Objek Kondisi 9 jalur A.....	100
Gambar 4.107 Kondisi 9 Jalur B	100
Gambar 4.108 Script Kondisi 9 Jalur B	101
Gambar 4.109 Monitoring Jarak Objek Kondisi 9 jalur B.....	101
Gambar 4.110 Kondisi 9 Jalur B	102
Gambar 4.111 Script Kondisi 9 Jalur B	103

Gambar 4.112 Monitoring Jarak Objek Kondisi 9 jalur B.....	103
Gambar 4.113 Ilustrasi Prototype	105
Gambar 4.114 Flowchart Kondisi 1	107
Gambar 4.115 Flowgraph Kondisi 1	108
Gambar 4.116 Flowchart Kondisi 2	109
Gambar 4.117 Flowgraph Kondisi 2	110
Gambar 4.118 Flowchart Kondisi 3	111
Gambar 4.119 Flowgraph Kondisi 3	112
Gambar 4.120 Flowchart Kondisi 4	113
Gambar 4.121 Flowgraph Kondisi 4	114
Gambar 4.122 Flowchart Kondisi 5	115
Gambar 4.123 Flowgraph Kondisi 5	116
Gambar 4.124 Flowchart Kondisi 6	117
Gambar 4.125 Flowgraph Kondisi 6	118
Gambar 4.126 Flowchart Kondisi 7	119
Gambar 4.127 Flowgraph Kondisi 7	120
Gambar 4.128 Flowchart Kondisi 8	121
Gambar 4.129 Flowgraph Kondisi 8	122
Gambar 4.130 Flowchart Kondisi 9	123
Gambar 4.131 Flowgraph Kondisi 9	124
Gambar 4.132 Ilustrasi Kondisi 1	125
Gambar 4.133 Ilustrasi Kondisi 2	126

Gambar 4.134 Ilustrasi Kondisi 3	127
Gambar 4.135 Ilustrasi Kondisi 4	127
Gambar 4.136 Ilustrasi Kondisi 5	128
Gambar 4.137 Ilustrasi Kondisi 6	128
Gambar 4.138 Ilustrasi Kondisi 7	129
Gambar 4.139 Ilustrasi Kondisi 8	129
Gambar 4.140 Ilustrasi Kondisi 9	130

BAB I

PENDAHULUAN

1.1 Latar Belakang

Pada era modern sekarang ini teknologi transportasi semakin pesat dan maju baik dari segi kualitas maupun kuantitas. Sarana transportasi kendaraan menyebabkan pengguna alat-alat transportasi bebas memilih alat transportasi yang sesuai kondisi dan kebutuhan.

Setiap tahun peningkatan jumlah kendaraan semakin bertambah, hal ini berakibat munculnya masalah lalu lintas kendaraan, misalnya kemacetan lalu lintas, kecelakaan lalu lintas, polusi udara, dan sebagainya. Kemacetan lalu lintas telah menjadi masalah signifikan. Beberapa hal yang telah dilakukan untuk menangani masalah kemacetan misalnya pelebaran trotoar, pelebaran jalan, dan pengadaan lampu lalu lintas (*Traffic Light*). Namun hal ini belum mampu mengurangi kemacetan, lampu lalu lintas (*Traffic Light*) pada persimpangan jalan sebenarnya sudah sangat membantu untuk mengurangi kemacetan kota, namun penggunaan *Traffic Light* memiliki kelemahan yaitu pengaturan kendaraan tidak berdasarkan kondisi *real-time* dari jumlah kendaraan tetapi berdasarkan batas waktu, sehingga terjadi penumpukan kendaraan pada lampu lalu lintas karena waktu nyala *traffic light* tidak sesuai dengan jumlah kendaraan tiap sisi persimpangan jalan yang dapat menimbulkan kepadatan kendaraan pada *traffic light*.

Sistem kendali atau sistem kontrol (*control system*) adalah suatu alat (kumpulan alat) untuk mengendalikan, memerintah, dan mengatur keadaan dari suatu sistem. Sistem kendali ini bisa menjadi salah satu solusi untuk mengatasi kelemahan dari penggunaan *traffic light*.

Beberapa penelitian yang pernah dilakukan sebelumnya tentang sistem kendali lampu lalu lintas diantaranya, penelitian yang dilakukan oleh rahmat hidayat, pada tahun 2014, dengan judul “Prototipe Sistem Kontrol Lampu Lalu Lintas Otomatis Menggunakan LDR Sebagai Sensor Kepadatan Kendaraan Berbasis Mikrokontroler Atmega16”. Sistem ini menggunakan sensor LDR sebagai pendeteksi kepadatan kendaraan. Ketika sensor mendeteksi jumlah arus

kendaraan dalam kondisi sedang atau ramai maka akan menghasilkan output nyala lampu hijau menjadi lebih lama, sedangkan untuk nyala lampu merah yang terdeteksi jumlah arus kendaraan sepi maka akan menyala lebih lama dan hijau lebih cepat.

Penelitian selanjutnya oleh Roria M. Kristina Sitorus, pada tahun 2018, dengan judul “Sistem Kontrol *Traffic Light* Berdasarkan Jumlah Kendaraan Untuk Mengurangi Kemacetan Berbasis Arduino”. Pada rancangan ini menggunakan sensor *Proximity* yang berfungsi untuk mendeteksi suatu obyek benda berdasarkan jarak benda tersebut terhadap sensor. Pada pengujian baca sensor ini akan menggunakan sensor *infrared proximity* yang berfungsi untuk mendeteksi setiap kendaraan yang lewat pada setiap persimpangan pada jarak tertentu dan setiap kendaraan yang sudah dideteksi oleh *Infrared* maka hasilnya akan dibaca oleh Fotodioda. Apabila Fotodioda terkena cahaya, maka akan bersifat sebagai sumber tegangan dan nilai resistansinya akan kecil tetapi apabila Fotodioda tidak terkena cahaya, maka nilai resistansinya akan besar atau diasumsikan tak terhingga.

Berdasarkan uraian sebelumnya, maka penelitian yang akan dilakukan dalam penelitian ini yaitu membuat sebuah prototype sistem kontrol lampu lalu lintas berbasis arduino UNO. Hal-hal yang membedakan penelitian ini dengan penelitian-penelitian sebelumnya yaitu, disini peneliti menggunakan sensor ultrasonik HC SR-04 untuk mendeteksi objek. Input dari sistem merupakan hasil pembacaan objek menggunakan sensor ultrasonik. Sensor ultrasonik mendeteksi objek kondisi kepadatan kendaraan pada lampu lalu lintas, kemudian Arduino UNO R3 mengolah hasil sensor ultrasonik, hasil dari proses pembacaan objek sensor ultrasonik menjadi input data yang akan digunakan untuk menentukan tingkat kepadatan kendaraan yang akan dijadikan sebagai parameter kendali lampu lalu lintas dengan mengatur lama waktu yang akan diberikan pada tiap sisi lampu lalu lintas. Diharapkan dengan adanya sistem ini dapat mengurangi kepadatan kendaraan pada lampu lalu lintas dan penggunaan lampu lalu lintas menjadi lebih efektif.

1.2 Rumusan Masalah

Sesuai dengan latar belakang masalah yang telah di kemukakan, maka dapat di uraikan rumusan masalah, sebagai berikut :

1. Bagaimana merancang sistem lampu lalu lintas berdasarkan panjang antrian kendaraan dimasing-masing jalur menggunakan sensor ultrasonik.
2. Bagaimana menentukan tingkat kepadatan kendaraan pada tiap jalur dalam tiga kategori yaitu sepi, cukup padat, dan padat.
3. Bagaimana menentukan dan menyesuaikan tingkat kepadatan kendaraan dengan waktu nyala traffic light, sehingga terdapat perbedaan pengaturan lampu lalu lintas berdasarkan tingkat kepadatan kendaraan.

1.3 Tujuan Penelitian

Tujuan dari penelitian ini adalah :

1. Membuat sisitem kendali lampu lalu lintas berdasarkan panjang antrian kendaraan dimasing-masing jalur menggunakan sensor ultrasonik.
2. Membuat sisitem untuk menentukan tingkat kepadatan kendaraan pada tiap jalur dalam tiga kategori yaitu sepi, cukup padat, dan padat.
3. Merancang lampu lalu lintas dengan pengaturan lama waktu lampu menyala berdasarkan tingkat kepadatan kendaraan.

1.4 Batasan Masalah

Dalam penelitian ini terdapat batasan masalah, yaitu :

1. Untuk mendeteksi panjang antrin menggunakan sensor ultrasonic HC-SR04.
2. Jarak jangkauan sensor ultrasonik untuk mendeteksi objek maksimal 10 cm.
3. Tingkat kepadatan tiap jalur dibagi dalam tiga kategori yaitu sepi, cukup padat, dan padat.
4. Menggunakan prototype peremptan jalan raya dengan jalur searah ukuran 40x40 cm atau dengan skala perbandingan 1:15 dengan jalan raya sesungguhnya.

5. Perempatan satu arah antara jalan manggarupi dan jalan andi tonro.

BAB II

LANDASAN TEORI

II. Kerangka Teori

II.1 Prototype

Menurut Darmawan S (2021), Prototype atau prototipe adalah sebuah metode dalam pengembangan produk dengan cara membuat rancangan, sampel, atau model dengan tujuan pengujian konsep atau proses kerja dari produk. Prototype sendiri bukanlah produk final yang nantinya akan diedarkan. Prototype dibuat untuk kebutuhan awal *development software* dan untuk mengetahui apakah fitur dan fungsi dalam program berjalan sesuai dengan kebutuhan yang telah direncanakan. Sehingga pengembang produk dapat mengetahui kekurangan dan kesalahan lebih awal sebelum mengimplementasikan fitur lain ke dalam produk dan merilis produk.

Tujuan utama dari prototype adalah mengembangkan model atau rancangan produk menjadi produk final yang dapat memenuhi permintaan pengguna. Dalam proses pengembangan produk, pengguna dapat ikut andil dalam proses pengembangan produk dengan cara mengevaluasi dan memberikan umpan balik. Umpan balik yang diberikan dapat digunakan sebagai acuan dalam pengembangan produk. Selain itu, penggunaan prototipe dapat memunculkan ide-ide baru yang bisa dikembangkan menjadi sebuah fitur untuk melengkapi produk.

Ada banyak manfaat dan keuntungan yang bisa kamu dapatkan saat menggunakan sistem prototyping ini. Berikut adalah manfaatnya.

1. Dapat menghemat waktu dan biaya pengembangan produk

Yang pertama adalah kamu dapat menekan biaya dan menghemat waktu dalam proses pengembangan produk. Dengan begitu, sumber daya yang tersisa dapat dialokasikan untuk kebutuhan yang lain.

2. **Dapat mengetahui kebutuhan pengguna terlebih dahulu**

Manfaat yang kedua yaitu dengan memanfaatkan sistem prototyping kamu dapat mengetahui kebutuhan pengguna terlebih dahulu. Sehingga kamu dan tim-mu dapat mengetahui apa saja prioritas dan kebutuhan pengguna. Dengan begitu proses pengembangan produk akan berlangsung lebih cepat.

3. Menjadi acuan untuk mengembangkan produk

Selanjutnya, model prototype dapat menjadi acuan atau patokan untuk kamu dalam mengembangkan sebuah produk. Kamu juga dapat menggunakan prototype untuk menemukan kekurangan dan mencari solusi untuk membuat produk kamu menjadi semakin baik lagi.

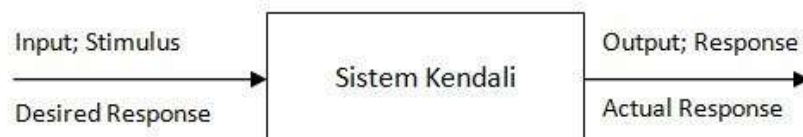
4. Dapat menjadi bahan presentasi

Terakhir, kamu dapat menggunakan prototype untuk mempresentasikan produk yang akan kamu luncurkan. Dengan adanya prototype akan memudahkan audiens mendapatkan gambaran tentang produk yang akan diluncurkan.

II.2 Sistem Kendali

Menurut Khamami F. (2017), Sistem adalah suatu susunan, set, atau sekumpulan sesuatu yang terhubung atau terkait sedemikian rupa sehingga membentuk sesuatu secara keseluruhan. Sistem adalah susunan komponen fisik yang terhubung atau terkait sedemikian rupa sehingga membentuk atau bertindak sebagai seluruh unit dalam satu kesatuan. Sedangkan kata kontrol atau kendali biasanya diartikan mengatur, mengarahkan, atau perintah. Dari kedua makna kata sistem dan kontrol/kendali, sistem kendali adalah suatu susunan komponen fisik yang terhubung atau terkait sedemikian rupa sehingga dapat memerintah, mengarahkan, atau mengatur diri sendiri atau sistem lain. Di dalam dunia *engineering* dan *science* sistem kendali cenderung dimaksudkan untuk sistem kendali dinamis.

Sistem kendali terdiri dari sub-sistem dan proses (atau *plants*) yang disusun untuk mendapatkan keluaran(output) dan kinerja yang diinginkan dari input yang diberikan. Gambar di bawah ini menunjukkan blok diagram untuk sistem kendali paling sederhana, sistem kendali membuat sistem dengan input yang diberikan menghasilkan output yang diharapkan.



Gambar 2. 1 *Sistem kendali*
(Sumber : te.unib.ac.id)

II.3 Lampu Lalu Lintas

Menurut UU No. 22/2009, Lampu lalu lintas adalah lampu yang mengendalikan arus [lalu lintas](#) yang terpasang di persimpangan [jalan](#), tempat penyeberangan [pejalan kaki](#) (*zebra cross*), dan tempat arus lalu lintas lainnya. Lampu ini yang menandakan kapan [kendaraan](#) harus berjalan dan berhenti secara bergantian dari berbagai [arah](#). Pengaturan lalu lintas di persimpangan jalan dimaksudkan untuk mengatur pergerakan kendaraan pada masing-masing kelompok pergerakan kendaraan agar dapat bergerak secara bergantian sehingga tidak saling mengganggu antar-arus yang ada.

Lampu lalu lintas telah diadopsi di hampir semua kota di dunia ini. Lampu ini menggunakan [warna](#) yang diakui secara universal; untuk menandakan berhenti adalah warna [merah](#), hati-hati yang ditandai dengan warna [kuning](#), dan [hijau](#) yang berarti dapat berjalan.

Menurut Pamor Gunoto, M. Irsyam, dan Toni Kusuma Wijaya (2016) kepadatan kendaraan dan panjang antrian kendaraan ditentukan berdasarkan 4 kondisi, keadaan kosong jika ketiga sensor tidak aktif (0 meter), keadaan sepi jika hanya sensor pertama yang aktif (0-4 meter), keadaan normal jika dua sensor aktif (4-8 meter), dan kondisi padat jika ketiga sensor aktif (8-12 meter).

II.4 Arduino Software (IDE)

Menurut Dewi, R. P(2018), IDE itu merupakan kependekan dari *Integrated Development Environment*, atau secara bahasa mudahnya merupakan lingkungan terintegrasi yang digunakan untuk melakukan pengembangan. Disebut sebagai lingkungan karena melalui software inilah Arduino dilakukan pemrograman untuk melakukan fungsi-fungsi yang dibenamkan melalui sintaks pemrograman. Arduino menggunakan bahasa pemrograman sendiri yang menyerupai bahasa C. Bahasa pemrograman Arduino (*Sketch*) sudah dilakukan perubahan untuk memudahkan pemula dalam melakukan pemrograman dari bahasa aslinya. Sebelum dijual ke pasaran, IC mikrokontroler Arduino telah ditanamkan suatu program bernama *Bootlader* yang berfungsi sebagai penengah antara *compiler* Arduino dengan mikrokontroler.

- **Sketch**

Program yang ditulis dengan menggunakan Arduino Software (IDE) disebut sebagai *sketch*. Sketch ditulis dalam suatu editor teks dan disimpan dalam file dengan ekstensi *.ino*. Teks editor pada Arduino Software memiliki fitur” seperti *cutting/paste* dan *seraching/replacing* sehingga memudahkan kamu dalam menulis kode program. Pada Software Arduino IDE, terdapat semacam *message box* berwarna hitam yang berfungsi menampilkan status, seperti pesan *error*, *compile*, dan *upload* program. Di bagian bawah paling kanan Software Arduino IDE, menunjukkan *board* yang terkonfigurasi beserta COM Ports yang digunakan.



Gambar 2. 2 tampilan dari Software Arduino IDE
(Sumber : sinauarduino.com)



Verify

Berfungsi untuk melakukan checking kode yang kamu buat apakah sudah sesuai dengan kaidah pemrograman yang ada atau belum

Upload



Berfungsi untuk melakukan kompilasi program atau kode yang kamu buat menjadi bahasa yang dapat dipahami oleh mesin alias si Arduino.



New

Berfungsi untuk membuat Sketch baru



Open

Berfungsi untuk membuka *sketch* yang pernah kamu buat dan membuka kembali untuk dilakukan editing atau sekedar upload ulang ke Arduino.



Save

Berfungsi untuk menyimpan *Sketch* yang telah kamu buat.



Serial Monitor

Berfungsi untuk membuka serial monitor. Serial monitor disini merupakan jendela yang menampilkan data apa saja

yang dikirimkan atau dipertukarkan antara arduino dengan sketch pada port serialnya. Serial Monitor ini sangat berguna sekali ketika kamu ingin membuat program atau melakukan debugging tanpa menggunakan LCD pada Arduino. Serial monitor ini dapat digunakan untuk menampilkan nilai proses, nilai pembacaan, bahkan pesan error.

II.5 Bahasa Pemrograman C++

Menurut Andre (2020), Bahasa Pemrograman C++ adalah bahasa pemrograman komputer yang bisa dipakai untuk membuat berbagai aplikasi (*general-purpose programming language*), dan merupakan pengembangan dari bahasa pemrograman C.

Bahasa C++ kadang disebut juga sebagai “C with Classes“. Ini karena perbedaan paling utama antara bahasa C dan C++ ada di dukungan ke pemrograman berbasis object (*object oriented programming*). Bahasa C++ sudah mendukung pemrograman berbasis object, sedangkan bahasa C belum. Karena termasuk *general-purpose programming language*, yakni bahasa pemrograman yang bisa membuat berbagai aplikasi, bahasa pemrograman C++ bisa dipakai untuk membuat aplikasi desktop seperti antivirus, software pengolah gambar (*image processing*), aplikasi pengolah kata (*word processing*), hingga untuk membuat compiler bahasa pemrograman lain.



Gambar 2. 3 Logo C++
(Sumber : duniaikom.com)

II.6 Pengujian Sistem

Menurut Peni Kurniawati (2018), Pengujian sistem adalah pengujian program perangkat lunak yang lengkap dan terintegrasi. Perangkat lunak atau yang sering dikenal dengan sebutan *software* hanyalah satuan elemen dari sistem berbasis komputer yang lebih besar. Biasanya, perangkat lunak dihubungkan dengan perangkat lunak dan perangkat keras lainnya.

Pengujian perangkat lunak *Black Box Testing* dan *white Box Testing*.

- ***Black Box Testing***



Gambar 2. 4 Sistem pengujian
(Sumber : medium.com)

Black Box Testing atau yang sering dikenal dengan sebutan pengujian fungsional merupakan metode pengujian Perangkat Lunak yang digunakan untuk menguji perangkat lunak tanpa mengetahui struktur internal kode atau Program. Dalam pengujian ini, *tester* menyadari apa yang harus dilakukan oleh program tetapi tidak memiliki pengetahuan tentang bagaimana melakukannya.

- ***White Box Testing***

White box testing atau yang dapat diartikan menjadi “pengujian kotak putih” adalah pengujian yang dilakukan untuk menguji

perangkat lunak dengan cara menganalisa dan meneliti struktur *internal* dan kode dari perangkat lunak. Lain halnya dengan *black box testing* yang hanya melihat hasil *input* dan *output* dari perangkat lunak, pengujian *white box testing* berfokus pada aliran *input* dan *output* dari perangkat lunak.

Untuk melakukan pengujian ini, penguji/tester perlu memiliki kemampuan dalam memahami kode dari suatu program sehingga pengujian ini tidak bisa dilakukan oleh sembarang orang.

Teknik-Teknik Pengujian

Berikut ini adalah teknik yang dapat digunakan untuk melakukan *white box testing* pada perangkat lunak:

1. Basis path testing

Teknik pertama adalah basis path testing. Teknik bertujuan untuk mengukur kompleksitas kode program dan mendefinisikan alur yang dieksekusi.

2. Branch coverage

Kemudian ada branch coverage. Pengujian ini dirancang agar setiap branch code diuji setidaknya satu kali.

3. Condition coverage

Selanjutnya adalah teknik condition coverage, tujuannya untuk menguji seluruh kode agar menghasilkan nilai TRUE atau FALSE. Dengan begitu, tester dapat memastikan perangkat lunak dapat bekerja dan mengeluarkan output sesuai dengan input dari pengguna.

4. Loop testing

Ada juga teknik loop testing. Pengujian ini yang wajib dilakukan untuk menguji berbagai perulangan/looping yang ada dalam program, seperti do-while, for, dan while. Dalam

pengujian ini kamu juga dapat memeriksa kondisi dari perulangan, apakah sudah berjalan dengan benar atau tidak.

5. Multiple condition coverage

Berikutnya adalah multiple condition coverage. Teknik ini dilakukan untuk menguji seluruh kombinasi dari kode yang mungkin digunakan dalam berbagai kondisi. Seluruh kombinasi harus diuji minimal satu kali, tujuannya untuk memastikan perangkat lunak agar berjalan dengan baik.

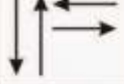















6. Statement coverage

Teknik terakhir adalah statement coverage. Teknik ini dilakukan minimal satu kali untuk menguji setiap statement dalam perangkat lunak. Dengan pengujian ini, kamu dapat mengetahui kode-kode yang error sehingga dapat segera memperbaikinya.

II.7 Flowchart

Menurut Wibawanto (2017:20), Flowchart adalah adalah suatu bagan dengan simbol-simbol tertentu yang menggambarkan urutan proses secara mendetail dan hubungan antara suatu proses (instruksi) dengan proses lainnya dalam suatu program.

Dalam perancangan flowchart sebenarnya tidak ada rumus atau patokan yang bersifat mutlak (pasti). Hal ini didasari oleh flowchart (bagan alir) adalah sebuah gambaran dari hasil pemikiran dalam menganalisa suatu permasalahan dalam komputer. Karena setiap analisa akan menghasilkan hasil yang bervariasi antara satu dan lainnya. Kendati begitu secara garis besar setiap perancangan flowchart selalu

	Flow Direction symbol Yaitu simbol yang digunakan untuk menghubungkan antara simbol yang satu dengan simbol yang lain. Simbol ini disebut juga connecting line.		Simbol Manual Input Simbol untuk memasukan data secara manual on-line keyboard
	Terminator Symbol Yaitu simbol untuk permulaan (start) atau akhir (stop) dari suatu kegiatan		Simbol Preparation Simbol untuk mempersiapkan penyimpanan yang akan digunakan sebagai tempat pengolahan di dalam storage.
	Connector Symbol Yaitu simbol untuk keluar - masuk atau penyambungan proses dalam lembar / halaman yang sama.		Simbol Predefine Proses Simbol untuk pelaksanaan suatu bagian (sub-program)/prosedure
	Connector Symbol Yaitu simbol untuk keluar - masuk atau penyambungan proses pada lembar / halaman yang berbeda.		Simbol Display Simbol yang menyatakan peralatan output yang digunakan yaitu layar, plotter, printer dan sebagainya.
	Processing Symbol Simbol yang menunjukkan pengolahan yang dilakukan oleh komputer		Simbol disk and On-line Storage Simbol yang menyatakan input yang berasal dari disk atau disimpan ke disk.
	Simbol Manual Operation Simbol yang menunjukkan pengolahan yang tidak dilakukan oleh computer		Simbol magnetik tape Unit Simbol yang menyatakan input berasal dari pita magnetik atau output disimpan ke pita magnetik.
	Simbol Decision Simbol pemilihan proses berdasarkan kondisi yang ada,		Simbol Punch Card Simbol yang menyatakan bahwa input berasal dari kartu atau output ditulis ke kartu
	Simbol Input-Output Simbol yang menyatakan proses input dan output tanpa tergantung dengan jenis peralatannya		Simbol Dokumen Simbol yang menyatakan input berasal dari dokumen dalam bentuk kertas atau output dicetak ke kertas.

terdiri dari tiga bagian, yaitu input proses dan output.
 Gambar 2.5 Simbol Flowchart
 (Sumber : informatikalogi.com)

II.8 Arduino Uno

Menurut R Pradana (2017), Arduino Uno adalah board mikrokontroler berbasis ATmega328 (datasheet). Memiliki 14 pin input dari output digital dimana 6 pin input tersebut dapat digunakan sebagai output PWM dan 6 pin input analog, 16 MHz osilator kristal, koneksi

USB, jack power, ICSP header, dan tombol reset. Untuk mendukung mikrokontroler agar dapat digunakan, cukup hanya menghubungkan Board Arduino Uno ke komputer dengan menggunakan kabel USB atau listrik dengan AC yang ke adaptor-DC atau baterai untuk menjalankannya.

Uno berbeda dengan semua board sebelumnya dalam hal koneksi USB-to-serial yaitu menggunakan fitur Atmega8U2 yang diprogram sebagai konverter USB-to-serial berbeda dengan board sebelumnya yang menggunakan chip FTDI driver USB-to-serial.

Nama “Uno” berarti *satu* dalam bahasa Italia, untuk menandai peluncuran Arduino 1.0. Uno dan versi 1.0 akan menjadi versi referensi dari Arduino. Uno adalah yang terbaru dalam serangkaian board USB Arduino, dan sebagai model referensi untuk platform Arduino, untuk perbandingan dengan versi sebelumnya, lihat indeks board Arduino.



Gambar 2. 6 Arduino UNO
(Sumber : ilearning.me)

II.9 Sensor Ultrasonik HC-SR04

HC-SR04 adalah sebuah modul sensor ultrasonik yang biasanya digunakan untuk alat pengukur jarak. Pada artikel ini kita akan mempelajari bagaimana cara kerja dari sensor HC-SR04 dan berikut contoh programnya dengan menggunakan Arduino.

Pada HC-SR04 terdapat sepasang transducer ultrasonik yang satu berfungsi sebagai transmitter yang bertugas untuk mengubah sinyal elektrik menjadi sinyal ultrasonik dengan frekuensi 40KHz, dan satu sebagai receiver yang bertugas untuk menerima sinyal



Gambar 2.7 Sensor HC-SR04
(Sumber : www.nn-digital.com)

- **Cara Kerja HC-SR04**

Sebuah sinyal pulsa dengan durasi setidaknya 10 μ S (10 mikrodetik) diterapkan ke pin *Trigger*. Setelah itu, sensor mentransmisikan gelombang ultrasonik delapan pulsa pada frekuensi 40 KHz. Pola 8-pulsa ini digunakan untuk sebuah penanda sinyal ultrasonik dari modul ini, yang memungkinkan receiver / penerima untuk membedakan pola yang ditransmisikan dari kebisingan ultrasonik sekitar.

