

LAPORAN TESIS

Prototipe *Smart Traffic Light* (STL) berdasar Panjang Antrian menggunakan *Internet of Things* (IoT)

**Jihan Athira Ramadhani
NIM. 2256102004**

**DOSEN PEMBIMBING
Dr. Agus Urip Ari Wibowo, S.T., M.T.
Muhammad Diono, S.S.T., M.T.**

**PROGRAM STUDI
MAGISTER TERAPAN TEKNIK KOMPUTER
POLITEKNIK CALTEX RIAU
2025**

TESIS

**Prototipe *Smart Traffic Light* (STL) berdasar
Panjang Antrian menggunakan *Internet of Things*
(*IoT*)**

JIHAN ATHIRA RAMADHANI
NIM. 2256102004

DOSEN PEMBIMBING
Dr. Agus Urip Ari Wibowo, S.T., M.T.
Muhammad Diono, S.S.T., M.T.

PROGRAM STUDI
MAGISTER TERAPAN TEKNIK KOMPUTER
POLITEKNIK CALTEX RIAU
2025

HALAMAN PENGESAHAN

Prototipe *Smart Traffic Light* (STL) berdasar Panjang Antrian menggunakan *Internet of Things* (IoT)

Oleh:

Jihan Athira Ramadhani

NRP. 2256102004

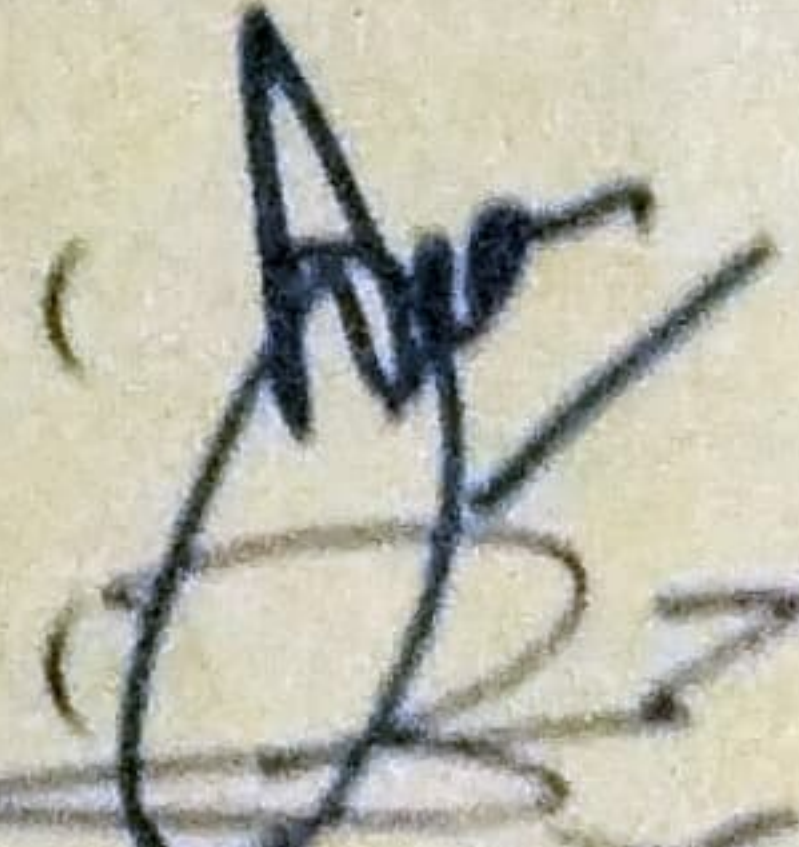
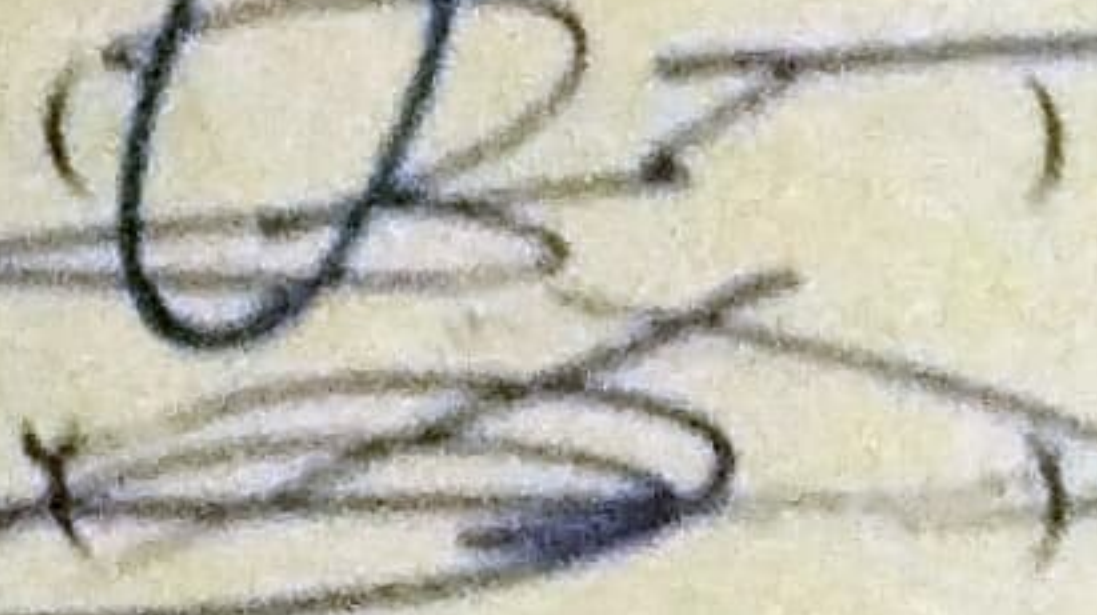
Tesis ini digunakan sebagai salah satu syarat untuk memperoleh Gelar Magister Terapan (M.Tr.Kom.)

di

Politeknik Caltex Riau

2025

Disetujui oleh:

Pembimbing Utama	:	<u>Dr. Agus Urip Ari Wibowo, S.T., M.T.</u>	()
		NIP. 007001	
Pembimbing Pendamping	:	<u>Muhammad Diono, S.S.T., M.T.</u>	()
		NIP. 128808	
Penguji	:	<u>Dr. Eng. Yoanda Alim Syahbana, ST.</u>	()
		<u>M.Sc NIP. 148809</u>	
Penguji	:	<u>Dr. Emansa Hasri Putra S.T., M.Eng.</u>	()
		NIP. 017501	
Penguji	:	<u>Wawan Yunanto S.Kom., M.T.</u>	()
		NIP. 037802	

Mengetahui,

Ketua Program Studi Magister Terapan Teknik Komputer
Politeknik Caltex Riau


Dr. Eng. Yoanda Alim Syahbana, ST., M.Sc.
PASCASARJANA
NIP. 148809

HALAMAN PERNYATAAN ORISINALITAS

Dengan ini saya menyatakan bahwa bagian atau keseluruhan tesis ini:

1. adalah hasil karya sendiri dan tidak mengandung unsur plagiat dari pihak lain
2. tidak pernah diajukan untuk mendapatkan gelar akademis pada suatu Perguruan Tinggi
3. tidak pernah dipublikasikan atau ditulis oleh pihak lain
4. mencantumkan rujukan dan kutipan dengan jujur dan benar terhadap sumber referensi lain yang menunjang pembahasan pada tesis.

Apabila ditemukan bukti bahwa pernyataan saya diatas tidak benar, maka saya bersedia menerima sanksi sesuai dengan ketentuan yang berlaku di Politeknik Caltex Riau.

Pekanbaru, 13 Oktober 2025

Yang menyatakan,

Jihan Athira Ramadhani

HALAMAN KESEPAKATAN PUBLIKASI

Demi pengembangan ilmu pengetahuan, dengan ini saya menyatakan:

1. memberikan persetujuan kepada Politeknik Caltex Riau untuk menyimpan, mengolah dalam bentuk pangkalan data, merawat, mengalih media/formatkan dan mempublikasikan tesis ini selama tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai pemilik Hak Cipta.
2. tidak melakukan alih media/format dan publikasi dalam bentuk makalah ilmiah dari bagian atau keseluruhan tesis ini ke suatu publikasi ilmiah, pada seminar ataupun jurnal, skala nasional maupun internasional, kecuali ada persetujuan dari saya dan Dosen Pembimbing Utama, dan mencantumkan nama saya, Dosen Pembimbing Utama dan nama-nama lain (jika ada) yang berkontribusi pada makalah.

Pekanbaru, 13 Oktober 2025

Yang menyatakan,

Jihan Athira Ramadhani

KATA PENGANTAR

Segala puji syukur kepada Allah Tuhan yang Maha Pencipta yang telah mulimpahkan rahmat dan hidayahnya kepada kita sebagai hambanya, serta Sholawat salam atas Nabi Muhammad Shallallahu Alaihi Wassalam. Penulis dapat menyelesaikan tesis yang berjudul *Prototipe Smart Traffic Light (STL)* berdasar Panjang Antrian menggunakan *Internet of Things (IoT)*. Tesis ini disusun untuk memenuhi salah satu syarat menyelesaikan pendidikan jenjang Magister Terapan pada program studi Teknik Komputer di Politeknik Caltex Riau. Pada kesempatan ini, penulis ingin mengucapkan terima kasih yang sebesar-besarnya kepada pihak yang telah banyak memberikan bantuan dan dukungan yang tiada terhingga baik secara langsung maupun tidak langsung. Ucapan terima kasih ditujukan kepada :

1. Kedua orang tua dan seluruh keluarga besar penulis atas dukungan dan kasih sayang yang tidak terhingga sehingga penulis mampu menyelesaikan tesis ini dengan tepat waktu.
2. Pak Dr. Dadang Syarif Sihabudin, S.Si., M.Sc. selaku Direktur Politeknik Caltex Riau.
3. Pak Dr. Eng. Yoanda Alim Syahbana, S.T., M.Sc. selaku ketua program studi Magister Terapan Teknik Komputer dan penguji yang telah memberikan masukan dan saran dalam menyelesaikan tesis ini.
4. Pak Dr. Agus Urip Ari Wibowo, S.T., M.T. selaku Dosen Pembimbing I dan pak Muhammad Diono, S.S.T., M.T. selaku Dosen Pembimbing II penulis yang telah banyak memberikan bimbingan dan arahan serta bantuan untuk menyelesaikan tesis ini dengan penuh kesabaran.
5. Pak Dr. Emansa Hasri Putra S.T., M.Eng dan pak Wawan Yunanto S.Kom., M.T selaku dosen penguji yang telah memberikan masukan dan saran dalam menyelesaikan tesis.
6. Seluruh dosen khususnya Program Studi Magister Terapan Teknik Komputer dan seluruh dosen di Politeknik Caltex Riau

yang telah memberikan bekal ilmu kepada penulis dalam menyelesaikan penulisan tesis.

7. Teman seperjuangan angkatan 22MTTKA Program Studi Magister Terapan Teknik Komputer yang telah memberikan dukungan serta masukan dalam penulisan tesis ini.
8. Dan semua pihak yang tidak bisa penulis sebutkan satu persatu.

Dalam penyusunan tesis ini penulis masih menyadari sepenuhnya bahwa laporan tesis ini jauh dari kata sempurna, oleh karena itu segala jenis kritik, saran dan masukan yang membangun diharapkan agar berguna dan memberikan wawasan untuk penulis sendiri. Akhir kata agar tesis ini berguna untuk semua pihak yang berkepentingan.

Pekanbaru, 13 Oktober 2025

Yang Menyatakan,

Jihan Athira Ramadhani

ABSTRAK

Sebagai upaya dalam pengaturan lalu lintas digunakanlah Alat Pengatur Instruksi Lalu lintas (APIL) yang berupa *Traffic Light*. *Traffic Light* dipasang di berbagai jenis persimpangan jalan maupun sarana penyeberangan. Fungsi lampu lalu lintas sangatlah penting sehingga lampu lalu lintas harus dapat dikontrol semudah dan seefisien mungkin guna memperlancar arus lalu lintas di suatu persimpangan jalan. Namun setelah digunakan *Traffic Light* masih terdapat kepadatan yang diakibatkan tidak serentak nya arah datang kendaraan dari masing-masing jalur. Hal ini mengakibatkan terjadinya antrian panjang di salah satu jalur *Traffic Light* yang membuat antrian semakin panjang yang diakibatkan tidak efektifnya lama lampu merah dari *Traffic Light* yang ada. Maka dari itu peneliti ingin mendesain suatu alat menggunakan 8 sensor IR sebagai pendeteksi antrian di masing-masing jalur, kemudian menyatukan dengan sistem *Traffic Light* otomatis dapat dioptimasi lampu merah dan lampu hijau sesuai dengan panjang antrian yang ada. Hasil eksperimen menemukan bahwa pada kondisi sepi atau tidak ada sensor yang mendeteksi kendaraan maka waktu lampu hijau nya yaitu 20 detik. Saat kondisi kemacetan 1 atau sensor IR depan mendeteksi kendaraan, maka lampu hijau nya bertambah 5 detik menjadi 25 detik. Saat kondisi kemacetan 2 atau kedua sensor IR mendeteksi kendaraan, maka lampu hijau nya bertambah 10 detik menjadi 30 detik.

Kata kunci

Traffic Light, IR, kemacetan, antrian, APIL.

ABSTRACT

As an effort to regulate traffic, a Traffic Instruction Control Device (APIL) in the form of a Traffic Light is used. Traffic Lights are installed at various types of road intersections and crossing facilities. The function of traffic lights is very important so that traffic lights must be controlled as easily and efficiently as possible to facilitate traffic flow at a road intersection. However, after the use of Traffic Lights, there is still congestion due to the direction of vehicle arrivals from each lane not being simultaneous. This results in long queues in one of the Traffic Light lanes which makes the queue even longer due to the ineffectiveness of the red light duration of the existing Traffic Light. Therefore, researchers want to design a tool using 8 IR sensors as queue detection in each lane, then combining it with an automatic Traffic Light system can optimize the red and green lights according to the existing queue length. The results of the experiment found that in quiet conditions or no sensors detecting vehicles, the green light time is 20 seconds. When the condition of congestion 1 or the front IR sensor detects a vehicle, the green light time is increased by 5 seconds to 25 seconds. When traffic jam condition 2 or both IR sensors detect a vehicle, the green light is extended by 10 seconds to 30 seconds.

Keywords

Traffic Light, IR, traffic jam, queue.

DAFTAR ISI

HALAMAN PENGESAHAN	ii
HALAMAN PERNYATAAN ORISINALITAS.....	i
HALAMAN KESEPAKATAN PUBLIKASI	ii
KATA PENGANTAR	iii
ABSTRAK	v
ABSTRACT	vi
DAFTAR ISI.....	vii
DAFTAR GAMBAR	x
DAFTAR TABEL.....	xii
BAB I PENDAHULUAN	1
1.1 Latar Belakang.....	1
1.2 Perumusan Masalah	3
1.3 Tujuan Penelitian	3
1.4 Batasan Masalah	3
1.5 Luaran yang Diharapkan.....	4
1.6 Manfaat Penelitian	4
1.7 Sistematika Penulisan	4
BAB II TINJAUAN PUSTAKA	6
2.1 Penelitian Terkait.....	6
2.2 Teori Dasar	9
2.2.1 Lampu Lalu Lintas (<i>Traffic Light</i>)	9
2.2.2 ESP32	10
2.2.3 Sensor Proximity	12
2.2.4 <i>Light Emitting Diode</i> (LED)	14
2.2.5 LCD I2C	15
2.2.6 Software Arduino IDE.....	16
2.2.7 Wi-Fi	17
2.2.8 <i>Protokol MQTT</i>	18
2.2.9 Protokol MQTT – SN	19

2.2.10 Rangkaian pada Modul Lampu Lalu Lintas	20
BAB III PERANCANGAN.....	21
3.1 Sumber Data dan Pengumpulan Data	21
3.2 Alat dan Bahan	22
3.3 Perancangan Sistem	22
3.3.1 Perancangan Sistem Kerja	22
3.3.2 Perancangan Hardwere	24
3.3.3 Perancangan Software	25
3.3.4 Perancangan Prototype	26
3.4 Skenario Pengujian	27
3.5 Rangkaian pada Modul Lampu Lalu Lintas	28
BAB IV EKSPERIMEN DAN ANALISIS	30
4.1 Hasil Perancangan Alat.....	30
4.1.1 Hasil Perancangan Mekanik	30
4.1.2 Hasil Perancangan Skematik Rangkaian.....	32
4.2 Pengujian dan Analisis Hasil	33
4.2.1 Hasil Pengujian Catu Daya.....	33
4.2.2 Hasil Pengujian Sensor IR.....	34
4.2.3 Hasil Pengujian LCD.....	37
4.3 Hasil Pengujian Skenario.....	38
4.3.1 Hasil Skenario Pertama	40
4.3.2 Hasil Skenario Kedua	43
4.3.3 Hasil Skenario Ketiga.....	46
4.3.4 Hasil Skenario Keempat	49
4.3.5 Hasil Skenario Kelima.....	52
4.4 Analisis Skenario	55
4.4.1 Analisa Respon Sistem terhadap Kepadatan Lalu Lintas	57
4.4.2 Analisa Akurasi dan Efisiensi Sensor IR	58
4.4.3 Analisa Komunikasi dan Kendali IoT.....	58
4.4.4 Kesesuaian Sistem terhadap Kebutuhan Dishub	58
4.4.5 Kelemahan dan Pengembangan Sistem	60

BAB V KESIMPULAN DAN SARAN.....	63
5.1 Kesimpulan.....	63
5.2 Saran.....	63
DAFTAR PUSTAKA	65

DAFTAR GAMBAR

Gambar 2. 1	ESP32.....	10
Gambar 2.3	pinout ESP32.....	11
Gambar 2.4	Sensor Proximity [14]	12
Gambar 2.5	LED [15]	14
Gambar 2.5	LCD I2C.....	16
Gambar 2.6	Tampilan Arduino IDE.....	17
Gambar 2.7	Sistem Sederhana MQTT [16].....	19
Gambar 2.8	Arsitektur MQTT-SN [17]	19
Gambar 2.9	Rangkaian Sistem.....	20
Gambar 3.1	Metodologi Penelitian	21
Gambar 3.2	Flowchart Sistem Kerja Alat	23
Gambar 3.3	Blok Diagram Penelitian	24
Gambar 3.4	Bentuk Perancangan Website Monitoring Kepadatan	26
Gambar 3. 5	Bentuk Perancangan Prototype.....	27
Gambar 3.6	Rangkaian Sistem.....	28
Gambar 4.1	Hasil Perancangan Mekanik.....	30
Gambar 4.2	Hasil Perancangan Keseluruhan	31
Gambar 4.3	Hasil Rangkaian	32
Gambar 4.4	Pengujian Catu Daya Adaptor <i>Power supply</i> DC	34
Gambar 4.5	Pengukuran Tahanan Output Sensor <i>Infrared</i>	35
Gambar 4.6	Pengujian Kalibrasi Sensor <i>Infrared</i>	36
Gambar 4.7	Hasil Kalibrasi Sensor <i>Infrared</i>	37
Gambar 4.8	Pengujian LCD.....	37
Gambar 4.9	Skenario Pertama	41
Gambar 4.10	Grafik Hasil Pengujian Skenario Pertama	42
Gambar 4.11	Tampilan MQTT Skenario Pertama	43
Gambar 4.12	Skenario Kedua	44
Gambar 4.13	Grafik Hasil Pengujian Skenario Kedua.....	45
Gambar 4.14	Hasil MQTT Skenario Kedua.....	46
Gambar 4.15	Skenario Ketiga.....	47
Gambar 4.16	Grafik Hasil Pengujian Skenario Ketiga.....	48
Gambar 4.17	Hasil MQTT Skenario Ketiga	49

Gambar 4.18 Skenario Keempat	50
Gambar 4.19 Grafik Hasil Pengujian Skenario Keempat	51
Gambar 4.20 Hasil MQTT Skenario Keempat.....	52
Gambar 4.21 Skenario Kelima.....	53
Gambar 4.22 Grafik Hasil Pengujian Skenario Kelima.....	54
Gambar 4.23 Hasil MQTT Skenario Kelima	55

DAFTAR TABEL

Tabel 2.1 Penelitian Terdahulu	8
Tabel 3.1 Alat dan Bahan	22
Tabel 3.2 Skenario Pengujian	28
Tabel 4.1 Skenario Pengujian	39
Tabel 4. 2 Durasi Lampu Hijau.....	39
Tabel 4.3 Hasil Pengujian Skenario Pertama	42
Tabel 4.4 Hasil Pengujian Skenario Kedua.....	45
Tabel 4.5 Hasil Pengujian Skenario Kedtiga	47
Tabel 4.6 Hasil Pengujian Skenario Ketujuh	51
Tabel 4.7 Hasil Pengujian Skenario Kelima	53
Tabel 4. 8 Analisa Perbandingan Kondisi Eksisting	59

BAB I

PENDAHULUAN

1.1 Latar Belakang

Setiap tahun jumlah kendaraan yang ada di Indonesia selalu mengalami penambahan. Berdasar data yang dikutip dari kompas.com. Puncak tertinggi dalam sejarah penjualan mobil di Indonesia pada tahun 2013 yaitu sebanyak 1.229.902 unit (Kompas.com). Menurut data dari Badan Pusat Statistik jumlah populasi kendaraan sampai dengan tahun 2013 sejumlah 104.118.969 dengan pertumbuhan jumlah kendaraan setiap tahunnya mencapai 1%. Saat ini diperkirakan sekitar 108 juta kendaraan di Indonesia (bps.go.id). Peningkatan jumlah kendaraan yang tidak diikuti dengan peningkatan luas jalan (infrastruktur) mengakibatkan kemacetan di kota-kota besar.

Sebagai upaya dalam pengaturan lalu lintas digunakanlah Alat Pengatur Instruksi Lalu lintas (APIL) yang berupa *Traffic Light*. *Traffic Light* dipasang di berbagai jenis persimpangan jalan maupun sarana penyeberangan. Berdasarkan hasil pengamatan lapangan pada saat jam sibuk, terutama pagi hari pada saat jam berangkat ke sekolah dan kantor serta sore hari pada saat jam pulang sekolah dan bekerja terjadi kemacetan di berbagai persimpangan jalan. Penumpukan kendaraan pada *Traffic Light* setiap persimpangan jalan terjadi sampai ratusan meter (Walean, 2019).

Traffic Light atau lampu lalu lintas merupakan lampu yang digunakan untuk mengatur kelancaran lalu lintas pada suatu persimpangan dengan cara memberi kesempatan pengguna jalan dari masing-masing arah untuk berjalan secara bergantian (Aria, 2017). Fungsi lampu lalu lintas sangatlah penting sehingga lampu lalu lintas harus dapat dikontrol semudah dan seefisien mungkin guna memperlancar arus lalu lintas di suatu persimpangan jalan. Saat ini sudah banyak jenis pengaturan lampu lalu lintas yang digunakan. Namun, masih terdapat banyak lampu lalu lintas menggunakan kabel bawah tanah untuk menghubungkan setiap

lampu lalu lintas di setiap persimpangan dengan kontroler utamanya sehingga dibutuhkan kabel yang cukup panjang. Selain pengaturan lampu lalu lintas menggunakan kabel, terdapat juga pengaturan lampu lalu lintas menggunakan sistem nirkabel (*wireless*).

Untuk mengatasi permasalahan tersebut, Sistem lampu lalu lintas untuk menerapkan pendeteksi kendaraan sebelumnya telah diteliti oleh siswaya (siswaya, 2021), peneliti ini untuk menghasilkan inovasi pengatur lampu lalu lintas yang lebih akurat sehingga bisa mengatasi baik kemacetan lalu lintas, adapun sensor yang digunakan yaitu menggunakan sensor proximity sebagai pendeteksi kemacetan. Kemudian peneliti purwanda melakukan penelitian yang hampir serupa tetapi sensor yang digunakan yaitu sensor LDR (Purwanda, 2023). Akan tetapi penelitian tersebut masih belum dijalankan secara online serta data yang ada belum dapat dipantau dari jarak jauh.

Sementara itu untuk pendeteksian kemacetan dengan penambahan waktu yang dikondisikan juga sudah diteliti oleh Alfith pada tahun 2019 (Alfith, 2019). Dengan menggunakan sensor LDR dan penambahan Timer sehingga kondisi lalu lintas dapat disesuaikan dengan tingkat kemacetan. Penelitian serupa juga diteliti oleh Reny pada tahun 2020 (Reny, 2020) dengan memaksimalkan pengaturan waktu yang lebih optimal dalam menangani tingkat kemacetan. Sistem lalu lintas pada persimpangan juga diteliti oleh Zulianti pada tahun 2021, zulianti lebih memfokuskan lalu lintas untuk kondisi darurat (Zulianti, 2021).

Sistem yang akan dibuat pada penelitian ini adalah *Smart Traffic Light* Berbasis Esp32. *Smart Traffic* Berbasis Esp32 adalah lampu lalu lintas yang dibuat dalam bentuk modul. Modul memiliki komponen yang terdiri dari satu mikrokontroler Esp32, tiga buah lampu LED yang terdiri dari satu LED merah, satu LED kuning dan satu LED hijau. Apabila modul hendak digunakan pada perempatan jalan, dibutuhkan empat LED yang ditempatkan pada setiap sisi jalan. Setiap LED yang digunakan memiliki karakteristik yang sama berupa tampilan fisik yang digunakan. Kelebihan dari hal ini yaitu memasang sensor IR di setiap persimpangan lalu lintas, sehingga memiliki optimasi persimpangan lebih efisien. Hal inilah yang menyebabkan modul lampu lalu lintas yang akan dibuat adalah modul yang '*smart*'.

Berdasarkan pernyataan yang telah dipaparkan, peneliti mengambil topik penelitian yang berjudul “**Prototipe *Smart Traffic Light (STL)* Berdasar Panjang Antrian Menggunakan *Internet Of Things (IoT)***” dengan harapan dapat mengurangi tingkat kepadatan dan dapat menginformasikan tingkat lalu lintas di simpang tiga tabek gadang.

1.2 Perumusan Masalah

Dalam penelitian ini terdapat beberapa rumusan masalah yang akan dibahas mengenai Rancang Bangun Sistem *Smart Traffic Light* Berbasis *IoT* menggunakan Mikrokontroler ESP32, yaitu:

1. Bagaimana membuat prototipe sistem lampu lalu lintas berbasis mikrokontroler Esp32 Sebagai sistem pendeteksi panjang antrian lalu lintas?
2. Bagaimana cara mendeteksi panjang antrian pada persimpangan lampu lalu lintas untuk mengetahui kondisi lalu lintas pada *Smart Traffic Light*?
3. Bagaimana mengatasi selisih waktu yang terjadi ketika adanya perbedaan lampu merah (lampu hijau) yang terjadi secara tiba-tiba pada persimpangan?

1.3 Tujuan Penelitian

Tujuan yang di harapkan dari penelitian Rancang Bangun Sistem *Smart Traffic Light* Berbasis *IoT* menggunakan Mikrokontroler ESP32 adalah:

1. Membuat prototipe sistem lampu lalu lintas berbasis mikrokontroler Esp32 Sebagai sistem pendeteksi panjang antrian lalu lintas.
2. Mendeteksi panjang antrian pada persimpangan lampu lalu lintas untuk mengetahui kondisi lalu lintas pada *Smart Traffic Light*
3. Mengatasi selisih waktu yang terjadi ketika adanya perbedaan lampu merah (lampu hijau) yang terjadi secara tiba-tiba pada persimpangan.

1.4 Batasan Masalah

Dalam penelitian ini ditetapkan batasan masalah untuk mempermudah pembahasan dan memfokuskan pada Rancang Bangun

Sistem *Smart Traffic Light* Berbasis *IoT* menggunakan Mikrokontroler ESP32, Adapun batasan yang di tetapkan adalah:

1. Menggunakan ESP32 Esp32 sebagai kontroler lampu lalu lintas.
2. Membuat kondisi lalu lintas dengan tiga kondisi yaitu normal, Macet Tingkat I dan Macet Tingkat II.
3. Menggunakan MQTT sebagai monitoring keadaan graffik yang sedang berlangsung menggunakan sensor IR sebanyak 8 sensor.
4. Memanipulasi waktu lampu hijau untuk mengoptimalkan panjang antrian berdasarkan kemacetan.
5. Sistem lampu lalu lintas yang dibuat digunakan pada perempatan.

1.5 Luaran yang Diharapkan

Luaran yang diharapkan dari penelitian ini yaitu menghasilkan Rancang Bangun Sistem *Smart Traffic Light* Berbasis *IoT* menggunakan Mikrokontroler ESP32 sebagai sistem yang dapat memonitoring keadaan graffik tingkat kepadatan melalui MQTT.

1.6 Manfaat Penelitian

Adapun manfaat penelitian ini adalah:

1. Dapat dijadikan salah satu sumber referensi dalam bidangnya.
2. Menambah pengetahuan tentang sistem *Smart Lamp*.
3. Menambah keterampilan dalam mendesain sistem *Smart Lamp* yang lebih efisien dan berteknologi terbaru sehingga bisa di monitoring melalui jarak jauh.
4. Menambah pengetahuan pada bidangnya.

1.7 Sistematika Penulisan

Penulisan Skripsi ini disajikan dengan sistematika penulisan sebagai berikut:

BAB I : Pendahuluan

Bab ini penulis memberikan penjelasan tentang latar belakang, perumusan masalah, batasan masalah, manfaat penelitian, tujuan penelitian dan sistematika penulisan.

BAB II : TINJAUAN PUSTAKA

Bab ini memberikan penjelasan mengenai penelitian terkait dan dasar teori yang penulis gunakan sebagai acuan dan referensi penulisan skripsi serta pelaksanaan penelitian.

BAB III : PERANCANGAN

Bab ini memberikan penjelasan tentang flowchart penelitian, perancangan simulasi dan blok diagram yang digunakan pada penelitian ini.

BAB IV : EKSPERIMEN DAN ANALISIS

Bab ini membahas hasil pengujian desain penelitian serta data data yang didapatkan pada saat pengujian.

BAB V : KESIMPULAN DAN SARAN

Bab ini berisikan kesimpulan dari penelitian serta saran yang diberikan oleh penulis.

BAB II

TINJAUAN PUSTAKA

2.1 Penelitian Terkait

Dalam penelitian terdahulu yang berhubungan dengan penelitian ini diharapkan peneliti dapat menunjang penelitian agar lebih baik lagi. Penelitian terkait pertama dilakukan oleh Alfith dengan judul “Pengembangan Perancangan Smart *Traffic Light* Berbasis LDR Sensor Dan Timer Delay System” tahun 2019 (Alfith, 2019). Dalam penelitian ini, Pengembangan ini dimaksudkan untuk mengetahui prinsip kerja dari smart traffic light dengan mikrokontroler dan mengurangi lampu merah para pengendara disetiap persimpangan. Ketika keempat simpang dalam keadaan kosong maka keempat simpang lampu akan tetap merah, apabila pada satu simpang ada kendaraan maka simpang tersebut maka sensor LDR akan mendeteksi kendaraan.

Penelitian keuda yang dilakukan oleh Reny Medikawati Taufiq dengan judul “Integrated Smart Traffic Control System Menuju Pekanbaru Sebagai Smart City” tahun 2020 (Reny, 2020). Penelitian ini memberikan rancangan solusi untuk tata kelola lampu lalu lintas cerdas (Smart Traffic Control System), berbasis teknologi object detection yang menggunakan deep learning untuk mendeteksi jumlah dan jenis kendaraan. Jumlah kendaraan menjadi dasar penentuan timer lampu hijau secara otomatis. Smart Traffic Control System (STCS) terintegrasi dengan sistem informasi geografis berbasis web (smart map) yang secara kontinu menerima informasi kepadatan (gambar terkini, jumlah kendaraan, level kepadatan), kemudian menampilkannya diatas peta Kota Pekanbaru..

Kemudian penelitian dengan penambahan timer dilakukan oleh Neci Purwanda dengan judul “Prototype Smart Time Scheduler Lampu Lalu Lintas Menggunakan Algoritma Haar Cascade” tahun 2023 (Purwanda, 2023). Implementasi Smart Time Scheduler menjadi solusi untuk mengatur durasi lampu lalu lintas berdasarkan tingkat kepadatan lalu lintas, dengan membangun sebuah

alat untuk mendeteksi dan menghitung jumlah antrian kendaraan pada *Traffic Light* kemudian memasukkan hasil perhitungan kedalam 3 kategori yaitu Kepadatan Normal, Sedang, dan Padat. Hasil perhitungan kendaraan oleh sistem dengan Hasil Penghitungan secara Manual dikali 100 %. Maka Error Perhitungan Sistem sebesar 16% dan tingkat akurasi alat adalah $100\% - 16\% = 84\%$, besarkecil nya nilai Error ini dipengaruhi oleh kualitas intensitas cahaya, dan jarak deteksi kamera dengan objek pada saat melakukan Deteksi.

Pengaturan lampu lalu lintas untuk keadaan darurat dilakukan oleh Febriyani Zulianti dengan judul “Smart *Traffic Light* Berbasis Internet Of Things Pada Keselamatan Ambulans” tahun 2021 (Zulianti, 2021). Dalam perancangan sistem Smart *Traffic Light* ini bekerja berbasis IoT dan di perlukan beberapa alat untuk menunjang dalam rancangan ini seperti ESP8266 yang sudah terhubung firebase melalui koneksi internet, dimana layanan firebase yang digunakan pada aplikasi Smart *Traffic Light* adalah authentication dan realtime database. Fitur GPS berguna untuk menyimpan data update lokasi perjalanan ke Firebase melalui koneksi internet. Ketika jarak mobil ambulans pada jarak tertentu mendekati lampu lalu lintas, maka lampu lalu lintas tersebut berubah menjadi warna hijau..

Sementara itu untuk memasukkan data secara realtime dilakukan oleh Ida Bagus dengan judul “Pengembangan Smart *Traffic Light* Berbasis IOT Dengan Mobile Backend As A Service Sebagai Wujud Smart City Bidang Transportasi” tahun 2022 (Bagus, 2022). Smart *Traffic Light* berbasis IoT dengan MbaaS menggabungkan sistem berupa aplikasi NodeMCU, Firebase dan android. Layanan Firebase yang digunakan pada Smart *Traffic Light* adalah autentikasi dan realtime database. Modul NodeMCU terhubung ke Firebase melalui komunikasi internet. Aplikasi android menggunakan layanan autentikasi dan database realtime. Referensi untuk menemukan rute perjalanan dan memprediksi jarak pengendara dengan lampu lalu lintas memanfaatkan layanan Google Map API (Application Program Interface). Aplikasi Smart *Traffic Light* berbasis Android mengupdate lokasi dengan memanfaatkan fitur GPS kemudian menyimpan data lokasi perjalanan yang terupdate ke Firebase.

Tabel 2. 1 Penelitian Terdahulu

No	Nama	Metode/Sensor	Kelemahan
1	Alfith, 2019	Mendeteksi tingkat kepadatan dengan sensor LDR	1. Sensor LDR hanya mendeteksi cahaya, tingkat error akan sangat tinggi ketika malam hari 2. Tidak berbasis <i>IoT</i>
2	Reny Medikawati Taufiq, 2020	Mendeteksi kepadatan dengan Citra Gambar Digital	1. Pendeteksian menggunakan citra gambar yang memiliki tingkat error yang tinggi 2. Tidak berbasis <i>IoT</i>
3	Siswaya, 2021	Mendeteksi tingkat kepadatan dengan sensor proximity	1. Tidak berbasis <i>IoT</i> 2. Masih menggunakan module Bluetooth yang membuat jarak pengontrolan rentan pendek
4	Febriyani Zulianti, 2021	Menempatkan GPS pada ambulan dan menjadikan prioritas pada persimpangan lalu lintas.	1. Hanya pada kendaraan ambulan
5	Ida Bagus, 2022	ESP32 yang terletak di <i>Smart Traffic Light</i> di persimpangan mengambil data dari Firebase menggunakan koneksi internet.	1. Hanya sebagai monitoring
6	Penelitian ini	Rancang Bangun Sistem <i>Smart Traffic Light</i> Berbasis <i>IoT</i> Sebagai Sistem Pendeteksi Kepadatan menggunakan sensor proximity dan mengatur lalu lintas menggunakan <i>Timer</i> serta membagikannya secara <i>online</i>	1. Diharapkan dapat mendeteksi kendaraan dengan tingkat error yang lebih rendah dari penelitian terdahulu 2. Mengatur lalu lintas sesuai dengan tingkat kepadatan yang terjadi 3. menampilkan pada sistem <i>IoT</i> agar dapat dipantau oleh setiap pengguna lalu lintas

2.2 Teori Dasar

2.2.1 Lampu Lalu Lintas (*Traffic Light*)

Lampu lalu lintas menurut UU No. 22/2009 tentang lalu lintas dan angkutan jalan ialah alat pemberi isyarat lalu lintas atau (APILL) merupakan lampu yang mengendalikan arus lalu lintas yang terpasang di persimpangan jalan, tempat penyeberangan pejalan kaki (zebra cross), dan tempat arus lalu lintas lainnya (Undang-Undang, 2009). Pemasangan lampu lalu lintas bertujuan untuk mengatur sirkulasi kendaraan pada persimpangan jalan agar tidak terjadi kecelakaan. Oleh karena itu, lampu lalu lintas menjadi bagian sangat penting guna memperlancar lalu lintas di setiap kota.

Terdapat dua jenis lampu lalu lintas yaitu lampu lalu lintas berdasarkan cakupan dan berdasarkan pengoperasian. Berdasarkan cakupan, lampu lalu lintas terdiri dari tiga jenis (Hozanna, 2021),, yaitu:

1. Lampu lalu lintas terpisah, yaitu lampu lalu lintas yang pemasangannya didasarkan pada suatu tempat persimpangan saja tanpa mempertimbangkan persimpangan lain.
2. Lampu lalu lintas terkoordinasi, yaitu lampu lalu lintas yang pemasangannya mempertimbangkan beberapa persimpangan yang terdapat pada arah tertentu.
3. Lampu lalu lintas jaringan, yaitu lalu lintas yang pemasangannya mempertimbangkan beberapa persimpangan yang terdapat dalam suatu jaringan yang masih dalam satu kawasan.

Berdasarkan pengoperasian, lampu lalu lintas terdiri dari dua jenis (Jatmika, 2014), yaitu:

1. *Fixed time traffic signal* , yaitu lampu lalu lintas yang pengoperasiannya menggunakan waktu yang tetap dan tidak mengalami perubahan.
2. *Actuated traffic signal*, yaitu lampu lalu lintas yang pengoperasiannya dengan pengaturan waktu tertentu dan

mengalami perubahan dari waktu ke waktu sesuai dengan kedatangan kendaraan dari berbagai persimpangan.

Secara umum, warna lampu yang digunakan untuk lampu lalu lintas adalah warna merah, warna kuning dan warna hijau. Merah menandakan berhenti atau sebuah tanda bahaya, kuning menandakan hati-hati dan hijau menandakan boleh memulai berjalan dengan hati-hati. Biasanya, lampu warna merah mengandung beberapa corak berwarna jingga dan lampu hijau mengandung beberapa warna biru. Ini dimaksudkan agar orang-orang yang buta warna merah dan hijau dapat mengerti sinyal lampu yang menyala.

2.2.2 ESP32

ESP32 adalah mikrokontroler yang dikenalkan oleh Espressif System merupakan penerus dari mikrokontroler ESP8266. Pada mikrokontroler ini sudah tersedia modul WiFi dalam chip sehingga sangat mendukung untuk membuat sistem aplikasi *Internet of Things*. Untuk spesifikasi dari ESP32 dapat dilihat pada Gambar 2.1 dan pada Gambar 2.2 merupakan pinout dari ESP32. Pin tersebut dapat dijadikan input atau output untuk menyalakan LCD, lampu, bahkan untuk menggerakkan motor DC (Romli, 2021).



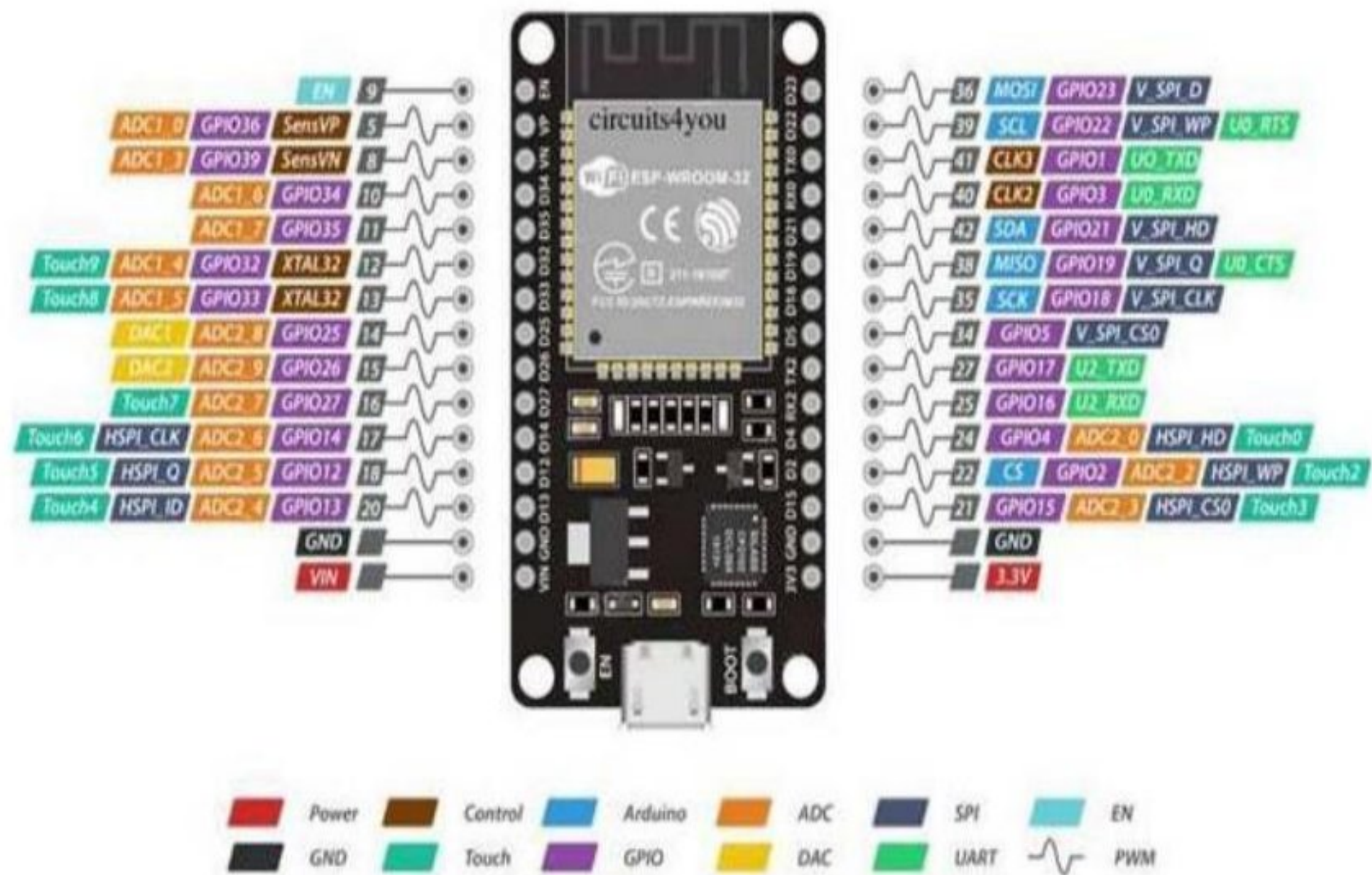
Gambar 2. 1 ESP32

Spesifikasi modul ESP32-WROOM-32 :

1. Microprosesor Xtensa Dual-Core 32 Bit LX6
2. Freq Clock up to 240 MHz
3. SRAM 520 kB
4. Flash memori 4 MB
5. 11b/g/n WiFi transceiver
6. Bluetooth 4.2/BLE
7. 48 pin GPIO
8. 15 pin channel ADC (Analog to Digital Converter)

9. 25 pin PWM (Pulse Width Modulation)
10. 2 pin channel DAC (Digital to Analog Converter)

Pada gambar 2.3 merupakan pinout dari ESP32 :



Gambar 2.2 pinout ESP32

ESP32 Dev Kit V1 adalah sebuah board pengembangan (development board) yang didukung oleh mikrokontroler Tensilica 32-bit Single-/Dual-core CPU Xtensa LX6 dengan kecepatan clock 240 Mhz. Board ini dilengkapi dengan 520 KiB SRAM dan 4 MB flash memory untuk menyimpan program dan data. Board ini juga memiliki 25 digital input/output (DIO) pins, 6 analog input (ADC) pin, dan 2 analog output (DAC) pin yang dapat digunakan untuk berbagai keperluan seperti mengendalikan perangkat elektronik atau membaca sensor. Selain itu, ESP32 Dev Kit V1 juga dilengkapi dengan 3 UARTs, 2 SPIs, dan 3 I2Cs, yang memungkinkan board ini untuk berkomunikasi dengan perangkat lain secara serial atau menggunakan protokol komunikasi seperti SPI dan I2C. Board ini juga dilengkapi dengan Wi-Fi yang mendukung standar IEEE 802.11 b/g/n/e/, sehingga board ini dapat terhubung ke jaringan Wi-Fi dan berkomunikasi dengan perangkat lain melalui jaringan tersebut. Dengan ukuran 51.5x29x5mm, board ini cukup kecil dan mudah untuk diintegrasikan ke dalam proyek-proyek yang lebih kompleks.