Delapan pulsa ultrasonik bergerak melalui udara menjauh dari transmitter / pemancar mengarah ke benda atau obyek yang ada di depannya. Sementara itu pin *Echo* menjadi **HIGH / TINGGI** untuk mulai membentuk awal sinyal gema.

Jika tidak ada sinyal ultrasonik yang dipantulkan atau diterima oleh receiver selama rentang 38 mS (mili detik), yang artinya tidak ada obyek atau benda maka sinyal *Echo* akan Timeout dan kembali menjadi **LOW / RENDAH**.

Sedangkan jika ada sinyal ultrasonik yang dipantulkan atau diterima oleh receiver, maka saat itu juga sinyal *Echo* langsung berubah menjadi **LOW / RENDAH**. Nah, lebar rentang waktu dari sinyal ECHO inilah yang digunakan untuk mengukur jarak antara sensor dengan obyek atau benda.

Dengan menggunakan persamaan *jarak* – *kecepatan* – *waktu* dari gelombang suara yang merambat pada udara, maka bisa dijabarkan sebagai berikut :

Jarak = Kecepatan x Waktu

Dimana kecepatan gelombang suara pada udara adalah = 340 m/s = 0.034 cm/ μ S. Karena jarak tempuh gelombang suara ultrasonik tadi adalah bolak-balik yaitu dari sensor (transmitter) ke obyek dan kembali ke sensor (receiver), maka rumusnya menjadi :

Jarak (cm) = Waktu (μ S) * 0.034 / 2

II.10 LED (Light Emitting Diode)

Menurut Sarif Hidayatullah (2020), Light Emitting Diode atau sering disingkat dengan LED adalah komponen elektronika yang dapat memancarkan cahaya monokromatik ketika diberikan tegangan maju. LED merupakan keluarga Dioda yang terbuat dari bahan semikonduktor. Warna-warna Cahaya yang dipancarkan oleh LED tergantung pada jenis bahan semikonduktor yang dipergunakannya. LED juga dapat memancarkan sinar inframerah yang tidak tampak oleh mata seperti yang sering kita jumpai pada Remote Control TV ataupun



Gambar 2.8 LED Remote Control perangkat elektronik lainnya.
(Sumber : teknikelektronika.com)

II.11 Kabel Jumper

Menurut Aldy Razor (2021), Kabel *jumper* adalah kabel elektrik yang memiliki pin konektor di setiap ujungnya dan memungkinkanmu

untuk menghubungkan dua komponen yang melibatkan Arduino tanpa memerlukan solder. Intinya kegunaan kabel *jumper* ini adalah sebagai konduktor listrik untuk menyambungkan rangkaian listrik.

Biasanya kabel *jumper* digunakan pada *breadboard* atau alat *prototyping* lainnya agar lebih mudah untuk mengutak-atik rangkaian. Konektor yang ada pada ujung kabel terdiri atas dua jenis yaitu konektor jantan (*male connector*) dan konektor betina (*female connector*). Konektor jantan fungsinya untuk menusuk dan konektor betina fungsinya untuk ditusuk.



Gambar 2. 9 Kabel Jumper
(Sumber : aldyrazor.com)

Gambar di atas menunjukkan seperti apa bentuk dari yang namanya *male connector* dan *female connector*.

1. Jenis jenis kabel jumper yang paling umum adalah sebagai berikut:

- **Kabel Jumper Male to Male**

Jenis yang pertama adalah kabel jumper *male male*. Kabel jumper male to male adalah jenis yang sangat yang sangat cocok untuk kamu yang mau membuat rangkaian elektronik di *breadboard*.



Gambar 2. 10 Kabel Jumper Male To Male
(Sumber : aldyrazor.com)

- **Kabel Jumper Male to Female**

Kabel jumper male female memiliki ujung konektor yang berbeda pada tiap ujungnya, yaitu *male* dan *female*. Biasanya kabel ini digunakan untuk menghubungkan komponen elektronika selain [Arduino](#) ke *breadboard*



Gambar 2. 11 Kabel Jumper Male To Female
(Sumber : aldyrazor.com)

- **Kabel Jumper Female to Female**

Jenis kabel jumper yang terakhir adalah kabel *female to female*. Kabel ini sangat cocok untuk menghubungkan antar komponen yang memiliki *header male*. contohnya seperti sensor ultrasonik *HC-SR04*, sensor suhu *DHT*, dan masih banyak lagi.



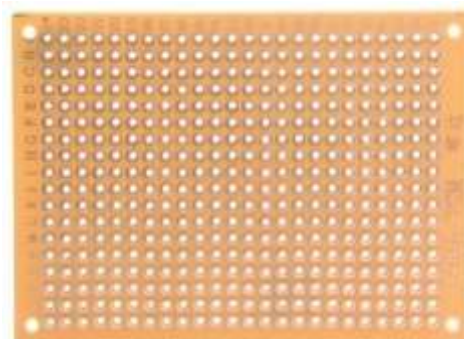
Gambar 2. 12 Kabel Jumper Female To Female
(Sumber : aldyrazor.com)

2. Cara Kerja Kabel *Jumper*

Singkatnya, prinsip kerja kabel jumper yaitu menghantarkan arus listrik dari satu komponen ke komponen lainnya yang dihubungkan. Ini terjadi karena di ujung dan di dalam kabel terdapat konduktor listrik kecil yang memang fungsinya untuk menghantarkan listrik.

II.12 Papan PCB

Menurut Dickson Kho (2021), Pengertian PCB (Printed Circuit Board) PCB adalah singkatan dari *Printed Circuit Board* yang dalam bahasa Indonesia sering diterjemahkan menjadi Papan Rangkaian Cetak atau Papan Sirkuit Cetak. Seperti namanya yaitu Papan Rangkaian Tercetak (*Printed Circuit Board*), PCB adalah Papan yang digunakan untuk menghubungkan komponen-komponen Elektronika dengan lapisan jalur konduktornya. PCB ditemukan oleh seorang ilmuwan Austria yang bernama Paul Eisler pada tahun 1936. Paul Eisler menggunakan PCB pertama kalinya di sebuah rangkaian Radio. Kemudian pada tahun 1943, Amerika Serikat mulai memanfaatkan teknologi PCB ini pada Radio Militer dalam skala yang lebih besar. Tiga tahun setelah perang dunia kedua yaitu pada tahun 1948, PCB mulai digunakan untuk produk-produk komersil oleh perusahaan-perusahaan Amerika Serikat.

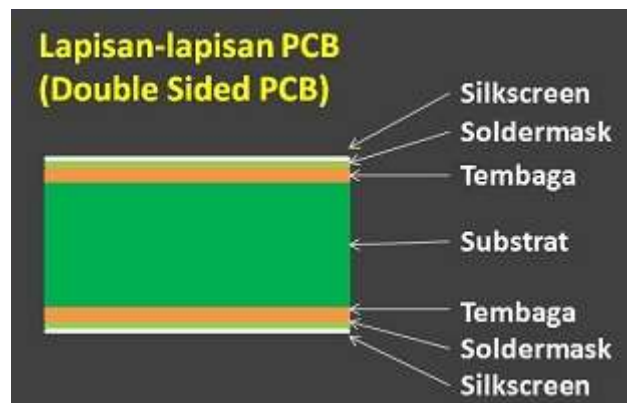


Gambar 2. 13 Papan PCB
(Sumber : pulangsore.com)

➤ Lapisan PCB dan Bahan-bahannya

Secara struktur, PCB seperti kue lapis yang terdiri dari beberapa lapisan dan dilaminasi menjadi satu kesatuan yang disebut dengan PCB. Ada PCB yang berlapis satu lapisan tembaga (Single Sided), ada juga yang berlapis dua lapisan tembaga (double sided) dan ada juga PCB yang memiliki beberapa lapisan tembaga atau sering disebut dengan Multilayer PCB.

Berikut ini adalah struktur dan komposisi standar dari PCB



(Printed Circuit Board)

- *Substrat (Lapisan Landasan)*

Lapisan dasar (landasan) PCB biasanya disebut dengan Substrat. Bahan Substrat yang paling umum digunakan adalah FR2 dan FR4. FR2 atau Flame Resistant 2 adalah kertas bonding resin sintesis (synthetic resin bonded paper) yaitu bahan komposit yang terbuat dari kertas yang diresapi dengan resin plastik formaldehida fenol (*plasticized phenol formaldehyde resin*). Sedangkan FR4 atau Flame Resistant 4 adalah anyaman Fiberglass yang dilapisi dengan resin epoksi (*epoxy resin*). FR4 memiliki daya serap air yang rendah, properti isolasi yang bagus serta tahan suhu panas hingga 140 derajat celcius. Namun, PCB yang berbahan FR4 lebih mahal jika dibandingkan dengan PCB yang berbahan FR2.

- *Tembaga (Copper)*

Lapisan PCB berikutnya adalah lapisan tembaga tipis yang dilaminasi ke lapisan substrat dengan suhu tinggi tertentu dan perekat. Tergantung pada jenis PCB-nya, lapisan tembaga tipis ini hanya akan dilapisi pada satu sisi substrat untuk jenis Single Sided PCB. Sedangkan Double Sided PCB terdapat lapisan tembaga tipis di dua sisi Substrat. Seiring dengan perkembangan Teknologi manufaktur PCB saat ini, PCB telah dapat dibuat hingga 16 lapisan atau bahkan lebih dari 16 lapisan tergantung pada perancangan PCB dan rangkaian yang diinginkan.

- *Soldermask*

Soldermask adalah lapisan diatas lapisan tembaga yang berfungsi melindungi tembaga atau jalur konduktor dari

hubungan atau kontak yang tidak disengaja. Lapisan soldermask ini hanya terdapat pada bagian-bagian PCB yang tidak disolder, sedangkan bagian yang akan disolder tidak ditutupi oleh lapisan soldermask. Lapisan soldermask ini juga dapat membantu para pengguna PCB untuk menyolder tepat pada tempatnya sehingga mencegah solder short (hubung singkat solder). Lapisan soldermask ini biasanya berwarna hijau, namun ada juga yang berwarna lain seperti warna biru dan merah.

- *Silkscreen*

Lapisan setelah soldermask adalah lapisan silkscreen yang biasanya berwarna putih atau hitam. Namun ada juga silkscreen yang berwarna lain seperti warna abu-abu, warna merah dan bahkan ada berwarna kuning keemasan. Silkscreen merupakan cetakan huruf, angka dan simbol pada PCB. Silkscreen ini berfungsi sebagai tanda atau indikator untuk komponen-komponen elektronika pada PCB sehingga mempermudah orang dalam merakitnya.

II.13 Penelitian Terdahulu (State of The Art)

Tabel 2. 1 Penelitian terdahulu

NO	Nama Peneliti	Judul Penelitian	Tahun Penelitian	Hasil/Output
1	Roria M. Kristina Sitorus	Sistem kontrol traffic light berdasarkan jumlah kendaraan untuk mengurangi kemacetan berbasis arduino	2018	Pada pengujian baca ini akan menggunakan sensor infrared proximity yang berfungsi untuk mendeteksi setiap kendaraan yang lewat

				<p>pada setiap persimpangan pada jarak tertentu dan setiap kendaraan yang sudah dideteksi oleh Infrared maka hasilnya akan dibaca oleh Fotodiode. Apabila Fotodiode terkena cahaya, maka akan bersifat sebagai sumber tegangan dan nilai resistansinya akan kecil tetapi apabila Fotodiode tidak terkena cahaya, maka nilai resistansinya akan besar atau diasumsikan tak terhingga.</p>
2	Rahmat Hidayat	<p>Prototipe sistem kontrol lampu lalu lintas otomatis menggunakan LDR sebagai sensor kepadatan kendaraan berbasis mikrokontroler atmega16</p>	2014	<p>Prototipe ini adalah simulasi sistem lampu lalu lintas otomatis. Sistem ini menggunakan sensor LDR sebagai pendeteksi kepadatan kendaraan. Ketika sensor mendeteksi jumlah arus</p>

				<p>kendaraan dalam kondisi sedang atau ramai maka akan menghasilkan output nyala lampu hijau menjadi lebih lama, sedangkan untuk nyala lampu merah yang terdeteksi jumlah arus kendaraan sepi maka akan menyala lebih lama dan hijau lebih cepat.</p>
3	IGAP. Raka Agung	<p>Imulator Pengatur Lampu Lalu Lintas Berdasarkan Waktu Dan Kepadatan Kendaraan Berbasis Mikrokontroler At89s52</p>	2019	<p>Hasil perhitungan nilai nyala lampu lalu lintas yang diperoleh dari persimpangan Jl WaribangWR.Supra tman Denpasar yaitu siklus optimum dan nyala lampu hijau masing-masing phase. Untuk Hari Senin – Kamis untuk kondisi padat 1 didapat 45 detik, 15 detik dan 20 detik, kondisi padat 2 didapat 38</p>

				<p>detik, 19 detik dan 9 detik, kondisi padat 3 didapat 39 detik, 10 detik, 19 detik dan untuk kondisi normal didapat 30 detik, 7 detik dan 13 detik. Untuk Hari Jumat – Minggu untuk kondisi padat 1 didapat 37 detik, 13 detik dan 14 detik, kondisi padat 2 didapat 39 detik, 9 detik dan 20 detik, kondisi padat 3 didapat 38 detik, 8 detik, 20 detik dan untuk kondisi normal didapat 28 detik, 8 detik dan 10 detik. Realisasi simulator sistem kontrol lampu lalu lintas berbasis mikrokontroler AT89S52 ini, diharapkan dapat</p>
--	--	--	--	--

				digunakan sesuai dengan kondisi di lapangan sehingga dapat mengurangi kemacetan.
4	Windarto dan Muhammad Haekal	Aplikasi Pengatur Lampu Lalu Lintas Berbasis Arduino Mega 2560 Menggunakan <i>Light Dependent Resistor</i> (LDR) Dan Laser	2012	Aplikasi Pengaturan Sistem Lampu Lalu Lintas berbasis Mikrokontroler Arduino Mega 2560 yang diajukan menggunakan komputer dimana memerlukan pengguna (user) dalam pemakaiannya. Disini pengguna memiliki otoritas untuk mengatur lampu lalu lintas pada salah satu persimpangan dimana aplikasi tersebut dijalankan. Aplikasi yang diajukan menggunakan komunikasi <i>serial port</i> antara komputer dengan Arduino sebagai mikrokontroler,

				sehingga arduino dapat mengakses <i>output</i> sensor LDR melalui pin analog.
5	Deva Okky Deltania, Djuniadi, Esa Apriaskar	Pengaturan Lampu Lalu Lintas (Traffic Light) Dengan Sensor Ultrasonik	2021	Sensor ultrasonik yang dipakai memiliki 2 fungsi, yaitu menghitung jarak dan mengubah satuan durasi mikrodetik menjadi centimeter. Fungsi menghitung jarak akan menghitung jarak dengan mengirim gelombang ultrasonik dan ketika dipantulkan kembali, echo akan menerimanya sehingga kita dapat menghitung lamanya pengiriman dan menerima pulsa dengan menggunakan pulsa sesuai perintah. Kemudian

				<p>kita akan mengirimkan durasi itu ke fungsi yang lain yaitu dari microdetik ke centimeter, dan setelah berubah menjadi centimeter nilai tersebut akan diubah lagi menjadi nilai meter. Pada fungsi loop, pertama-tama akan dihitung jarak kedua gelombang dengan menggunakan kedua sensor ultrasonik, dan kemudian akan dibandingkan jaraknya.</p> <p>Jika jarak yang 1 lebih besar daripada jarak yang ke 2, maka kedua led kuning akan menyala selama 200 milidetik, kemudian jika jarak yang ke 1 lebih besar dari jarak</p>
--	--	--	--	---

				<p>yang ke 2, led merah pertama (traffic light yang ke 1) akan menyala dan led hijau kedua (traffic light yang ke 2) akan menyala. Dan kasus yang kedua yaitu jika jarak yang ke 2 lebih besar daripada jarak yang ke 1 maka kedua led kuning akan menyala selama 200 milidetik, kemudian jika jarak yang ke 2 lebih besar dari jarak yang ke 1 maka led merah kedua (traffic light yang ke 2) akan menyala dan led hijau pertama (traffic light yang ke 1) akan menyala.</p>
--	--	--	--	---

II.14 Kerangka Pikir

Berikut ini kerangka pikir dari prototype sistem kontrol lampu lalu lintas berbasis Arduino UNO



Gambar 2. 14 Kerangka pikir

BAB III

METODE PENELITIAN

III.1 Waktu dan Tempat Penelitian

Waktu penelitian dilaksanakan selama 2 bulan, dimulai pada bulan Januari 2022 sampai dengan Februari 2022. Tempat penelitian dilaksanakan di Laboratorium Teknik Elektro, Universitas Fajar.

III.2 Alat dan Bahan Penelitian

Alat dan bahan yang digunakan untuk merancang dan membuat sistem adalah sebagai berikut:

II.1.1 Alat

Alat yang digunakan dalam penelitian ini adalah :

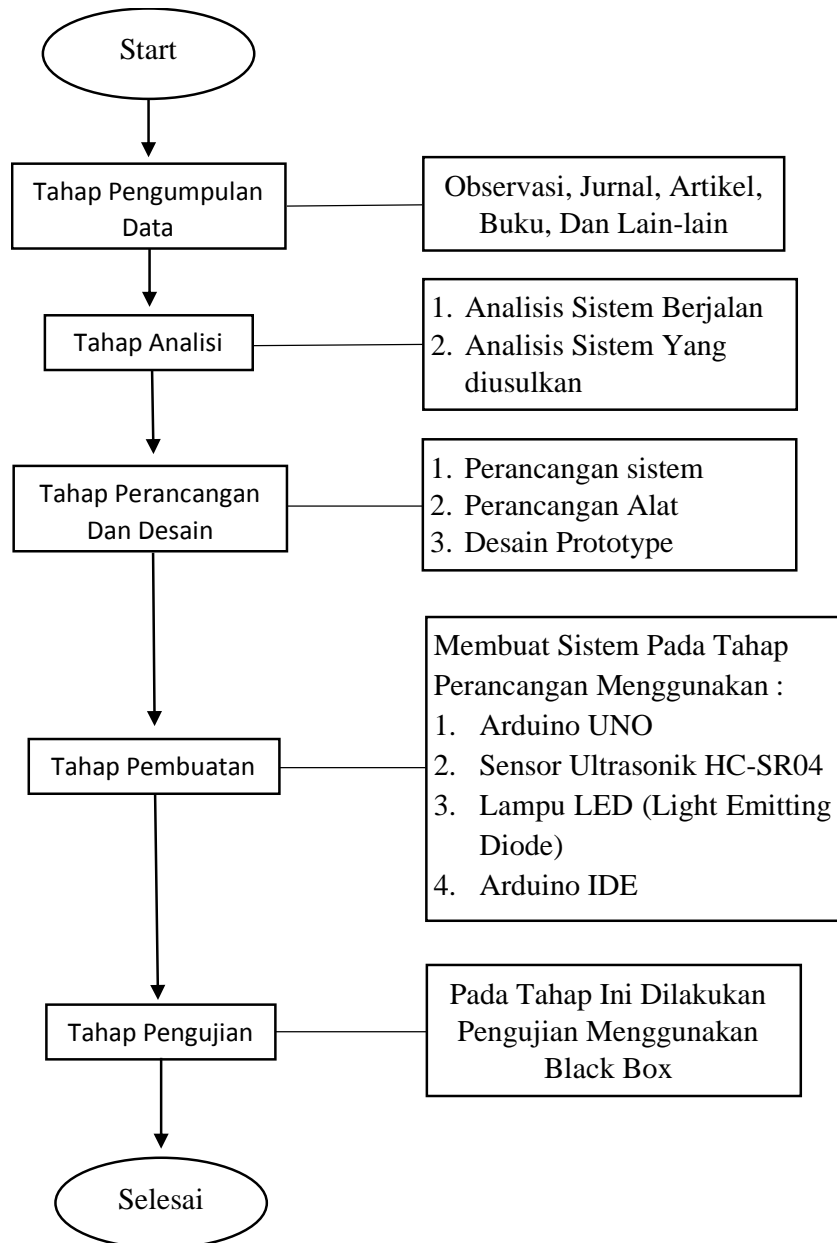
1. Arduino IDE
2. Solder
3. Penghisap Timah

II.1.2 Bahan

1. Arduino UNO
2. Sensor ultrasonic HC-SR04
3. Lampu LED
 - Merah
 - Kuning
 - Hijau
4. Kabel Jumper
5. Timah

III.3 Tahapan Penelitian

Tahapan yang dilakukan dalam penelitian ini adalah sebagai berikut:



Gambar 3. 1 Bagan Prosedur Penelitian
(Sumber : Dokumentasi Pribadi)

III.4 Metode Pengumpulan Data

Pengumpulan data adalah salah satu hal yang penting dilakukan dalam memperoleh data yang diinginkan. Dengan adanya data yang diambil tersebut, sangat membantu dalam menghasilkan informasi yang diinginkan. Pengumpulan data yang dilakukan dengan cara observasi dan studi literatur.

III.4.1 Data Primer

Pengumpulan data primer, yaitu data yang dikumpulkan dan diolah sendiri oleh peneliti langsung dari subjek atau objek penelitian dan dilakukan dengan cara observasi di lokasi penelitian dengan melakukan metode pengamatan langsung dan mengambil data tersebut. Observasi dilakukan di perempatan jalan manggarupi, jalan Sirajuddin rani, dan jalan andi tonro. Observasi yang dilakukan yaitu mengukur luas lebar jalan.

III.4.2 Data Sekunder

Data sekunder, yaitu data yang didapatkan tidak secara langsung dari objek atau penelitian dengan mengolah data dari dokumen-dokumen yang sudah ada sebelumnya baik berasal dari artikel, jurnal, buku, internet dan berbagai sumber lain yang berhubungan dengan masalah yang penulis teliti untuk memperoleh landasan teori yang dapat menunjang penelitian.

III.5 Analisis Data/Analisis Sistem

Metode pengujian yang digunakan pada penelitian ini yaitu metode *Black Box Testing* dan *White Box Testing*. Pengujian *Black-Box* berfokus pada persyaratan fungsional dan suatu sistem.

III.6 Bagan Alur Penelitian

III.6.1 Sistem Berjalan

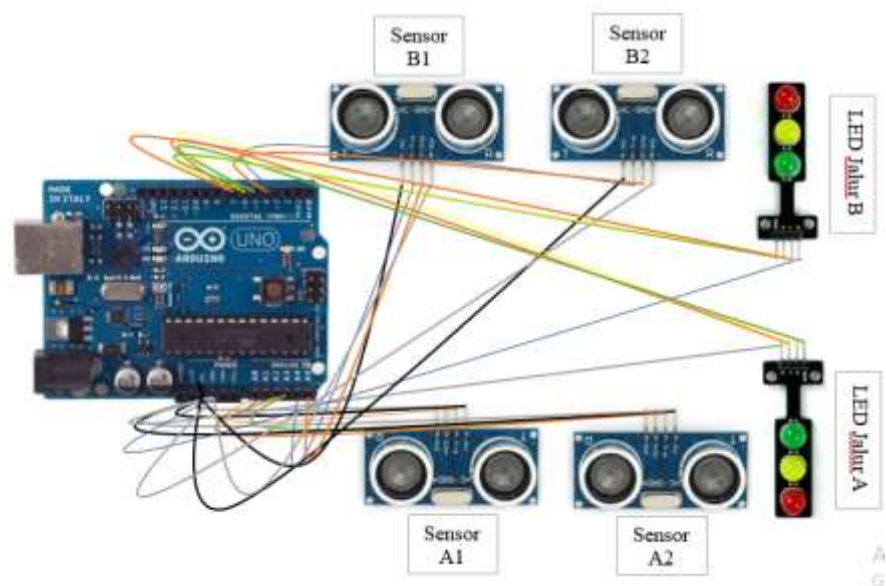
1. pengoperasian lampu lalu lintas yang pemasangannya didasarkan pada suatu tempat persimpangan saja tanpa mempertimbangkan persimpangan lain dan kondisi waktu yang tidak real time.
2. lampu lalu lintas yang pengoperasiannya menggunakan waktu yang tepat dan tidak mengalami perubahan

III.6.2 Rancangan Penelitian Yang Diusulkan

Pengoperasin lampu lalu lintas berdasarkan kondisi masing-masing jalur, sehingga waktu nyala lampu lalu lintas berdasarkan kondisi *real-time*, sesuai dengan jumlah dan panjang antrian kendaraan.

III.6.3 Perancangan Penelitian

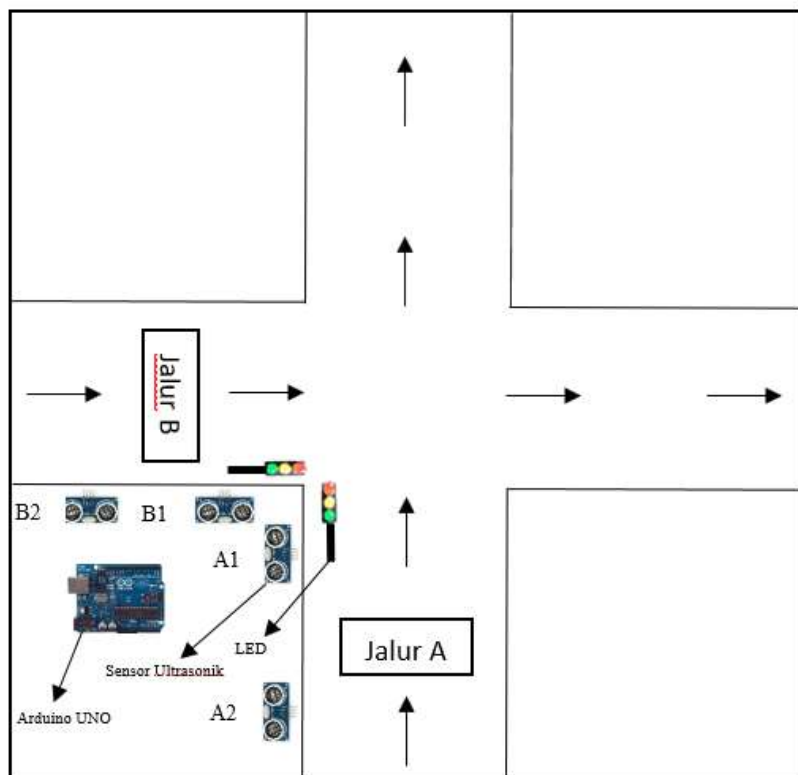
A. Desain Alat



Gambar 3. 2 Desain Alat
(Sumber : Dokumentasi Pribadi)

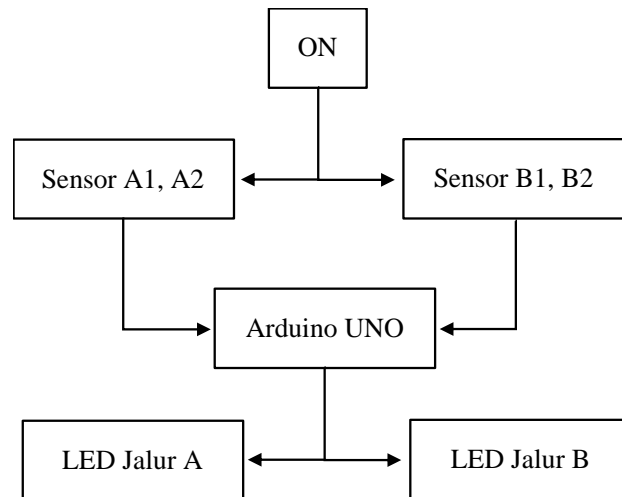
1. Sensor A1 mendeteksi objek (kendaraan) dengan kondisi cukup padat kemudian diproses untuk mendapatkan nyala lampu lalu lintas menggunakan arduino UNO. Output nyala lampu lalu lintas (LED hijau) pada kondisi cukup padat selama 20 detik.
2. Sensor A2 mendeteksi objek (kendaraan) dengan kondisi cukup padat kemudian diproses untuk mendapatkan nyala lampu lalu lintas menggunakan arduino UNO. Output nyala lampu lalu lintas (LED hijau) pada kondisi padat selama 30 detik.
3. LED jalur A sebagai output dari sensor jalur A, dimana jika kondisi cukup padat maka LED hijau menyala selama 20 detik dan pada kondisi padat LED hijau menyala selama 30 detik. Kemudian LED kuning delay selama 2 detik pada masing-masing kondisi. Kemudian LED merah menyala tergantung kondisi jalur B, jika kondisi jalur B cukup padat maka LED merah pada jalur A akan menyala selama 20 detik, dan jika kondisi jalur B padat maka LED merah pada jalur A akan menyala selama 30 detik
4. Sensor B1 mendeteksi objek (kendaraan) dengan kondisi cukup padat kemudian diproses untuk mendapatkan nyala lampu lalu lintas menggunakan arduino UNO. Output nyala lampu lalu lintas (LED hijau) pada kondisi cukup padat selama 20 detik.