Keseluruhan, ESP32 Dev Kit V1 adalah board pengembangan yang kuat dan serbaguna, yang cocok untuk berbagai aplikasi *Internet of Things* (IoT) atau proyek-proyek elektronika yang membutuhkan konektivitas Wi-Fi.

2.2.3 Sensor Proximity

Proximity Sensor (Sensor Proksimitas) atau dalam bahasa Indonesia disebut dengan Sensor Jarak adalah sensor elektronik yang mampu mendeteksi keberadaan objek di sekitarnya tanpa adanya sentuhan fisik. Dapat juga dikatakan bahwa Sensor Proximity adalah perangkat yang dapat mengubah informasi tentang gerakan atau keberadaan objek menjadi sinyal listrik.

Salah satu contoh bentuk Sensor Proximity seperti ditunjukkan pada Gambar 2.3 berikut:



IR Proximity Sensor

Gambar 2.3 Sensor Proximity (Kurniawan, 2019)

Sensor Proximity dapat diklasifikasikan menjadi 4 jenis, yaitu Inductive Proximity Sensor, Capacitive Proximity Sensor, Ultrasonic Proximity Sensor dan Photoelectric Sensor (Kurniawan, 2019).

1. Inductive Proximity Sensor (Sensor Jarak Induktif)

Sensor Jarak Induktif atau Inductive Proximity Sensor adalah Sensor Jarak yang digunakan untuk Sensor Jarak yang digunakan untuk mendeteksi keberadaan logam baik logam jenis Ferrous maupun logam jenis non-ferrous. Sensor ini

dapat digunakan untuk mendeteksi keberadaan (ada atau tidak adanya objek logam), menghitung objek logam dan aplikasi pemosisian. Sensor induktif sering digunakan sebagai pengganti saklar mekanis karena kemampuannya yang dapat beroperasi pada kecepatan yang lebih tinggi dari saklar mekanis biasa. Sensor Jarak Induktif ini juga lebih andal dan lebih kuat.

2. Capacitive Proximity Sensor (Sensor Jarak Kapasitif)

Sensor Jarak Kapasitif atau Capacitive Proximity Sensor adalah Sensor Jarak yang dapat mendeteksi gerakan, komposisi kimia, tingkat dan komposisi cairan maupun tekanan. Sensor Jarak Kapasitif dapat mendeteksi bahan-bahan dielektrik rendah seperti plastik atau kaca dan bahan-bahan dielektrik yang lebih tinggi seperti cairan sehingga memungkinkan sensor jenis ini untuk mendeteksi tingkat banyak bahan melalui kaca, plastik maupun komposisi kontainer lainnya.

Sensor Jarak Kapasitif ini pada dasarnya mirip dengan Sensor Jarak Induktif, perbedaannya adalah sensor kapasitif menghasilkan medan elektrostatik sedangkan sensor induktif menghasilkan medan elektromagnetik. Sensor Jarak Kapasitif ini dapat digerakan oleh bahan konduktif dan bahan non-konduktif. Elemen aktif Sensor Jarak Kapasitif dibentuk oleh dua elektroda logam yang diposisikan untuk membentuk ekuivalen (sama dengan) dengan Kapasitor Terbuka. Elektroda ini ditempatkan di rangkaian osilasi yang berfrekuensi tinggi. Ketika objek mendekati permukaan sensor jarak kapasitif ini, medan elektrostatik pelat logam akan terinterupsi sehingga mengubah kapasitansi sensor jarak. Perubahan ini akan mengubah kondisi dalam pengoperasian sensor jarak sehingga dapat mendeteksi keberadaan objek tersebut.

3. Ultrasonic Proximity Sensor (Sensor Jarak Ultrasonik)

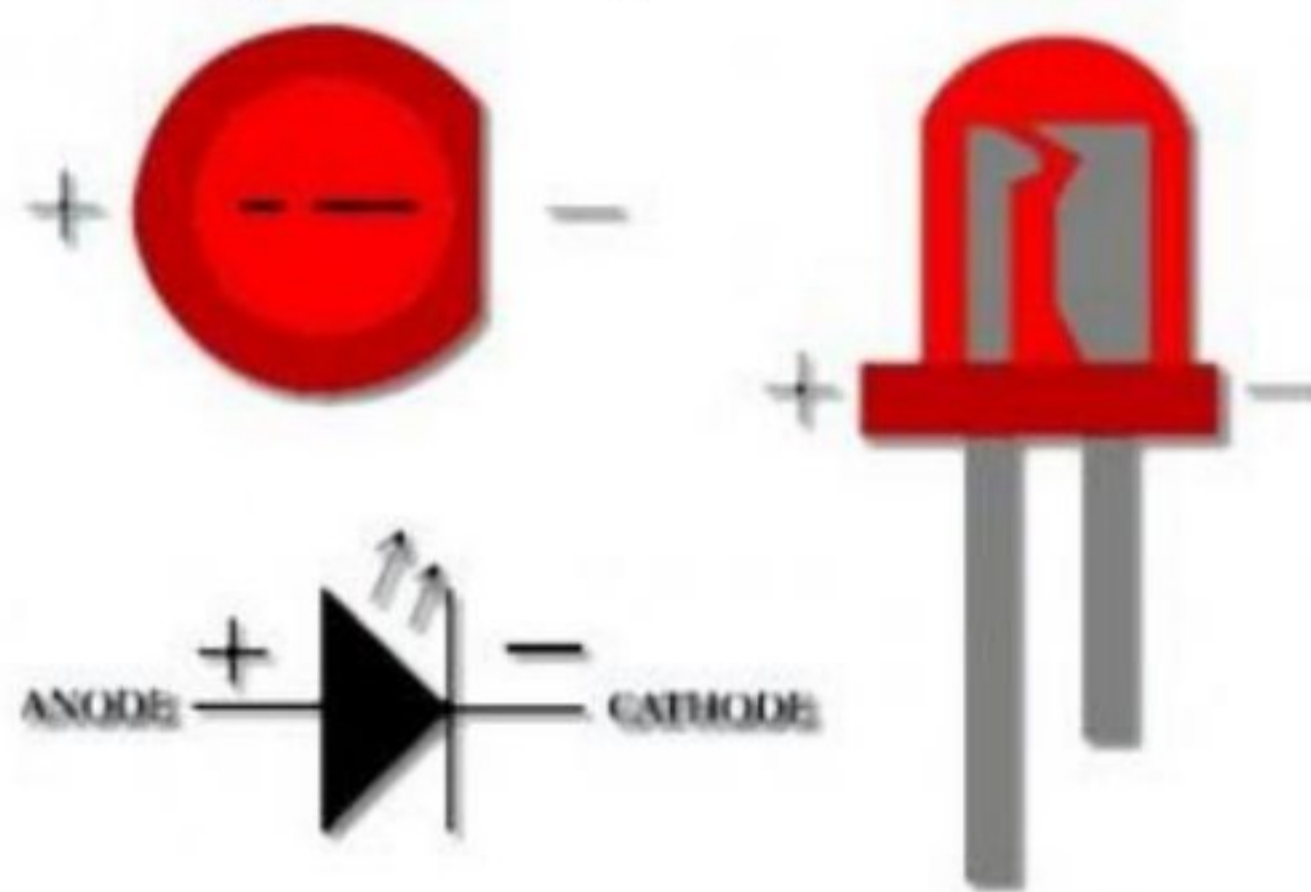
Sensor Jarak Ultrasonik atau Ultrasonic Proximity Sensor adalah sensor jarak yang menggunakan prinsip operasi yang mirip dengan radar atau sonar yaitu dengan menghasilkan gelombang frekuensi tinggi untuk menganalisis gema yang

diterima setelah terpantul dari objek yang mendekatnya. Sensor Proximity Ultrasonik ini akan menghitung waktu antara pengiriman sinyal dengan penerimaan sinyal untuk menentukan jarak objek yang bersangkutan. Sering digunakan untuk mendeteksi keberadaan objek dan mengukur jarak objek di proses otomatisasi pabrik.

4. Photoelectric Proximity Sensor (Sensor Jarak Fotolistrik)
Sensor Jarak Fotolistrik atau Photoelectric Proximity Sensor adalah sensor jarak yang menggunakan elemen peka cahaya untuk mendeteksi obyek. Sensor Proximity Fotolistrik terdiri sumber cahaya (atau disebut dengan Emitter) dan Penerima (Receiver).

2.2.4 *Light Emitting Diode (LED)*

LED adalah dioda yang dapat memancarkan cahaya jika diberikan arus bias maju (*forward bias*). LED dapat memancarkan cahaya karena menggunakan dopping galium, arsenic dan phosporus. Jenis dopping yang berbeda di atas dapat menghasilkan cahaya dengan warna yang berbeda. Setiap warna LED memiliki tegangan kerja yang berbeda. Arus bias maju puncak LED sebesar 60 mA dengan rata-rata arus bias maju tipikal sebesar 20 mA (Sidik, 2007). Pada rangkaian LED dipasang sebuah resistor sebagai pembatas arus. Simbol LED dan bentuk fisik led dapat dilihat pada Gambar 2.4.



Gambar 2.4 LED (Sidik, 2007)

Teknik pemasangan LED agar dapat bekerja adalah dengan memberikan tegangan bias maju (tegangan positif) pada kaki anoda dan tegangan negatif pada kaki katoda. Pembatas arus pada dioda

dilakukan dengan memasang resistor secara seri pada salah satu kaki LED.

Besarnya rata-rata arus bias maju tipikal pada LED adalah 20 mA, sehingga nilai resistor dapat ditentukan. Besarnya nilai resistor berbanding lurus dengan besarnya tegangan sumber yang digunakan. Secara matematis besarnya nilai resistor pembatas arus LED dapat ditentukan menggunakan persamaan berikut:

$$R = \frac{V - V_{ab}(\text{Volt})}{0.02 \text{ Ampere}} \dots\dots\dots(1)$$

Di mana:

R = resistor pembatas arus (Ohm)

V = tegangan sumber yang digunakan untuk men-supply tegangan ke LED (Volt)

V_{ab} = tegangan kerja LED (Volt)

0.02 Ampere = rata-rata arus bias maju tipikal LED (20 mA)

2.2.5 LCD I2C

LCD I2C adalah singkatan dari Liquid Crystal Display Inter-Integrated Circuit. LCD I2C adalah modul display karakter yang dapat digunakan untuk menampilkan teks pada layar. Modul ini dikendalikan melalui protokol I2C, yang memungkinkan data untuk ditransfer antara mikrokontroler dan modul dengan kecepatan yang relatif tinggi. Dalam modul LCD I2C, konversi dari karakter digital ke karakter tampilan pada layar dilakukan oleh driver LCD yang terpasang di dalam modul. Keuntungan dari penggunaan modul LCD I2C adalah kemudahan penggunaannya dan jumlah pin yang dibutuhkan untuk mengendalikan modul yang lebih sedikit dibandingkan dengan modul LCD standar.

Dengan menggunakan modul LCD I2C, proses pengembangan perangkat elektronik yang memerlukan tampilan karakter menjadi lebih mudah dan efisien karena hanya membutuhkan sedikit pin dan wiring yang dibutuhkan. Oleh karena itu, modul LCD I2C sangat cocok digunakan dalam proyek-proyek yang membutuhkan display tampilan karakter dengan kompleksitas yang relatif rendah, seperti tampilan suhu dan kelembapan pada sistem monitoring atau tampilan pesan pada alat ukur sederhana. Beberapa karakteristik dari modul LCD I2C antara lain dan spesifikasi LCD I2C dapat dilihat pada gambar 2.17.

1. Memiliki dua baris display, masing-masing dengan 16 karakter.
2. Mampu menampilkan karakter khusus dengan menggunakan RAM karakter.
3. Tersedia banyak library untuk berbagai jenis mikrokontroler
4. Memiliki konsumsi daya yang relatif rendah.



Gambar 2. 5 LCD I2C

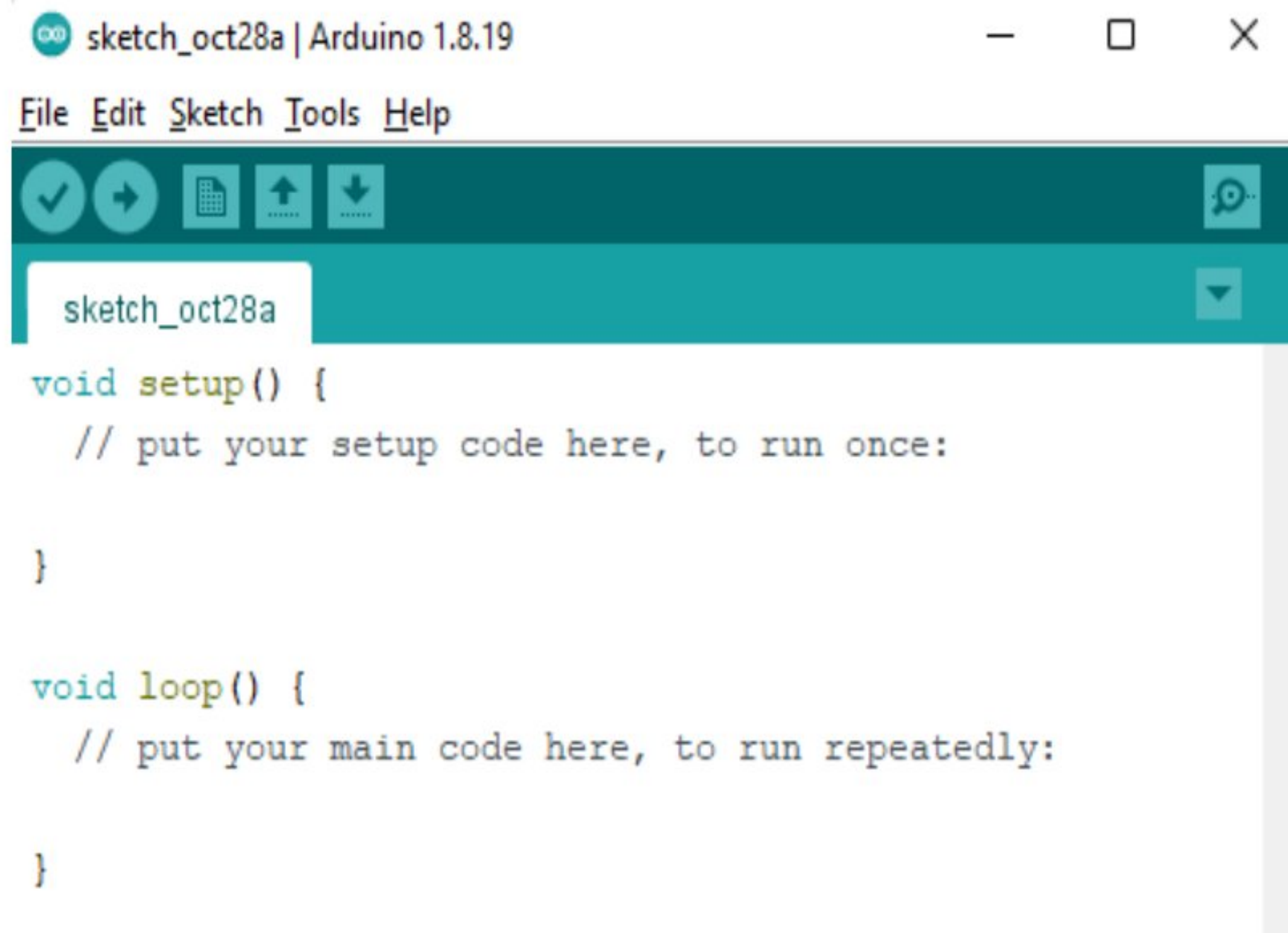
2.2.6 Software Arduino IDE

IDE (Integrated Development Environment) adalah program komputer yang memiliki beberapa fasilitas yang diperlukan dalam pembangunan perangkat lunak. Tampilan software Arduino IDE dapat dilihat pada Gambar 2.5. Tujuan dari IDE adalah untuk menyediakan semua utilitas yang diperlukan dalam membangun perangkat lunak. IDE Arduino digunakan untuk membuat program atau source code, melakukan pengecekan kesalahan, kompilasi, upload program dan menguji hasil kerja arduino melalui serial monitor.

Pada Adruino IDE memiliki toolbars IDE yang memberikan akses instan ke fungsifungsi yang penting yaitu :

1. Verify, untuk mengkompilasi program yang saat ini dikerjakan.
2. Upload, untuk mengkompilasi program dan meng-upload ke papan adruino atau ESP32.

3. New, menciptakan lembar kerja baru.
4. Open, untuk membuka program yang ada di file sistem.
5. Save, untuk menyimpan program yang dikerjakan.
6. Stop, untuk menghentikan serial monitor yang sedang dijalankan.



Gambar 2.6 Tampilan Arduino IDE

2.2.7 Wi-Fi

Wi-Fi atau Wireless Fidelity adalah suatu standar Wireless Networking tanpa kabel, hanya dengan komponen yang sesuai dapat terkoneksi ke jaringan. Teknologi Wi-Fi memiliki standar yang ditetapkan oleh institusi internasional yang bernama Institute of Electrical and Electronic Engineers (IEEE). Secara umum, standar teknologi Wi-Fi adalah sebagai berikut:

1. Standar IEEE 802.11a yaitu Wi-Fi dengan frekuensi 5 GHz yang memiliki kecepatan 54 Mbps dan jangkauan jaringan 300 m.
2. Standar IEEE 802.11b yaitu Wi-Fi dengan frekuensi 2,4 GHz yang memiliki kecepatan 11 Mbps dan jangkauan jaringan 100 m.

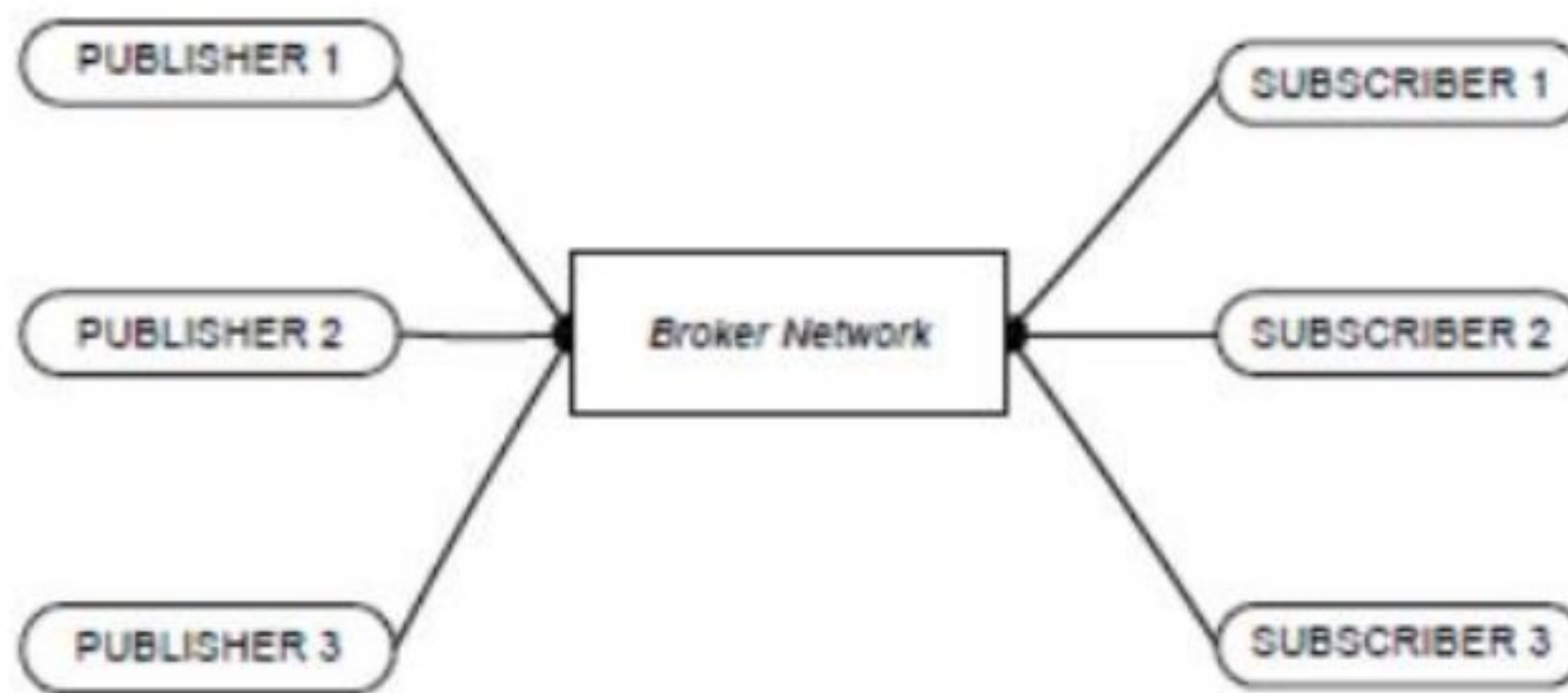
3. Standar IEEE 802.11g yaitu Wi-Fi dengan frekuensi 2,4 GHz yang memiliki kecepatan 54 Mbps dan jangkauan jaringan 300 m.

Access point merupakan salah satu perangkat dalam jaringan komputer yang berguna untuk membuat jaringan nirkabel yang sifatnya lokal atau disebut dengan istilah *Wireless Local Area Network* (WLAN). Pada *Access point* terdapat antena dan transceiver, komponen ini bertugas untuk memancarkan dan menerima sinyal dari client dan server ataupun menuju client dan server.

Access point memiliki fungsi utama sebagai pemancar sinyal internet. *Access point* juga memiliki beberapa fungsi lainnya seperti sebagai *Dynamic Host Configuration Protocol* (DHCP) server sehingga mampu memberikan IP address di setiap perangkat yang terhubung. Fungsi lainnya yakni menggantikan fungsi hub yang menghubungkan jaringan lokal nirkabel dengan jaringan kabel. Disamping ketiga fungsi tersebut, *Access point* dapat memberikan fitur keamanan *Wired Equivalent Privacy* (WEP) dan *Wireless Application Protocol* (WAP).

2.2.8 Protokol MQTT

Message Queue Telemetry Transport (MQTT) merupakan protokol yang lightweight atau terbilang ringan dibanding dengan beberapa protokol lainnya seperti HTTP dan COAP dan lain-lain. MQTT hampir sama dengan HTTP yang berjalan pada TCP/IP, namun HTTP merupakan protokol simetris sedangkan MQTT merupakan protokol asimetris. MQTT merupakan protokol yang dirancang oleh IBM dengan sistem publish/subscribe yang cocok digunakan untuk sistem berbasis Machine To Machine (M2M). Ditengah dari Publisher serta Subscriber terdapat broker server, di mana Publisher akan mengirim data menuju broker dan selanjutnya data akan dikirim kepada Subscriber yang telah tersambung dan disesuaikan dengan topic yang tersedia.

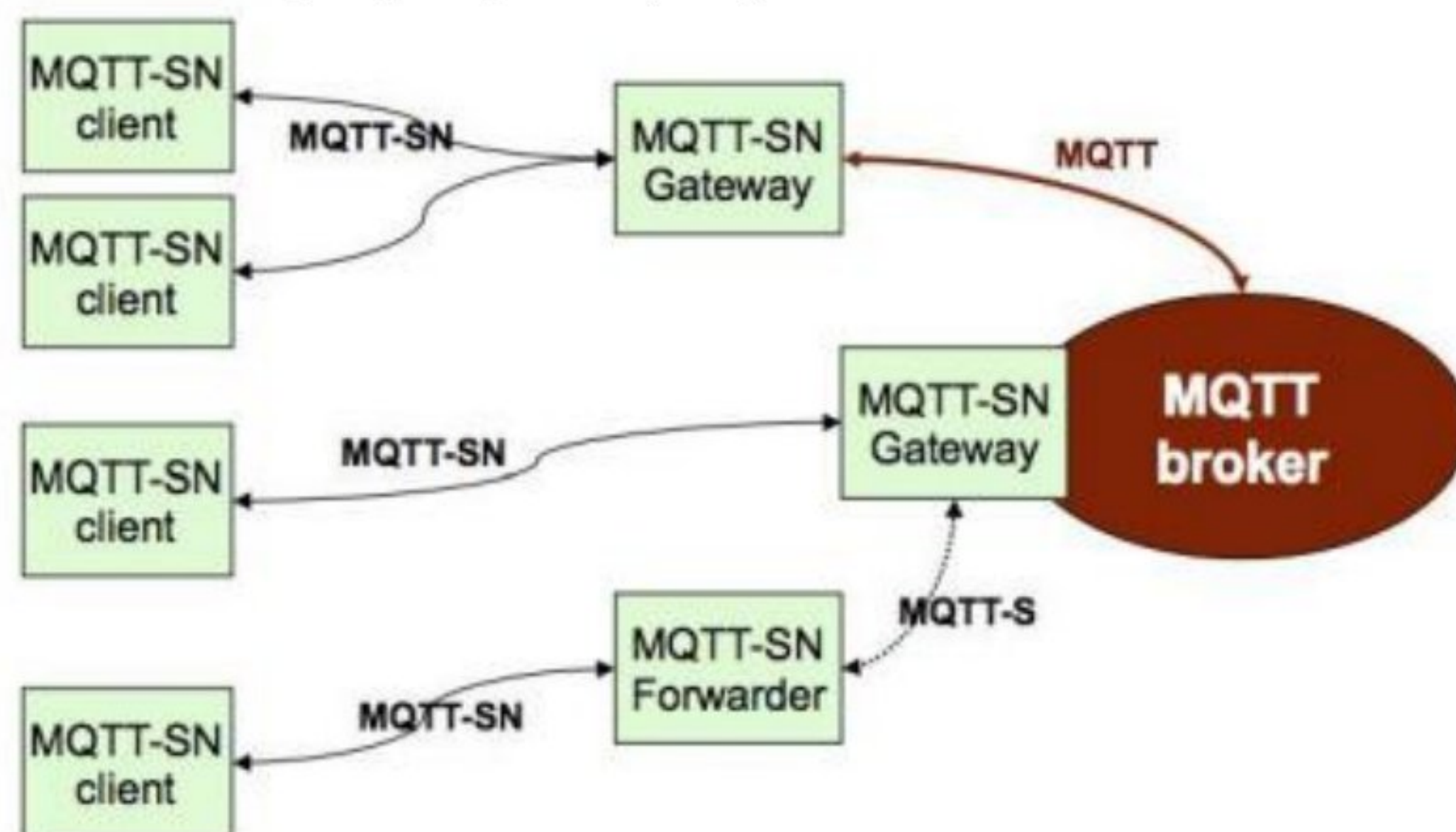


Gambar 2.7 Sistem Sederhana MQTT

Pada Gambar 2.6 merupakan konfigurasi sistem dari MQTT, ketika broker server yang menjembatani node publisherserta subscriber. Ada beberapa platform yang mendukung MQTT broker seperti Trafero Stack, HiveMQ, mosquitto, MQTT.js, Apache Apollo, dan lain-lain.

2.2.9 Protokol MQTT – SN

Message Queue Telemetry Transport for Sensor Network(MQTT–SN) merupakan versi MQTT yang dapat menyesuaikan dengan lingkungan komunikasi network. MQTT-SN sangat baik digunakan untuk implementasi cost yang murah, pemrosesan dan penyimpanan yang terbatas.



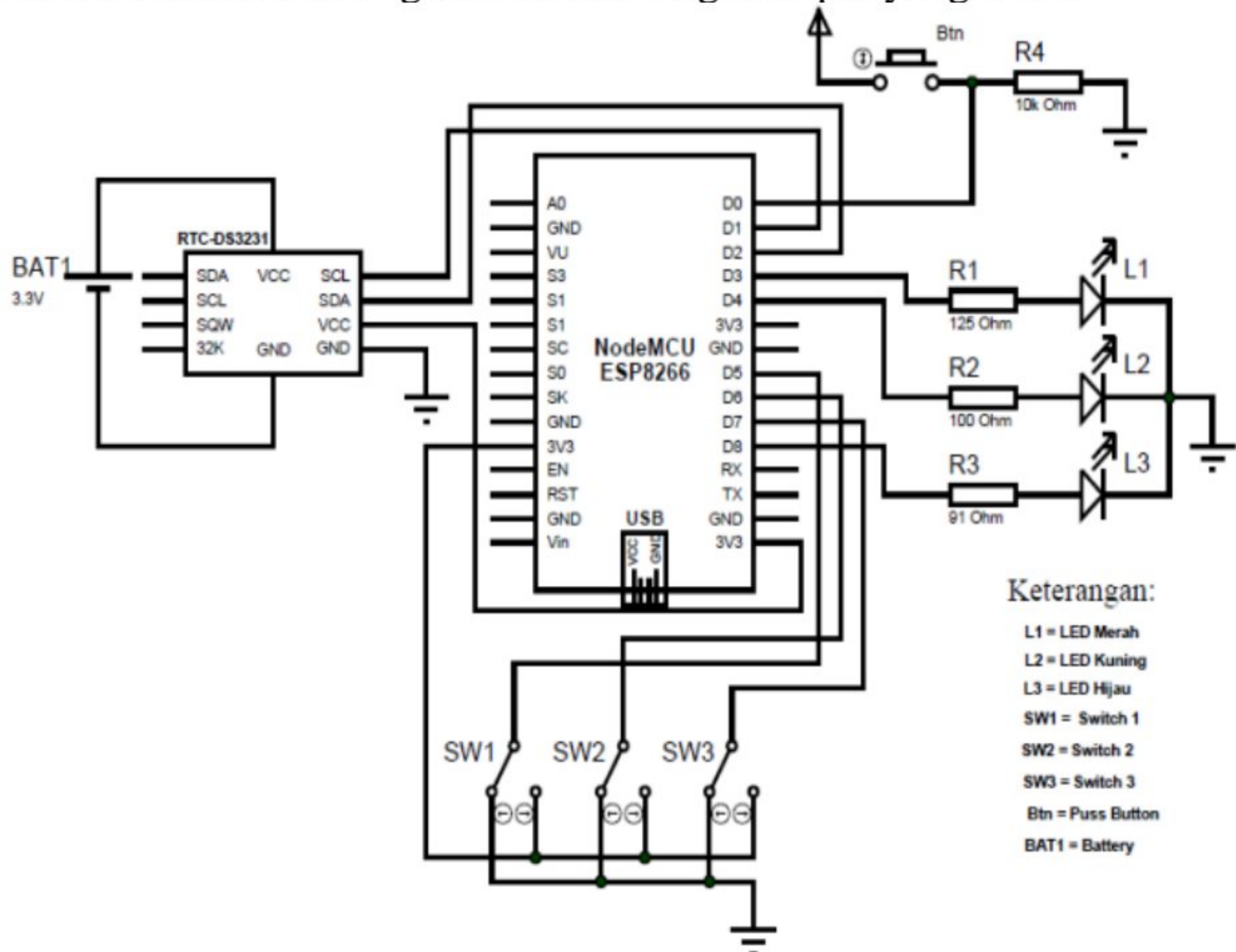
Gambar 2.8 Arsitektur MQTT-SN

Pada Gambar 2.7 MQTT-SN terdapat beberapa komponen seperti clients, gateway, dan forwarders. Node client terhubung dapat terhubung dengan broker dengan terhubung lebih dulu dengan

gateway. Dari Gateway menuju broker dapat menggunakan MQTT. Dalam MQTT-SN terdapat fitur yang dapat menghemat penggunaan energi pada node client yaitu sleeping client.

2.2.10 Rangkaian pada Modul Lampu Lalu Lintas

Rangkaian lampu lalu lintas yang terdapat di dalam modul lampu lalu lintas dapat dilihat pada Gambar 2.8. Ketiga modul lampu lalu lintas memiliki rangkaian dan konfigurasi pin yang sama.

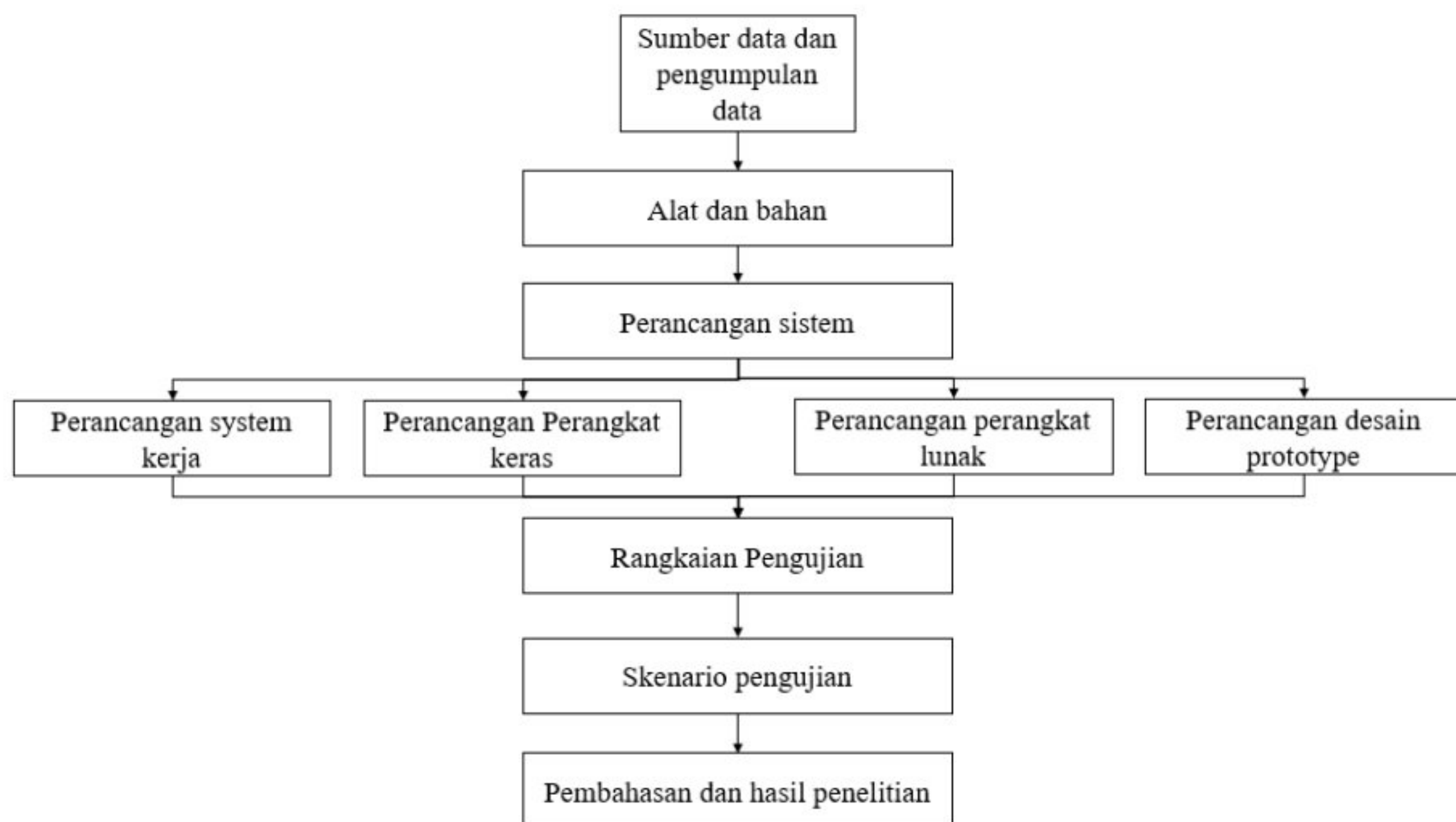


Gambar 2.9 Rangkaian Sistem

ESP32 diberi catu daya melalui port USB. Tengan keluaran port I/O modul ESP32 sebesar 3.3 V. Tegangan keluaran port I/O digunakan untuk menyalakan lampu LED merah, lampu LED kuning dan lampu LED hijau. Besar arus tipikal yang digunakan untuk menyalakan LED adalah 20mA. Oleh karena itu, terdapat resistor yang dihubungkan seri terhadap lampu LED yang berfungsi untuk membatasi arus pada lampu LED.

BAB III PERANCANGAN

Sistem yang akan dibuat pada penelitian ini adalah *Smart Modular Traffic Light* Berbasis ESP32. *Smart Modular Traffic* Berbasis ESP32 adalah lampu lalu lintas yang dibuat dalam bentuk modul. Modul memiliki komponen yang terdiri dari satu mikrokontroler ESP32, satu push button, tiga buah lampu LED yang terdiri dari satu LED merah, satu LED kuning dan satu LED hijau dan dua buah sensor IR. Setiap modul mewakili satu simpang lampu lalu lintas.



Gambar 3.1 Metodologi Penelitian

3.1 Sumber Data dan Pengumpulan Data

Sumber data yang digunakan pada penelitian ini merupakan data yang diambil langsung dari referensi-referensi terkait baik melalui media penelitian seperti buku, jurnal dan referensi terkait lainnya. Teknik pengumpulan data yang penulis gunakan dalam penelitian ini disebut dengan “studi kepustakaan, yang merujuk pada

proses pencarian data terkait dengan pertanyaan penelitian yang penulis ajukan.

3.2 Alat dan Bahan

Adapun alat-alat serta bahan yang digunakan pada penelitian ini dapat dilihat pada Tabel 3.1 :

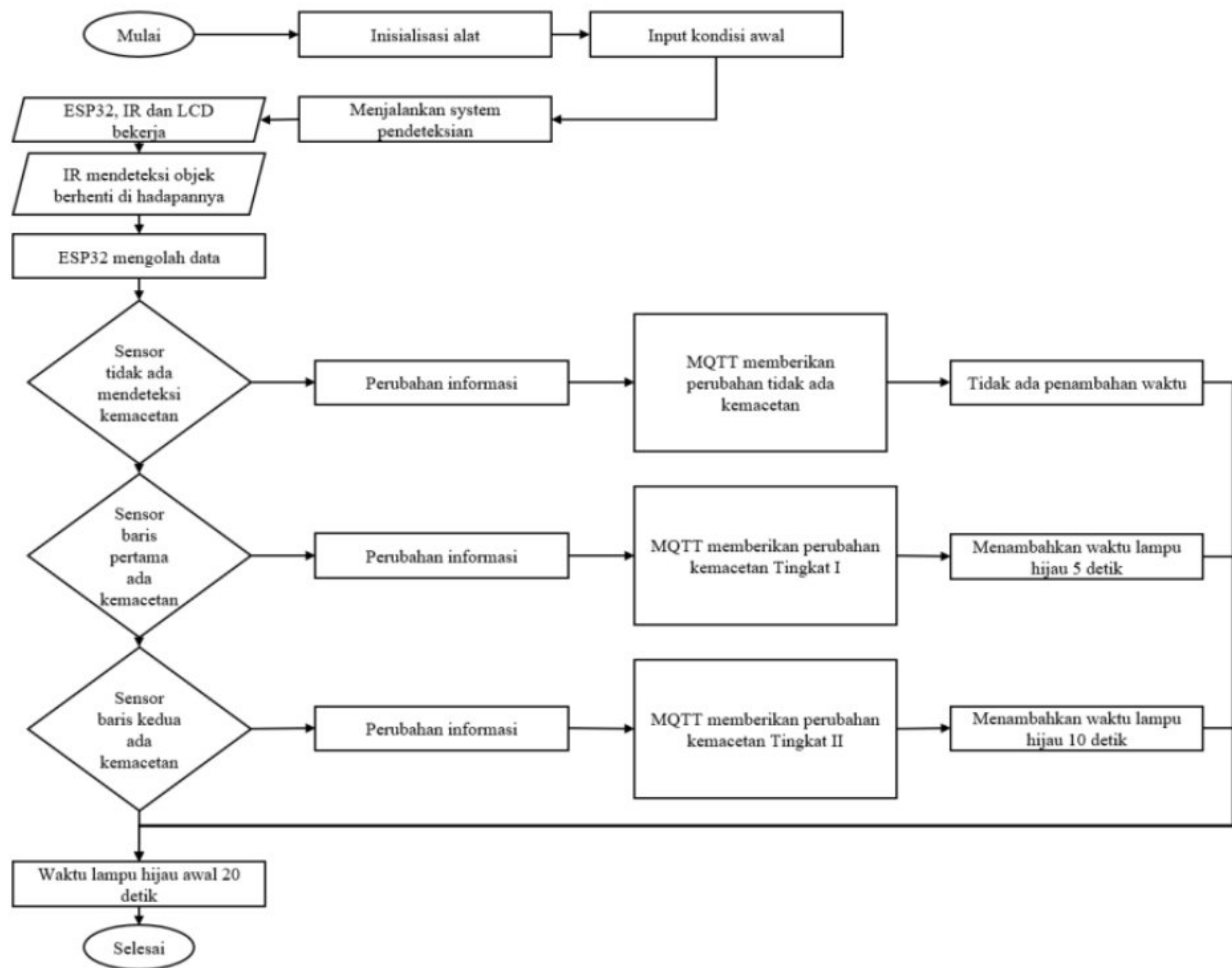
Tabel 3.1 Alat dan Bahan

No	Alat dan Bahan	Keterangan
1	Laptop Asus	Untuk memprogram mikrokontroler
2	ESP32	Sebagai mikrokontroler
3	Sensor Inframerah dan Photodiode	Sebagai sensor
4	LED	Sebagai lampu lalu lintas
5	Kabel	Sebagai penghubung komponen
6	Bread Board	Sebagai penghubung konduktor
7	Liquid Crystal Display (LCD) 4x20 dan dudukan	Sebagai tampilan data
8	Buzzer atau Bepper	Sebagai indikator
9	Power Supply	Sebagai input tegangan
10	PVC	Sebagai wadah
11	Multipleks	Sebagai media
12	Mobil mainan	Sebagai objek uji coba

3.3 Perancangan Sistem

3.3.1 Perancangan Sistem Kerja

Flowchart system kerja dari prototipe *Smart Traffic Light* berbasis *IoT* untuk optimalisasi penanggulangan kepadatan dapat dilihat pada Gambar 3.2.



Gambar 3.2 *Flowchart* Sistem Kerja Alat

Perancangan system kerja alat dapat dilihat pada Gambar 3.2. Adapun parameter dan penjelasan *Flowchart* pada penelitian ini merupakan parameter yang sama pada masing-masing simpang yang meliputi :

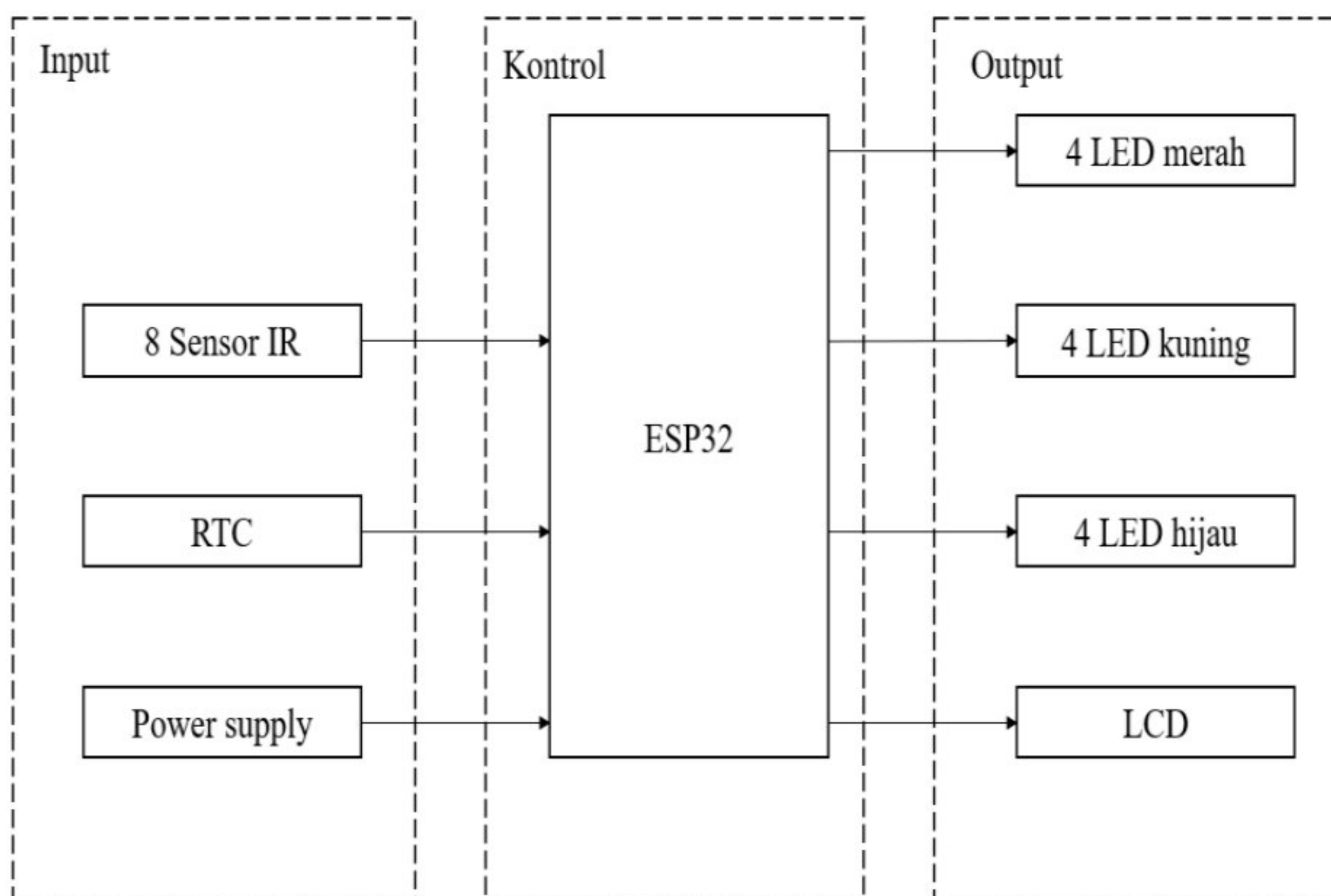
1. Inisialisasi sensor yang digunakan, serta mengatur control melalui Arduino dan ESP32.
2. Pada saat 2 sensor IR tidak mendeteksi kendaraan, maka kondisi lalu lintas dikatakan normal dan lampu hijau akan dibuat tetap tanpa perubahan yaitu 20 detik.
3. Pada saat 1 sensor IR yang di depan mendeteksi kendaraan, maka kondisi lalu lintas dikatakan kepadatan Macet Tingkat I dan lampu hijau akan dibuat penambahan selama 5 detik dan total menjadi 25 detik.
4. Pada saat 2 sensor IR mendeteksi kendaraan, maka kondisi lalu lintas dikatakan Macet Tingkat II dan lampu hijau akan

dibuat penambahan selama 10 detik dan total jalan menjadi 30 detik.

perancangan “Prototipe *Smart Traffic Light* Berbasis *IoT* untuk Optimalisasi Penanggulangan Kepadatan” yang terdiri dari blok diagram, perancangan hardware dan perancangan software. Prototipe yang dibuat adalah sistem lampu lalu lintas yang terdiri dari empat modul yang digunakan pada simpang empat. Setiap modul memiliki karakteristik yang sama berupa tampilan fisik dan software yang digunakan. Salah satu modul berfungsi sebagai *server* dan modul lainnya sebagai *client*. Keempat modul berkomunikasi secara *wireless* menggunakan Wi-Fi. Pengaturan pewaktuan untuk setiap modul lampu lalu lintas dilakukan melalui web.