5. Sensor B2 mendeteksi objek (kendaraan) dengan kondisi cukup padat kemudian diproses untuk mendapatkan nyala lampu lalu lintas menggunakan arduino UNO. Output nyala lampu lalu lintas (LED hijau) pada kondisi cukup padat selama 30 detik.
6. LED jalur B sebagai output dari sensor jalur B, dimana jika kondisi cukup padat maka LED hijau menyala selama 20



detik dan pada kondisi padat LED hijau menyala selama 30 detik. Kemudian LED kuning delay selama 2 detik pada masing-masing kondisi. Kemudian LED merah menyala tergantung kondisi jalur A, jika kondisi jalur A cukup padat maka LED merah pada jalur B akan menyala selama 20 detik, dan jika kondisi jalur A padat maka LED merah pada jalur B akan menyala selama 30 detik

B. Bagan Kerja Sistem

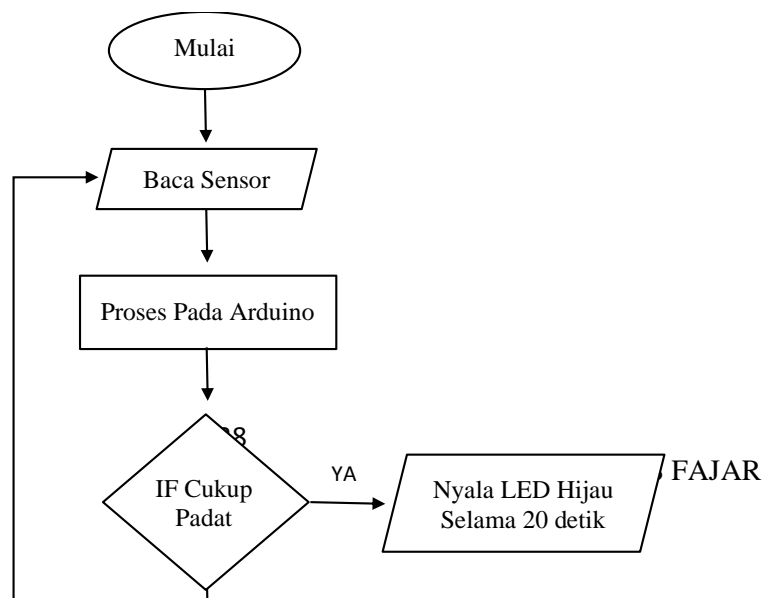


Gambar 3. 3 Rancangan Prototype sistem kontrol lampu lalu lintas berbasis Arduino UNO
(Sumber : Dokumentasi Pribadi)

Penjelasan bagan kerja sistem Gambar 3.4 adalah sebagai berikut :

1. Sensor Ultrasonik jalur A dan B mendeteksi objek.
2. Kemudian arduino memproses objek yang dikirm dari sensor.
3. Hasil dari proses kerja arduino akan menjadi output ke LED jalur A dan jalur B.

C. Flowchart Sistem



Gambar 3.5. *Flow Chart Sistem*
(Sumber : Dokumentasi Pribadi)

Penjelasan flowchart sistem Gambar 3.5 adalah sebagai berikut:

1. Mulai
2. Sensor ultrasonik mendeteksi objek (kendaraan).
3. Arduino memproses data yang dikirim sensor ultrasonik.
4. IF tmelakukan pengambilan keputusan untuk nyala lampu lalu lintas, jika masuk pada kondisi cukup padat maka nyala lampu hijau selama 20 detik.
5. IF tmelakukan pengambilan keputusan untuk nyala lampu lalu lintas, jika masuk pada kondisi padat maka nyala lampu hijau selama 30 detik.
6. Jika tidak masuk pada kondisi cukup padat atau padat, maka nyala LED hijau selama 10 detik.
7. Kembali baca sensor

BAB IV

HASIL DAN PEMBAHASAN

4.1 Hasil

Berdasarkan penelitian yang telah dilakukan, berikut adalah hasil yang diperoleh :

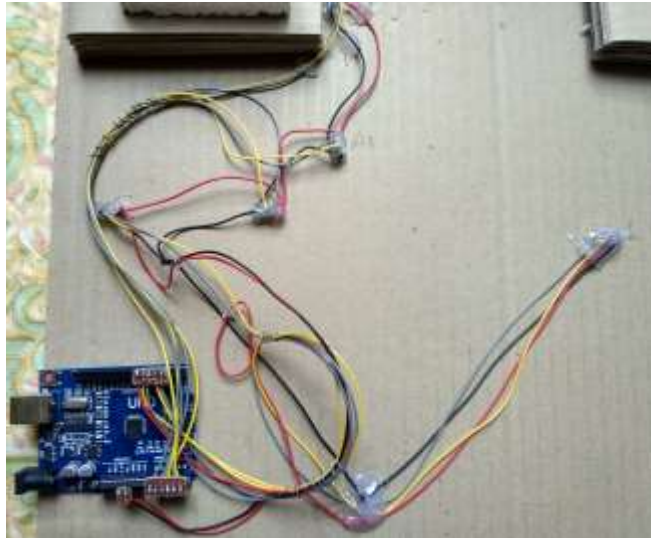
4.1.1 Tampilan Prototype Jalan Raya



Gambar 4.1 Prototype jalan raya
Sumber ; Dokumentasi Pribadi

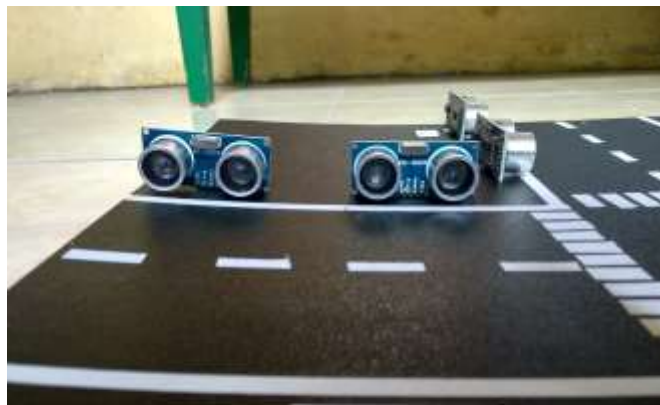
Gambar 4.1 menunjukkan tampilan alat dengan kondisi prototype jalan raya dengan keadaan tidak aktif.

4.1.2 Tampilan Alat



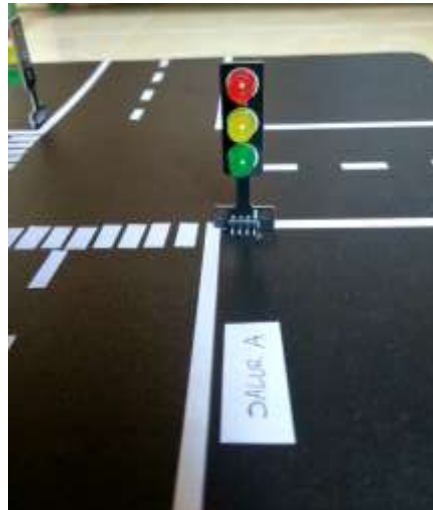
Gambar 4.2 Tampilan alat (Arduino UNO)
Sumber ; Dokumentasi Pribadi

Gambar 4.2 menunjukkan tampilan komponen Arduino pada prototype jalan raya yang berfungsi untuk memproses objek yang diterima dari sensor.



Gambar 4.3 Tampilan alat (Sensor)
Sumber ; Dokumentasi Pribadi

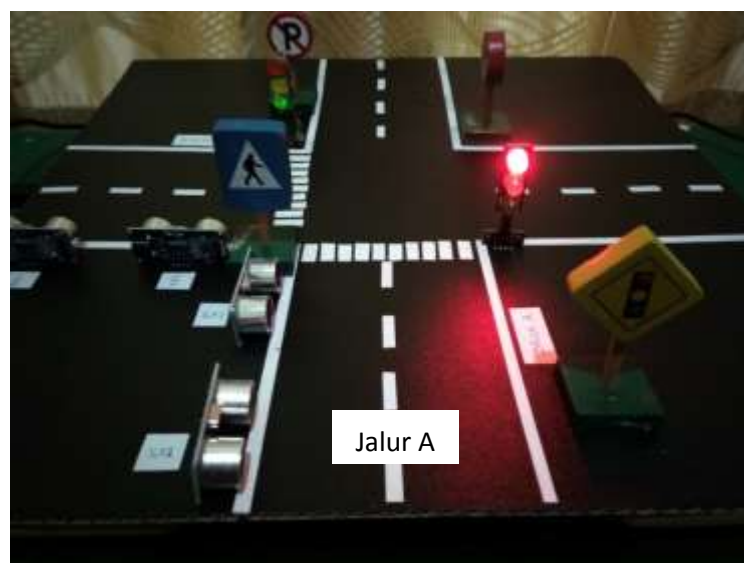
Gambar 4.3 menunjukkan tampilan komponen sensor ultrasonik pada prototype jalan raya yang berfungsi untuk mendeteksi objek.



Gambar 4.4 Tampilan alat (LED)
Sumber : Dokumentasi Pribadi

Gambar 4.4 menunjukkan tampilan komponen LED pada prototype jalan raya yang berfungsi sebagai output untuk nyala lampu lalu lintas.

4.1.3 Tampilan Kondisi 1 Jalur A



Gambar 4.5 Kondisi 1 jalur A Lampu Merah

Sumber : Dokumentasi Pribadi

Gambar 4.5 menunjukkan kondisi 1 jalur A dimana kondisi ini merupakan kondisi awal yang tampil ketika prototype diaktifkan. Pada kondisi ini ada 2 output yaitu lampu hijau dan lampu merah, dan masing-masing memiliki durasi selama 10 detik. Pada kondisi ini, jika pada jalur A lampu merah menyala dengan durasi 10 detik, maka lampu hijau pada jalur B menyala dengan durasi 10 detik. Script dari gambar diatas :

```
else
{
Serial.println("kondisi 1");

digitalWrite(led_R1, HIGH);
digitalWrite(led_Y1, LOW);
digitalWrite(led_G1, LOW);
delay(10000);

digitalWrite(led_R1, LOW);
digitalWrite(led_Y1, HIGH);
digitalWrite(led_G1, LOW);
delay(2000);

digitalWrite(led_R1, LOW);
digitalWrite(led_Y1, LOW);
digitalWrite(led_G1, HIGH);
delay(10000);

digitalWrite(led_R1, LOW);
digitalWrite(led_Y1, HIGH);
digitalWrite(led_G1, LOW);
```

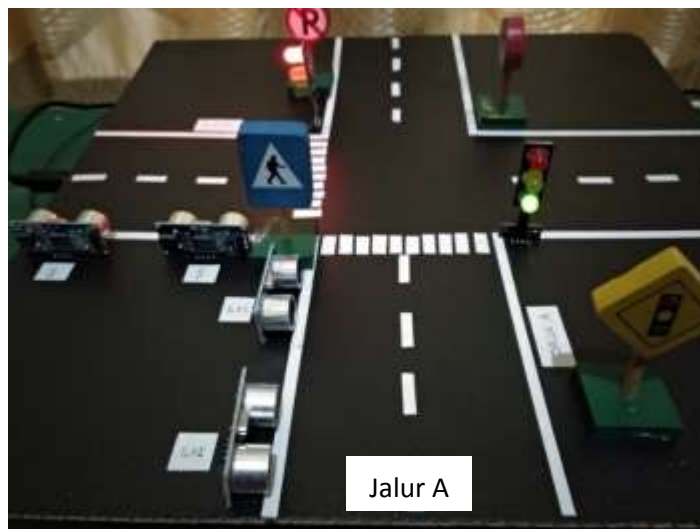
Gambar 4.6 Script Kondisi 1 Jalur A

Sumber ; Dokumentasi Pribadi

Sensor A1 = 86 CM	Sensor A2 = 11 CM	Sensor B1 = 17 CM	Sensor B2 = 52 CM
kondisi 1			
Sensor A1 = 85 CM	Sensor A2 = 11 CM	Sensor B1 = 17 CM	Sensor B2 = 41 CM
kondisi 1			
Sensor A1 = 83 CM	Sensor A2 = 11 CM	Sensor B1 = 17 CM	Sensor B2 = 43 CM
kondisi 1			
Sensor A1 = 85 CM	Sensor A2 = 11 CM	Sensor B1 = 17 CM	Sensor B2 = 40 CM
kondisi 1			
Sensor A1 = 85 CM	Sensor A2 = 11 CM	Sensor B1 = 17 CM	Sensor B2 = 39 CM
kondisi 1			
Sensor A1 = 84 CM	Sensor A2 = 11 CM	Sensor B1 = 17 CM	Sensor B2 = 41 CM
kondisi 1			
Sensor A1 = 85 CM	Sensor A2 = 11 CM	Sensor B1 = 17 CM	Sensor B2 = 41 CM
kondisi 1			
Sensor A1 = 84 CM	Sensor A2 = 11 CM	Sensor B1 = 17 CM	Sensor B2 = 40 CM
kondisi 1			
Sensor A1 = 86 CM	Sensor A2 = 11 CM	Sensor B1 = 17 CM	Sensor B2 = 41 CM
kondisi 1			
Sensor A1 = 85 CM	Sensor A2 = 11 CM	Sensor B1 = 17 CM	Sensor B2 = 40 CM
kondisi 1			

Gambar 4.7 Monitoring jarak objek Kondisi 1 Jalur A

Sumber ; Dokumentasi Pribadi



Gambar 4.8 Kondisi 1 jalur A Lampu

Sumber ; Dokumentasi Pribadi

Gambar 4.8 menunjukkan kondisi 1 jalur A dimana kondisi ini merupakan kondisi awal tampil ketika prototype diaktifkan. Pada kondisi ini ada 2 output yaitu lampu hijau dan lampu merah, dan masing-masing memiliki durasi selama 10 detik. Pada kondisi ini, jika pada jalur A lampu hijau menyala dengan durasi 10 detik, maka lampu

merah pada jalur B menyala dengan durasi 10 detik. Script dari gambar diatas :

```
else
{
Serial.println("kondisi 1");

digitalWrite(led_R1, HIGH);
digitalWrite(led_Y1, LOW);
digitalWrite(led_G1, LOW);
delay(10000);

digitalWrite(led_R1, LOW);
digitalWrite(led_Y1, HIGH);
digitalWrite(led_G1, LOW);
delay(2000);

digitalWrite(led_R1, LOW);
digitalWrite(led_Y1, LOW);
digitalWrite(led_G1, HIGH);
delay(10000);

digitalWrite(led_R1, LOW);
digitalWrite(led_Y1, HIGH);
digitalWrite(led_G1, LOW);
delay(2000);
}
```

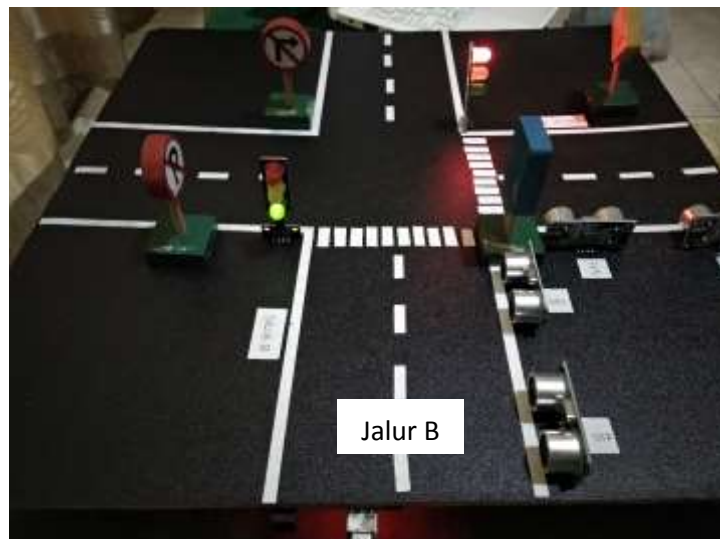
Gambar 4.9 Script Kondisi 1 Jalur A
Sumber ; Dokumentasi Pribadi

Sensor A1 = 86 CM	Sensor A2 = 11 CM	Sensor B1 = 17 CM	Sensor B2 = 52 CM
kondisi 1			
Sensor A1 = 85 CM	Sensor A2 = 11 CM	Sensor B1 = 17 CM	Sensor B2 = 41 CM
kondisi 1			
Sensor A1 = 83 CM	Sensor A2 = 11 CM	Sensor B1 = 17 CM	Sensor B2 = 43 CM
kondisi 1			
Sensor A1 = 85 CM	Sensor A2 = 11 CM	Sensor B1 = 17 CM	Sensor B2 = 40 CM
kondisi 1			
Sensor A1 = 85 CM	Sensor A2 = 11 CM	Sensor B1 = 17 CM	Sensor B2 = 39 CM
kondisi 1			
Sensor A1 = 84 CM	Sensor A2 = 11 CM	Sensor B1 = 17 CM	Sensor B2 = 41 CM
kondisi 1			
Sensor A1 = 85 CM	Sensor A2 = 11 CM	Sensor B1 = 17 CM	Sensor B2 = 41 CM
kondisi 1			
Sensor A1 = 84 CM	Sensor A2 = 11 CM	Sensor B1 = 17 CM	Sensor B2 = 40 CM
kondisi 1			
Sensor A1 = 86 CM	Sensor A2 = 11 CM	Sensor B1 = 17 CM	Sensor B2 = 41 CM
kondisi 1			
Sensor A1 = 85 CM	Sensor A2 = 11 CM	Sensor B1 = 17 CM	Sensor B2 = 40 CM
kondisi 1			

Gambar 4.10 Monitoring jarak objek Kondisi 1 Jalur A

Sumber ; Dokumentasi Pribadi

4.1.4 Tampilan Kondisi 1 Jalur B



Gambar 4.11 Kondisi 1 Jalur B

Sumber ; Dokumentasi Pribadi

Gambar 4.11 menunjukkan kondisi 1 jalur B dimana kondisi ini merupakan kondisi awal yang tampil ketika prototype diaktifkan. Pada kondisi ini ada 2 output yaitu lampu hijau dan lampu merah, dan masing-masing memiliki durasi selama 10 detik. Pada kondisi ini, jika pada jalur B lampu hijau menyala dengan durasi 10 detik, maka lampu merah pada jalur A menyala dengan durasi 10 detik. Script dari gambar diatas :

```
else
{
Serial.println("kondisi 1");

digitalWrite(led_R1, HIGH);
digitalWrite(led_Y1, LOW);
digitalWrite(led_G1, LOW);
delay(10000);

digitalWrite(led_R1, LOW);
digitalWrite(led_Y1, HIGH);
digitalWrite(led_G1, LOW);
delay(2000);

digitalWrite(led_R1, LOW);
digitalWrite(led_Y1, LOW);
digitalWrite(led_G1, HIGH);
delay(10000);

digitalWrite(led_R1, LOW);
digitalWrite(led_Y1, HIGH);
digitalWrite(led_G1, LOW);
delay(2000);
}
```

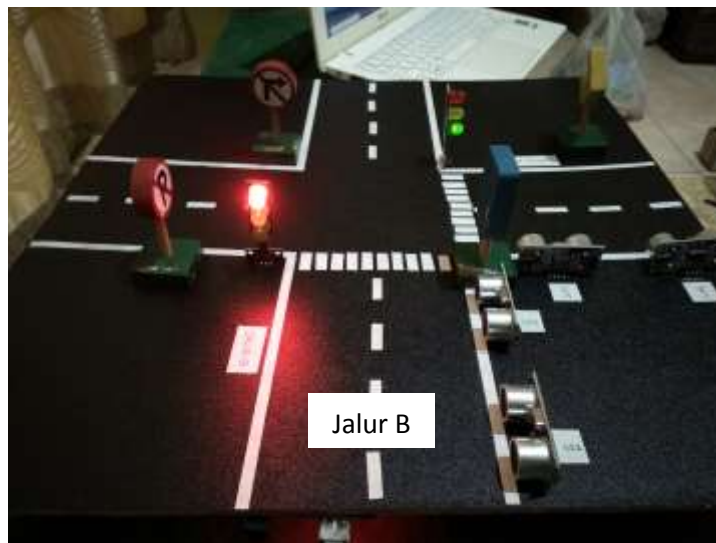
Gambar 4.12 Script Kondisi 1 Jalur B

Sumber^{4,7}, Dokumentasi Pribadi

Sensor A1 = 86 CM	Sensor A2 = 11 CM	Sensor B1 = 17 CM	Sensor B2 = 52 CM
kondisi 1			
Sensor A1 = 85 CM	Sensor A2 = 11 CM	Sensor B1 = 17 CM	Sensor B2 = 41 CM
kondisi 1			
Sensor A1 = 83 CM	Sensor A2 = 11 CM	Sensor B1 = 17 CM	Sensor B2 = 43 CM
kondisi 1			
Sensor A1 = 85 CM	Sensor A2 = 11 CM	Sensor B1 = 17 CM	Sensor B2 = 40 CM
kondisi 1			
Sensor A1 = 85 CM	Sensor A2 = 11 CM	Sensor B1 = 17 CM	Sensor B2 = 39 CM
kondisi 1			
Sensor A1 = 84 CM	Sensor A2 = 11 CM	Sensor B1 = 17 CM	Sensor B2 = 41 CM
kondisi 1			
Sensor A1 = 85 CM	Sensor A2 = 11 CM	Sensor B1 = 17 CM	Sensor B2 = 41 CM
kondisi 1			
Sensor A1 = 84 CM	Sensor A2 = 11 CM	Sensor B1 = 17 CM	Sensor B2 = 40 CM
kondisi 1			
Sensor A1 = 86 CM	Sensor A2 = 11 CM	Sensor B1 = 17 CM	Sensor B2 = 41 CM
kondisi 1			
Sensor A1 = 85 CM	Sensor A2 = 11 CM	Sensor B1 = 17 CM	Sensor B2 = 40 CM
kondisi 1			

Gambar 4.13 Monitoring jarak objek Kondisi 1 Jalur B

Sumber ; Dokumentasi Pribadi



Gambar 4.14 Kondisi 1 Jalur B

Sumber ; Dokumentasi Pribadi

Gambar 4.14 menunjukkan kondisi 1 jalur B dimana kondisi ini merupakan kondisi awal yang tampil ketika prototype diaktifkan. Pada kondisi ini ada 2 output yaitu lampu hijau dan lampu merah, dan masing-masing memiliki durasi selama 10 detik. Pada kondisi ini, jika

pada jalur B lampu merah menyala dengan durasi 10 detik, maka lampu hijau pada jalur A menyala dengan durasi 10 detik. Script dari gambar diatas :

```
else
{
Serial.println("kondisi 1");

digitalWrite(led_R1, HIGH);
digitalWrite(led_Y1, LOW);
digitalWrite(led_G1, LOW);
delay(10000);

digitalWrite(led_R1, LOW);
digitalWrite(led_Y1, HIGH);
digitalWrite(led_G1, LOW);
delay(2000);

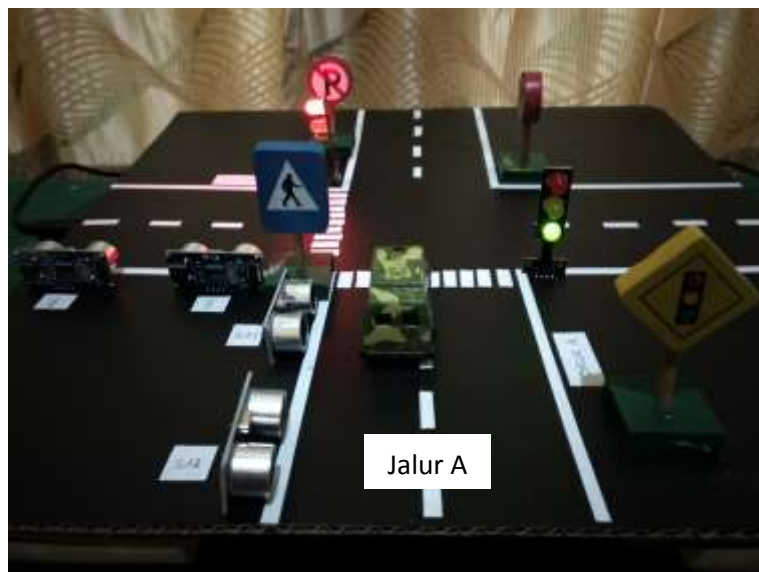
digitalWrite(led_R1, LOW);
digitalWrite(led_Y1, LOW);
digitalWrite(led_G1, HIGH);
delay(10000);

digitalWrite(led_R1, LOW);
digitalWrite(led_Y1, HIGH);
digitalWrite(led_G1, LOW);
delay(2000);
}
```

Sensor A1 = 86 CM	Sensor A2 = 11 CM	Sensor B1 = 17 CM	Sensor B2 = 52 CM
kondisi 1			
Sensor A1 = 85 CM	Sensor A2 = 11 CM	Sensor B1 = 17 CM	Sensor B2 = 41 CM
kondisi 1			
Sensor A1 = 83 CM	Sensor A2 = 11 CM	Sensor B1 = 17 CM	Sensor B2 = 43 CM
kondisi 1			
Sensor A1 = 85 CM	Sensor A2 = 11 CM	Sensor B1 = 17 CM	Sensor B2 = 40 CM
kondisi 1			
Sensor A1 = 85 CM	Sensor A2 = 11 CM	Sensor B1 = 17 CM	Sensor B2 = 39 CM
kondisi 1			
Sensor A1 = 84 CM	Sensor A2 = 11 CM	Sensor B1 = 17 CM	Sensor B2 = 41 CM
kondisi 1			
Sensor A1 = 85 CM	Sensor A2 = 11 CM	Sensor B1 = 17 CM	Sensor B2 = 41 CM
kondisi 1			
Sensor A1 = 84 CM	Sensor A2 = 11 CM	Sensor B1 = 17 CM	Sensor B2 = 40 CM
kondisi 1			
Sensor A1 = 86 CM	Sensor A2 = 11 CM	Sensor B1 = 17 CM	Sensor B2 = 41 CM
kondisi 1			
Sensor A1 = 85 CM	Sensor A2 = 11 CM	Sensor B1 = 17 CM	Sensor B2 = 40 CM
kondisi 1			

Gambar 4.16 Monitoring jarak objek Kondisi 1 Jalur B
Sumber ; Dokumentasi Pribadi

4.1.5 Tampilan Kondisi 2 Jalur A



Gambar 4.17 Kondisi 2 Jalur A
Sumber ; Dokumentasi Pribadi

Gambar 4.17 menunjukkan kondisi 2 jalur A, Pada kondisi ini Jalur A masuk kategori cukup padat, dan durasi lampu hijau pada jalur A selama 20 detik. Script dari gambar diatas :

```
if (distance1 <= 10 && distance2 >= 10 && distance3 >= 10 && distance4 >=
10) // SENSOR A1
{
Serial.println("kondisi 2");

digitalWrite(led_R1, HIGH);
digitalWrite(led_Y1, LOW);
digitalWrite(led_G1, LOW);
delay(10000); //

digitalWrite(led_R1, LOW);
digitalWrite(led_Y1, HIGH);
digitalWrite(led_G1, LOW);
delay(2000);

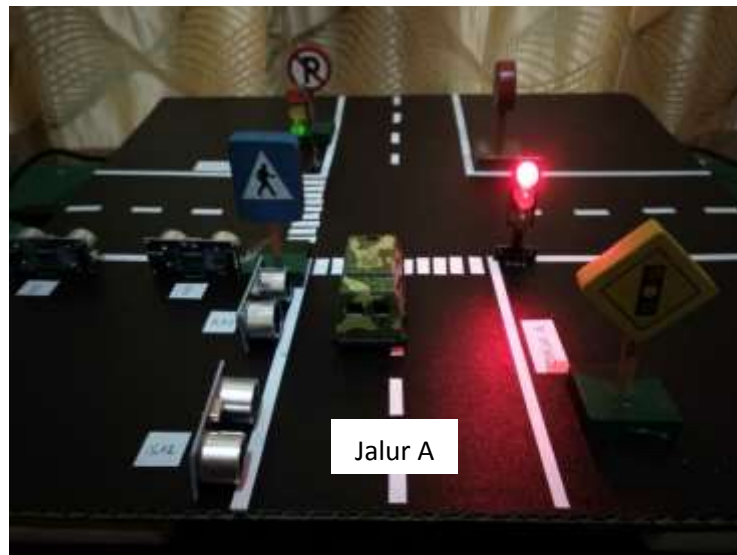
digitalWrite(led_R1, LOW);
digitalWrite(led_Y1, LOW);
digitalWrite(led_G1, HIGH);
delay(20000); // 20 DETIK

digitalWrite(led_R1, LOW);
digitalWrite(led_Y1, HIGH);
digitalWrite(led_G1, LOW);
delay(2000);
}
```

Gambar 4.18 Script Kondisi 2 Jalur A
Sumber ; Dokumentasi Pribadi

kondisi 2
 Sensor A1 = 2 CM Sensor A2 = 12 CM Sensor B1 = 17 CM Sensor B2 = 41 CM
 kondisi 2
 Sensor A1 = 2 CM Sensor A2 = 12 CM Sensor B1 = 17 CM Sensor B2 = 40 CM
 kondisi 2
 Sensor A1 = 2 CM Sensor A2 = 12 CM Sensor B1 = 17 CM Sensor B2 = 44 CM
 kondisi 2
 Sensor A1 = 2 CM Sensor A2 = 12 CM Sensor B1 = 17 CM Sensor B2 = 40 CM
 kondisi 2
 Sensor A1 = 2 CM Sensor A2 = 12 CM Sensor B1 = 17 CM Sensor B2 = 40 CM
 kondisi 2

Gambar 4.19 Monitoring jarak objek Kondisi 2 Jalur A
 Sumber ; Dokumentasi Pribadi



Gambar 4.20 Kondisi 2 Jalur A
 Sumber ; Dokumentasi Pribadi

Gambar 4.20 menunjukkan kondisi 2 jalur A, Pada kondisi ini Jalur A masuk kategori cukup padat, dan durasi lampu merah pada jalur A selama 10 detik. Script dari gambar diatas :

```

if (distance1 <= 10 && distance2 >= 10 && distance3 >= 10 && distance4 >=
10) // SENSOR A1
{
Serial.println("kondisi 2");

digitalWrite(led_R1, HIGH);
digitalWrite(led_Y1, LOW);
digitalWrite(led_G1, LOW);
delay(10000); //

digitalWrite(led_R1, LOW);
digitalWrite(led_Y1, HIGH);
digitalWrite(led_G1, LOW);
delay(2000);

digitalWrite(led_R1, LOW);
digitalWrite(led_Y1, LOW);
digitalWrite(led_G1, HIGH);
delay(20000); // 20 DETIK

digitalWrite(led_R1, LOW);
digitalWrite(led_Y1, HIGH);
digitalWrite(led_G1, LOW);
delay(2000);
}

```

Gambar 4.21 Script Kondisi 2 Jalur A
Sumber ; Dokumentasi Pribadi

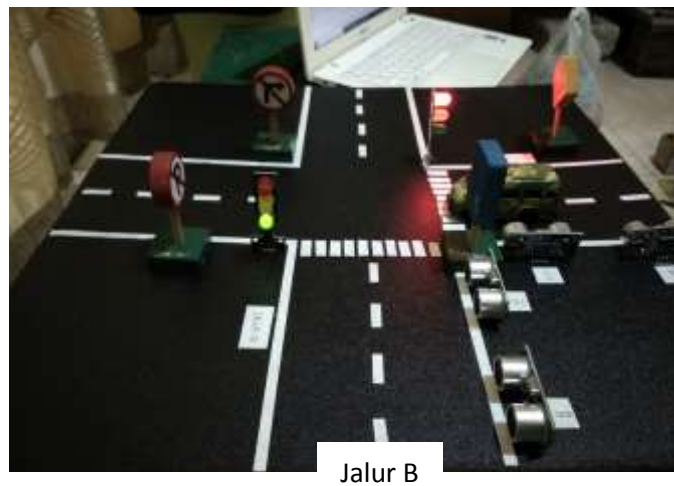
```

kondisi 2
Sensor A1 = 2 CM   Sensor A2 = 12 CM   Sensor B1 = 17 CM   Sensor B2 = 41 CM
kondisi 2
Sensor A1 = 2 CM   Sensor A2 = 12 CM   Sensor B1 = 17 CM   Sensor B2 = 40 CM
kondisi 2
Sensor A1 = 2 CM   Sensor A2 = 12 CM   Sensor B1 = 17 CM   Sensor B2 = 44 CM
kondisi 2
Sensor A1 = 2 CM   Sensor A2 = 12 CM   Sensor B1 = 17 CM   Sensor B2 = 40 CM
kondisi 2
Sensor A1 = 2 CM   Sensor A2 = 12 CM   Sensor B1 = 17 CM   Sensor B2 = 40 CM
kondisi 2

```

Gambar 4.22 Monitoring jarak objek Kondisi 2 Jalur A
Sumber ; Dokumentasi Pribadi

4.1.6 Tampilan Kondisi 2 Jalur B



Gambar 4.23 Kondisi 2 Jalur B

Gambar 4.23 menunjukkan Kondisi 2 Jalur B, Pada kondisi ini Jalur B tidak masuk dalam kondisi cukup padat maupun padat, dan durasi lampu hijau pada jalur B selama 10 detik. Script dari gambar diatas:

```
if (distance1 <= 10 && distance2 >= 10 && distance3 >= 10 && distance4 >=
10) // SENSOR A1
{
Serial.println("kondisi 2");

digitalWrite(led_R1, HIGH);
digitalWrite(led_Y1, LOW);
digitalWrite(led_G1, LOW);
delay(10000); //

digitalWrite(led_R1, LOW);
digitalWrite(led_Y1, HIGH);
```



```

digitalWrite(led_G1, LOW);
delay(2000);
digitalWrite(led_R1, LOW);
digitalWrite(led_Y1, LOW);
digitalWrite(led_G1, HIGH);
delay(20000); // 20 DETIK

digitalWrite(led_R1, LOW);
digitalWrite(led_Y1, HIGH);
digitalWrite(led_G1, LOW);
delay(2000);
}

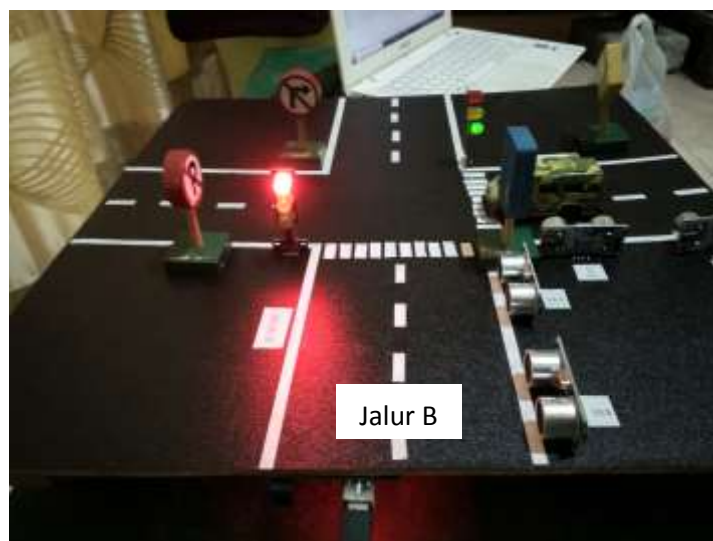
```

```

kondisi 2
Sensor A1 = 2 CM   Sensor A2 = 12 CM   Sensor B1 = 17 CM   Sensor B2 = 41 CM
kondisi 2
Sensor A1 = 2 CM   Sensor A2 = 12 CM   Sensor B1 = 17 CM   Sensor B2 = 40 CM
kondisi 2
Sensor A1 = 2 CM   Sensor A2 = 12 CM   Sensor B1 = 17 CM   Sensor B2 = 44 CM
kondisi 2
Sensor A1 = 2 CM   Sensor A2 = 12 CM   Sensor B1 = 17 CM   Sensor B2 = 40 CM
kondisi 2
Sensor A1 = 2 CM   Sensor A2 = 12 CM   Sensor B1 = 17 CM   Sensor B2 = 40 CM
kondisi 2

```

Gambar 4.25 Monitoring jarak objek Kondisi 2 Jalur B
 Sumber ; Dokumentasi Pribadi



Gambar 4.26 Kondisi 2 Jalur B
 Sumber ; Dokumentasi Pribadi

Gambar 4.26 menunjukkan kondisi 2 jalur B, Pada kondisi ini Jalur B tidak masuk dalam kondisi cukup padat maupun padat, dan durasi lampu merah pada jalur B selama 20 detik. Script dari gambar diatas:

```

if (distance1 <= 10 && distance2 >= 10 && distance3 >= 10 && distance4 >= 10)
// SENSOR A1
{
Serial.println("kondisi 2");

digitalWrite(led_R1, HIGH);
digitalWrite(led_Y1, LOW);
digitalWrite(led_G1, LOW);
delay(10000); //

digitalWrite(led_R1, LOW);
digitalWrite(led_Y1, HIGH);
digitalWrite(led_G1, LOW);
delay(2000);

digitalWrite(led_R1, LOW);
digitalWrite(led_Y1, LOW);
digitalWrite(led_G1, HIGH);
delay(20000); // 20 DETIK

digitalWrite(led_R1, LOW);
digitalWrite(led_Y1, HIGH);
digitalWrite(led_G1, LOW);
delay(2000);
}

```

Gambar 4.27 Script Kondisi 2 Jalur B
Sumber : Dokumentasi Pribadi

```

kondisi 2
Sensor A1 = 2 CM   Sensor A2 = 12 CM   Sensor B1 = 17 CM   Sensor B2 = 41 CM
kondisi 2
Sensor A1 = 2 CM   Sensor A2 = 12 CM   Sensor B1 = 17 CM   Sensor B2 = 40 CM
kondisi 2
Sensor A1 = 2 CM   Sensor A2 = 12 CM   Sensor B1 = 17 CM   Sensor B2 = 44 CM
kondisi 2
Sensor A1 = 2 CM   Sensor A2 = 12 CM   Sensor B1 = 17 CM   Sensor B2 = 40 CM
kondisi 2
Sensor A1 = 2 CM   Sensor A2 = 12 CM   Sensor B1 = 17 CM   Sensor B2 = 40 CM
kondisi 2
Sensor A1 = 2 CM   Sensor A2 = 12 CM   Sensor B1 = 17 CM   Sensor B2 = 40 CM

```

Gambar 4.28 Monitoring jarak objek Kondisi 2 Jalur B
Sumber ; Dokumentasi Pribadi

4.1.7 Tampilan Kondisi 3 Jalur A



Gambar 4.29 Kondisi 3 Jalur A
Sumber ; Dokumentasi Pribadi

Gambar 4.29 menunjukkan kondisi 3 jalur A, Pada kondisi ini Jalur A masuk kategori padat, dan durasi lampu hijau pada jalur A selama 30 detik. Script dari gambar diatas :

```

else if (distance1 <= 10 && distance2 <= 10 && distance3 >= 10 && distance4
>= 10) // SENSOR A1 & A2
{
Serial.println("kondisi 3");

digitalWrite(led_R1, HIGH);
digitalWrite(led_Y1, LOW);
digitalWrite(led_G1, LOW);
delay(10000);

digitalWrite(led_R1, LOW);
digitalWrite(led_Y1, HIGH);
digitalWrite(led_G1, LOW);
delay(2000);

```

```

digitalWrite(led_R1, LOW);
digitalWrite(led_Y1, LOW);
digitalWrite(led_G1, HIGH);
delay(30000);

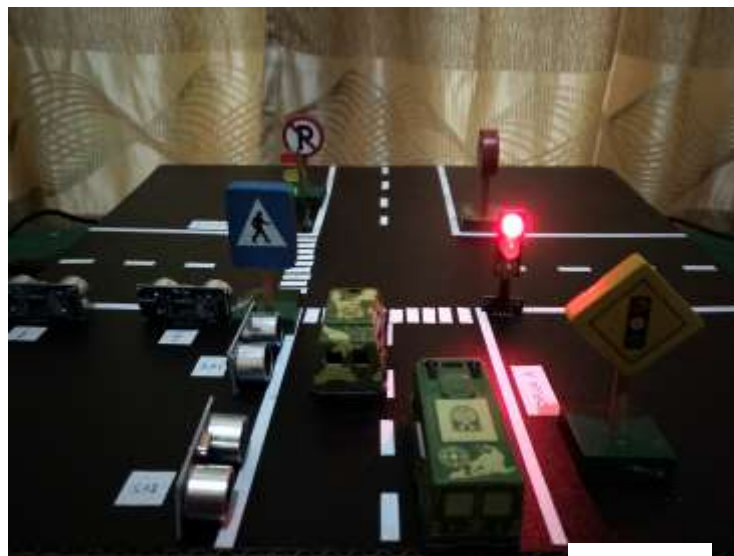
digitalWrite(led_R1, LOW);
digitalWrite(led_Y1, HIGH);
digitalWrite(led_G1, LOW);
delay(2000);
}

```

Gambar 4.30 Script Kondisi 3 Jalur A
Sumber ; Dokumentasi Pribadi

Sensor A1 = 2 CM	Sensor A2 = 6 CM	Sensor B1 = 17 CM	Sensor B2 = 40 CM
kondisi 3			
Sensor A1 = 2 CM	Sensor A2 = 6 CM	Sensor B1 = 17 CM	Sensor B2 = 40 CM
kondisi 3			
Sensor A1 = 2 CM	Sensor A2 = 6 CM	Sensor B1 = 17 CM	Sensor B2 = 41 CM
kondisi 3			
Sensor A1 = 2 CM	Sensor A2 = 6 CM	Sensor B1 = 17 CM	Sensor B2 = 40 CM
kondisi 3			
Sensor A1 = 6 CM	Sensor A2 = 3 CM	Sensor B1 = 17 CM	Sensor B2 = 40 CM
kondisi 3			

Gambar 4.31 Monitoring jarak objek Kondisi 3 Jalur A
Sumber ; Dokumentasi Pribadi



Jalur A

Gambar 4.32 Kondisi 3 Jalur A
Sumber ; Dokumentasi Pribadi

Gambar 4.32 menunjukkan kondisi 3 jalur A, Pada kondisi ini Jalur A masuk kategori padat, dan durasi lampu merah pada jalur A selama 10 detik. Script dari gambar diatas :

```
else if (distance1 <= 10 && distance2 <= 10 && distance3 >= 10 && distance4  
>= 10) // SENSOR A1 & A2  
{  
  Serial.println("kondisi 3");  
  
  digitalWrite(led_R1, HIGH);
```

```

digitalWrite(led_Y1, LOW);
digitalWrite(led_G1, LOW);
delay(10000);

digitalWrite(led_R1, LOW);
digitalWrite(led_Y1, HIGH);
digitalWrite(led_G1, LOW);
delay(2000);

digitalWrite(led_R1, LOW);
digitalWrite(led_Y1, LOW);
digitalWrite(led_G1, HIGH);
delay(30000);

digitalWrite(led_R1, LOW);
digitalWrite(led_Y1, HIGH);
digitalWrite(led_G1, LOW);
delay(2000);
}

```

Gambar 4.33 Script Kondisi 3 Jalur A
Sumber ; Dokumentasi Pribadi

Sensor A1 = 2 CM	Sensor A2 = 6 CM	Sensor B1 = 17 CM	Sensor B2 = 40 CM
kondisi 3			
Sensor A1 = 2 CM	Sensor A2 = 6 CM	Sensor B1 = 17 CM	Sensor B2 = 40 CM
kondisi 3			
Sensor A1 = 2 CM	Sensor A2 = 6 CM	Sensor B1 = 17 CM	Sensor B2 = 41 CM
kondisi 3			
Sensor A1 = 2 CM	Sensor A2 = 6 CM	Sensor B1 = 17 CM	Sensor B2 = 40 CM
kondisi 3			
Sensor A1 = 6 CM	Sensor A2 = 3 CM	Sensor B1 = 17 CM	Sensor B2 = 40 CM
kondisi 3			

Gambar 4.34 Monitoring jarak objek Kondisi 3 Jalur A
Sumber ; Dokumentasi Pribadi

4.1.8 Tampilan Kondisi 3 Jalur B



Jalur B

Gambar 4.35 Kondisi 3 Jalur B
Sumber ; Dokumentasi Pribadi

Gambar 4.35 menunjukkan kondisi 3 jalur B, Pada kondisi ini Jalur B tidak masuk dalam kondisi cukup padat maupun padat, dan durasi lampu hijau pada jalur B selama 10 detik. Script dari gambar diatas:

```
else if (distance1 <= 10 && distance2 <= 10 && distance3 >= 10 && distance4  
>= 10) // SENSOR A1 & A2  
{  
  Serial.println("kondisi 3");  
  
  digitalWrite(led_R1, HIGH);  
  digitalWrite(led_Y1, LOW);  
  digitalWrite(led_G1, LOW);  
  delay(10000);  
  
  digitalWrite(led_R1, LOW);  
  digitalWrite(led_Y1, HIGH);  
  digitalWrite(led_G1, LOW);  
  delay(2000);  
  
  digitalWrite(led_R1, LOW);  
  digitalWrite(led_Y1, LOW);
```

```

digitalWrite(led_G1, HIGH);
delay(30000);

digitalWrite(led_R1, LOW);
digitalWrite(led_Y1, HIGH);
digitalWrite(led_G1, LOW);
delay(2000);
}

```

Gambar 4.36 Script Kondisi 3 Jalur B
 Sumber ; Dokumentasi Pribadi

Sensor A1 = 2 CM	Sensor A2 = 6 CM	Sensor B1 = 17 CM	Sensor B2 = 40 CM
kondisi 3			
Sensor A1 = 2 CM	Sensor A2 = 6 CM	Sensor B1 = 17 CM	Sensor B2 = 40 CM
kondisi 3			
Sensor A1 = 2 CM	Sensor A2 = 6 CM	Sensor B1 = 17 CM	Sensor B2 = 41 CM
kondisi 3			
Sensor A1 = 2 CM	Sensor A2 = 6 CM	Sensor B1 = 17 CM	Sensor B2 = 40 CM
kondisi 3			
Sensor A1 = 6 CM	Sensor A2 = 3 CM	Sensor B1 = 17 CM	Sensor B2 = 40 CM
kondisi 3			

Gambar 4.37 Monitoring jarak objek Kondisi 3 Jalur B
 Sumber ; Dokumentasi Pribadi



Jalur B

Gambar 4.38 Kondisi 3 Jalur B
Sumber ; Dokumentasi Pribadi

Gambar 4.38 menunjukkan kondisi 3 jalur B, Pada kondisi ini Jalur B tidak masuk dalam kondisi cukup padat maupun padat, dan durasi lampu merah pada jalur B selama 30 detik. Script dari gambar diatas:

```
else if (distance1 <= 10 && distance2 <= 10 && distance3 >= 10 && distance4 >=
10) // SENSOR A1 & A2
{
Serial.println("kondisi 3");

digitalWrite(led_R1, HIGH);
digitalWrite(led_Y1, LOW);
digitalWrite(led_G1, LOW);
delay(10000);

digitalWrite(led_R1, LOW);
digitalWrite(led_Y1, HIGH);
digitalWrite(led_G1, LOW);
delay(2000);

digitalWrite(led_R1, LOW);
digitalWrite(led_Y1, LOW);
digitalWrite(led_G1, HIGH);
delay(30000);
```

```

digitalWrite(led_R1, LOW);
digitalWrite(led_Y1, HIGH);
digitalWrite(led_G1, LOW);
delay(2000);
}

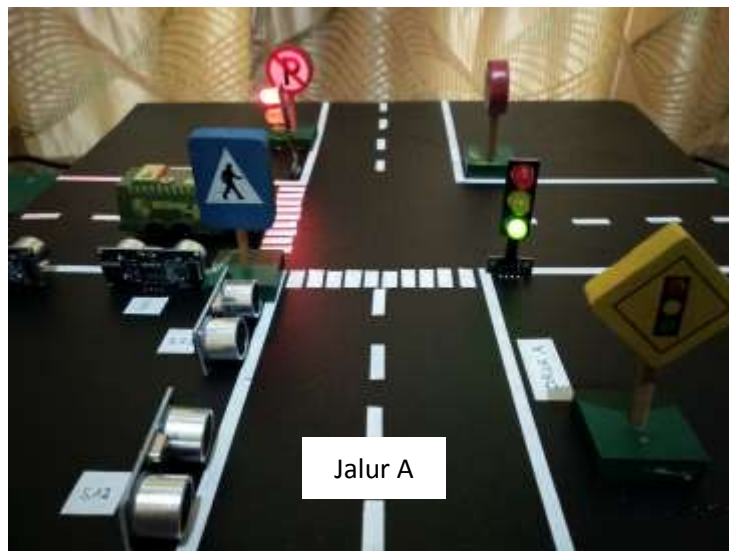
```

Gambar 4.39 Script Kondisi 3 Jalur B
 Sumber ; Dokumentasi Pribadi

Sensor A1 = 2 CM	Sensor A2 = 6 CM	Sensor B1 = 17 CM	Sensor B2 = 40 CM
kondisi 3			
Sensor A1 = 2 CM	Sensor A2 = 6 CM	Sensor B1 = 17 CM	Sensor B2 = 40 CM
kondisi 3			
Sensor A1 = 2 CM	Sensor A2 = 6 CM	Sensor B1 = 17 CM	Sensor B2 = 41 CM
kondisi 3			
Sensor A1 = 2 CM	Sensor A2 = 6 CM	Sensor B1 = 17 CM	Sensor B2 = 40 CM
kondisi 3			
Sensor A1 = 6 CM	Sensor A2 = 3 CM	Sensor B1 = 17 CM	Sensor B2 = 40 CM
kondisi 3			

Gambar 4.40 Monitoring jarak objek Kondisi 3 Jalur B
 Sumber ; Dokumentasi Pribadi

4.1.9 Tampilan Kondisi 4 Jalur A



Gambar 4.41 Kondisi 4 Jalur A
 Sumber ; Dokumentasi Pribadi

Gambar 4.41 menunjukkan kondisi 4 jalur A, Pada kondisi ini Jalur A tidak masuk dalam kondisi cukup padat maupun padat, dan durasi lampu hijau pada jalur A selama 10 detik. Script dari gambar diatas:

```
else if (distance1 >= 10 && distance2 >= 10 && distance3 <= 10 && distance4
>= 10) // SENSOR B1
{
Serial.println("kondisi 4");

digitalWrite(led_R1, HIGH);
digitalWrite(led_Y1, LOW);
digitalWrite(led_G1, LOW);
delay(20000); // 10 DETIK

digitalWrite(led_R1, LOW);
digitalWrite(led_Y1, HIGH);
digitalWrite(led_G1, LOW);
delay(2000);

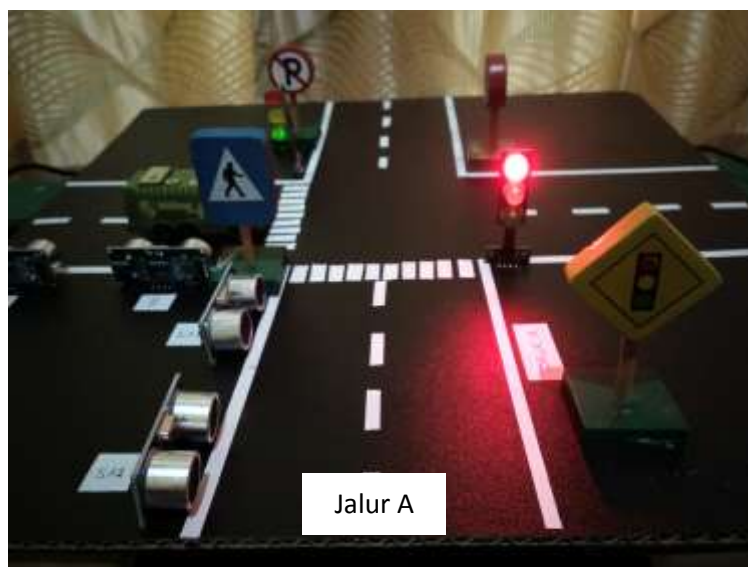
digitalWrite(led_R1, LOW);
digitalWrite(led_Y1, LOW);
digitalWrite(led_G1, HIGH);
delay(10000); // 20 DETIK

digitalWrite(led_R1, LOW);
digitalWrite(led_Y1, HIGH);
digitalWrite(led_G1, LOW);
delay(2000);
}
```