3.3.2 Perancangan Hardware

Blok diagram penelitian ini dapat dilihat pada Gambar 3.3



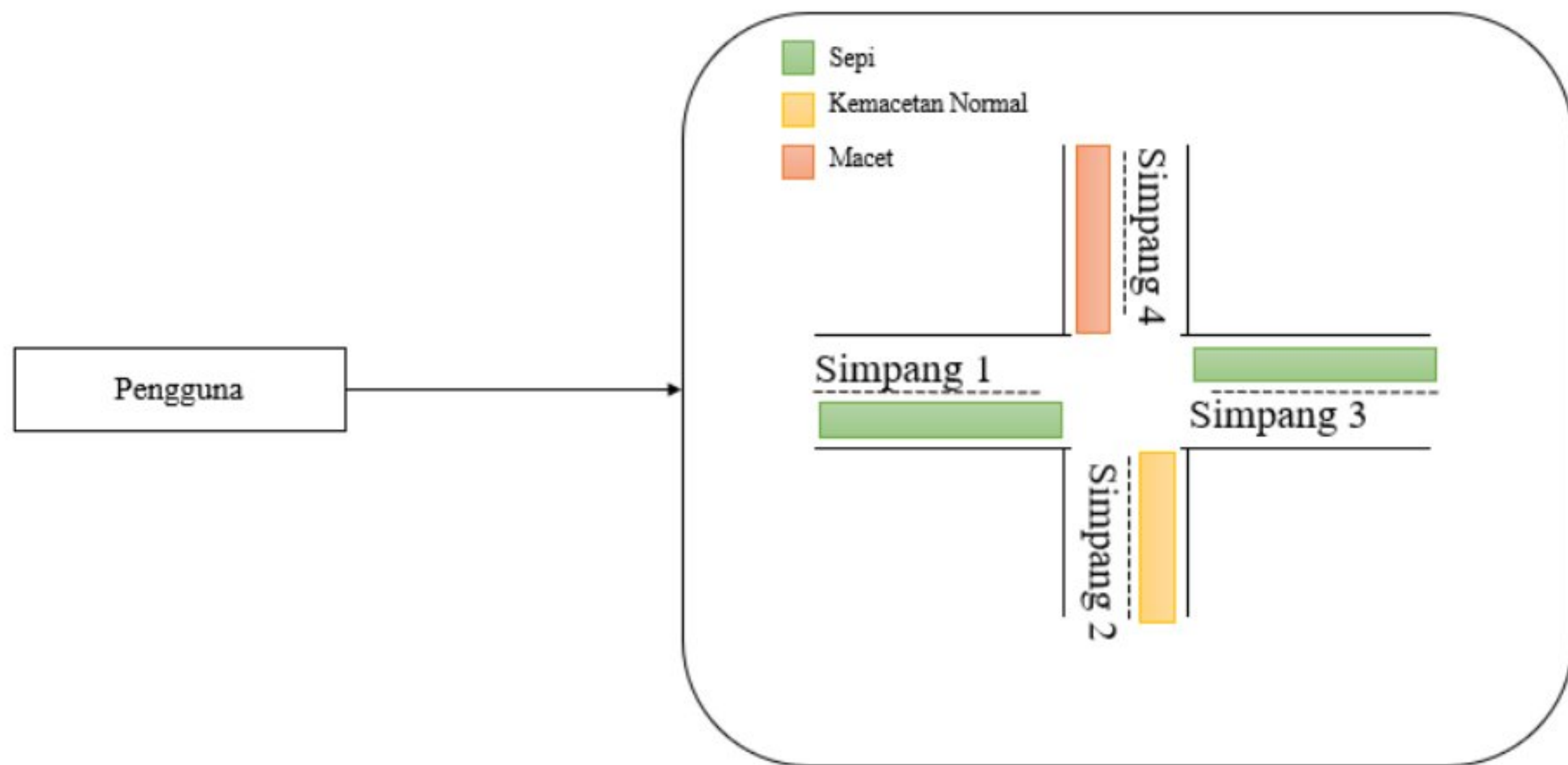
Gambar 3.3 Blok Diagram Penelitian

Berdasarkan blok diagram pada Gambar 3.3 menunjukkan input dan output dari penelitian ini. Adapun komponen yang digunakan sebagai input dan output pada penelitian ini sebagai berikut :

1. 8 Sensor IR digunakan sebagai input. Sensor IR digunakan untuk mendeteksi kepadatan, tiap simpang menggunakan 2 sensor IR sebagai pendeteksi kepadatan.
2. RTC digunakan sebagai input. RTC digunakan untuk mengatur waktu yang digunakan di dalam kontroller.
3. Power Supply digunakan sebagai input. Power Suplly digunakan untuk menyuplai tegangan pada ESP32.
4. ESP32 sebagai Kontrol. ESP32 adalah otak dari setiap komponen, ESP32 akan melakukan perintah pada masing-masing komponen berdasarkan program yang telah dimasukkan.
5. LED sebagai output digunakan sebagai tanda jalan kendaraan. Ketika LED berwarna merah, maka kendaraan berhenti, LED kuning menandakan hati-hati dan LED hijau menandakan jalan.
6. LCD sebagai output yang digunakan untuk melihat sisa lampu hijau pada masing-masing simpang.

3.3.3 Perancangan Software

Gambar 3.4 merupakan desain dari aplikasi optimasi *Traffic Light* berbasis website. Pengguna aplikasi ini adalah pengguna jalan yang akan melewati simpang lampumerah dari arah manapun. Pada saat pengguna jalan membuka website, pengguna akan ditampilkan monitoring kondisi kepadatan yang ada di simpang lampu merah. Tampilan website untuk monitoring kepadatan akan ditampilkan pada Gambar 3.4.

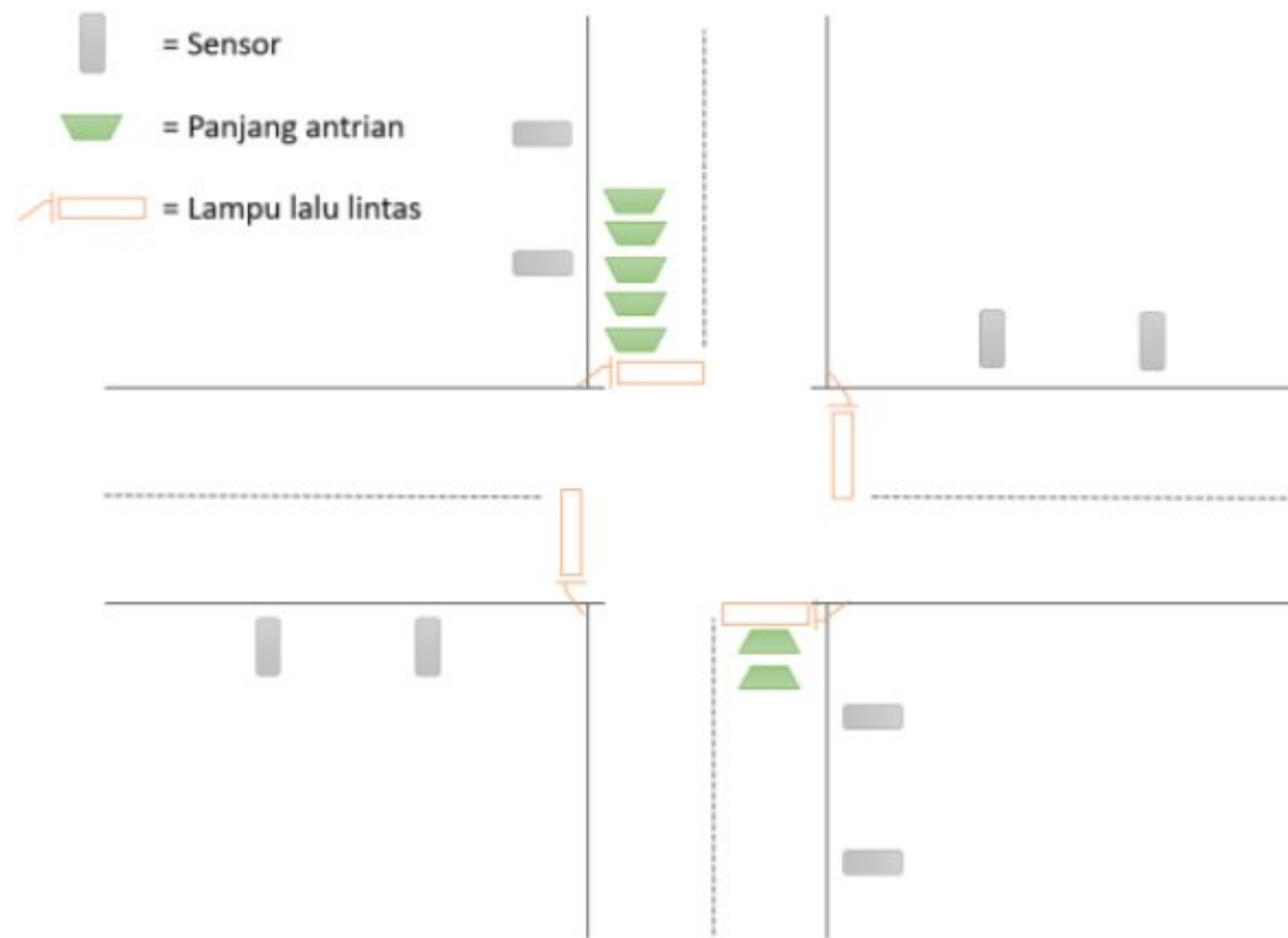


Gambar 3.4 Bentuk Perancangan Website Monitoring Kepadatan

Gambar 3.4 menunjukkan bentuk perancangan website monitoring kepadatan. Website ini digunakan bagi pengguna yang akan melalui simpang lalu lintas. Dalam website berisi kondisi kepadatan yang ada pada simpang lalu lintas. Perancangan digunakan menggunakan aplikasi VSCode Setelah itu kita perlu membuat file project baru di software VSCode, File yang dibuat disimpan dalam format php. Hal pertama membuat file koneksi ke database agar data pemantauan bisa ditampilkan pada website. membuat file indeks yaitu tampilan website untuk menampilkan data hasil monitoring. Jika file koneksi dan indeks berhasil dijalankan maka langkah selanjutnya membuat file indeks tabel yang mana data dari tabel yang akan ditampilkan berasal dari tabel database yang sudah dibuat selanjutnya.

3.3.4 Perancangan Prototype

Tahapan perancangan *Prototype* dilakukan untuk membuat gambaran dari *Prototype* yang akan dibuat. Rancangan ini dibuat berdasarkan hasil studi literatur dan studi bimbingan yang telah dilakukan sebelumnya dan dari berbagai permasalahan yang dikaji. Adapun bentuk rancangan *Prototype* yang akan dibuat seperti pada Gambar 3.5.



Gambar 3. 5 Bentuk Perancangan *Prototype*

Gambar 3.5 merupakan bentuk perancangan *Prototype*. Bentuk ini didesain dengan ukuran 80cm x 80 cm, dengan jarak satu buah simpang berjarak 30 cm dan lebar jalan sebesar 20 cm. Sensor yang digunakan pada satu simpang berjumlah dua sensor, yang di mana masing-masing sensor digunakan dengan jarak 10 cm. Hitungan jarak sensor pertama dimulai dari lampu merah dengan jarak 10 cm dan diikuti dengan sensor kedua dengan jarak 10 cm. Perancangan *Prototype* dirancang sama di setiap simpangnya sehingga bentuk simpang satu dengan yang lainnya akan dibuat sama.

3.4 Skenario Pengujian

Pengujian skenario dilakukan dengan mensimulasikan skenario berdasarkan kondisi nyata yang terjadi. Peneliti membuat tujuh skenario yang akan diuji pada Rancang Bangun *Sistem Smart Traffic Light Berbasis IoT* Sebagai Sistem Pendeteksi, diantaranya:

1. Kondisi normal seluruh jalur kendaraan tidak ada menyentuh satupun sensor
2. Kondisi kemacetan tingkat pertama pada jalur 4 dan jalur lainnya kondisi kendaraan tidak ada menyentuh satupun sensor
3. Kondisi kemacetan tingkat pertama pada jalur 3 dan 4 kemudian jalur lainnya kondisi kendaraan tidak ada menyentuh satupun sensor

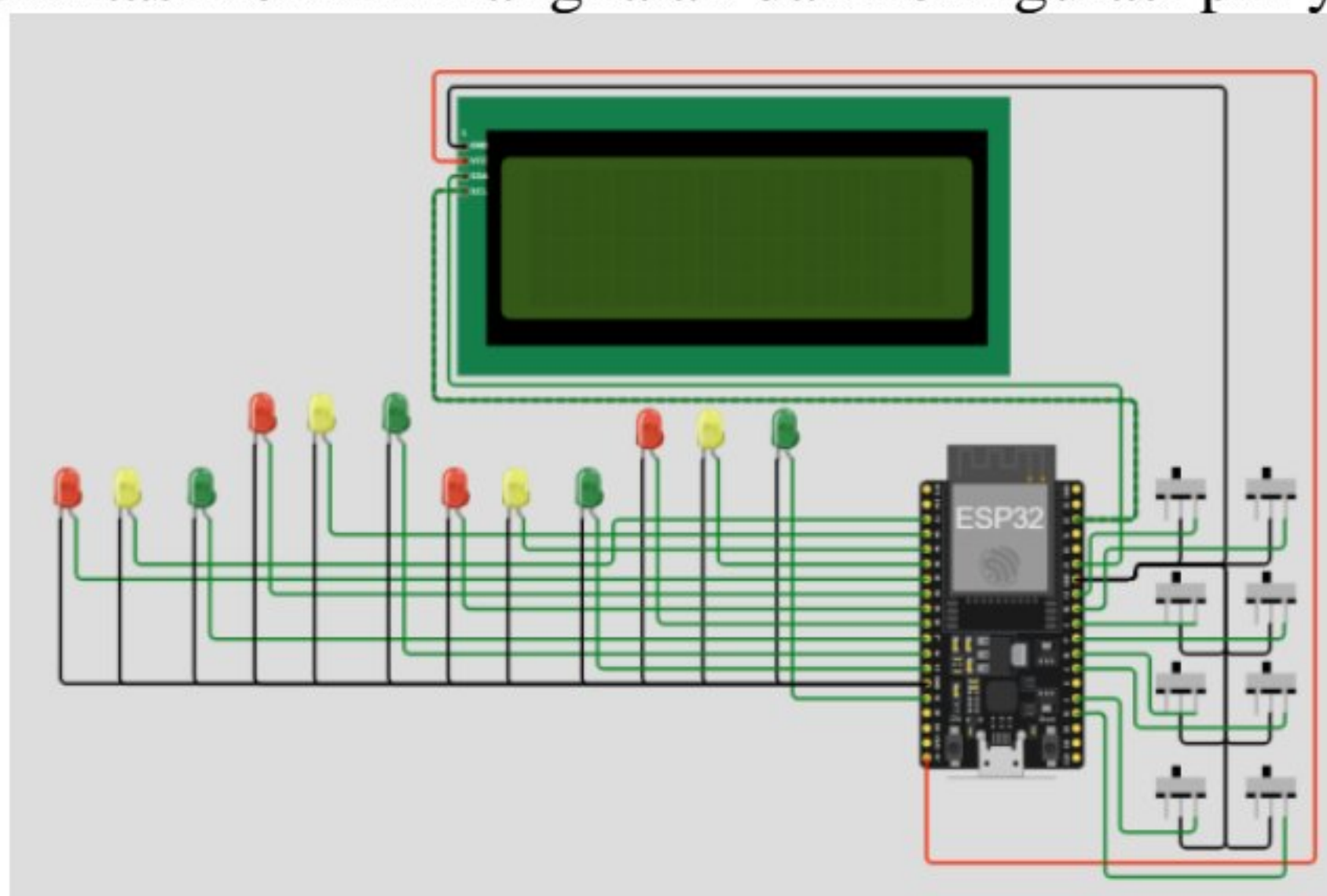
4. Kondisi kemacetan tingkat pertama pada jalur 2,3 dan 4 kemudian jalur 1 kondisi kendaraan tidak ada menyentuh satupun sensor
5. Kondisi kemacetan tingkat kedua pada jalur 4, kemacetan tingkat pertama pada jalur 2 dan 3 kemudian jalur 1 kondisi kendaraan tidak ada menyentuh satupun sensor

Tabel 3.2 Skenario Pengujian

Skenario	Jalur 1	Jalur 2	Jalur 3	Jalur 4
1	Sepi	Sepi	Sepi	Sepi
2	Sepi	Sepi	Sepi	Macet Tingkat I
3	Sepi	Sepi	Macet Tingkat I	Macet Tingkat I
4	Sepi	Macet Tingkat I	Macet Tingkat I	Macet Tingkat I
5	Sepi	Macet Tingkat I	Macet Tingkat I	Macet Tingkat II

3.5 Rangkaian pada Modul Lampu Lalu Lintas

Rangkaian lampu lalu lintas yang terdapat di dalam modul lampu lalu lintas dapat dilihat pada Gambar 3.6. Keempat modul lampu lalu lintas memiliki rangkaian dan konfigurasi pin yang sama.



Gambar 3.6 Rangkaian Sistem

ESP32 diberi catu daya melalui port USB. Tengan keluaran port I/O modul ESP32 sebesar 3.3 V. Tegangan keluaran port I/O digunakan untuk menyalakan lampu LED merah, lampu LED kuning

dan lampu LED hijau. Besar arus tipikal yang digunakan untuk menyalakan LED adalah 20mA. Oleh karena itu, terdapat resistor yang dihubungkan seri terhadap lampu LED yang berfungsi untuk membatasi arus pada lampu LED, rangkaian dilengkapi dengan dua buah sensor IR. Rangkaian penelitian di atas dibuat sebanyak empat jenis yang akan diletakkan di masing-masing simpang lalu lintas.

BAB IV

EKSPERIMEN DAN ANALISIS

Pada bab ini menjelaskan hasil dan pembahasan mengenai analisis dari Prototipe *Smart Traffic Light* Berbasis *Internet of Things (IoT)* untuk Penanggulangan Kemacetan.

Sub judul pada pembahasan ini yaitu hasil perancangan alat, Pengujian dan analisis masing-masing komponen, program, skenario pengujian dan hasil pengujian berdasarkan skenario pengujian.

4.1 Hasil Perancangan Alat

Hasil perancangan alat ini membahas mengenai semua analisis yang sebelumnya telah dirancang pada bab perancangan meliputi perancangan mekanik, perancangan skematik rangkaian dan perancangan *Interface*.