Gambar 4.42 Script Kondisi 4 Jalur A
Sumber ; Dokumentasi Pribadi

Sensor A1 = 84 CM	Sensor A2 = 11 CM	Sensor B1 = 2 CM	Sensor B2 = 41 CM
kondisi 4			
Sensor A1 = 84 CM	Sensor A2 = 11 CM	Sensor B1 = 7 CM	Sensor B2 = 41 CM
kondisi 4			
Sensor A1 = 83 CM	Sensor A2 = 11 CM	Sensor B1 = 6 CM	Sensor B2 = 40 CM
kondisi 4			
Sensor A1 = 85 CM	Sensor A2 = 11 CM	Sensor B1 = 5 CM	Sensor B2 = 40 CM
kondisi 4			

Gambar 4.43 Monitoring jarak objek Kondisi 4 Jalur A
 Sumber ; Dokumentasi Pribadi



Gambar 4.44 Kondisi 4 Jalur A
 Sumber ; Dokumentasi Pribadi

Gambar 4.44 menunjukkan kondisi 4 jalur A, Pada kondisi ini Jalur A tidak masuk dalam kondisi cukup padat maupun padat, dan durasi lampu merah pada jalur A selama 20 detik. Script dari gambar diatas:

```

else if (distance1 >= 10 && distance2 >= 10 && distance3 <= 10 && distance4
>= 10) // SENSOR B1
{
Serial.println("kondisi 4");

digitalWrite(led_R1, HIGH);

```

```

digitalWrite(led_Y1, LOW);
digitalWrite(led_G1, LOW);
delay(20000); // 10 DETIK

digitalWrite(led_R1, LOW);
digitalWrite(led_Y1, HIGH);
digitalWrite(led_G1, LOW);
delay(2000);

digitalWrite(led_R1, LOW);
digitalWrite(led_Y1, LOW);
digitalWrite(led_G1, HIGH);
delay(10000); // 20 DETIK

digitalWrite(led_R1, LOW);
digitalWrite(led_Y1, HIGH);
digitalWrite(led_G1, LOW);
delay(2000);
}

```

Gambar 4.45 Script Kondisi 4 Jalur A
Sumber ; Dokumentasi Pribadi

Sensor A1 = 84 CM	Sensor A2 = 11 CM	Sensor B1 = 2 CM	Sensor B2 = 41 CM
kondisi 4			
Sensor A1 = 84 CM	Sensor A2 = 11 CM	Sensor B1 = 7 CM	Sensor B2 = 41 CM
kondisi 4			
Sensor A1 = 83 CM	Sensor A2 = 11 CM	Sensor B1 = 6 CM	Sensor B2 = 40 CM
kondisi 4			
Sensor A1 = 85 CM	Sensor A2 = 11 CM	Sensor B1 = 5 CM	Sensor B2 = 40 CM
kondisi 4			

Gambar 4.46 Monitoring jarak objek Kondisi 4 Jalur A
Sumber ; Dokumentasi Pribadi

4.1.10 Tampilan Kondisi 4 Jalur B



Gambar 4.47 Kondisi 4 Jalur B
Sumber ; Dokumentasi Pribadi

Gambar 4.47 menunjukkan kondisi 4 jalur B, Pada kondisi ini Jalur B masuk dalam kondisi cukup padat, dan durasi lampu merah pada jalur B selama 10 detik. Script dari gambar diatas:

```
else if (distance1 >= 10 && distance2 >= 10 && distance3 <= 10 && distance4  
>= 10) // SENSOR B1  
{  
  Serial.println("kondisi 4");  
  
  digitalWrite(led_R1, HIGH);  
  digitalWrite(led_Y1, LOW);  
  digitalWrite(led_G1, LOW);  
  delay(20000); // 10 DETIK
```

```

digitalWrite(led_R1, LOW);
digitalWrite(led_Y1, HIGH);
digitalWrite(led_G1, LOW);
delay(2000);

digitalWrite(led_R1, LOW);
digitalWrite(led_Y1, LOW);
digitalWrite(led_G1, HIGH);
delay(10000); // 20 DETIK

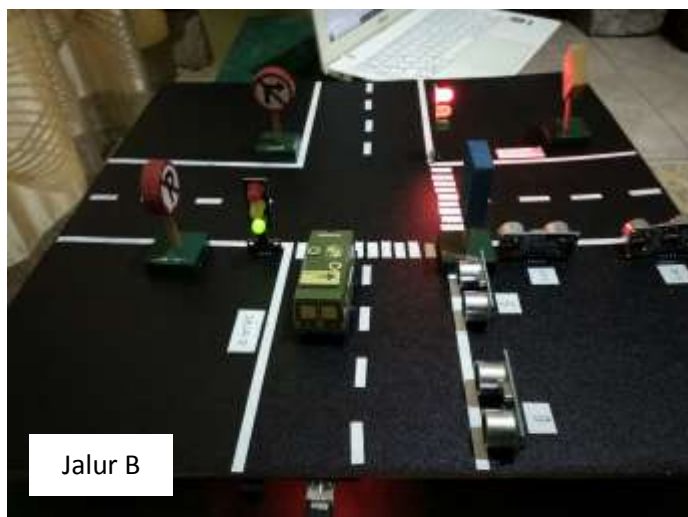
digitalWrite(led_R1, LOW);
digitalWrite(led_Y1, HIGH);
digitalWrite(led_G1, LOW);
delay(2000);
}

```

Gambar 4.48 Script Kondisi 4 Jalur B
 Sumber ; Dokumentasi Pribadi

Sensor A1 = 84 CM	Sensor A2 = 11 CM	Sensor B1 = 2 CM	Sensor B2 = 41 CM
kondisi 4			
Sensor A1 = 84 CM	Sensor A2 = 11 CM	Sensor B1 = 7 CM	Sensor B2 = 41 CM
kondisi 4			
Sensor A1 = 83 CM	Sensor A2 = 11 CM	Sensor B1 = 6 CM	Sensor B2 = 40 CM
kondisi 4			
Sensor A1 = 85 CM	Sensor A2 = 11 CM	Sensor B1 = 5 CM	Sensor B2 = 40 CM
kondisi 4			

Gambar 4.49 Monitoring jarak objek Kondisi 4 Jalur B
 Sumber ; Dokumentasi Pribadi



Gambar 4.50 Kondisi 4 Jalur B
Sumber ; Dokumentasi Pribadi

Gambar 4.50 menunjukkan kondisi 4 jalur B, Pada kondisi ini Jalur B masuk dalam kondisi cukup padat, dan durasi lampu hijau pada jalur B selama 20 detik. Script dari gambar diatas:

```
else if (distance1 >= 10 && distance2 >= 10 && distance3 <= 10 && distance4
>= 10) // SENSOR B1
{
Serial.println("kondisi 4");

digitalWrite(led_R1, HIGH);
digitalWrite(led_Y1, LOW);
digitalWrite(led_G1, LOW);
delay(20000); // 10 DETIK

digitalWrite(led_R1, LOW);
digitalWrite(led_Y1, HIGH);
digitalWrite(led_G1, LOW);
delay(2000);

digitalWrite(led_R1, LOW);
digitalWrite(led_Y1, LOW);
digitalWrite(led_G1, HIGH);
delay(10000); // 20 DETIK

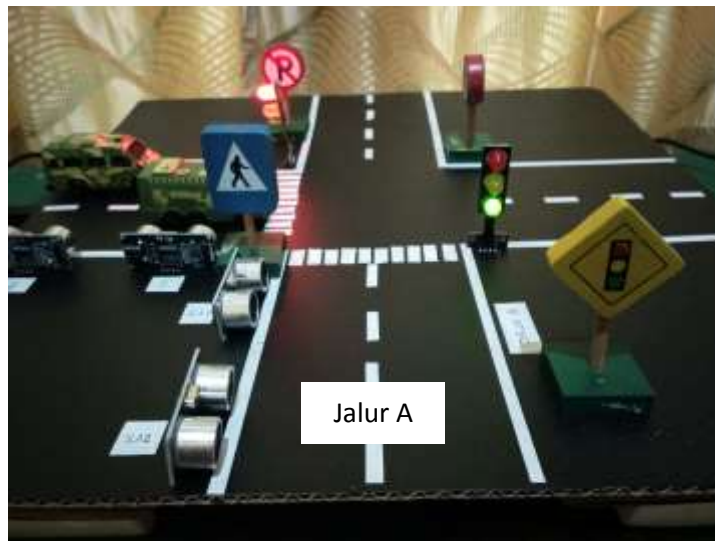
digitalWrite(led_R1, LOW);
digitalWrite(led_Y1, HIGH);
digitalWrite(led_G1, LOW);
delay(2000);
}
```


Gambar 4.51 Script Kondisi 4 Jalur B
Sumber ; Dokumentasi Pribadi

Sensor A1 = 84 CM	Sensor A2 = 11 CM	Sensor B1 = 2 CM	Sensor B2 = 41 CM
kondisi 4			
Sensor A1 = 84 CM	Sensor A2 = 11 CM	Sensor B1 = 7 CM	Sensor B2 = 41 CM
kondisi 4			
Sensor A1 = 83 CM	Sensor A2 = 11 CM	Sensor B1 = 6 CM	Sensor B2 = 40 CM
kondisi 4			
Sensor A1 = 85 CM	Sensor A2 = 11 CM	Sensor B1 = 5 CM	Sensor B2 = 40 CM
kondisi 4			

Gambar 4.52 Monitoring jarak objek Kondisi 4 Jalur B
Sumber ; Dokumentasi Pribadi

4.1.11 Tampilan Kondisi 5 Jalur A



Gambar 4.53 Kondisi 5 Jalur A
Sumber ; Dokumentasi Pribadi

Gambar 4.53 menunjukkan kondisi 5 jalur A, Pada kondisi ini Jalur A tidak masuk dalam kondisi cukup padat maupun padat, dan

durasi lampu hijau pada jalur A selama 10 detik. Script dari gambar diatas:

```

else if (distance1 >= 10 && distance2 >= 10 && distance3 <= 10 &&
distance4 <= 10) // SENSOR B1 & B2
{
Serial.println("kondisi 5");

digitalWrite(led_R1, HIGH);
digitalWrite(led_Y1, LOW);
digitalWrite(led_G1, LOW);
delay(30000); // 10 DETIK

digitalWrite(led_R1, LOW);
digitalWrite(led_Y1, HIGH);
digitalWrite(led_G1, LOW);
delay(2000);

digitalWrite(led_R1, LOW);
digitalWrite(led_Y1, LOW);
digitalWrite(led_G1, HIGH);
delay(10000);

digitalWrite(led_R1, LOW);
digitalWrite(led_Y1, HIGH);
digitalWrite(led_G1, LOW);
delay(2000);
}

```

Gambar 4.54 Script Kondisi 5 Jalur A
Sumber ; Dokumentasi Pribadi

Sensor A1 = 86 CM	Sensor A2 = 12 CM	Sensor B1 = 8 CM	Sensor B2 = 6 CM
kondisi 5			
Sensor A1 = 137 CM	Sensor A2 = 12 CM	Sensor B1 = 5 CM	Sensor B2 = 3 CM
kondisi 5			
Sensor A1 = 84 CM	Sensor A2 = 11 CM	Sensor B1 = 10 CM	Sensor B2 = 3 CM
kondisi 5			
Sensor A1 = 85 CM	Sensor A2 = 12 CM	Sensor B1 = 8 CM	Sensor B2 = 233 CM
kondisi 4			
Sensor A1 = 85 CM	Sensor A2 = 12 CM	Sensor B1 = 8 CM	Sensor B2 = 5 CM
kondisi 5			
Sensor A1 = 85 CM	Sensor A2 = 12 CM	Sensor B1 = 8 CM	Sensor B2 = 8 CM
kondisi 5			

Gambar 4.55 Monitoring jarak objek Kondisi 5 Jalur A
Sumber ; Dokumentasi Pribadi



Gambar 4.56 Kondisi 5 Jalur A
Sumber ; Dokumentasi Pribadi

Gambar 4.56 menunjukkan kondisi 5 jalur A, Pada kondisi ini Jalur A tidak masuk dalam kondisi cukup padat maupun padat, dan durasi lampu merah pada jalur A selama 30 detik. Script dari gambar diatas:

```

else if (distance1 >= 10 && distance2 >= 10 && distance3 <= 10 && distance4
<= 10) // SENSOR B1 & B2
{
Serial.println("kondisi 5");

digitalWrite(led_R1, HIGH);
digitalWrite(led_Y1, LOW);
digitalWrite(led_G1, LOW);
delay(30000); // 10 DETIK
digitalWrite(led_R1, LOW);
digitalWrite(led_Y1, HIGH);
digitalWrite(led_G1, LOW);
delay(2000);

digitalWrite(led_R1, LOW);
digitalWrite(led_Y1, LOW);
digitalWrite(led_G1, HIGH);
delay(10000);

digitalWrite(led_R1, LOW);
digitalWrite(led_Y1, HIGH);
digitalWrite(led_G1, LOW);
delay(2000);
}

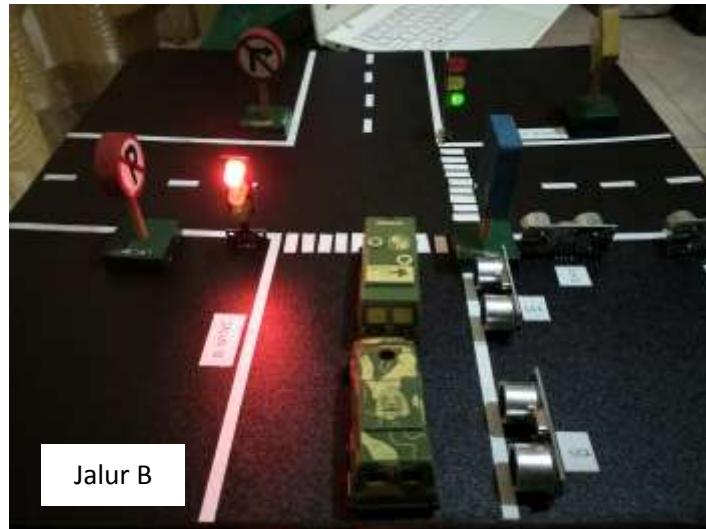
```

Gambar 4.57 Script Kondisi 5 Jalur A
Sumber ; Dokumentasi Pribadi

Sensor A1 = 84 CM	Sensor A2 = 11 CM	Sensor B1 = 2 CM	Sensor B2 = 41 CM
kondisi 4			
Sensor A1 = 84 CM	Sensor A2 = 11 CM	Sensor B1 = 7 CM	Sensor B2 = 41 CM
kondisi 4			
Sensor A1 = 83 CM	Sensor A2 = 11 CM	Sensor B1 = 6 CM	Sensor B2 = 40 CM
kondisi 4			
Sensor A1 = 85 CM	Sensor A2 = 11 CM	Sensor B1 = 5 CM	Sensor B2 = 40 CM
kondisi 4			

Gambar 4.58 Monitoring jarak objek Kondisi 5 Jalur A
Sumber ; Dokumentasi Pribadi

4.1.12 Tampilan Kondisi 5 Jalur B



Gambar 4.59 Kondisi 5 Jalur B
Sumber ; Dokumentasi Pribadi

Gambar 4.59 menunjukkan kondisi 5 jalur B, Pada kondisi ini Jalur B masuk dalam kondisi padat, dan durasi lampu merah pada jalur B selama 10 detik. Script dari gambar diatas:

```
else if (distance1 >= 10 && distance2 >= 10 && distance3 <= 10 &&
distance4 <= 10) // SENSOR B1 & B2
{
Serial.println("kondisi 5");

digitalWrite(led_R1, HIGH);
digitalWrite(led_Y1, LOW);
digitalWrite(led_G1, LOW);
delay(30000); // 10 DETIK

digitalWrite(led_R1, LOW);
digitalWrite(led_Y1, HIGH);
digitalWrite(led_G1, LOW);
```

```

delay(2000);

digitalWrite(led_R1, LOW);
digitalWrite(led_Y1, LOW);
digitalWrite(led_G1, HIGH);
delay(10000);

digitalWrite(led_R1, LOW);
digitalWrite(led_Y1, HIGH);
digitalWrite(led_G1, LOW);
delay(2000);
}

```

Gambar 4.60 Script Kondisi 5 Jalur B
 Sumber ; Dokumentasi Pribadi

Sensor A1 = 84 CM	Sensor A2 = 11 CM	Sensor B1 = 2 CM	Sensor B2 = 41 CM
kondisi 4			
Sensor A1 = 84 CM	Sensor A2 = 11 CM	Sensor B1 = 7 CM	Sensor B2 = 41 CM
kondisi 4			
Sensor A1 = 83 CM	Sensor A2 = 11 CM	Sensor B1 = 6 CM	Sensor B2 = 40 CM
kondisi 4			
Sensor A1 = 85 CM	Sensor A2 = 11 CM	Sensor B1 = 5 CM	Sensor B2 = 40 CM
kondisi 4			

Gambar 4.61 Monitoring jarak objek Kondisi 5 Jalur B
 Sumber ; Dokumentasi Pribadi



Jalur B

Gambar 4.62 Kondisi 5 Jalur B
Sumber ; Dokumentasi Pribadi

Gambar 4.62 menunjukkan kondisi 5 jalur B, Pada kondisi ini Jalur B masuk dalam kondisi padat, dan durasi lampu merah pada jalur B selama 30 detik. Script dari gambar diatas:

```
else if (distance1 >= 10 && distance2 >= 10 && distance3 <= 10 && distance4
<= 10) // SENSOR B1 & B2
{
Serial.println("kondisi 5");

digitalWrite(led_R1, HIGH);
digitalWrite(led_Y1, LOW);
digitalWrite(led_G1, LOW);
delay(30000); // 10 DETIK

digitalWrite(led_R1, LOW);
digitalWrite(led_Y1, HIGH);
digitalWrite(led_G1, LOW);
delay(2000);

digitalWrite(led_R1, LOW);
digitalWrite(led_Y1, LOW);
digitalWrite(led_G1, HIGH);
delay(10000);

digitalWrite(led_R1, LOW);
digitalWrite(led_Y1, HIGH);
digitalWrite(led_G1, LOW);
```

```
delay(2000);  
}
```

Gambar 4.63 Script Kondisi 5 Jalur B
Sumber ; Dokumentasi Pribadi

Sensor A1 = 84 CM	Sensor A2 = 11 CM	Sensor B1 = 2 CM	Sensor B2 = 41 CM
kondisi 4			
Sensor A1 = 84 CM	Sensor A2 = 11 CM	Sensor B1 = 7 CM	Sensor B2 = 41 CM
kondisi 4			
Sensor A1 = 83 CM	Sensor A2 = 11 CM	Sensor B1 = 6 CM	Sensor B2 = 40 CM
kondisi 4			
Sensor A1 = 85 CM	Sensor A2 = 11 CM	Sensor B1 = 5 CM	Sensor B2 = 40 CM
kondisi 4			

Gambar 4.64 Monitoring jarak objek Kondisi 5 Jalur B
Sumber ; Dokumentasi Pribadi

4.1.13 Tampilan Kondisi 6 Jalur A



Jalur A

Gambar 4.65 Kondisi 6 Jalur A
Sumber ; Dokumentasi Pribadi

Gambar 4.65 menunjukkan kondisi 6 jalur A, Pada kondisi ini Jalur A masuk dalam kondisi cukup padat, dan durasi lampu hijau pada jalur A selama 20 detik. Script dari gambar diatas:

```
else if (distance1 <= 10 && distance2 >= 10 && distance3 <= 10 && distance4  
>= 10) // SENSOR A1 & B1  
{  
  Serial.println("kondisi 6");  
  
  digitalWrite(led_R1, HIGH);  
  digitalWrite(led_Y1, LOW);  
  digitalWrite(led_G1, LOW);  
  delay(20000); //  
  
  digitalWrite(led_R1, LOW);  
  digitalWrite(led_Y1, HIGH);  
  digitalWrite(led_G1, LOW);  
  delay(2000);  
  
  digitalWrite(led_R1, LOW);  
  digitalWrite(led_Y1, LOW);  
  digitalWrite(led_G1, HIGH);  
  delay(20000); //  
  
  digitalWrite(led_R1, LOW);  
  digitalWrite(led_Y1, HIGH);  
  digitalWrite(led_G1, LOW);  
  delay(2000);  
}
```

Gambar 4.66 Script Kondisi 6 Jalur A
Sumber ; Dokumentasi Pribadi

```
Sensor A1 = 3 CM      Sensor A2 = 12 CM      Sensor B1 = 7 CM      Sensor B2 = 42 CM  
kondisi 6  
Sensor A1 = -94 CM   Sensor A2 = 12 CM      Sensor B1 = 4 CM      Sensor B2 = 41 CM  
kondisi 6  
Sensor A1 = 7 CM     Sensor A2 = 12 CM      Sensor B1 = 4 CM      Sensor B2 = 40 CM  
kondisi 6  
Sensor A1 = 3 CM     Sensor A2 = 12 CM      Sensor B1 = 6 CM      Sensor B2 = 41 CM  
kondisi 6
```

Gambar 4.67 Monitoring jarak objek Kondisi 6 Jalur A
Sumber ; Dokumentasi Pribadi



Gambar 4.68 Kondisi 6 Jalur A
Sumber ; Dokumentasi Pribadi

Gambar 4.64 menunjukkan kondisi 6 jalur A, Pada kondisi ini Jalur A masuk dalam kondisi cukup padat, dan durasi lampu merah pada jalur A selama 20 detik. Script dari gambar diatas:

```
else if (distance1 <= 10 && distance2 >= 10 && distance3 <= 10 && distance4  
>= 10) // SENSOR A1 & B1  
{  
  Serial.println("kondisi 6");  
  
  digitalWrite(led_R1, HIGH);  
  digitalWrite(led_Y1, LOW);  
  digitalWrite(led_G1, LOW);  
  delay(20000); //
```

```

digitalWrite(led_R1, LOW);
digitalWrite(led_Y1, HIGH);
digitalWrite(led_G1, LOW);
delay(2000);

digitalWrite(led_R1, LOW);
digitalWrite(led_Y1, LOW);
digitalWrite(led_G1, HIGH);
delay(20000); //

digitalWrite(led_R1, LOW);
digitalWrite(led_Y1, HIGH);
digitalWrite(led_G1, LOW);
delay(2000);
}

```

Gambar 4.69 Script Kondisi 6 Jalur A
 Sumber ; Dokumentasi Pribadi

Sensor A1 = 3 CM	Sensor A2 = 12 CM	Sensor B1 = 7 CM	Sensor B2 = 42 CM
kondisi 6			
Sensor A1 = -94 CM	Sensor A2 = 12 CM	Sensor B1 = 4 CM	Sensor B2 = 41 CM
kondisi 6			
Sensor A1 = 7 CM	Sensor A2 = 12 CM	Sensor B1 = 4 CM	Sensor B2 = 40 CM
kondisi 6			
Sensor A1 = 3 CM	Sensor A2 = 12 CM	Sensor B1 = 6 CM	Sensor B2 = 41 CM
kondisi 6			

Gambar 4.70 Monitoring jarak objek Kondisi 6 Jalur A
 Sumber ; Dokumentasi Pribadi

4.1.14 Tampilan Kondisi 6 Jalur B



Jalur B

Gambar 4.71 Kondisi 6 Jalur B
Sumber ; Dokumentasi Pribadi

Gambar 4.71 menunjukkan kondisi 6 jalur B, Pada kondisi ini Jalur B masuk dalam kondisi cukup padat, dan durasi lampu hijau pada jalur B selama 20 detik. Script dari gambar diatas:

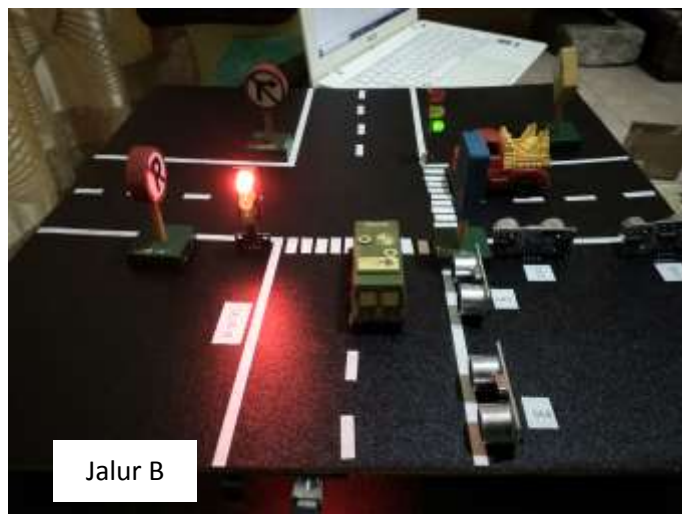
```
else if (distance1 <= 10 && distance2 >= 10 && distance3 <= 10 && distance4  
>= 10) // SENSOR A1 & B1  
{  
  Serial.println("kondisi 6");  
  
  digitalWrite(led_R1, HIGH);  
  digitalWrite(led_Y1, LOW);  
  digitalWrite(led_G1, LOW);  
  delay(20000); //  
  
  digitalWrite(led_R1, LOW);  
  digitalWrite(led_Y1, HIGH);  
  digitalWrite(led_G1, LOW);  
  delay(2000);  
  
  digitalWrite(led_R1, LOW);  
  digitalWrite(led_Y1, LOW);  
  digitalWrite(led_G1, HIGH);  
  delay(20000); //  
  
  digitalWrite(led_R1, LOW);
```

```
digitalWrite(led_Y1, HIGH);  
digitalWrite(led_G1, LOW);  
delay(2000);  
}
```

Gambar 4.72 Script Kondisi 6 Jalur B
Sumber ; Dokumentasi Pribadi

Sensor A1 = 3 CM	Sensor A2 = 12 CM	Sensor B1 = 7 CM	Sensor B2 = 42 CM
kondisi 6			
Sensor A1 = -94 CM	Sensor A2 = 12 CM	Sensor B1 = 4 CM	Sensor B2 = 41 CM
kondisi 6			
Sensor A1 = 7 CM	Sensor A2 = 12 CM	Sensor B1 = 4 CM	Sensor B2 = 40 CM
kondisi 6			
Sensor A1 = 3 CM	Sensor A2 = 12 CM	Sensor B1 = 6 CM	Sensor B2 = 41 CM
kondisi 6			

Gambar 4.73 Monitoring jarak objek Kondisi 6 Jalur B
Sumber ; Dokumentasi Pribadi