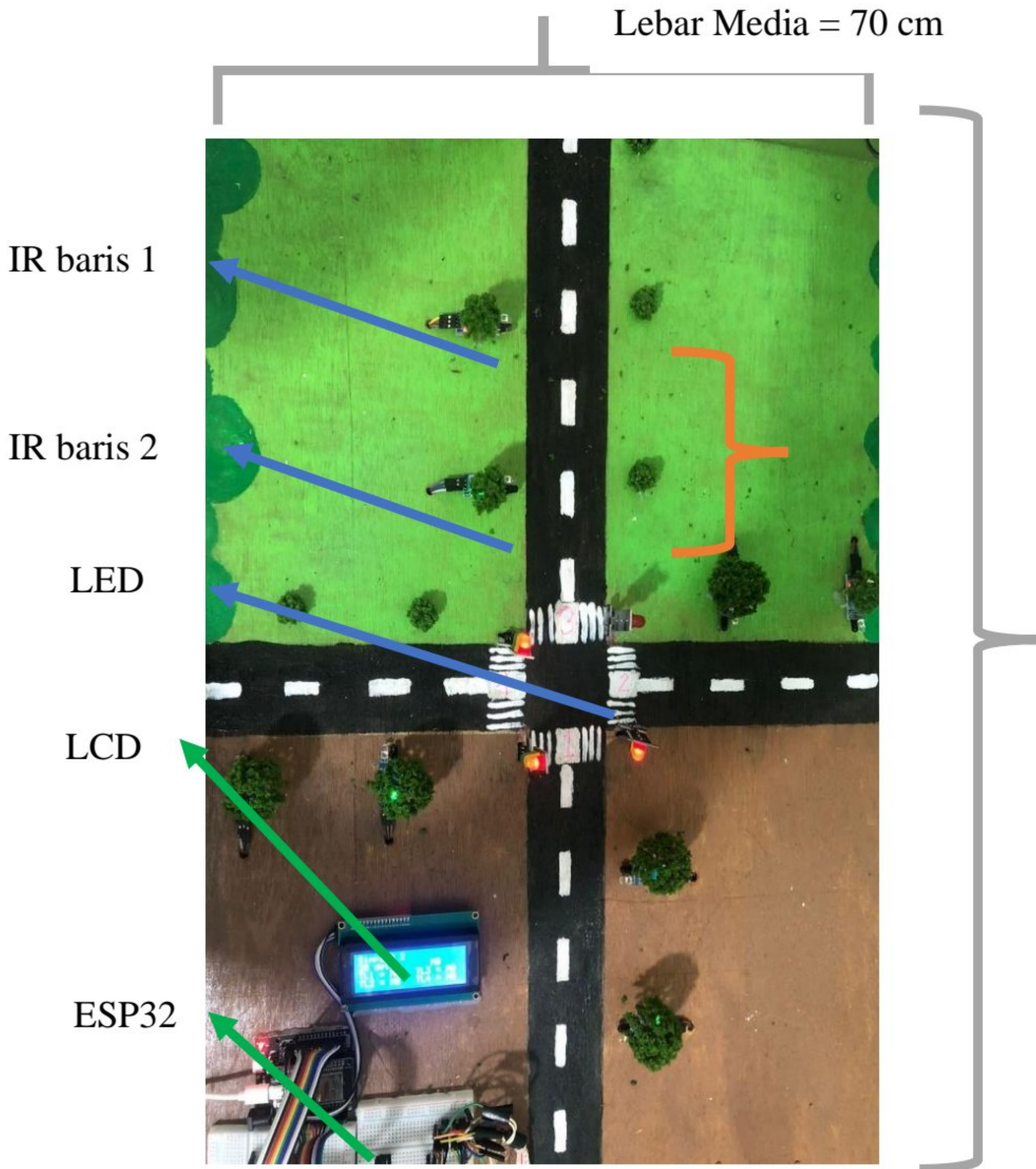
4.1.1 Hasil Perancangan Mekanik

Berdasarkan perancangan mekanik yang telah dilakukan, maka hasil perancangan dapat dilihat pada Gambar 4.1



Gambar 4.1 Hasil Perancangan Mekanik

Gambar 4.1 merupakan hasil perancangan mekanik sistem secara keseluruhan. Secara rinci perancangan pemodelan sistem dapat dilihat pada Gambar 4.2



Gambar 4.2 Hasil Perancangan Keseluruhan

Pada Gambar 4.2 Hasil perancangan alat berupa sistem terdiri dari model utama yaitu komponen elektrik, sensor IR sebagai pendeteksi kendaraan, LED sebagai lampu *Traffic Light* dan LCD sebagai interface jalur mana yang sedang berada di posisi jalan ataupun berhenti.

4.1.2 Hasil Perancangan Skematik Rangkaian

Berdasarkan hasil dari ilustrasi perancangan rangkaian skematik yang telah dirancang sebelumnya maka dapat dilihat pada Gambar 4.3 yang merupakan hasil rangkaian yang telah dirancang.



Gambar 4.3 Hasil Rangkaian

Pada hasil rancangan rangkaian Gambar 4.3, untuk merangkai komponen - komponen menggunakan papan PCB Board titik dengan ukuran 9 cm x 7 cm yang berfungsi untuk tempat meletakkan dan membuat jalur antar komponen – komponen. Di atas PCB Board lubang terdapat *Pin Header Male* dan *Pin Header Female* sebagai tempat meletakkan mikrokontroler ESP8622, LCD dan converter jack DC power. Peletakan sensor IR ditempatkan pada masing-masing jalur sebanyak 2 sensor di tiap jalurnya, total penggunaan sensor IR

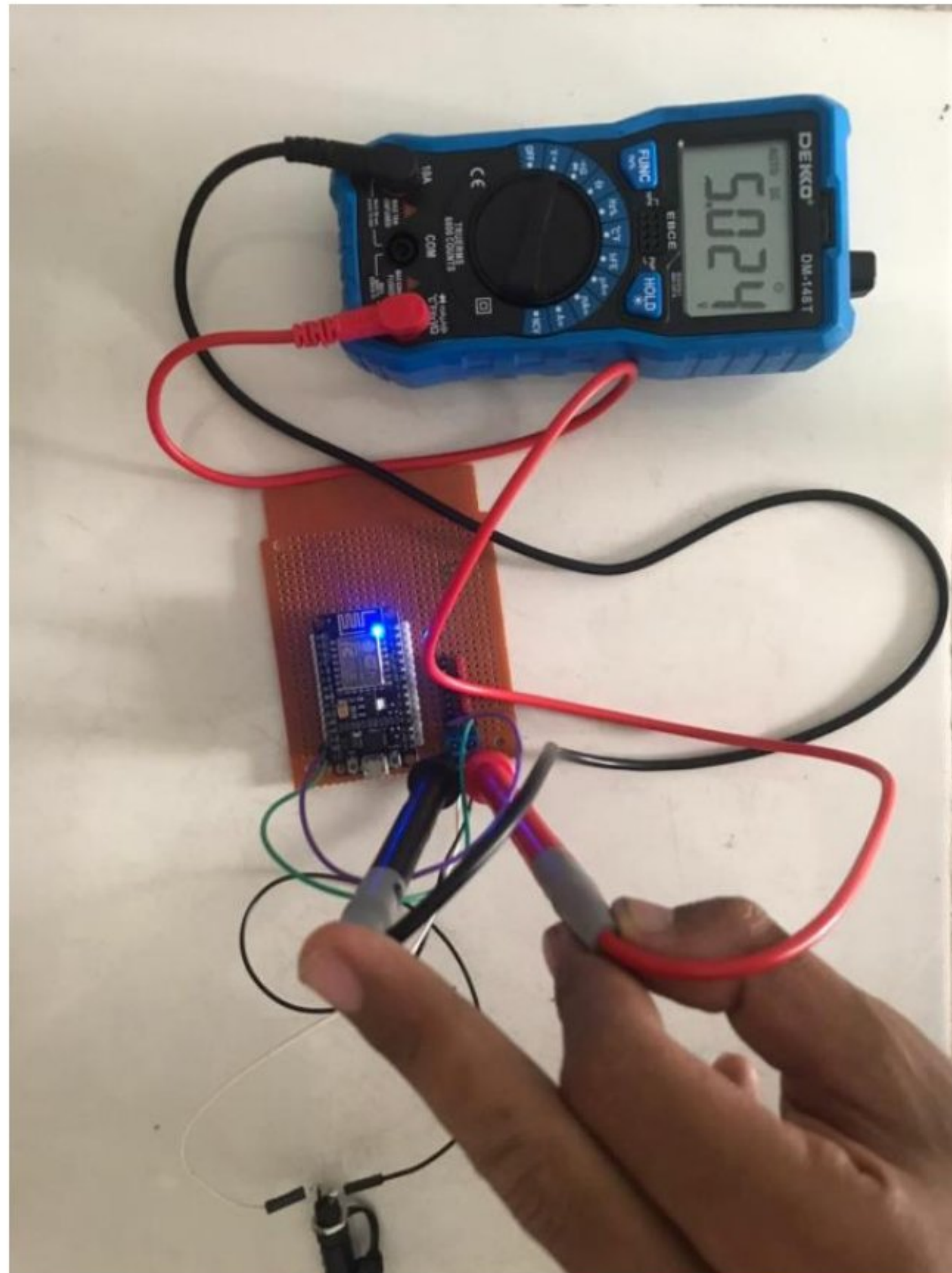
berjumlah 8 sensor. Untuk menyambungkan arus dari ESP32 ke komponen lainnya digunakan kabel *jumper* dengan ukuran 20cm dan 30cm. *Power supply* dari rangkaian ini menggunakan adaptor *Power supply* 5V DC.

4.2 Pengujian dan Analisis Hasil

Hasil pengujian alat ini membahas mengenai kinerja alat baik dari masing-masing komponen maupun keseluruhan alat dan untuk mengetahui efektifitas alat secara keseluruhan untuk objek yang diteliti.

4.2.1 Hasil Pengujian Catu Daya

Catu daya merupakan instrumen listrik yang berfungsi untuk menyuplai listrik ke suatu beban listrik. *Power supply* memiliki fungsi utama yaitu mengubah arus listrik dari sumber menjadi tegangan, arus, dan frekuensi yang sesuai untuk memberi daya pada beban. Adaptor *Power supply* merupakan adaptor untuk mengubah tegangan searah menjadi tegangan bolak-balik. Adaptor yang digunakan yaitu adaptor *Power supply* DC, di mana adaptor ini mengubah tegangan 220V (dari PLN) menjadi 5V tegangan DC yang digunakan untuk menghidupkan mikrokontroler dan komponen – komponen. Gambar 4.4 merupakan pengujian catu daya adaptor *Power supply*.



Gambar 4.4 Pengujian Catu Daya Adaptor *Power supply* DC

Pengujian catu daya adaptor *Power supply* DC ini dilakukan untuk mengukur tegangan output dari adaptor *Power supply* di mana pada data sheet tegangan output pada adaptor *Power supply* sebesar 5 V DC sesuai dengan pengukuran tegangan DC menggunakan multimeter yaitu 5.024 V DC. Berikut kebutuhan tegangan dan sumber tegangan yang dibutuhkan pada masing-masing komponen yang dibutuhkan.

4.2.2 Hasil Pengujian Sensor IR

Pengujian resistansi sensor *Infrared* dilakukan dengan cara mengukur resistansi yang terdapat pada sensor *Infrared*. Sensor

Infrared yang digunakan terdiri dari 4 buah kabel yaitu kabel merah sebagai input tegangan dari sensor *Infrared*, kabel hitam sebagai output tegangan dari sensor *Infrared*, kabel hijau sebagai output sinyal dari sensor *Infrared*, dan kabel putih sebagai input sinyal dari sensor *Infrared*.

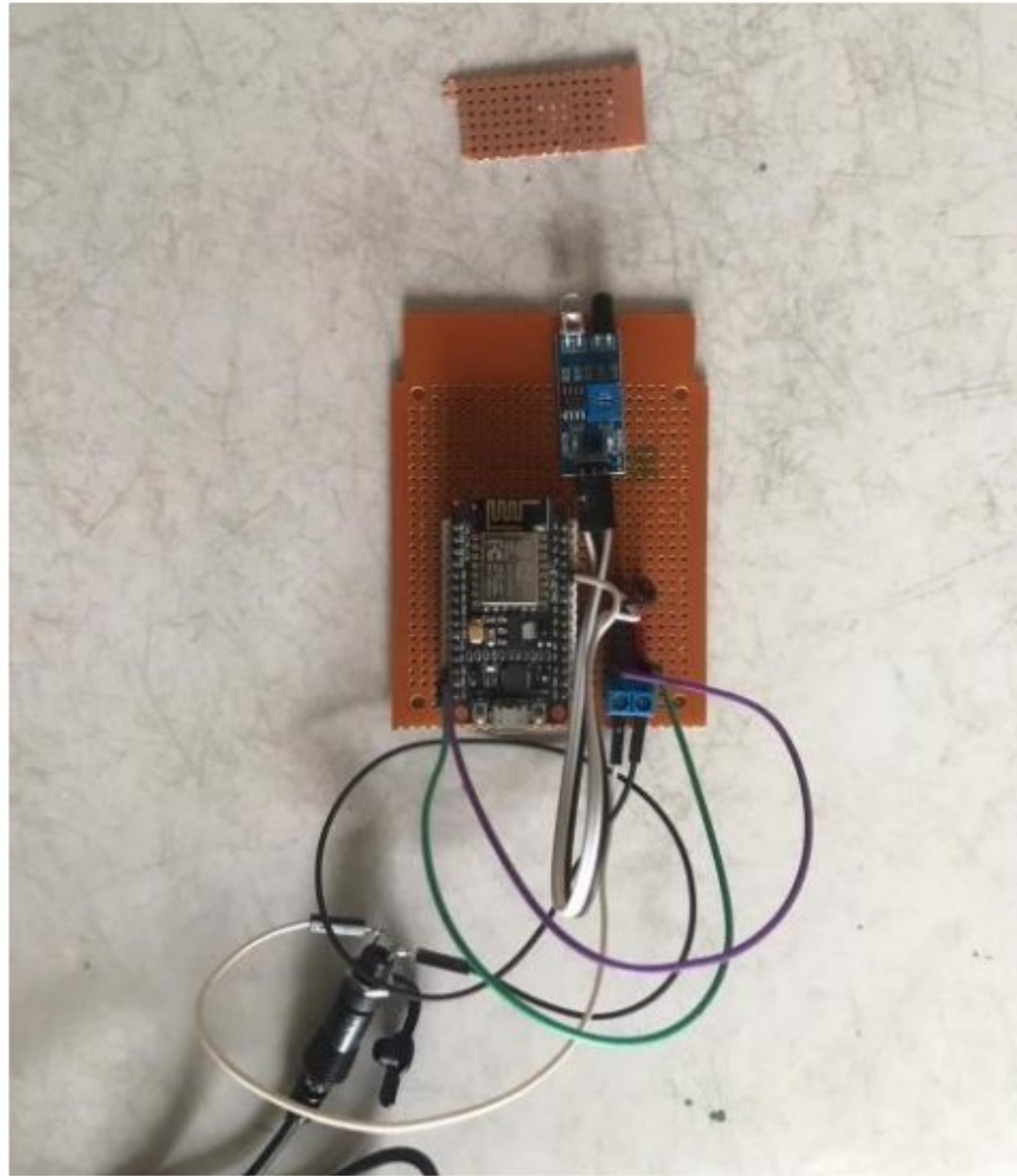


Gambar 4.5 Pengukuran Tahanan Output Sensor *Infrared*

Pada Gambar 4.5 merupakan hasil pengukuran tahanan output pada sensor *Infrared*. Di mana pada pengukuran tahanan input pada kabel merah dan hitam tahanan yang dihasilkan 10.19 Ω .

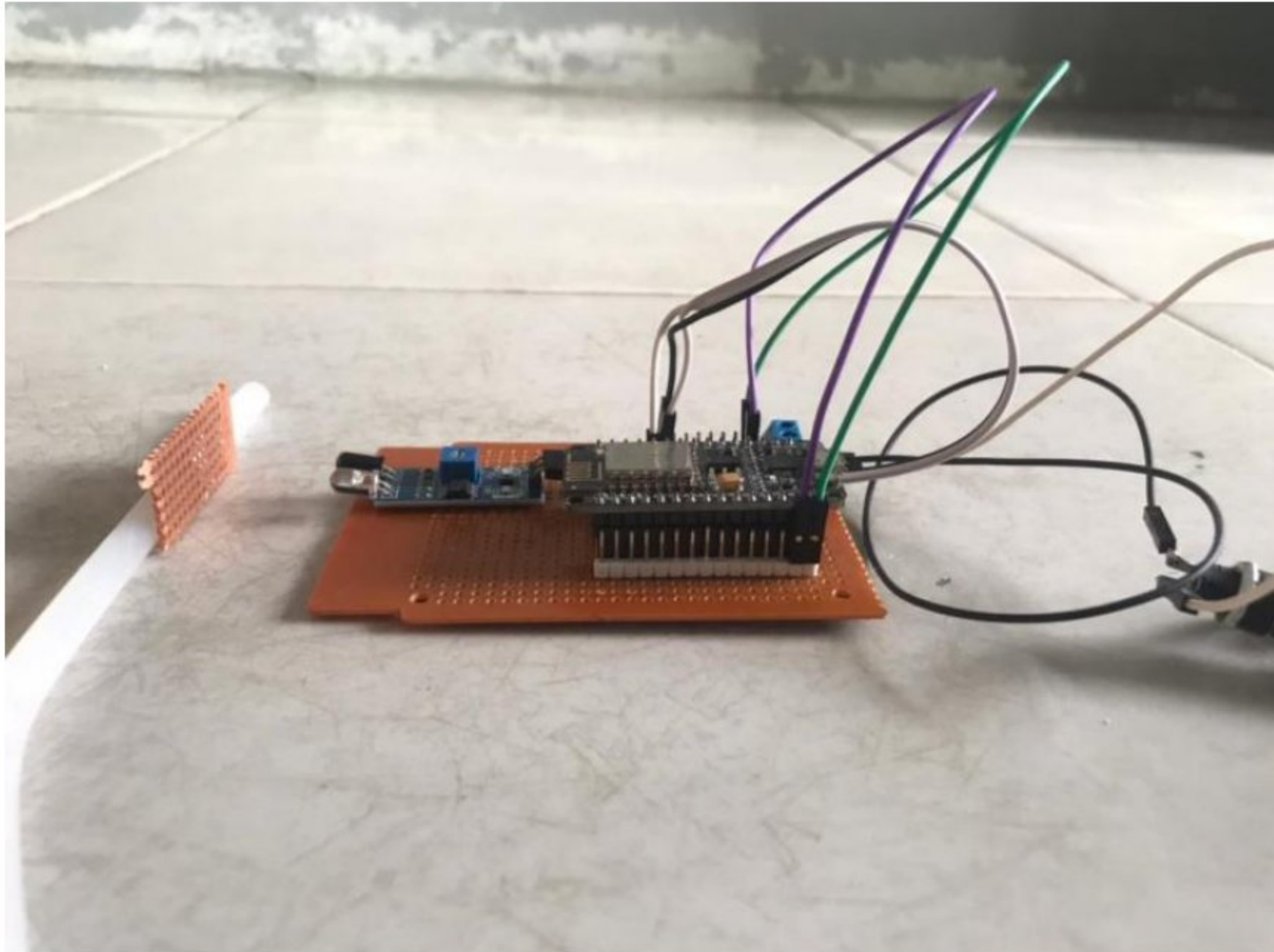
Pengujian tegangan keluaran sensor *Infrared* dilakukan untuk mengetahui perubahan keluaran tegangan yang dihasilkan sensor *Infrared* terhadap benda yang dilewatkan. Pada pengujian ini menggunakan beberapa tumpukan pakan ikan untuk mendeteksi pakan ikat yang sudah dilewatkan.

Pengujian kalibrasi sensor *Infrared* dilakukan untuk mendapatkan ketetapan antara kondisi kendaraan. Pada proses pengujian kalibrasi ini menggunakan beberapa objek penghalang.



Gambar 4. 6 Pengujian Kalibrasi Sensor *Infrared*

Proses pengujian kalibrasi sensor *Infrared* ini dilakukan dengan cara melewati objek yang diibaratkan kendaraan yang melewati sebuah jalur pada sensor *Infrared*. Melalui program kalibrasi ini disesuaikan dengan nilai faktor kalibrasi sensor *Infrared*.



Gambar 4. 7 Hasil Kalibrasi Sensor Infrared

4.2.3 Hasil Pengujian LCD

Pengujian LCD ini dilakukan dengan cara menjalankan alat. Jika LCD menampilkan tulisan atau angka yang sesuai dengan proses yang dilakukan alat maka dinyatakan berhasil. Pada Gambar 4.8 merupakan salah satu tampilan ketika alat bekerja.



Gambar 4.8 Pengujian LCD

Dapat dilihat bahwa pada LCD sudah bisa menampilkan tulisan yang sesuai dengan proses yang dilakukan. Dapat dinyatakan bahwa LCD bekerja sesuai dengan yang diharapkan.

4.3 Hasil Pengujian Skenario

Hasil wawancara dengan pihak Dinas Perhubungan Kota menunjukkan bahwa pengaturan durasi lampu hijau idealnya tidak bersifat tetap, melainkan perlu disesuaikan secara dinamis berdasarkan kondisi aktual di lapangan, terutama pada simpang-simpang padat seperti Simpang Empat Jalan Durian. Salah satu indikator kemacetan yang mudah diterapkan dan terukur secara teknis adalah panjang antrian kendaraan. Dalam wawancara, diperoleh klasifikasi kondisi lalu lintas sebagai berikut:

1. Normal: Antrian kendaraan < 10 meter
2. Macet Tingkat I: Antrian kendaraan sekitar 10–30 meter
3. Macet Tingkat II: Antrian kendaraan > 30 meter

Skenario ini kemudian disimulasikan dalam prototipe *Smart Traffic Light* (STL) yang dibangun menggunakan sensor Infrared (IR) dan modul IoT (ESP32 + Arduino), dengan logika kontrol yang menyesuaikan durasi lampu hijau secara otomatis berdasarkan jumlah sensor yang aktif.

Pengujian skenario dilakukan dengan mensimulasikan skenario berdasarkan kondisi nyata yang terjadi. Peneliti membuat tujuh skenario yang akan diuji pada Rancang Bangun *Sistem Smart Traffic Light* Berbasis *IoT* Sebagai Sistem Pendeteksi, diantaranya:

1. Kondisi normal seluruh jalur sepi
2. Kondisi kemacetan tingkat pertama pada jalur 4 dan jalur lainnya kondisi sepi
3. Kondisi kemacetan tingkat pertama pada jalur 3 dan 4 kemudian jalur lainnya kondisi sepi
4. Kondisi kemacetan tingkat pertama pada jalur 2,3 dan 4 kemudian jalur 1 kondisi sepi
5. Kondisi kemacetan tingkat kedua pada jalur 4, kemacetan tingkat pertama pada jalur 2 dan 3 kemudian jalur 1 kondisi sepi

Skenario tersebut juga bisa dilihat pada Tabel 4.1 Skenario pengujian.

Tabel 4.1 Skenario Pengujian

Skenario	Jalur 1	Jalur 2	Jalur 3	Jalur 4
1	Sepi	Sepi	Sepi	Sepi
2	Sepi	Sepi	Sepi	Macet tingkat 1
3	Sepi	Sepi	Macet tingkat 1	Macet tingkat 1
4	Sepi	Macet tingkat 1	Macet tingkat 1	Macet tingkat 1
5	Sepi	Macet tingkat 1	Macet tingkat 1	Macet tingkat 2

Untuk menyederhanakan implementasi dan menjaga keterandalan sistem pada tahap prototipe, digunakan dua buah sensor IR yang diletakkan pada dua titik berbeda di jalur pendekat (approach) simpang.