Gambar 4.74 Kondisi 6 Jalur B
Sumber ; Dokumentasi Pribadi

Gambar 4.74 menunjukkan kondisi 6 jalur B, Pada kondisi ini Jalur B masuk dalam kondisi cukup padat, dan durasi lampu merah pada jalur B selama 20 detik. Script dari gambar diatas:

```

else if (distance1 <= 10 && distance2 >= 10 && distance3 <= 10 && distance4
>= 10) // SENSOR A1 & B1
{
Serial.println("kondisi 6");

digitalWrite(led_R1, HIGH);
digitalWrite(led_Y1, LOW);
digitalWrite(led_G1, LOW);
delay(20000); //

digitalWrite(led_R1, LOW);
digitalWrite(led_Y1, HIGH);
digitalWrite(led_G1, LOW);
delay(2000);

digitalWrite(led_R1, LOW);
digitalWrite(led_Y1, LOW);
digitalWrite(led_G1, HIGH);
delay(20000); //

digitalWrite(led_R1, LOW);
digitalWrite(led_Y1, HIGH);
digitalWrite(led_G1, LOW);
delay(2000);
}

```

Gambar 4.75 Script Kondisi 6 Jalur B
Sumber ; Dokumentasi Pribadi

Sensor A1 = 3 CM	Sensor A2 = 12 CM	Sensor B1 = 7 CM	Sensor B2 = 42 CM
kondisi 6			
Sensor A1 = -94 CM	Sensor A2 = 12 CM	Sensor B1 = 4 CM	Sensor B2 = 41 CM
kondisi 6			
Sensor A1 = 7 CM	Sensor A2 = 12 CM	Sensor B1 = 4 CM	Sensor B2 = 40 CM
kondisi 6			
Sensor A1 = 3 CM	Sensor A2 = 12 CM	Sensor B1 = 6 CM	Sensor B2 = 41 CM
kondisi 6			

Gambar 4.76 Monitoring jarak objek Kondisi 6 Jalur B
Sumber ; Dokumentasi Pribadi

4.1.15 Tampilan Kondisi 7 Jalur A



Gambar 4.77 Kondisi 7 Jalur A
Sumber ; Dokumentasi Pribadi

Gambar 4.77 menunjukkan kondisi 7 jalur A, Pada kondisi ini Jalur A masuk dalam kondisi padat, dan durasi lampu Hijau pada jalur A selama 30 detik. Script dari gambar diatas:

```
else if (distance1 <= 10 && distance2 <= 10 && distance3 <= 10 && distance4  
>= 10) // SENSOR A1 & A2 & B1  
{  
Serial.println("kondisi 7");  
  
digitalWrite(led_R1, HIGH);  
digitalWrite(led_Y1, LOW);  
digitalWrite(led_G1, LOW);  
delay(20000); //
```

```

digitalWrite(led_R1, LOW);
digitalWrite(led_Y1, HIGH);
digitalWrite(led_G1, LOW);
delay(2000);

digitalWrite(led_R1, LOW);
digitalWrite(led_Y1, LOW);
digitalWrite(led_G1, HIGH);
delay(30000); //

digitalWrite(led_R1, LOW);
digitalWrite(led_Y1, HIGH);
digitalWrite(led_G1, LOW);
delay(2000);
}

```

Gambar 4.78 Sumber ; Dokumentasi Pribadi
Script Kondisi 7 Jalur A

Sensor A1 = 4 CM	Sensor A2 = 2 CM	Sensor B1 = 3 CM	Sensor B2 = 40 CM
kondisi 7			
Sensor A1 = 7 CM	Sensor A2 = 2 CM	Sensor B1 = 2 CM	Sensor B2 = 40 CM
kondisi 7			
Sensor A1 = 7 CM	Sensor A2 = 2 CM	Sensor B1 = 3 CM	Sensor B2 = 40 CM
kondisi 7			
Sensor A1 = 6 CM	Sensor A2 = 6 CM	Sensor B1 = 5 CM	Sensor B2 = 41 CM
kondisi 7			
Sensor A1 = 6 CM	Sensor A2 = 6 CM	Sensor B1 = 5 CM	Sensor B2 = 40 CM
kondisi 7			

Gambar 4.79 Monitoring jarak objek Kondisi 7 Jalur A
Sumber ; Dokumentasi Pribadi



Jalur A

Gambar 4.80 Kondisi 7 Jalur A
Sumber ; Dokumentasi Pribadi

Gambar 4.80 menunjukkan kondisi 7 jalur A, Pada kondisi ini Jalur A masuk dalam kondisi padat, dan durasi lampu merah pada jalur A selama 20 detik. Script dari gambar diatas:

```
else if (distance1 <= 10 && distance2 <= 10 && distance3 <= 10 && distance4
>= 10) // SENSOR A1 & A2 & B1
{
Serial.println("kondisi 7");

digitalWrite(led_R1, HIGH);
digitalWrite(led_Y1, LOW);
digitalWrite(led_G1, LOW);
delay(20000); //

digitalWrite(led_R1, LOW);
digitalWrite(led_Y1, HIGH);
digitalWrite(led_G1, LOW);
delay(2000);

digitalWrite(led_R1, LOW);
digitalWrite(led_Y1, LOW);
digitalWrite(led_G1, HIGH);
delay(30000); //

digitalWrite(led_R1, LOW);
digitalWrite(led_Y1, HIGH);
```

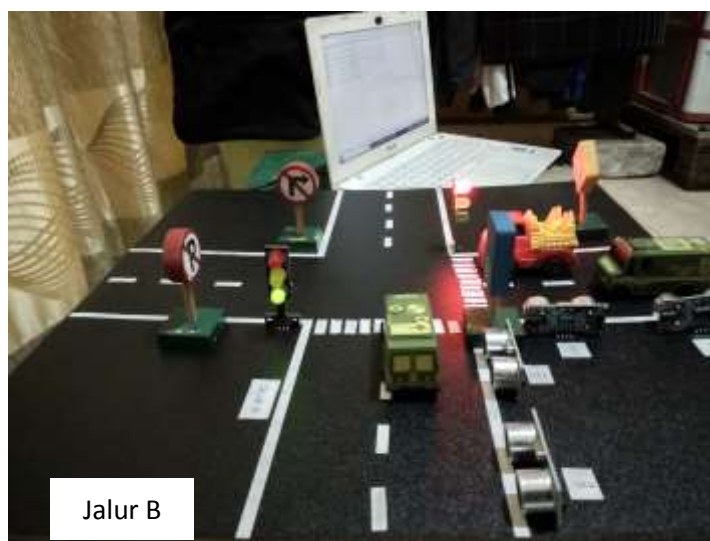
```
digitalWrite(led_G1, LOW);  
delay(2000);  
}
```

Gambar 4.81 Script Kondisi 7 Jalur A
Sumber ; Dokumentasi Pribadi

Sensor A1 = 4 CM	Sensor A2 = 2 CM	Sensor B1 = 3 CM	Sensor B2 = 40 CM
kondisi 7			
Sensor A1 = 7 CM	Sensor A2 = 2 CM	Sensor B1 = 2 CM	Sensor B2 = 40 CM
kondisi 7			
Sensor A1 = 7 CM	Sensor A2 = 2 CM	Sensor B1 = 3 CM	Sensor B2 = 40 CM
kondisi 7			
Sensor A1 = 6 CM	Sensor A2 = 6 CM	Sensor B1 = 5 CM	Sensor B2 = 41 CM
kondisi 7			
Sensor A1 = 6 CM	Sensor A2 = 6 CM	Sensor B1 = 5 CM	Sensor B2 = 40 CM
kondisi 7			

Gambar 4.82 Monitoring jarak objek Kondisi 7 Jalur A
Sumber ; Dokumentasi Pribadi

4.1.16 Tampilan Kondisi 7 Jalur B



Gambar 4.83 Kondisi 7 Jalur B
Sumber ; Dokumentasi Pribadi

Gambar 4.83 menunjukkan kondisi 7 jalur B, Pada kondisi ini Jalur B masuk dalam kondisi cukup padat, dan durasi lampu hijau pada jalur B selama 20 detik. Script dari gambar diatas:

```

else if (distance1 <= 10 && distance2 <= 10 && distance3 <= 10 && distance4
>= 10) // SENSOR A1 & A2 & B1
{
Serial.println("kondisi 7");

digitalWrite(led_R1, HIGH);
digitalWrite(led_Y1, LOW);
digitalWrite(led_G1, LOW);
delay(20000); //

digitalWrite(led_R1, LOW);
digitalWrite(led_Y1, HIGH);
digitalWrite(led_G1, LOW);
delay(2000);

digitalWrite(led_R1, LOW);
digitalWrite(led_Y1, LOW);
digitalWrite(led_G1, HIGH);
delay(30000); //

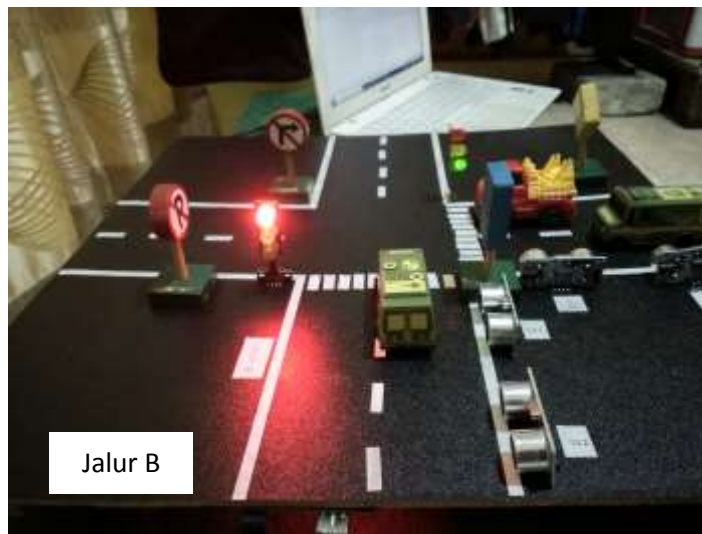
digitalWrite(led_R1, LOW);
digitalWrite(led_Y1, HIGH);
digitalWrite(led_G1, LOW);
delay(2000);
}

```

Gambar 4.84 Script Kondisi 7 Jalur B
Sumber: Dokumentasi Pribadi

Sensor A1 = 4 CM	Sensor A2 = 2 CM	Sensor B1 = 3 CM	Sensor B2 = 40 CM
kondisi 7			
Sensor A1 = 7 CM	Sensor A2 = 2 CM	Sensor B1 = 2 CM	Sensor B2 = 40 CM
kondisi 7			
Sensor A1 = 7 CM	Sensor A2 = 2 CM	Sensor B1 = 3 CM	Sensor B2 = 40 CM
kondisi 7			
Sensor A1 = 6 CM	Sensor A2 = 6 CM	Sensor B1 = 5 CM	Sensor B2 = 41 CM
kondisi 7			
Sensor A1 = 6 CM	Sensor A2 = 6 CM	Sensor B1 = 5 CM	Sensor B2 = 40 CM
kondisi 7			

Gambar 4.85 Monitoring jarak objek Kondisi 7 Jalur B
Sumber ; Dokumentasi Pribadi



Gambar 4.86 Kondisi 7 Jalur B
Sumber ; Dokumentasi Pribadi

Gambar 4.86 menunjukkan kondisi 7 jalur B, Pada kondisi ini Jalur B masuk dalam kondisi cukup padat, dan durasi lampu merah pada jalur B selama 20 detik. Script dari gambar diatas:

```
else if (distance1 <= 10 && distance2 <= 10 && distance3 <= 10 && distance4  
>= 10) // SENSOR A1 & A2 & B1  
{  
Serial.println("kondisi 7");  
  
digitalWrite(led_R1, HIGH);  
digitalWrite(led_Y1, LOW);  
digitalWrite(led_G1, LOW);  
delay(20000); //
```

```

digitalWrite(led_R1, LOW);
digitalWrite(led_Y1, HIGH);
digitalWrite(led_G1, LOW);
delay(2000);

digitalWrite(led_R1, LOW);
digitalWrite(led_Y1, LOW);
digitalWrite(led_G1, HIGH);
delay(30000); //

digitalWrite(led_R1, LOW);
digitalWrite(led_Y1, HIGH);
digitalWrite(led_G1, LOW);
delay(2000);
}

```

Gambar 4.87 Script Kondisi 7 Jalur B
 Sumber ; Dokumentasi Pribadi

Sensor A1 = 4 CM	Sensor A2 = 2 CM	Sensor B1 = 3 CM	Sensor B2 = 40 CM
kondisi 7			
Sensor A1 = 7 CM	Sensor A2 = 2 CM	Sensor B1 = 2 CM	Sensor B2 = 40 CM
kondisi 7			
Sensor A1 = 7 CM	Sensor A2 = 2 CM	Sensor B1 = 3 CM	Sensor B2 = 40 CM
kondisi 7			
Sensor A1 = 6 CM	Sensor A2 = 6 CM	Sensor B1 = 5 CM	Sensor B2 = 41 CM
kondisi 7			
Sensor A1 = 6 CM	Sensor A2 = 6 CM	Sensor B1 = 5 CM	Sensor B2 = 40 CM
kondisi 7			

Gambar 4.88 Monitoring jarak objek Kondisi 7 jalur B
 Sumber ; Dokumentasi Pribadi

4.1.17 Tampilan Kondisi 8 Jalur A



Jalur A

Gambar 4.89 Kondisi 8 Jalur A
Sumber ; Dokumentasi Pribadi

Gambar 4.89 menunjukkan kondisi 8 jalur A, Pada kondisi ini Jalur A masuk dalam kondisi cukup padat, dan durasi lampu hijau pada jalur A selama 20 detik. Script dari gambar diatas:

```
else if (distance1 <= 10 && distance2 >= 10 && distance3 <= 10 &&
distance4 <= 10) // SENSOR A1 & B1 & B2
{
Serial.println("kondisi 8");

digitalWrite(led_R1, HIGH);
digitalWrite(led_Y1, LOW);
digitalWrite(led_G1, LOW);
delay(30000); //

digitalWrite(led_R1, LOW);
digitalWrite(led_Y1, HIGH);
digitalWrite(led_G1, LOW);
delay(2000);

digitalWrite(led_R1, LOW);
digitalWrite(led_Y1, LOW);
digitalWrite(led_G1, HIGH);
delay(20000); //
```

```

digitalWrite(led_R1, LOW);
digitalWrite(led_Y1, HIGH);
digitalWrite(led_G1, LOW);
delay(2000);
}

```

Gambar 4.90 Script Kondisi 8 Jalur A
 Sumber ; Dokumentasi Pribadi

Sensor A1 = 7 CM	Sensor A2 = 12 CM	Sensor B1 = 3 CM	Sensor B2 = 6 CM
kondisi 8			
Sensor A1 = 3 CM	Sensor A2 = 12 CM	Sensor B1 = 3 CM	Sensor B2 = 5 CM
kondisi 8			
Sensor A1 = 3 CM	Sensor A2 = 12 CM	Sensor B1 = 4 CM	Sensor B2 = 5 CM
kondisi 8			
Sensor A1 = 7 CM	Sensor A2 = 12 CM	Sensor B1 = 4 CM	Sensor B2 = 5 CM
kondisi 8			

Gambar 4.91 Monitoring jarak objek Kondisi 8 Jalur A
 Sumber ; Dokumentasi Pribadi



Gambar 4.92 Kondisi 8 Jalur A
Sumber ; Dokumentasi Pribadi

Gambar 4.92 menunjukkan kondisi 8 jalur A, Pada kondisi ini Jalur A masuk dalam kondisi cukup padat, dan durasi lampu merah pada jalur A selama 30 detik. Script dari gambar diatas:

```
else if (distance1 <= 10 && distance2 >= 10 && distance3 <= 10 && distance4
<= 10) // SENSOR A1 & B1 & B2
{
Serial.println("kondisi 8");

digitalWrite(led_R1, HIGH);
digitalWrite(led_Y1, LOW);
digitalWrite(led_G1, LOW);
delay(30000); //

digitalWrite(led_R1, LOW);
digitalWrite(led_Y1, HIGH);
digitalWrite(led_G1, LOW);
delay(2000);

digitalWrite(led_R1, LOW);
digitalWrite(led_Y1, LOW);
digitalWrite(led_G1, HIGH);
delay(20000); //

digitalWrite(led_R1, LOW);
digitalWrite(led_Y1, HIGH);
digitalWrite(led_G1, LOW);
delay(2000);
}
```

Gambar 4.93 Script Kondisi 8 Jalur A
Sumber ; Dokumentasi Pribadi

Sensor A1 = 7 CM	Sensor A2 = 12 CM	Sensor B1 = 3 CM	Sensor B2 = 6 CM
kondisi 8			
Sensor A1 = 3 CM	Sensor A2 = 12 CM	Sensor B1 = 3 CM	Sensor B2 = 5 CM
kondisi 8			
Sensor A1 = 3 CM	Sensor A2 = 12 CM	Sensor B1 = 4 CM	Sensor B2 = 5 CM
kondisi 8			
Sensor A1 = 7 CM	Sensor A2 = 12 CM	Sensor B1 = 4 CM	Sensor B2 = 5 CM
kondisi 8			

Gambar 4.94 Monitoring jarak objek Kondisi 8 Jalur A
 Sumber ; Dokumentasi Pribadi

4.1.18 Tampilan Kondisi 8 Jalur B



Gambar 4.95 Kondisi 8 Jalur B
 Sumber ; Dokumentasi Pribadi

Gambar 4.95 menunjukkan kondisi 8 jalur B, Pada kondisi ini Jalur B masuk dalam kondisi cukup padat, dan durasi lampu hijau pada jalur B selama 30 detik. Script dari gambar diatas:

```

else if (distance1 <= 10 && distance2 >= 10 && distance3 <= 10 && distance4
<= 10) // SENSOR A1 & B1 & B2
{
Serial.println("kondisi 8");

digitalWrite(led_R1, HIGH);
digitalWrite(led_Y1, LOW);

```

```

digitalWrite(led_G1, LOW);
delay(30000); //

digitalWrite(led_R1, LOW);
digitalWrite(led_Y1, HIGH);
digitalWrite(led_G1, LOW);
delay(2000);

digitalWrite(led_R1, LOW);
digitalWrite(led_Y1, LOW);
digitalWrite(led_G1, HIGH);
delay(20000); //

digitalWrite(led_R1, LOW);
digitalWrite(led_Y1, HIGH);
digitalWrite(led_G1, LOW);
delay(2000);
}

```

Gambar 4.96 Script Kondisi 8 Jalur B
 Sumber ; Dokumentasi Pribadi

Sensor A1 = 7 CM	Sensor A2 = 12 CM	Sensor B1 = 3 CM	Sensor B2 = 6 CM
kondisi 8			
Sensor A1 = 3 CM	Sensor A2 = 12 CM	Sensor B1 = 3 CM	Sensor B2 = 5 CM
kondisi 8			
Sensor A1 = 3 CM	Sensor A2 = 12 CM	Sensor B1 = 4 CM	Sensor B2 = 5 CM
kondisi 8			
Sensor A1 = 7 CM	Sensor A2 = 12 CM	Sensor B1 = 4 CM	Sensor B2 = 5 CM
kondisi 8			

Gambar 4.97 Monitoring jarak objek Kondisi 8 Jalur B
 Sumber ; Dokumentasi Pribadi



Jalur B

Gambar 4.98 Kondisi 8 Jalur B
Sumber ; Dokumentasi Pribadi

Gambar 4.98 menunjukkan kondisi 8 jalur B, Pada kondisi ini Jalur B masuk dalam kondisi cukup padat, dan durasi lampu merah pada jalur B selama 30 detik. Script dari gambar diatas:

```
else if (distance1 <= 10 && distance2 >= 10 && distance3 <= 10 && distance4
<= 10) // SENSOR A1 & B1 & B2
{
Serial.println("kondisi 8");

digitalWrite(led_R1, HIGH);
digitalWrite(led_Y1, LOW);
digitalWrite(led_G1, LOW);
delay(30000); //

digitalWrite(led_R1, LOW);
digitalWrite(led_Y1, HIGH);
digitalWrite(led_G1, LOW);
delay(2000);

digitalWrite(led_R1, LOW);
digitalWrite(led_Y1, LOW);
digitalWrite(led_G1, HIGH);
delay(20000); //

digitalWrite(led_R1, LOW);
```

```
digitalWrite(led_Y1, HIGH);  
digitalWrite(led_G1, LOW);  
delay(2000);  
}
```

Gambar 4.99 Script Kondisi 8 Jalur B
Sumber ; Dokumentasi Pribadi

Sensor A1 = 7 CM	Sensor A2 = 12 CM	Sensor B1 = 3 CM	Sensor B2 = 6 CM
kondisi 8			
Sensor A1 = 3 CM	Sensor A2 = 12 CM	Sensor B1 = 3 CM	Sensor B2 = 5 CM
kondisi 8			
Sensor A1 = 3 CM	Sensor A2 = 12 CM	Sensor B1 = 4 CM	Sensor B2 = 5 CM
kondisi 8			
Sensor A1 = 7 CM	Sensor A2 = 12 CM	Sensor B1 = 4 CM	Sensor B2 = 5 CM
kondisi 8			

Gambar 4.100 Monitoring jarak objek Kondisi 8 Jalur B
Sumber ; Dokumentasi Pribadi

4.1.19 Tampilan Kondisi 9 Jalur A



Jalur A

Gambar 4.101 Kondisi 9 Jalur A
Sumber ; Dokumentasi Pribadi

Gambar 4.101 menunjukkan kondisi 9 jalur A, Pada kondisi ini Jalur A masuk dalam kondisi padat, dan durasi lampu hijau pada jalur A selama 30 detik. Script dari gambar diatas:

```
else if (distance1 <= 10 && distance2 <= 10 && distance3 <= 10 && distance4
<= 10) // SENSOR A1 & A2 & B1 & B2
{
Serial.println("kondisi 9");

digitalWrite(led_R1, HIGH);
digitalWrite(led_Y1, LOW);
digitalWrite(led_G1, LOW);
delay(30000); //

digitalWrite(led_R1, LOW);
digitalWrite(led_Y1, HIGH);
digitalWrite(led_G1, LOW);
delay(2000);

digitalWrite(led_R1, LOW);
digitalWrite(led_Y1, LOW);
digitalWrite(led_G1, HIGH);
delay(30000); //

digitalWrite(led_R1, LOW);
digitalWrite(led_Y1, HIGH);
digitalWrite(led_G1, LOW);
delay(2000);
}
```

Gambar 4.102 Script Kondisi 9 Jalur A
Sumber ; Dokumentasi Pribadi

Sensor A1 = 6 CM	Sensor A2 = 2 CM	Sensor B1 = 9 CM	Sensor B2 = 4 CM
kondisi 9			
Sensor A1 = 6 CM	Sensor A2 = 2 CM	Sensor B1 = 10 CM	Sensor B2 = 4 CM
kondisi 9			
Sensor A1 = 6 CM	Sensor A2 = 2 CM	Sensor B1 = 6 CM	Sensor B2 = 4 CM
kondisi 9			
Sensor A1 = 3 CM	Sensor A2 = 5 CM	Sensor B1 = 5 CM	Sensor B2 = 5 CM
kondisi 9			
Sensor A1 = 4 CM	Sensor A2 = 5 CM	Sensor B1 = 5 CM	Sensor B2 = 6 CM
kondisi 9			
Sensor A1 = 3 CM	Sensor A2 = 2 CM	Sensor B1 = 2 CM	Sensor B2 = 3 CM
kondisi 9			
Sensor A1 = 3 CM	Sensor A2 = 2 CM	Sensor B1 = 2 CM	Sensor B2 = 5 CM
kondisi 9			

Gambar 4.103 Monitoring jarak objek Kondisi 9 Jalur A
Sumber ; Dokumentasi Pribadi



Gambar 4.104 Kondisi 9 Jalur A
Sumber ; Dokumentasi Pribadi

Gambar 4.104 menunjukkan kondisi 9 jalur A, Pada kondisi ini Jalur A masuk dalam kondisi padat, dan durasi lampu merah pada jalur A selama 30 detik. Script dari gambar diatas:

```

else if (distance1 <= 10 && distance2 <= 10 && distance3 <= 10 && distance4
<= 10) // SENSOR A1 & A2 & B1 & B2
{
Serial.println("kondisi 9");
}

```

```

digitalWrite(led_R1, HIGH);
digitalWrite(led_Y1, LOW);
digitalWrite(led_G1, LOW);
delay(30000); //

digitalWrite(led_R1, LOW);
digitalWrite(led_Y1, HIGH);
digitalWrite(led_G1, LOW);
delay(2000);

digitalWrite(led_R1, LOW);
digitalWrite(led_Y1, LOW);
digitalWrite(led_G1, HIGH);
delay(30000); //

digitalWrite(led_R1, LOW);
digitalWrite(led_Y1, HIGH);
digitalWrite(led_G1, LOW);
delay(2000);
}

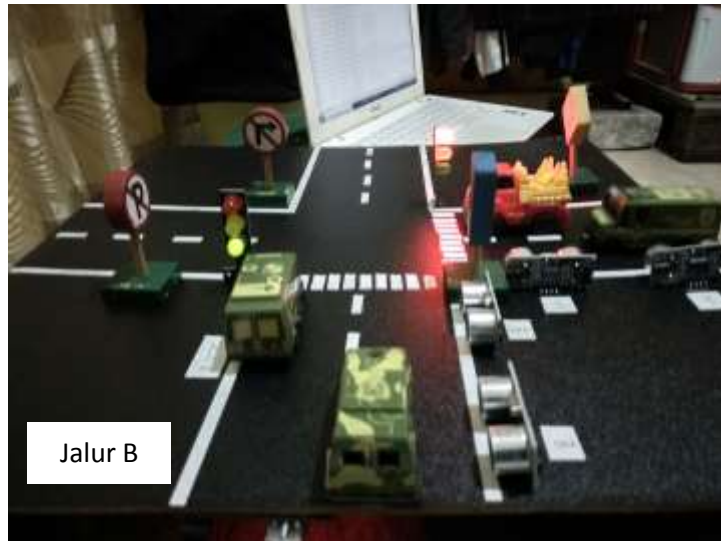
```

Gambar 4.105 Script Kondisi 9 Jalur A
Sumber ; Dokumentasi Pribadi

Sensor A1 = 6 CM	Sensor A2 = 2 CM	Sensor B1 = 9 CM	Sensor B2 = 4 CM
kondisi 9			
Sensor A1 = 6 CM	Sensor A2 = 2 CM	Sensor B1 = 10 CM	Sensor B2 = 4 CM
kondisi 9			
Sensor A1 = 6 CM	Sensor A2 = 2 CM	Sensor B1 = 6 CM	Sensor B2 = 4 CM
kondisi 9			
Sensor A1 = 3 CM	Sensor A2 = 5 CM	Sensor B1 = 5 CM	Sensor B2 = 5 CM
kondisi 9			
Sensor A1 = 4 CM	Sensor A2 = 5 CM	Sensor B1 = 5 CM	Sensor B2 = 6 CM
kondisi 9			
Sensor A1 = 3 CM	Sensor A2 = 2 CM	Sensor B1 = 2 CM	Sensor B2 = 3 CM
kondisi 9			
Sensor A1 = 3 CM	Sensor A2 = 2 CM	Sensor B1 = 2 CM	Sensor B2 = 5 CM
kondisi 9			

Gambar 4.106 Monitoring jarak objek Kondisi 9 Jalur A
Sumber ; Dokumentasi Pribadi