1. Sensor 1 (S1): Ditempatkan pada titik terdepan, 10 meter dari lampu lalu lintas
2. Sensor 2 (S2): Ditempatkan pada titik lebih jauh, 30 meter dari lampu lalu lintas

Logika pengambilan keputusan sistem terhadap kondisi lalu lintas berdasarkan sensor dapat dijelaskan sebagai berikut:

Tabel 4. 2 Durasi Lampu Hijau

Sensor S1	Sensor S2	Kategori Kepadatan	Durasi Lampu Hijau
Tidak Aktif	Tidak Aktif	Normal	20 detik
Aktif	Tidak Aktif	Macet Tingkat I	25 detik
Aktif	Aktif	Macet Tingkat II	30 detik

Berdasarkan skenario tersebut, dapat dianalisis bahwa sistem dirancang untuk beradaptasi secara real-time terhadap dinamika lalu lintas tanpa intervensi manual. Beberapa poin analisis teknis sebagai berikut:

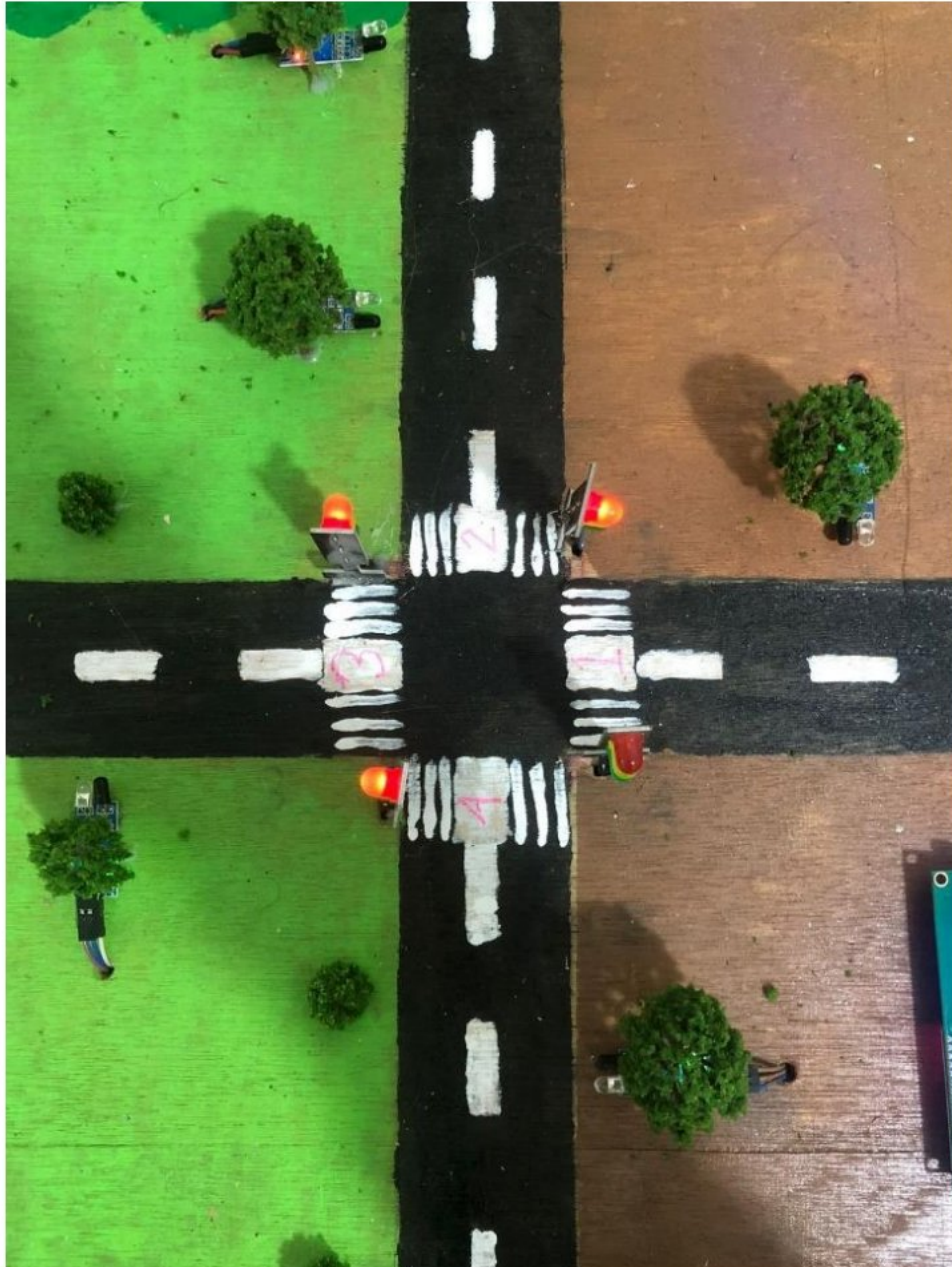
1. Sensor IR dipilih karena kecepatan responnya yang tinggi dan kemudahan implementasi. Namun, sensor IR memiliki keterbatasan dalam hal kondisi lingkungan (terganggu hujan, kabut, atau bayangan), sehingga akurasi dapat bervariasi. Untuk implementasi lebih lanjut di lapangan, perlu dipertimbangkan penggunaan sensor LIDAR atau kamera berbasis vision dengan pengolahan citra.
2. ESP32 memiliki kemampuan koneksi WiFi dan pengolahan logika kontrol secara lokal melalui Arduino IDE. Ini memungkinkan integrasi ke sistem monitoring pusat Dishub di masa depan, seperti dashboard status kemacetan.
3. Penambahan waktu lampu hijau dilakukan secara incremental (5 detik dan 10 detik). Meskipun sederhana, metode ini efektif sebagai sistem reaktif terhadap antrian.

Dari hasil wawancara, petugas menyatakan bahwa sistem saat ini bersifat statis dan kerap tidak mampu menyesuaikan dengan kondisi nyata. Dengan skenario STL ini, sistem mulai mendekati bentuk pengendalian lalu lintas berbasis situasi (situation-aware traffic control), yang sesuai dengan arah kebijakan Dinas Perhubungan menuju Smart City.

Sistem ini juga dapat mengurangi beban kerja manual serta meningkatkan efisiensi waktu perjalanan dan pengurangan bahan bakar kendaraan akibat idle time saat menunggu lampu merah terlalu lama di jalur yang sepi.

4.3.1 Hasil Skenario Pertama

Skenario pertama dilakukan dengan kondisi seluruh jalur dalam kondisi sepi, skenario ini adalah skenario di mana lampu lalu lintas yang ada biasanya tanpa adanya optimasi apapun. Skenario pengujian ini dapat dilihat pada Gambar 4.9.



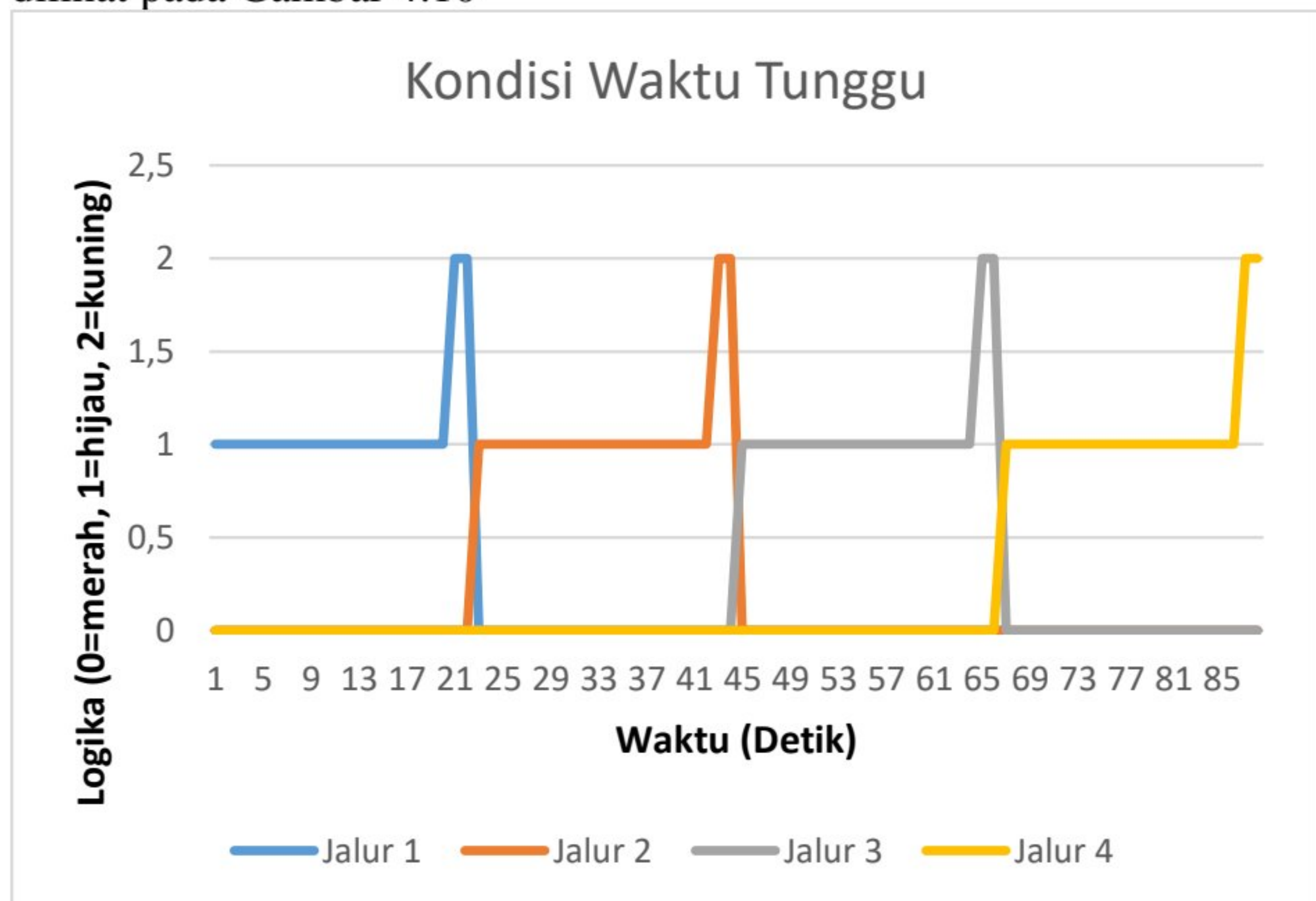
Gambar 4.9 Skenario Pertama

Gambar 4.9 Menunjukkan bahwa seluruh jalur dalam keadaan sepi, tidak ada kendaraan yang menghambat sensor atau terdeteksi kemacetan. Hasil lampu merah dalam satu kali siklus dapat dilihat pada Tabel 4.2.

Tabel 4.3 Hasil Pengujian Skenario Pertama

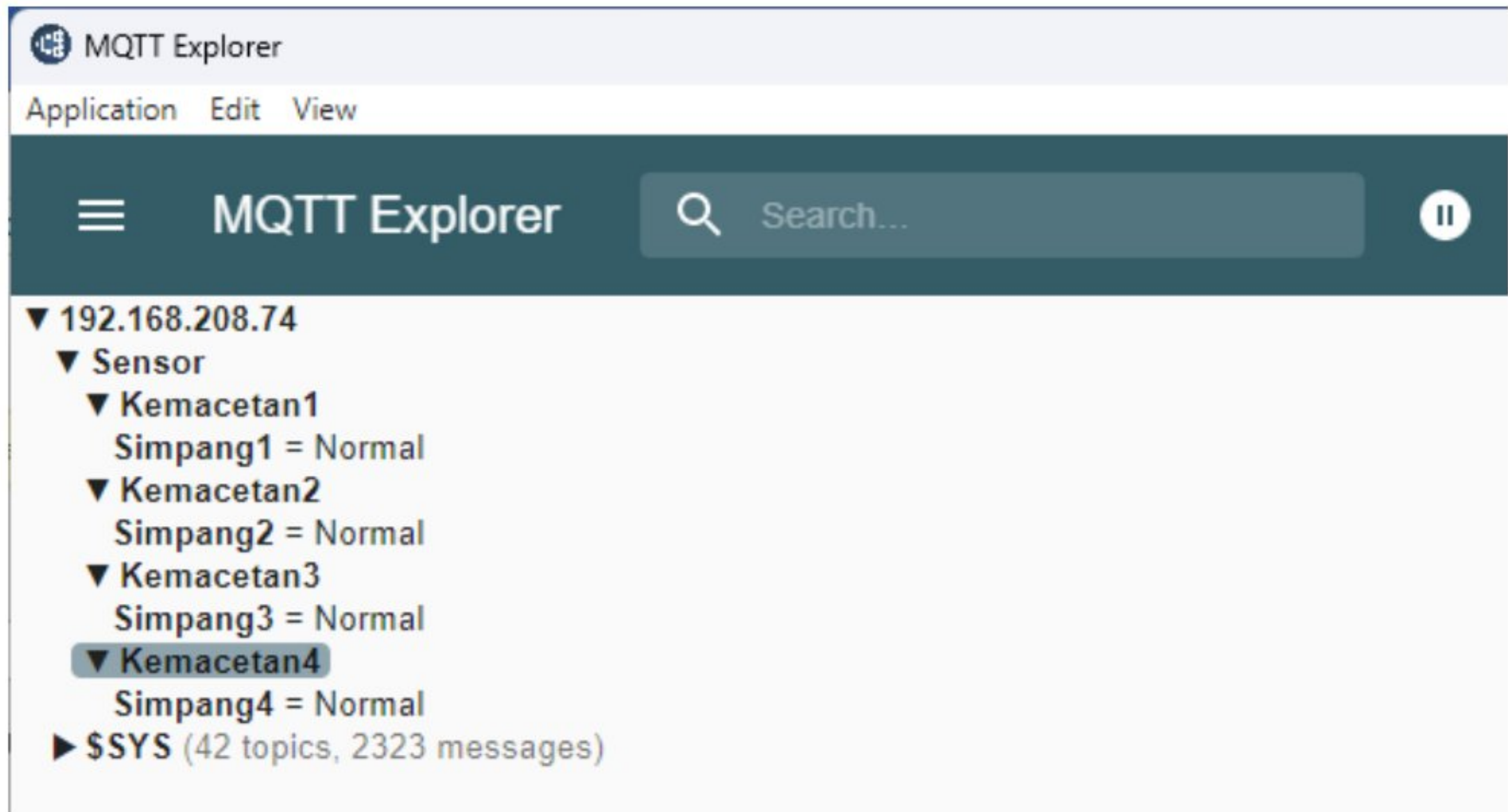
Jalur	Lalu Lintas	Lampu hijau (s)	Lampu merah (s)
Jalur 1	Sepi	20	60
Jalur 2	Sepi	20	60
Jalur 3	Sepi	20	60
Jalur 4	Sepi	20	60

Pada Tabel 4.4 menjelaskan bahwa seluruh jalur dalam keadaan sepi, sehingga lampu merah di setiap jalur sama yaitu pada nilai 20 detik, kemudian lampu merah di setiap jalur juga sama yaitu pada nilai 60 detik. Grafik hasil pengujian skenario pertama dapat dilihat pada Gambar 4.10



Gambar 4.10 Grafik Hasil Pengujian Skenario Pertama

Gambar 4.10 Menunjukkan bahwa jalur dalam kondisi stabil, tidak ada kemacetan yang terjadi, sehingga lampu hijau seluruh jalur sama. Tampilan pada MQTT dapat dilihat pada Gambar 4.11 :



Gambar 4.11 Tampilan MQTT Skenario Pertama

4.3.2 Hasil Skenario Kedua

Skenario kedua dilakukan dengan kondisi jalur 4 kemacetan tingkat pertama dan jalur lainnya dalam keadaan sepi, skenario ini adalah skenario di mana terjadi kemacetan tingkat pertama pada jalur 4. Kemacetan terjadi karena ada kendaraan yang berhenti pada sensor kemacetan pertama pada jalur 4 sehingga sistem akan mendeteksi adanya kemacetan tingkat pertama pada jalur 4. Skenario pengujian ini dapat dilihat pada Gambar 4.12



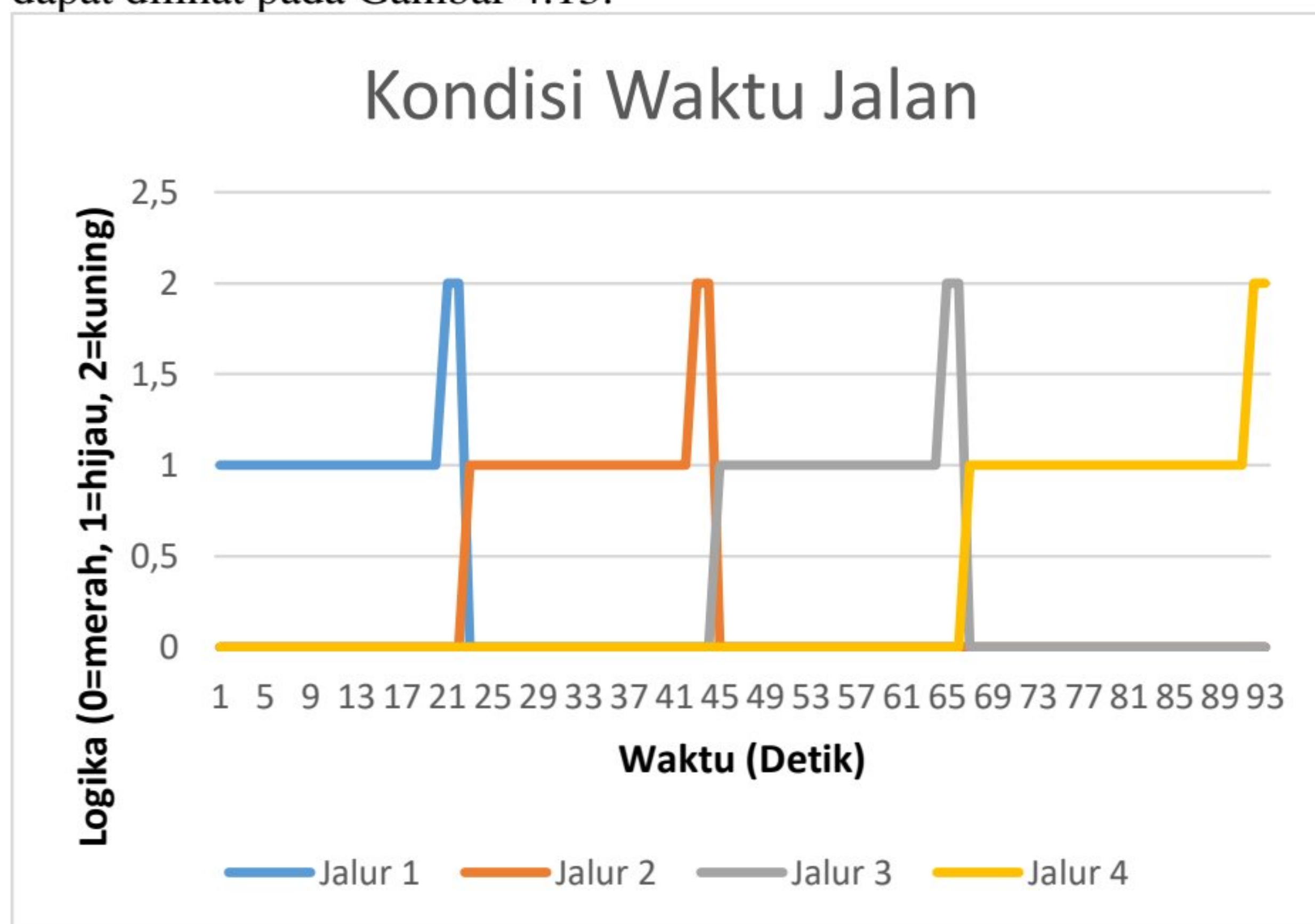
Gambar 4.12 Skenario Kedua

Gambar 4.12 Menunjukkan bahwa jalur 4 terjadi kemacetan tingkat 1 dan jalur lainnya dalam keadaan sepi, terdapat satu jalur yaitu jalur 4 yang berhenti dan menunggu pada sensor kemacetan tingkat 1 di jalur 4. Hasil lampu merah dalam skenario ini dapat dilihat pada Tabel 4.3.

Tabel 4.4 Hasil Pengujian Skenario Kedua

Jalur	Lalu Lintas	Lampu hijau (s)	Lampu merah (s)
Jalur 1	Sepi	20	65
Jalur 2	Sepi	20	65
Jalur 3	Sepi	20	65
Jalur 4	Macet Tingkat I	25	60

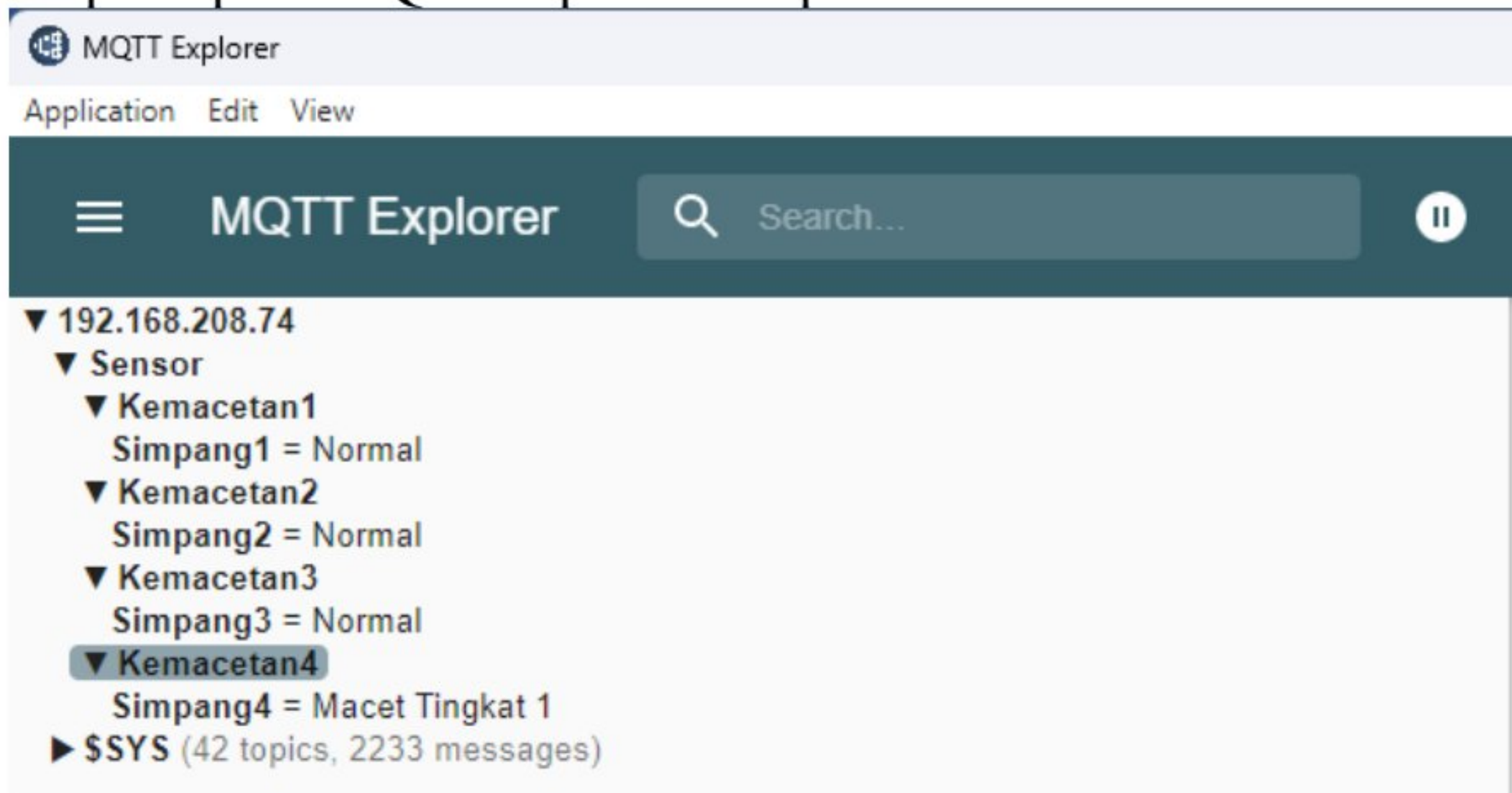
Tabel 4.5 menjelaskan bahwa jalur 4 terjadi kemacetan tingkat pertama dan jalur lainnya dalam keadaan sepi, sehingga lampu hijau pada jalur 4 menjadi lebih lama dari jalur yang lainnya menjadi total lampu hijau bernilai 25 detik. Grafik hasil pengujian skenario kedua dapat dilihat pada Gambar 4.13.