4.1.20 Tampilan Kondisi 9 Jalur B



Gambar 4.107 Kondisi 9 Jalur B
Sumber ; Dokumentasi Pribadi

Gambar 4.107 menunjukkan kondisi 9 jalur B, Pada kondisi ini Jalur B masuk dalam kondisi padat, dan durasi lampu hijau pada jalur B selama 30 detik. Script dari gambar diatas:

```
else if (distance1 <= 10 && distance2 <= 10 && distance3 <= 10 && distance4
<= 10) // SENSOR A1 & A2 & B1 & B2
{
Serial.println("kondisi 9");

digitalWrite(led_R1, HIGH);
digitalWrite(led_Y1, LOW);
digitalWrite(led_G1, LOW);
delay(30000); //

digitalWrite(led_R1, LOW);
digitalWrite(led_Y1, HIGH);
digitalWrite(led_G1, LOW);
delay(2000);
```



```

digitalWrite(led_R1, LOW);
digitalWrite(led_Y1, LOW);
digitalWrite(led_G1, HIGH);
delay(30000); //

digitalWrite(led_R1, LOW);
digitalWrite(led_Y1, HIGH);
digitalWrite(led_G1, LOW);
delay(2000);
}

```

Gambar 4.108 Script Kondisi 9 Jalur B
 Sumber ; Dokumentasi Pribadi

Sensor A1 = 6 CM	Sensor A2 = 2 CM	Sensor B1 = 9 CM	Sensor B2 = 4 CM
kondisi 9			
Sensor A1 = 6 CM	Sensor A2 = 2 CM	Sensor B1 = 10 CM	Sensor B2 = 4 CM
kondisi 9			
Sensor A1 = 6 CM	Sensor A2 = 2 CM	Sensor B1 = 6 CM	Sensor B2 = 4 CM
kondisi 9			
Sensor A1 = 3 CM	Sensor A2 = 5 CM	Sensor B1 = 5 CM	Sensor B2 = 5 CM
kondisi 9			
Sensor A1 = 4 CM	Sensor A2 = 5 CM	Sensor B1 = 5 CM	Sensor B2 = 6 CM
kondisi 9			
Sensor A1 = 3 CM	Sensor A2 = 2 CM	Sensor B1 = 2 CM	Sensor B2 = 3 CM
kondisi 9			
Sensor A1 = 3 CM	Sensor A2 = 2 CM	Sensor B1 = 2 CM	Sensor B2 = 5 CM
kondisi 9			

Gambar 4.109 Monitoring jarak objek Kondisi 9 Jalur B
 Sumber ; Dokumentasi Pribadi



Jalur B

Gambar 4.110 Kondisi 9 Jalur B
Sumber ; Dokumentasi Pribadi

Gambar 4.110 menunjukkan kondisi 9 jalur B, Pada kondisi ini Jalur B masuk dalam kondisi padat, dan durasi lampu merah pada jalur B selama 30 detik. Script dari gambar diatas:

```
else if (distance1 <= 10 && distance2 <= 10 && distance3 <= 10 && distance4
<= 10) // SENSOR A1 & A2 & B1 & B2
{
Serial.println("kondisi 9");

digitalWrite(led_R1, HIGH);
digitalWrite(led_Y1, LOW);
digitalWrite(led_G1, LOW);
delay(30000); //

digitalWrite(led_R1, LOW);
digitalWrite(led_Y1, HIGH);
digitalWrite(led_G1, LOW);
delay(2000);

digitalWrite(led_R1, LOW);
digitalWrite(led_Y1, LOW);
digitalWrite(led_G1, HIGH);
delay(30000); //

digitalWrite(led_R1, LOW);
digitalWrite(led_Y1, HIGH);
digitalWrite(led_G1, LOW);
delay(2000);
}
```

Gambar 4.111 Script Kondisi 9 Jalur B
Sumber ; Dokumentasi Pribadi

```

Sensor A1 = 6 CM   Sensor A2 = 2 CM   Sensor B1 = 9 CM   Sensor B2 = 4 CM
kondisi 9
Sensor A1 = 6 CM   Sensor A2 = 2 CM   Sensor B1 = 10 CM  Sensor B2 = 4 CM
kondisi 9
Sensor A1 = 6 CM   Sensor A2 = 2 CM   Sensor B1 = 6 CM   Sensor B2 = 4 CM
kondisi 9
Sensor A1 = 3 CM   Sensor A2 = 5 CM   Sensor B1 = 5 CM   Sensor B2 = 5 CM
kondisi 9
Sensor A1 = 4 CM   Sensor A2 = 5 CM   Sensor B1 = 5 CM   Sensor B2 = 6 CM
kondisi 9
Sensor A1 = 3 CM   Sensor A2 = 2 CM   Sensor B1 = 2 CM   Sensor B2 = 3 CM
kondisi 9
Sensor A1 = 3 CM   Sensor A2 = 2 CM   Sensor B1 = 2 CM   Sensor B2 = 5 CM
kondisi 9

```

Gambar 4.112 Monitoring jarak objek Kondisi 9 Jalur B
Sumber ; Dokumentasi Pribadi

4.2.1 Pengujian Black Box

Pengujian alat sistem kontrol lampu lalu lintas berbasis Arduino UNO menggunakan metode pengujian blackbox. Adapun yang akan diuji adalah peralatan atau komponen seperti pada tabel 4.1

Tabel IV.1 Data kendaraan

No	Kendaraan	Jenis	Ukuran
1	Mobil Besar	Mainan (container)	7 cm

Berdasarkan data panjang kendaraan pada table 4.1 dapat dikategorikan kondisi kepadatan kendaraan pada sebuah perempatan lampu merah pada tabel berikut :

T

No	Kondisi	Panjang Penumpukan Kendaraan
b ₁	Padat	7.5 cm
e ₂	Cukup Padat	15 cm

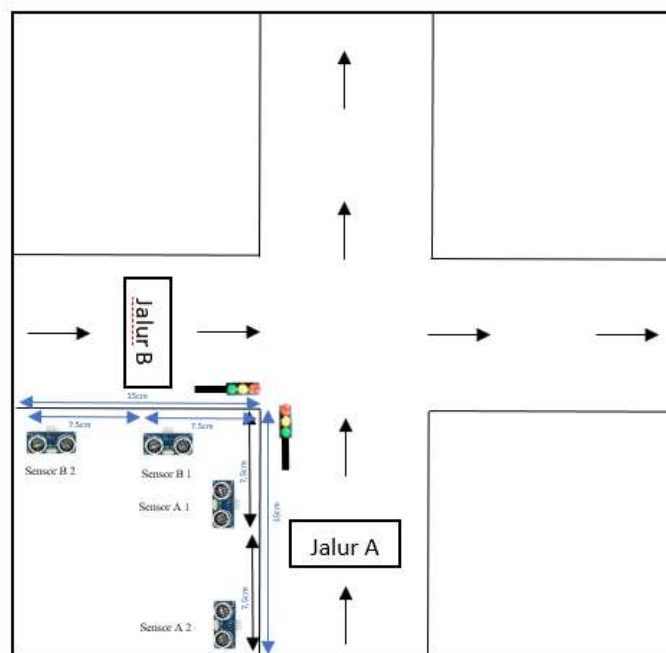
IV.2 Kondisi Kepadatan Lalu Lintas

Dari kondisi kepadatan kendaraan pada lampu lalu lintas, maka peletakkan sensor sebagai pendeteksi kendaraan masing-masing berjarak 7.5 cm. sensor 1 dipasang dengan jarak 7.5 cm, sensor 2 dipasang 15 cm dari lampu lalu lintas.

Tabel IV.3 Data Prototype

NO	Prototype	Ukuran/Jarak
1	Prototype	40x40 cm
2	Lebar Jalur A dan B	10 cm
3	Panjang jalur A dan B	15 cm
4	Jarak Antara Sensor	7.5 cm

Dari data prototype, maka dibuat prototype dengan ukuran 40x40 cm dengan lebar jalur A dan B 10 cm dan panjang jalur A dan B 15 cm dengan jarak antara sensor 7.5cm. Ilustrasi prototype dapat dilihat pada gambar berikut :



Gambar 4.113 Ilustrasi prototype
Sumber ; Dokumentasi Pribadi

Hasil uji coba prototype dibuat dengan hitungan waktu nyala pada kondisi tidak padat adalah hijau 10 detik dan merah 10 detik. Dan waktu nyala lampu lalu lintas pada kondisi cukup padat dan padat dapat dilihat pada table uji coba berikut :

Tabel IV.4 Pengujian Sistem Kerja Prototype

NO	Jalur	Sensor yang bekerja		Kondisi		Waktu Nyala Lampu	
		A1/A2	B1/B2	Cukup Padat	Padat	Hijau	Merah
Kondisi 1							
1	A	-	-	-	-	10 detik	10 detik
	B	-	-	-	-	10 detik	10 detik
Kondisi 2							
2	A	A1	-	Cukup padat	-	20 detik	10 detik
	B	-	-	-	-	10 detik	20 detik
Kondisi 3							
3	A	A1/A2	-	-	Padat	30 detik	10 detik
	B	-	-	-	-	10 detik	30 detik
Kondisi 4							
4	A	-	-	-	-	10 detik	20 detik
	B	-	B1	Cukup padat	-	20 detik	10 detik

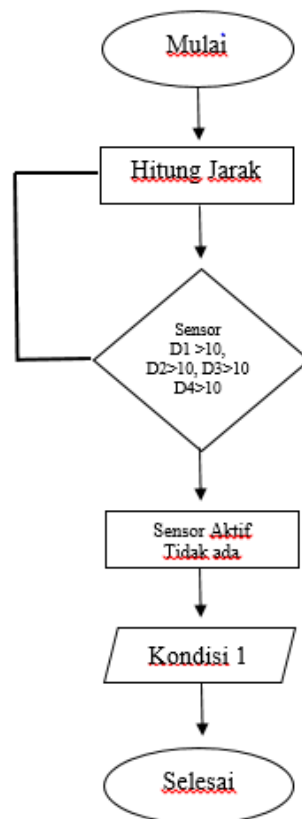
Kondisi 5							
5	A	-	-	-	-	10 detik	30 detik
	B	-	B1/B2	-	Padat	30 detik	10 detik
Kondisi 6							
6	A	A1	-	Cukup padat	-	20 detik	20 detik
	B	-	B1	Cukup padat	-	20 detik	20 detik
Kondisi 7							
7	A	A1/A2	-	-	Padat	30 detik	20 detik
	B	-	B1	Cukup padat	-	20 detik	30 detik
Kondisi 8							
8	A	A1	-	Cukup padat	-	20 detik	30 detik
	B	-	B1/B2	-	Padat	30 detik	20 detik
Kondisi 9							
9	A	A1/A2	-	-	Padat	30 detik	30 detik
	B	-	B1/B2	-	Padat	30 detik	30 detik

4.1.21 Pengujian White Box

1. Kondisi 1

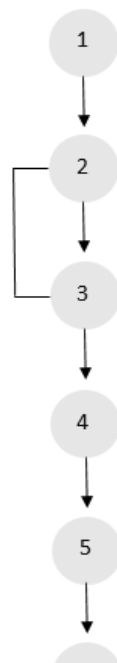
1. *Flowchart* Kondisi 1

Adapun flowchart pada kondisi 1 dapat dilihat pada gambar 4.114 sebagai berikut



2. *Flowgraph* kondisi 1

Berdasarkan flowchart pada kondisi 1 maka dapat ditentukan *flowgraph* sebagai berikut



Gambar 4.115 *Flowgraph* Kondisi 1
Sumber ; Dokumentasi Pribadi

3. Menghitung *Cyclomatic Complexity*

Hasil perhitungan *Cyclomatic Complexity* dari *flowgraph* di atas memiliki *region* = 2

1. Menghitung *Cyclomatic Complexity* dari *Edge* dan *Node*

$$\text{Rumus : } V(G) = E - N + 2$$

Dimana : E (jumlah *edge* pada *flowgraph*) = 6

N (jumlah *node* pada *flowgraph*) = 6

$$\text{Penyelesaian : } V(G) = 6 - 6 + 2$$

$$V(G) = 2$$

2. Menghitung *Cyclomatic Complexity* dari P

P adalah jumlah titik yang menyatakan logika dalam diagram alir

$$\text{Rumus : } V(G) = P + 1$$

Dimana : P = 1

$$\text{Penyelesaian : } V(G) = 1 + 1$$

$$V(G) = 2$$

4. Independent Path

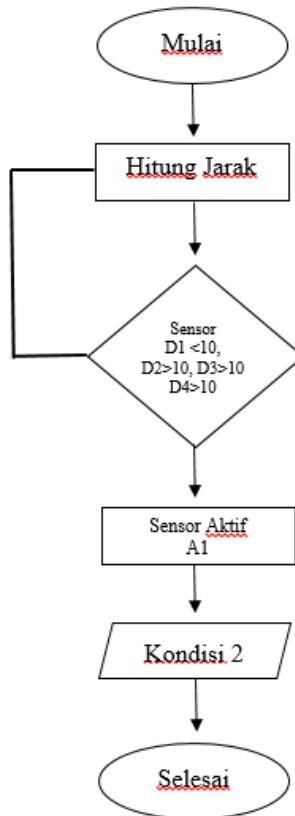
$$\text{Path 1} = 1 - 2 - 3 - 4 - 5 - 6$$

$$\text{Path 2} = 1 - 2 - 3 - 2 - 3 - 4 - 5 - 6$$

2. Kondisi 2

1. *Flowchart* Kondisi 2

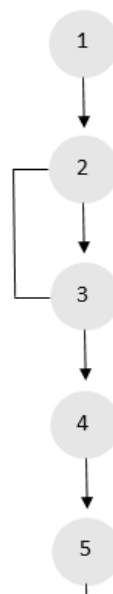
Adapun flowchart pada kondisi 1 dapat dilihat pada gambar 4.116 sebagai berikut



Gambar 4.116 Flochart Kondisi 2
Sumber ; Dokumentasi Pribadi

2. Flowgraph kondisi 2

Berdasarkan flowchart pada kondisi 2 maka dapat ditentukan *flowgraph* sebagai berikut



Gambar 4.117 *Flowgraph* Kondisi 2
Sumber ; Dokumentasi Pribadi

3. Menghitung *Cyclomatic Complexity*

Hasil perhitungan *Cyclomatic Complexity* dari *flowgraph* di atas memiliki *region* = 2

1. Menghitung *Cyclomatic Complexity* dari *Edge* dan *Node*

$$\text{Rumus : } V(G) = E - N + 2$$

$$\text{Dimana : } E \text{ (jumlah } \textit{edge} \text{ pada } \textit{flowgraph}) = 6$$

$$N \text{ (jumlah } \textit{node} \text{ pada } \textit{flowgraph}) = 6$$

$$\text{Penyelesaian : } V(G) = 6 - 6 + 2$$

$$V(G) = 2$$

2. Menghitung *Cyclomatic Complexity* dari P

P adalah jumlah titik yang menyatakan logika dalam diagram

alir

$$\text{Rumus : } V(G) = P + 1$$

$$\text{Dimana : } P = 1$$

$$\text{Penyelesaian : } V(G) = 1 + 1$$

$$V(G) = 2$$

4. Independent Path

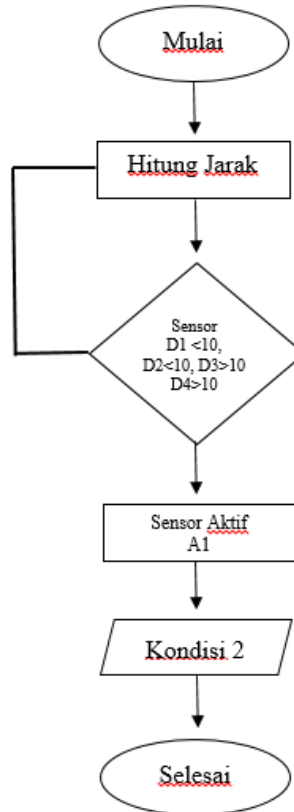
$$\text{Path 1} = 1 - 2 - 3 - 4 - 5 - 6$$

$$\text{Path 2} = 1 - 2 - 3 - 2 - 3 - 4 - 5 - 6$$

3. Kondisi 3

1. *Flowchart* Kondisi 3

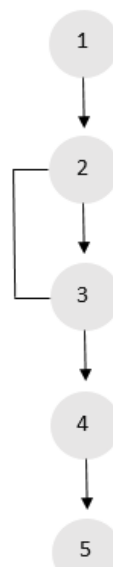
Adapun flowchart pada kondisi 3 dapat dilihat pada gambar 4.118 sebagai berikut



Gambar 4.118 *Flochart* Kondisi 3
Sumber ; Dokumentasi Pribadi

2. *Flowgraph* kondisi 3

Berdasarkan flowchart pada kondisi 3 maka dapat ditentukan *flowgraph* sebagai berikut



Gambar 4.119 *Flowgraph* Kondisi 3
Sumber ; Dokumentasi Pribadi

3. Menghitung *Cyclomatic Complexity*

Hasil perhitungan *Cyclomatic Complexity* dari *flowgraph* di atas memiliki *region* = 2

1. Menghitung *Cyclomatic Complexity* dari *Edge* dan *Node*

$$\text{Rumus : } V(G) = E - N + 2$$

$$\text{Dimana : } E \text{ (jumlah } \textit{edge} \text{ pada } \textit{flowgraph}) = 6$$

$$N \text{ (jumlah } \textit{node} \text{ pada } \textit{flowgraph}) = 6$$

$$\text{Penyelesaian : } V(G) = 6 - 6 + 2$$

$$V(G) = 2$$

2. Menghitung *Cyclomatic Complexity* dari P

P adalah jumlah titik yang menyatakan logika dalam diagram alir

$$\text{Rumus : } V(G) = P + 1$$

$$\text{Dimana : } P = 1$$

$$\text{Penyelesaian : } V(G) = 1 + 1$$

$$V(G) = 2$$

4. Independent Path

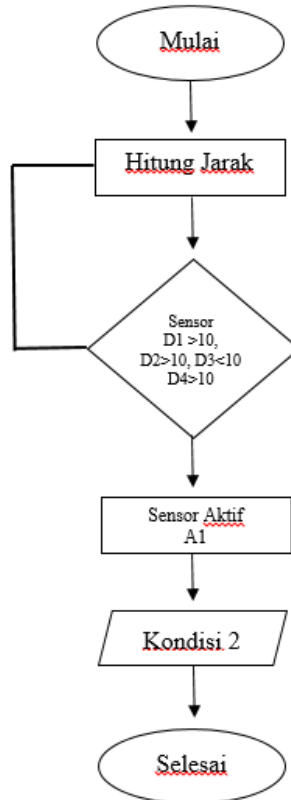
$$\text{Path 1} = 1 - 2 - 3 - 4 - 5 - 6$$

$$\text{Path 2} = 1 - 2 - 3 - 2 - 3 - 4 - 5 - 6$$

4. Kondisi 4

1. *Flowchart* Kondisi 4

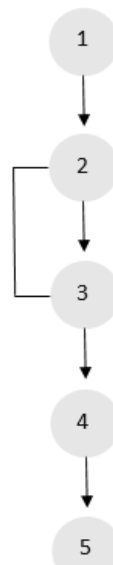
Adapun flowchart pada kondisi 4 dapat dilihat pada gambar 4.120 sebagai berikut



Gambar 4.120 Flochart Kondisi 4
Sumber ; Dokumentasi Pribadi

2. Flowgraph kondisi 4

Berdasarkan flowchart pada kondisi 4 maka dapat ditentukan *flowgraph* sebagai berikut



Gambar 4.121 *Flowgraph* Kondisi 4
Sumber ; Dokumentasi Pribadi

3. Menghitung *Cyclomatic Complexity*

Hasil perhitungan *Cyclomatic Complexity* dari *flowgraph* di atas memiliki *region* = 2

1. Menghitung *Cyclomatic Complexity* dari *Edge* dan *Node*

$$\text{Rumus : } V(G) = E - N + 2$$

$$\text{Dimana : } E \text{ (jumlah } \textit{edge} \text{ pada } \textit{flowgraph}) = 6$$

$$N \text{ (jumlah } \textit{node} \text{ pada } \textit{flowgraph}) = 6$$

$$\text{Penyelesaian : } V(G) = 6 - 6 + 2$$

$$V(G) = 2$$

2. Menghitung *Cyclomatic Complexity* dari P

P adalah jumlah titik yang menyatakan logika dalam diagram alir

$$\text{Rumus : } V(G) = P + 1$$

$$\text{Dimana : } P = 1$$

$$\text{Penyelesaian : } V(G) = 1 + 1$$

$$V(G) = 2$$

4. Independent Path

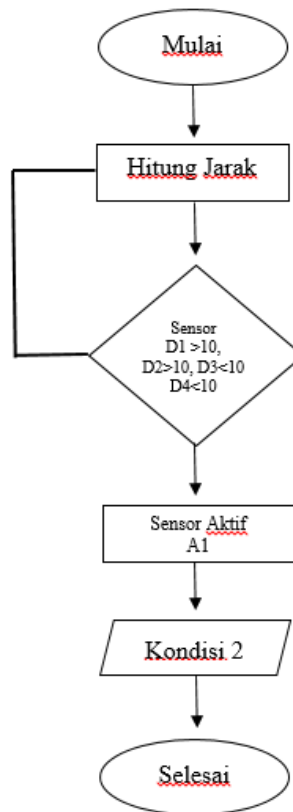
$$\text{Path 1} = 1 - 2 - 3 - 4 - 5 - 6$$

$$\text{Path 2} = 1 - 2 - 3 - 2 - 3 - 4 - 5 - 6$$

5. Kondisi 5

1. *Flowchart* Kondisi 5

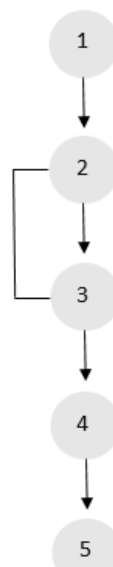
Adapun flowchart pada kondisi 5 dapat dilihat pada gambar 4.122 sebagai berikut



Gambar 4.122 Flochart Kondisi 5
Sumber ; Dokumentasi Pribadi

2. Flowgraph kondisi 5

Berdasarkan flowchart pada kondisi 5 maka dapat ditentukan *flowgraph* sebagai berikut



Gambar 4.123 *Flowgraph* Kondisi 5
Sumber ; Dokumentasi Pribadi

3. Menghitung *Cyclomatic Complexity*

Hasil perhitungan *Cyclomatic Complexity* dari *flowgraph* di atas memiliki *region* = 2

3. Menghitung *Cyclomatic Complexity* dari *Edge* dan *Node*

$$\text{Rumus : } V(G) = E - N + 2$$

$$\text{Dimana : } E \text{ (jumlah } \textit{edge} \text{ pada } \textit{flowgraph}) = 6$$

$$N \text{ (jumlah } \textit{node} \text{ pada } \textit{flowgraph}) = 6$$

$$\text{Penyelesaian : } V(G) = 6 - 6 + 2$$

$$V(G) = 2$$

4. Menghitung *Cyclomatic Complexity* dari P

P adalah jumlah titik yang menyatakan logika dalam diagram alir

$$\text{Rumus : } V(G) = P + 1$$

$$\text{Dimana : } P = 1$$

$$\text{Penyelesaian : } V(G) = 1 + 1$$

$$V(G) = 2$$

4. Independent Path

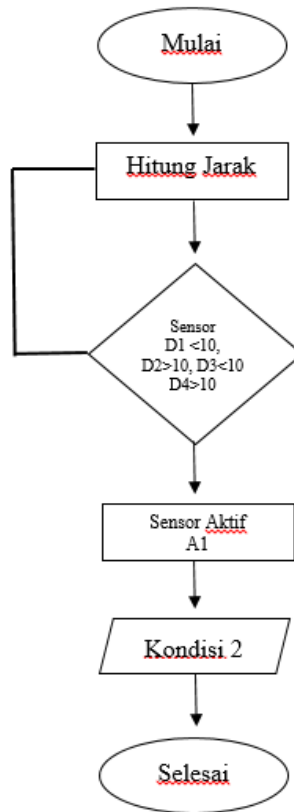
$$\text{Path 1} = 1 - 2 - 3 - 4 - 5 - 6$$

$$\text{Path 2} = 1 - 2 - 3 - 2 - 3 - 4 - 5 - 6$$

6. Kondisi 6

1. *Flowchart* Kondisi 6

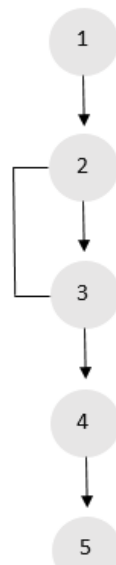
Adapun flowchart pada kondisi 6 dapat dilihat pada gambar 4.124 sebagai berikut



Gambar 4.124 Flochart Kondisi 6
Sumber ; Dokumentasi Pribadi

2. Flowgraph kondisi 6

Berdasarkan flowchart pada kondisi 6 maka dapat ditentukan *flowgraph* sebagai berikut



Gambar 4.125 *Flowgraph* Kondisi 6
Sumber ; Dokumentasi Pribadi

3. Menghitung *Cyclomatic Complexity*

Hasil perhitungan *Cyclomatic Complexity* dari *flowgraph* di atas memiliki *region* = 2

1. Menghitung *Cyclomatic Complexity* dari *Edge* dan *Node*

$$\text{Rumus : } V(G) = E - N + 2$$

$$\text{Dimana : } E \text{ (jumlah } \textit{edge} \text{ pada } \textit{flowgraph}) = 6$$

$$N \text{ (jumlah } \textit{node} \text{ pada } \textit{flowgraph}) = 6$$

$$\text{Penyelesaian : } V(G) = 6 - 6 + 2$$

$$V(G) = 2$$

2. Menghitung *Cyclomatic Complexity* dari P

P adalah jumlah titik yang menyatakan logika dalam diagram alir

$$\text{Rumus : } V(G) = P + 1$$

$$\text{Dimana : } P = 1$$

$$\text{Penyelesaian : } V(G) = 1 + 1$$

$$V(G) = 2$$

4. Independent Path

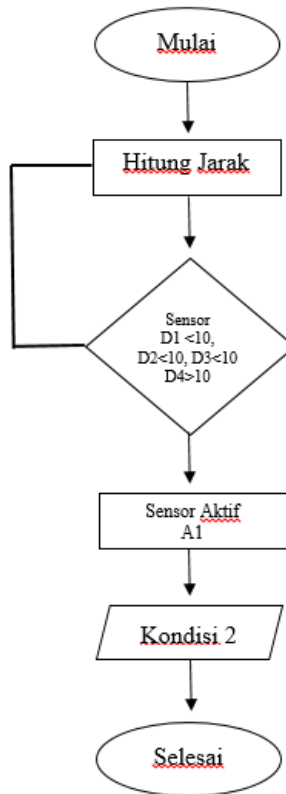
$$\text{Path 1} = 1 - 2 - 3 - 4 - 5 - 6$$

$$\text{Path 2} = 1 - 2 - 3 - 2 - 3 - 4 - 5 - 6$$

7. Kondisi 7

1. *Flowchart* Kondisi 7

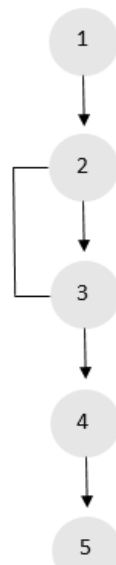
Adapun flowchart pada kondisi 7 dapat dilihat pada gambar 4.126 sebagai berikut



Gambar 4.126 Flochart Kondisi 7
Sumber ; Dokumentasi Pribadi

2. Flowgraph kondisi 7

Berdasarkan flowchart pada kondisi 7 maka dapat ditentukan *flowgraph* sebagai berikut



Gambar 4.127 *Flowgraph* Kondisi 7
Sumber ; Dokumentasi Pribadi

3. Menghitung *Cyclomatic Complexity*

Hasil perhitungan *Cyclomatic Complexity* dari *flowgraph* di atas memiliki *region* = 2

1. Menghitung *Cyclomatic Complexity* dari *Edge* dan *Node*

$$\text{Rumus : } V(G) = E - N + 2$$

$$\text{Dimana : } E \text{ (jumlah } \textit{edge} \text{ pada } \textit{flowgraph}) = 6$$

$$N \text{ (jumlah } \textit{node} \text{ pada } \textit{flowgraph}) = 6$$

$$\text{Penyelesaian : } V(G) = 6 - 6 + 2$$

$$V(G) = 2$$

2. Menghitung *Cyclomatic Complexity* dari P

P adalah jumlah titik yang menyatakan logika dalam diagram alir

$$\text{Rumus : } V(G) = P + 1$$

$$\text{Dimana : } P = 1$$

$$\text{Penyelesaian : } V(G) = 1 + 1$$

$$V(G) = 2$$

4. Independent Path

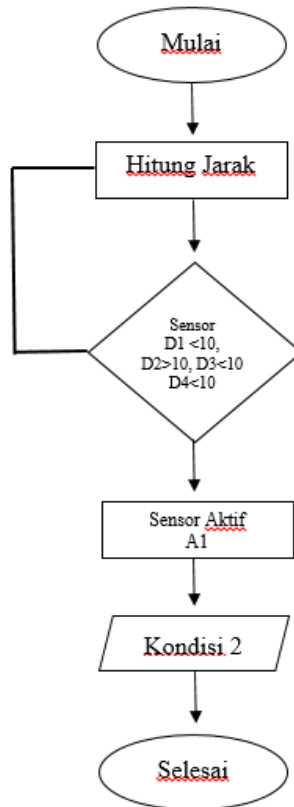
$$\text{Path 1} = 1 - 2 - 3 - 4 - 5 - 6$$

$$\text{Path 2} = 1 - 2 - 3 - 2 - 3 - 4 - 5 - 6$$

8. Kondisi 8

1. *Flowchart* Kondisi 8

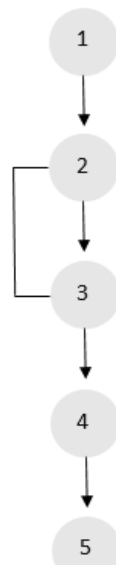
Adapun flowchart pada kondisi 8 dapat dilihat pada gambar 4.128 sebagai berikut



Gambar 4.128 Flochart Kondisi 8
Sumber ; Dokumentasi Pribadi

2. Flowgraph kondisi 8

Berdasarkan flowchart pada kondisi 8 maka dapat ditentukan *flowgraph* sebagai berikut



Gambar 4.129 *Flowgraph* Kondisi 8
Sumber ; Dokumentasi Pribadi

3. Menghitung *Cyclomatic Complexity*

Hasil perhitungan *Cyclomatic Complexity* dari *flowgraph* di atas memiliki *region* = 2

1. Menghitung *Cyclomatic Complexity* dari *Edge* dan *Node*

$$\text{Rumus : } V(G) = E - N + 2$$

$$\text{Dimana : } E \text{ (jumlah } \textit{edge} \text{ pada } \textit{flowgraph}) = 6$$

$$N \text{ (jumlah } \textit{node} \text{ pada } \textit{flowgraph}) = 6$$

$$\text{Penyelesaian : } V(G) = 6 - 6 + 2$$

$$V(G) = 2$$

2. Menghitung *Cyclomatic Complexity* dari P

P adalah jumlah titik yang menyatakan logika dalam diagram alir

$$\text{Rumus : } V(G) = P + 1$$

$$\text{Dimana : } P = 1$$

$$\text{Penyelesaian : } V(G) = 1 + 1$$

$$V(G) = 2$$

4. Independent Path

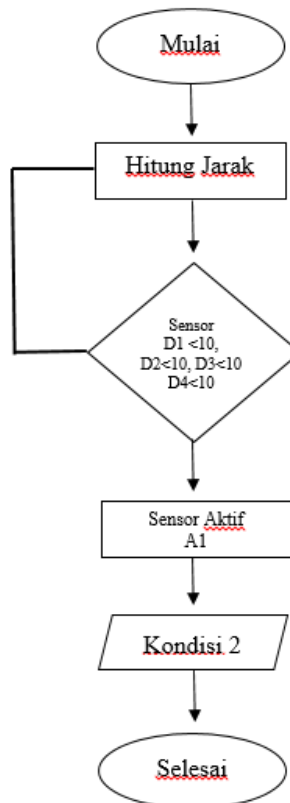
$$\text{Path 1} = 1 - 2 - 3 - 4 - 5 - 6$$

$$\text{Path 2} = 1 - 2 - 3 - 2 - 3 - 4 - 5 - 6$$

9. Kondisi 9

1. *Flowchart* Kondisi 9

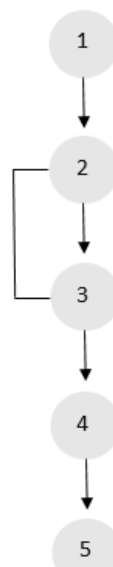
Adapun flowchart pada kondisi 9 dapat dilihat pada gambar 4.130 sebagai berikut



Gambar 4.130 Flochart Kondisi 9
Sumber ; Dokumentasi Pribadi

2. Flowgraph kondisi 9

Berdasarkan flowchart pada kondisi 9 maka dapat ditentukan *flowgraph* sebagai berikut



Gambar 4.131 *Flowgraph* Kondisi 9
Sumber ; Dokumentasi Pribadi

3. Menghitung *Cyclomatic Complexity*

Hasil perhitungan *Cyclomatic Complexity* dari *flowgraph* di atas memiliki *region* = 2

1. Menghitung *Cyclomatic Complexity* dari *Edge* dan *Node*

$$\text{Rumus : } V(G) = E - N + 2$$

$$\text{Dimana : } E \text{ (jumlah } \textit{edge} \text{ pada } \textit{flowgraph}) = 6$$

$$N \text{ (jumlah } \textit{node} \text{ pada } \textit{flowgraph}) = 6$$

$$\text{Penyelesaian : } V(G) = 6 - 6 + 2$$

$$V(G) = 2$$

2. Menghitung *Cyclomatic Complexity* dari P

P adalah jumlah titik yang menyatakan logika dalam diagram alir

$$\text{Rumus : } V(G) = P + 1$$

$$\text{Dimana : } P = 1$$

$$\text{Penyelesaian : } V(G) = 1 + 1$$

$$V(G) = 2$$

4. Independent Path

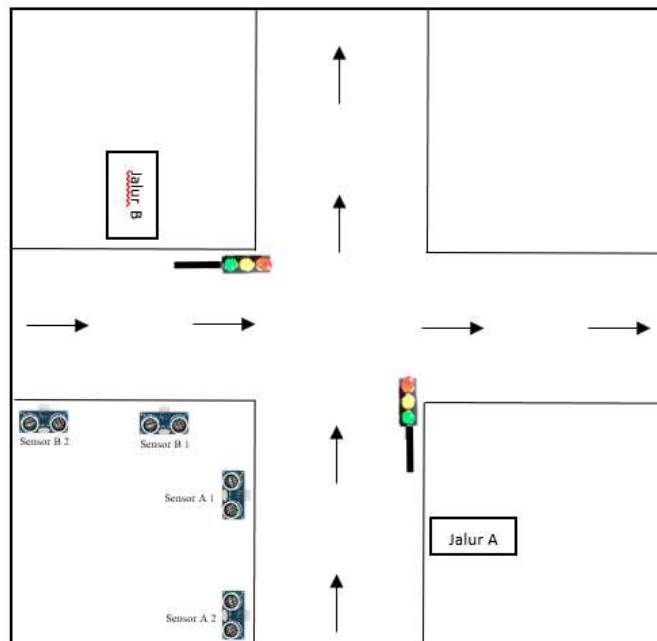
$$\text{Path 1} = 1 - 2 - 3 - 4 - 5 - 6$$

$$\text{Path 2} = 1 - 2 - 3 - 2 - 3 - 4 - 5 - 6$$

4.2 Pembahasan

Pada penelitian ini prototype sistem kontrol lampu lalu lintas berbasis arduino uno ini dapat mengurangi kepadatan kendaraan pada lampu lalu lintas dan penggunaan lampu lalu lintas menjadi lebih efektif. Pada pengujian prototype perempatan 2 jalur ini terdapat 4 sensor, tiap jalur dipasangkan 2 sensor, dan terdapat 9 kondisi.

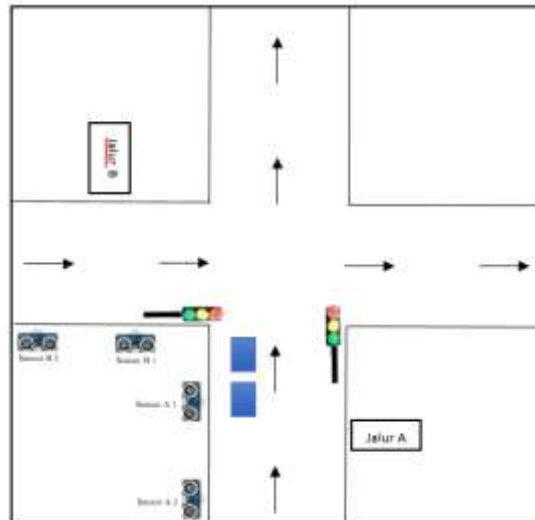
1. Kondisi 1 : Pada kondisi ini masing-masing jalur berada pada kondisi sepi, dan semua sensor kurang intens, maka waktu nyala lampu jalur A hijau 10 detik pada jalur B merah 10 detik, dan lampu merah jalur A 10 detik dan hijau jalur B 10 detik.



Gambar 4.132 Ilustrasi prototype kondisi 1
Sumber ; Dokumentasi Pribadi

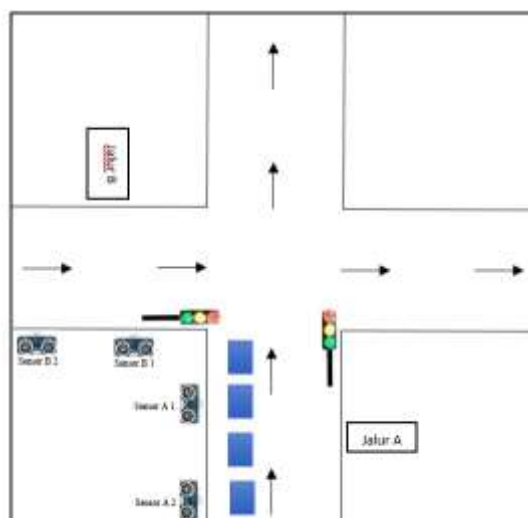
2. Kondisi 2 : Pada kondisi ini jalur A berada pada kondisi cukup padat, dan sensor A1 intens, sensor A2, B1, B2 kurang intens, maka waktu nyala lampu

3. jalur A hijau 20 detik pada jalur B merah 20 detik, dan lampu merah jalur A 10 detik dan hijau jalur B 10 detik.



Gambar 4.133 Ilustrasi prototype kondisi 2
Sumber ; Dokumentasi Pribadi

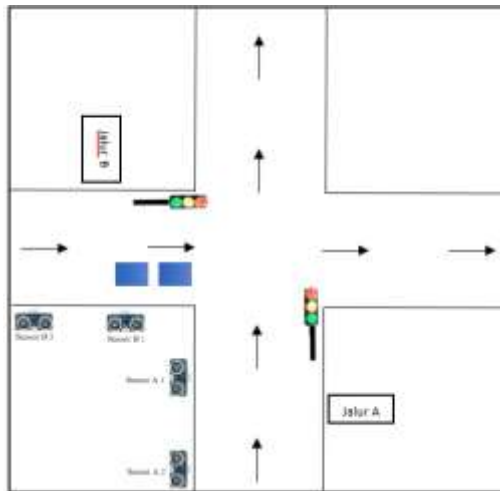
4. Kondisi 3 : Pada kondisi ini jalur A berada pada kondisi padat, dan sensor A1 dan A2 intens, sensor B1 dan B2 kurang intens, maka waktu nyala lampu jalur A hijau 30 detik pada jalur B merah 30 detik, dan lampu merah jalur A 10 detik dan hijau jalur B 10 detik.



Gambar 4.134 Ilustrasi prototype kondisi 3
Sumber ; Dokumentasi Pribadi

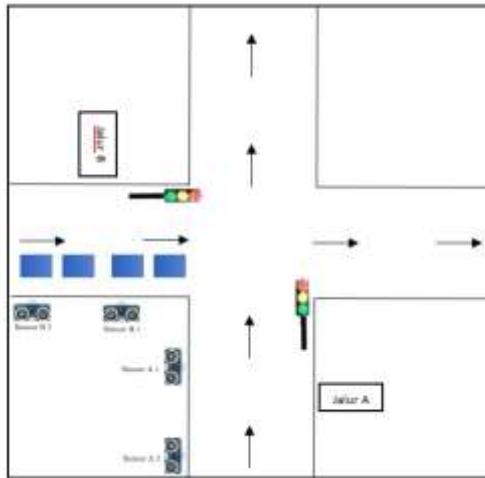
Gambar 4.134 Ilustrasi prototype kondisi 3
Sumber ; Dokumentasi Pribadi

5. Kondisi 4 : Pada kondisi ini jalur B berada pada kondisi cukup padat, dan sensor B1 intens, sensor B2, A1, A2 kurang intens, maka waktu nyala lampu jalur B hijau 20 detik pada jalur A merah 20 detik, dan lampu merah jalur B 10 detik dan hijau jalur A 10 detik.



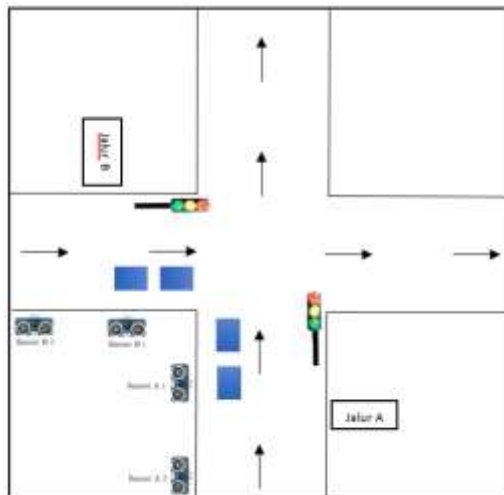
Gambar 4.135 Ilustrasi prototype kondisi 4
Sumber ; Dokumentasi Pribadi

5. Kondisi 5 : Pada kondisi ini jalur B berada pada kondisi padat, dan sensor B1 dan B2 intens, sensor A1 dan A2 kurang intens, maka waktu nyala lampu jalur B hijau 30 detik pada jalur A merah 30 detik, dan lampu merah jalur B 10 detik dan hijau jalur A 10 detik.



Gambar 4.136 Ilustrasi prototype kondisi 5
Sumber ; Dokumentasi Pribadi

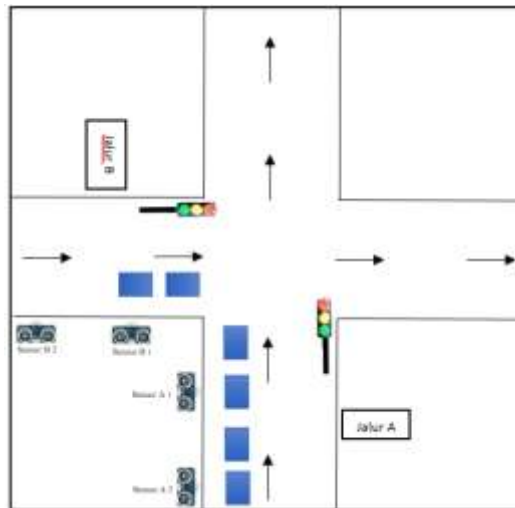
6. Kondisi 6 : Pada kondisi ini jalur A dan B berada pada kondisi cukup padat, dan sensor A1 dan B1 intens, sensor A2 dan B2 kurang intens, maka waktu nyala lampu jalur A hijau 20 detik pada jalur B merah 20 detik, dan lampu merah jalur A 20 detik dan hijau jalur B 20 detik.



Gambar 4.137 Ilustrasi prototype kondisi 6
Sumber ; Dokumentasi Pribadi

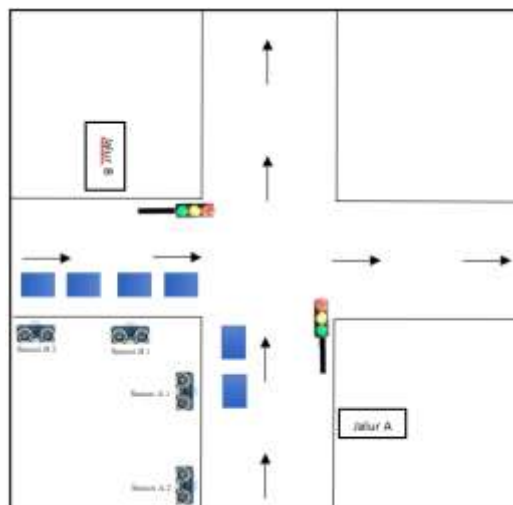
7. Kondisi 7 : Pada kondisi ini jalur A berada pada kondisi padat, jalur B berada pada kondisi cukup padat, dan sensor A1, A2, dan B1 intens, sensor

B2 kurang intens, maka waktu nyala lampu jalur A hijau 30 detik pada jalur B merah 30 detik, dan lampu merah jalur A 20 detik dan hijau jalur B 20 detik.



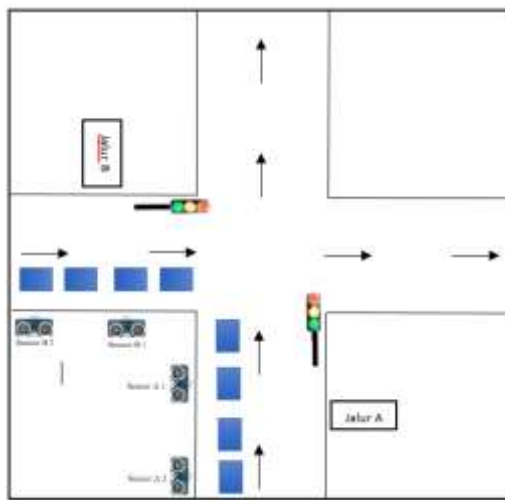
Gambar 4.138 Ilustrasi prototype kondisi 7
Sumber ; Dokumentasi Pribadi

8. Kondisi 8 : Pada kondisi ini jalur A berada pada kondisi cukup padat, jalur B berada pada kondisi padat, dan sensor A1, B1, dan B2 intens, sensor A2 kurang intens, maka waktu nyala lampu jalur B hijau 30 detik pada jalur A merah 30 detik, dan lampu merah jalur B 20 detik dan hijau jalur A 20 detik.



Gambar 4.139 Ilustrasi prototype kondisi 8
 Sumber ; Dokumentasi Pribadi

9. Kondisi 9 : Pada kondisi ini jalur A dan B berada pada kondisi padat, dan sensor A1, A2, B1, dan B2 intens, maka waktu nyala lampu jalur A hijau 30 detik pada jalur B merah 30 detik, dan lampu merah jalur A 30 detik dan hijau jalur B 30 detik.



Gambar 4.140 Ilustrasi prototype kondisi 9
 Sumber ; Dokumentasi Pribadi

Tabel IV.5 Rangkuman ilustras

Waktu Menyala Lampu							
Kondisi	State	Jalur A			Jalur B		
		Merah	Kuning	Hijau	Merah	Kuning	Hijau
1 (Sepi jalur A dan B)	Awal	-	-	10 detik	10 detik	-	-
	Peralihan	-	2 detik	-	-	2 detik	-
	Akhir	10 detik	-	-	-	-	10 detik

2 (Padat jalur A, jalur B Sepi)	Awal	-	-	20 detik	20 detik	-	-
	Peralihan	-	2 detik	-	-	2 detik	-
	Akhir	10 detik	-	-	-	-	10 detik
3 (Cukup padat jalur A, jalur B Sepi)	Awal	-	-	30 detik	30 detik	-	-
	Peralihan	-	2 detik	-	-	2 detik	-
	Akhir	10 detik	-	-	-	-	10 detik
4 (Padat jalur B, jalur A Sepi)	Awal	-	-	10 detik	10 detik	-	-
	Peralihan	-	2 detik	-	-	2 detik	-
	Akhir	20 detik	-	-	-	-	20 detik
5 (Cukup padat jalur B, jalur A Sepi)	Awal	-	-	10 detik	10 detik	-	-
	Peralihan	-	2 detik	-	-	2 detik	-
	Akhir	30 detik	-	-	-	-	30 detik
6 (Padat jalur A dan B)	Awal	-	-	20 detik	20 detik	-	-
	Peralihan	-	2 detik	-	-	2 detik	-
	Akhir	20 detik	-	-	-	-	20 detik
7 (Cukup padat jalur A, jalur B padat)	Awal	-	-	30 detik	30 detik	-	-
	Peralihan	-	2 detik	-	-	2 detik	-
	Akhir	20 detik	-	-	-	-	20 detik

8 (Cukup padat jalur B, jalur A padat)	Awal	-	-	20 detik	20 detik	-	-
	Peralihan	-	2 detik	-	-	2 detik	-
	Akhir	30 detik	-	-	-	-	30 detik
9 (Cukup padat jalur A dan B)	Awal	-	-	30 detik	30 detik	-	-
	Peralihan	-	2 detik	-	-	2 detik	-
	Akhir	30 detik	-	-	-	-	30 detik

Berdasarkan hasil yang telah didapatkan dalam perancangan prototype sistem kontrol lampu lalu lintas berbasis arduino uno ini telah dibuat sesuai dengan perancangan-perancangan dan pengujian yang telah dilakukan sesuai dengan yang direncanakan sebelumnya.

BAB V

PENUTUP

5.1 Kesimpulan

Berdasarkan hasil penelitian yang telah dilakukan dapat disimpulkan bahwa:

1. Alat ini dibuat berdasarkan panjang antrian kendaraan dimasing-masing jalur menggunakan sensor ultrasonik.
2. Sebagai alat untuk menentukan tingkat kepadatan kendaraan pada tiap jalur dalam dua kategori yaitu cukup padat dan padat.
3. Membuat lampu lalu lintas dengan pengaturan lama waktu lampu menyala berdasarkan tingkat kepadatan kendaraan.

5.2 Saran

Prototype sistem kontrol lampu lalu lintas untuk pengendalian kemacetan ini masih jauh dari kesempurnaan. Oleh karena itu, perlu adanya pengembangan lebih lanjut yang mampu memperbaiki segala kekurangan yang ada pada sistem yang telah dibuat. Adapun saran-saran yang dapat diberikan, sebagai berikut :

1. Sebaiknya selain menggunakan sensor dapat dikembangkan dengan menggunakan alat yang mampu membedakan objek (motor dan mobil) agar nyala lampu lalu lintas lebih realtime.
2. Sebaiknya menggunakan sensor ditiap sisi jalan agar akurasi sensor lebih efektif untuk membaca objek.
3. Sistem dapat dikembangkan dengan menggunakan open CV untuk proses sistem.
4. Diharapkan pada penelitian selanjutnya dapat menggunakan kamera untuk mendeteksi objek.

DAFTAR PUSTAKA

- Darmawan, S. (2021). PERANCANGAN UI/UX PADA PROTOTYPE APLIKASI MOBILE PEMBELAJARAN ONLINE SMARTER. In *Science and Engineering National Seminar* (Vol. 6, No. 1, pp. 441-448).
- Deva Okky Deltania, Djuniadi, Esa Apriaskar. (2021). Pengaturan Lampu Lalu Lintas (Traffic Light) Dengan Sensor Ultrasonik, Volume 19, No.1, Agustus 2021 P-ISSN 1412-0372, E-ISSN 2541-089X
- DEWI, Luh Joni Erawati. (2021) Media Pembelajaran Bahasa Pemrograman C++. *Jurnal Pendidikan Teknologi dan Kejuruan*.
- Huda, N., & Khamami, F. (2017). Modifikasi Sistem Kendali Sepeda Listrik Listrik Hybrid. *Cahaya Bagaskara: Jurnal Ilmiah Teknik Elektronika*, 1(1), 30-35.
- IGAP. Raka Agung (2009). Imulator Pengaturan Lampu Lalu Lintas Berdasarkan Waktu Dan Kepadatan Kendaraan Berbasis Mikrokontroler At89s52, Vol.8 No.2 Juli - Desember 2009
- Manto. Perangkat Pengatur Timer Lampu Lalu Lintas Berdasarkan Antrian Kendaraan.
- Muchlisin Riadi. (2016). Pengolahan Citra Digital. Pengolahan citra (*image Processing*) merupakan proses mengolah piksel-piksel di dalam citra digital untuk tujuan tertentu. 21 Apr, 2016
- Pamor Gunoto, M. Irsyam dan Toni Kusuma Wijaya (2016) Pengembangan Sistem Traffic Lights berdasarkan Kepadatan Kendaraan Menggunakan PLC
- Rahmat Hidayat. (2014). Prototipe sistem control lampu lalu lintas otomatis menggunakan LDR sebagai sensor kepadatan kendaraan berbasis mikrokontroler Atmega16, 2014
- Roria M Kristina Sitorus. (2015). Sistem Kontrol Traffic Light Berdasarkan Jumlah Kendaraan Untuk Mengurangi Kemacetan Berbasis Arduino. Juli 2015.
- Setiawan, A., & Dewi, R. P. (2018). Simulasi Mikrokontroler Pengukur Jarak Berbasis Arduino Uno Sebagai Media Pembelajaran Mahasiswa Diii Teknik Elektronika Politeknik Harapan Bersama Tegal. *Power Elektronik: Jurnal Orang Elektro*,

- Windarto, Muhammad Haekal. (2012). Aplikasi Pengaturan Lampu Lalu Lintas Berbasis Arduino Mega 2560 Menggunakan Light Dependent Resistor (LDR) dan laser, Vol. 3 No. 2 Desember 2012.
- Menurut Sarif Hidayatullah (2020). Sistem pendeteksi objek pada area *blind spot dump truck* menggunakan fuzzy logic dengan metode sugeno (Bachelor's thesis, Fakultas Sains dan Teknologi Universitas Islam Negeri Syarif Hidayatullah Jakarta).
- Sumarno, M. D. A., & Setyawan, R. T. (2021). Pengaruh Rain Sensor FR-04 Terhadap Wiper Otomatis Berbasis Mikrokontroler.