Gambar 4.13 Grafik Hasil Pengujian Skenario Kedua

Gambar 4.13. Menunjukkan bahwa jalur 4 dalam kondisi terjadi kemacetan tingkat 1 sementara itu jalur lainnya masih dalam kondisi sepi. Hal ini membuat lampu hijau jalur 4 menjadi lebih lama

sehingga kemacetan yang terjadi pada jalur 4 tidak bertambah parah. Tampilan pada MQTT dapat dilihat pada Gambar 4.14 :



Gambar 4.14 Hasil MQTT Skenario Kedua

4.3.3 Hasil Skenario Ketiga

Skenario ketiga dilakukan dengan kondisi jalur 3 dan 4 berada pada kemacetan tingkat pertama, jalur 1 dan 2 dalam keadaan sepi, skenario ini adalah skenario di mana terjadi kemacetan tingkat pertama pada jalur 3 dan 4. Kemacetan terjadi karena ada kendaraan yang berhenti pada sensor kemacetan pertama pada jalur 3 dan 4 sehingga sistem akan mendeteksi adanya kemacetan tingkat pertama pada jalur 3 dan 4. Skenario pengujian ini dapat dilihat pada Gambar 4.15



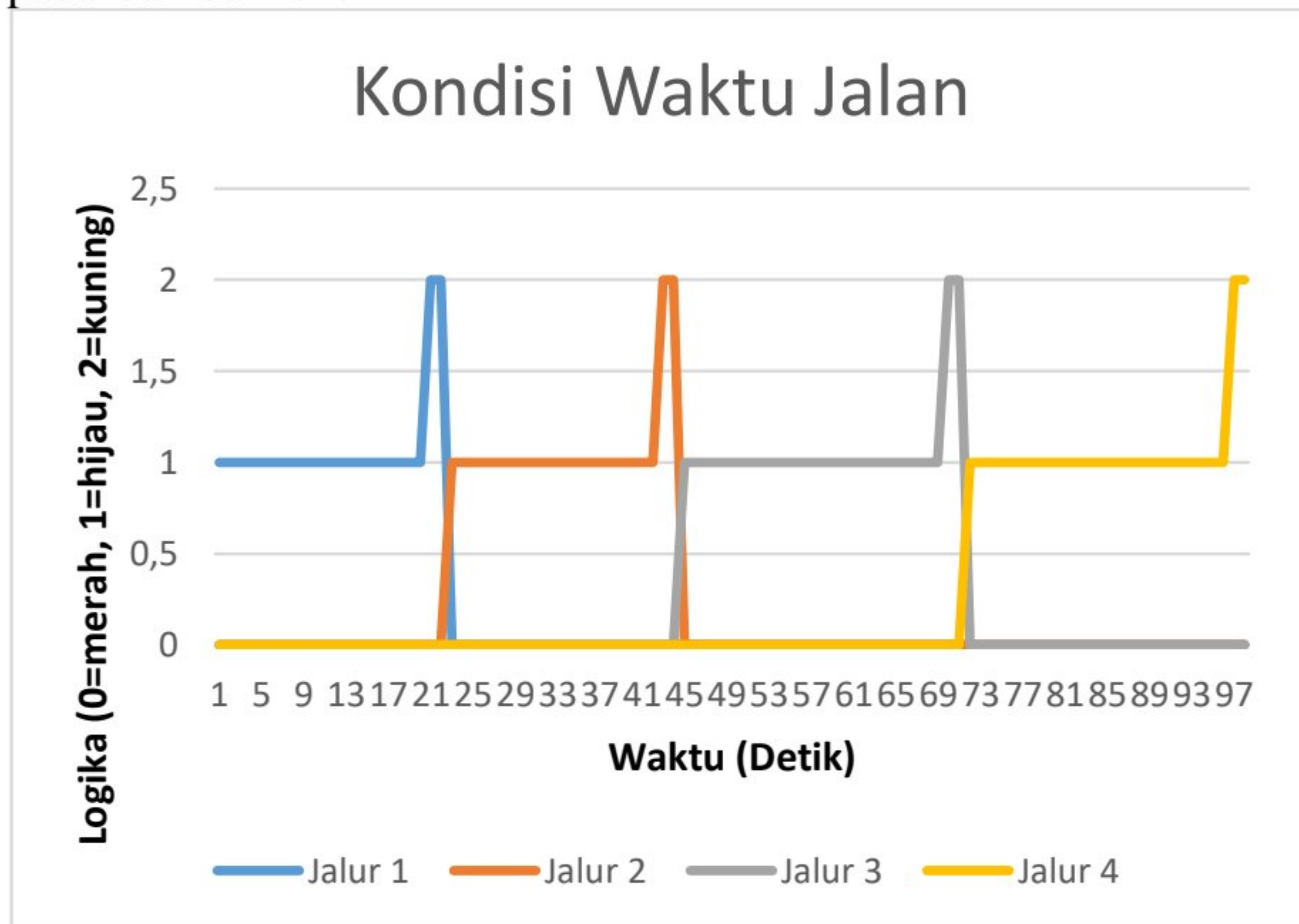
Gambar 4.15 Skenario Ketiga

Gambar 4.15 Menunjukkan bahwa jalur 3 dan 4 terjadi kemacetan tingkat 1, jalur 1 dan 2 dalam keadaan sepi, terdapat dua jalur yaitu jalur 3 dan 4 yang diamana ada kendaraan berhenti dan menunggu pada sensor kemacetan tingkat 1. Hasil lampu merah dalam skenario ini dapat dilihat pada Tabel 4.4.

Tabel 4.5 Hasil Pengujian Skenario Keditga

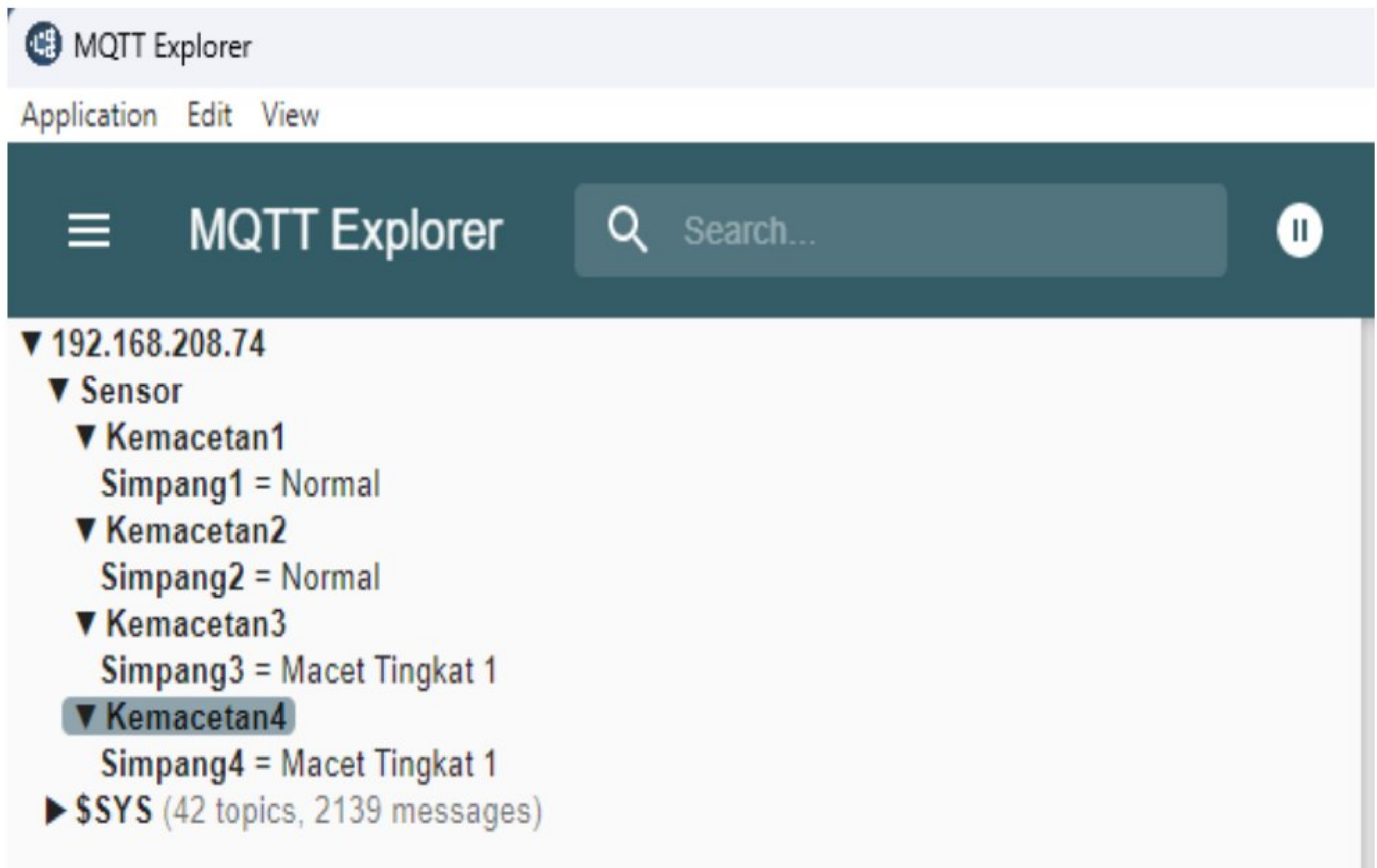
Jalur	Lalu Lintas	Lampu hijau (s)	Lampu merah (s)
Jalur 1	Sepi	20	70
Jalur 2	Sepi	20	70
Jalur 3	Macet Tingkat I	25	65
Jalur 4	Macet Tingkat I	25	65

Tabel 4.6 menjelaskan bahwa jalur 3 dan 4 terjadi kemacetan tingkat pertama. jalur 1 dan 2 dalam keadaan sepi, sehingga lampu hijau pada jalur 3 dan 4 menjadi lebih lama dengan total lampu hijau bernilai 25 detik. Grafik hasil pengujian skenario ketiga dapat dilihat pada Gambar 4.16



Gambar 4.16 Grafik Hasil Pengujian Skenario Ketiga

Gambar 4.16 Menunjukkan bahwa jalur 3 dan 4 dalam kondisi terjadi kemacetan tingkat 1 sementara itu jalur lainnya masih dalam kondisi sepi. Hal ini membuat lampu hijau jalur 3 dan 4 menjadi lebih lama sehingga kemacetan tingkat pertama yang terjadi pada jalur 3 dan 4 tidak bertambah parah. Tampilan pada MQTT dapat dilihat pada Gambar 4.17 :



Gambar 4.17 Hasil MQTT Skenario Ketiga

4.3.4 Hasil Skenario Keempat

Skenario keempat dilakukan dengan kondisi jalur 2, 3 dan 4 berada pada kemacetan tingkat pertama, jalur 1 dalam keadaan sepi, skenario ini adalah skenario di mana terjadi kemacetan tingkat pertama pada jalur 2, 3 dan 4. Kemacetan terjadi karena ada kendaraan yang berhenti pada sensor kemacetan pertama pada jalur 2, 3 dan 4 sehingga sistem akan mendeteksi adanya kemacetan tingkat pertama pada jalur 2, 3 dan 4. Skenario pengujian ini dapat dilihat pada Gambar 4.18.



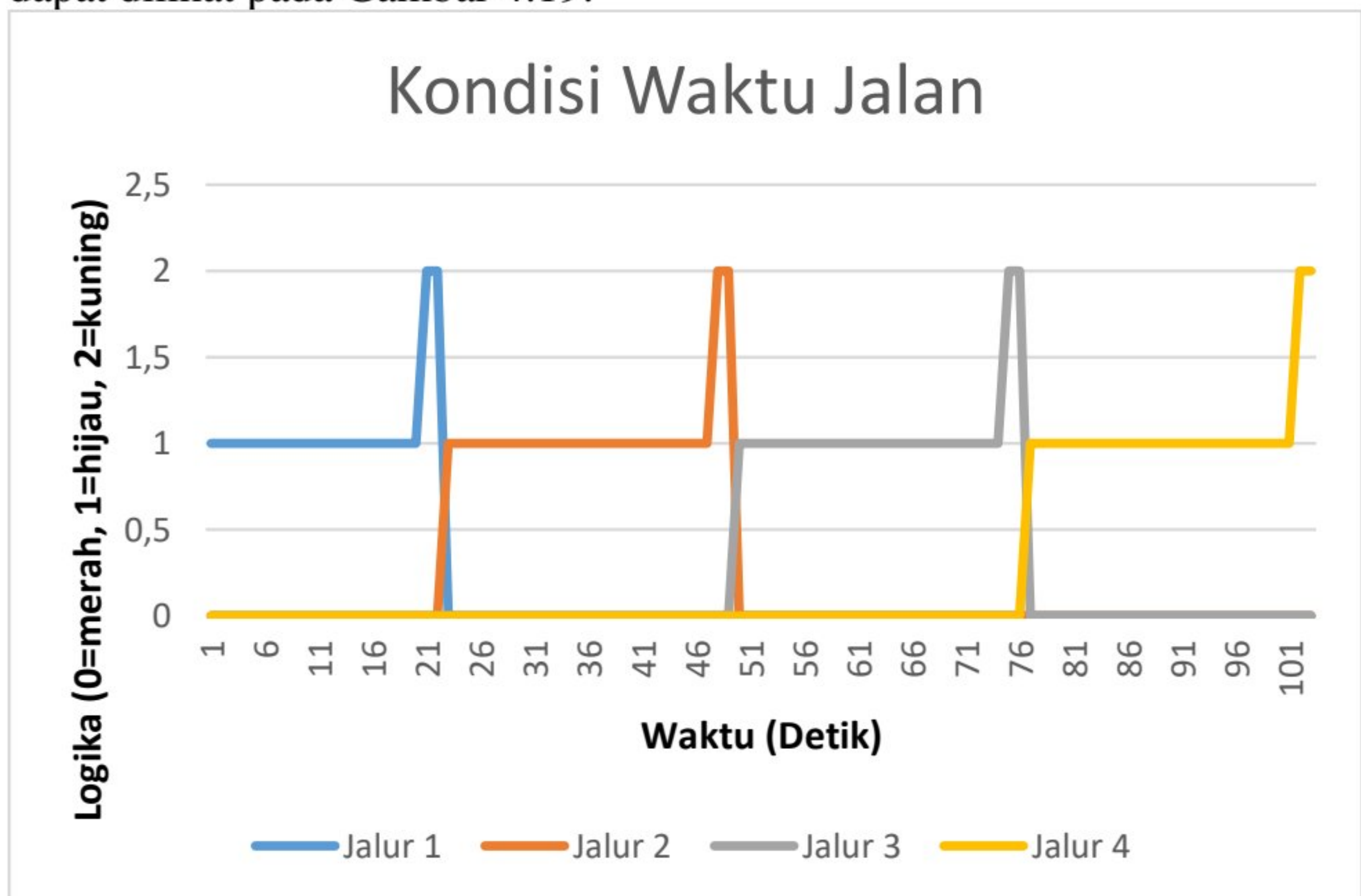
Gambar 4.18 Skenario Keempat

Gambar 4.18 Menunjukkan bahwa jalur 2, 3 dan 4 terjadi kemacetan tingkat 1, jalur 1 dalam keadaan sepi, terdapat tiga jalur yaitu jalur 2, 3 dan 4 yang diamana ada kendaraan berhenti dan menunggu pada sensor kemacetan tingkat 1. Hasil lampu merah dalam skenario ini dapat dilihat pada Tabel 4.5.

Tabel 4.6 Hasil Pengujian Skenario Ketujuh

Jalur	Lalu Lintas	Lampu hijau (s)	Lampu merah (s)
Jalur 1	Sepi	20	75
Jalur 2	Macet Tingkat I	25	70
Jalur 3	Macet Tingkat I	25	70
Jalur 4	Macet Tingkat I	25	70

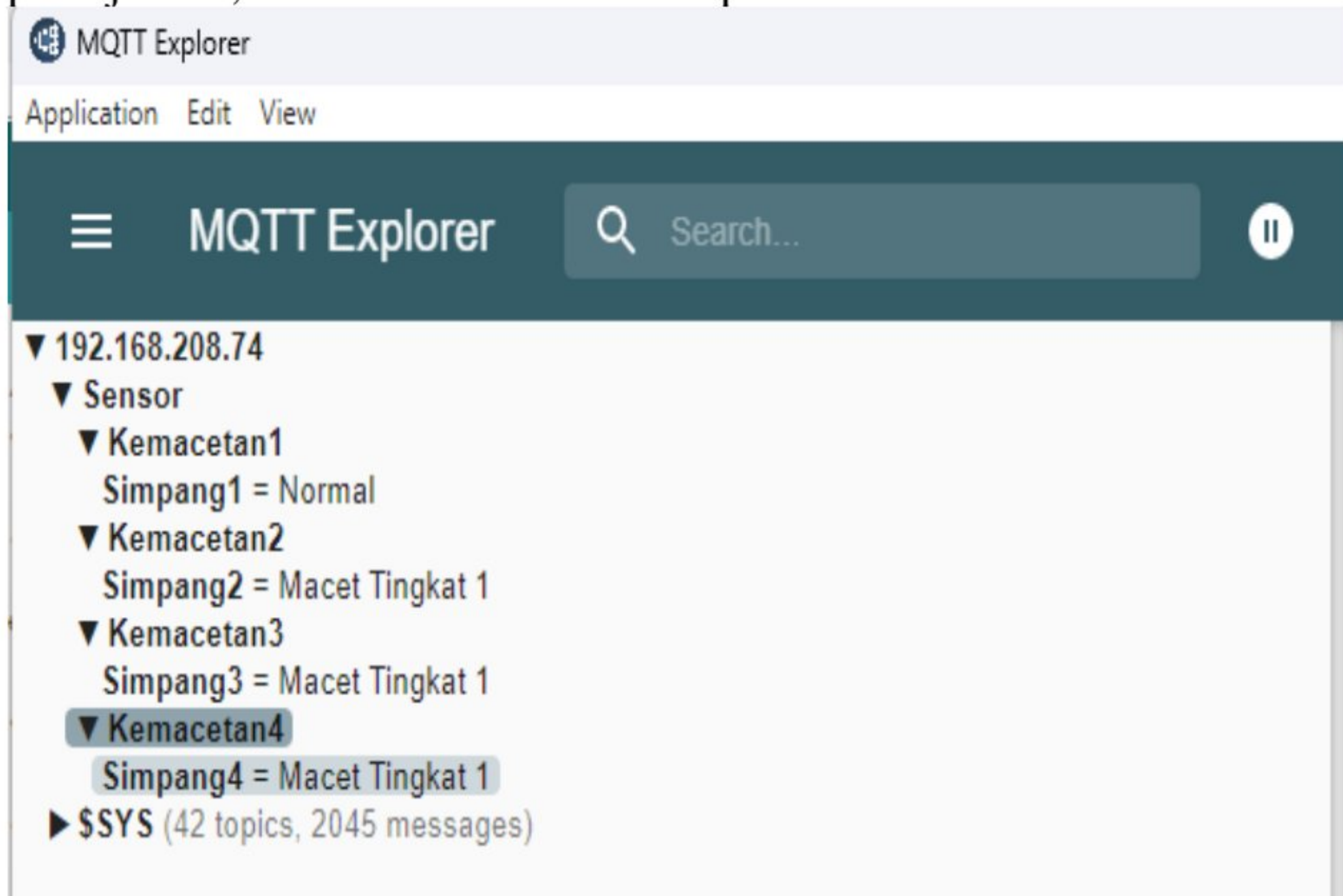
Tabel 4.7 menjelaskan bahwa jalur 2, 3 dan 4 terjadi kemacetan tingkat pertama. jalur 1 dalam keadaan sepi, sehingga lampu hijau pada jalur 2, 3 dan 4 menjadi lebih lama dengan total lampu hijau bernilai 25 detik. Grafik hasil pengujian skenario keempat dapat dilihat pada Gambar 4.19.



Gambar 4.19 Grafik Hasil Pengujian Skenario Keempat

Gambar 4.19 Menunjukkan bahwa jalur 2, 3 dan 4 dalam kondisi terjadi kemacetan tingkat 1 sementara itu jalur lainnya masih dalam kondisi sepi. Hal ini membuat lampu hijau jalur 2, 3 dan 4

menjadi lebih lama sehingga kemacetan tingkat pertama yang terjadi pada jalur 2, 3 dan 4 tidak bertambah parah.



Gambar 4.20 Hasil MQTT Skenario Keempat

4.3.5 Hasil Skenario Kelima

Skenario kelima dilakukan dengan kondisi jalur 2 dan 3 berada pada kemacetan tingkat pertama, jalur 4 pada kemacetan tingkat kedua, jalur 1 dalam keadaan sepi, skenario ini adalah skenario di mana terjadi kemacetan tingkat pertama pada jalur 2 dan 3, terjadi kemacetan yang lebih parah yaitu kemacetan tingkat kedua pada jalur 4. Kemacetan terjadi karena ada kendaraan yang berhenti pada sensor kemacetan pertama pada jalur 2 dan 3 sekaligus juga terjadi pendeteksian kendaraan pada jalur 4 sehingga sistem akan mendeteksi adanya kemacetan tingkat pertama pada jalur 2 dan 3 sekaligus kemacetan tingkat kedua pada jalur 4. Skenario pengujian ini dapat dilihat pada Gambar 4.21.



Gambar 4.21 Skenario Kelima

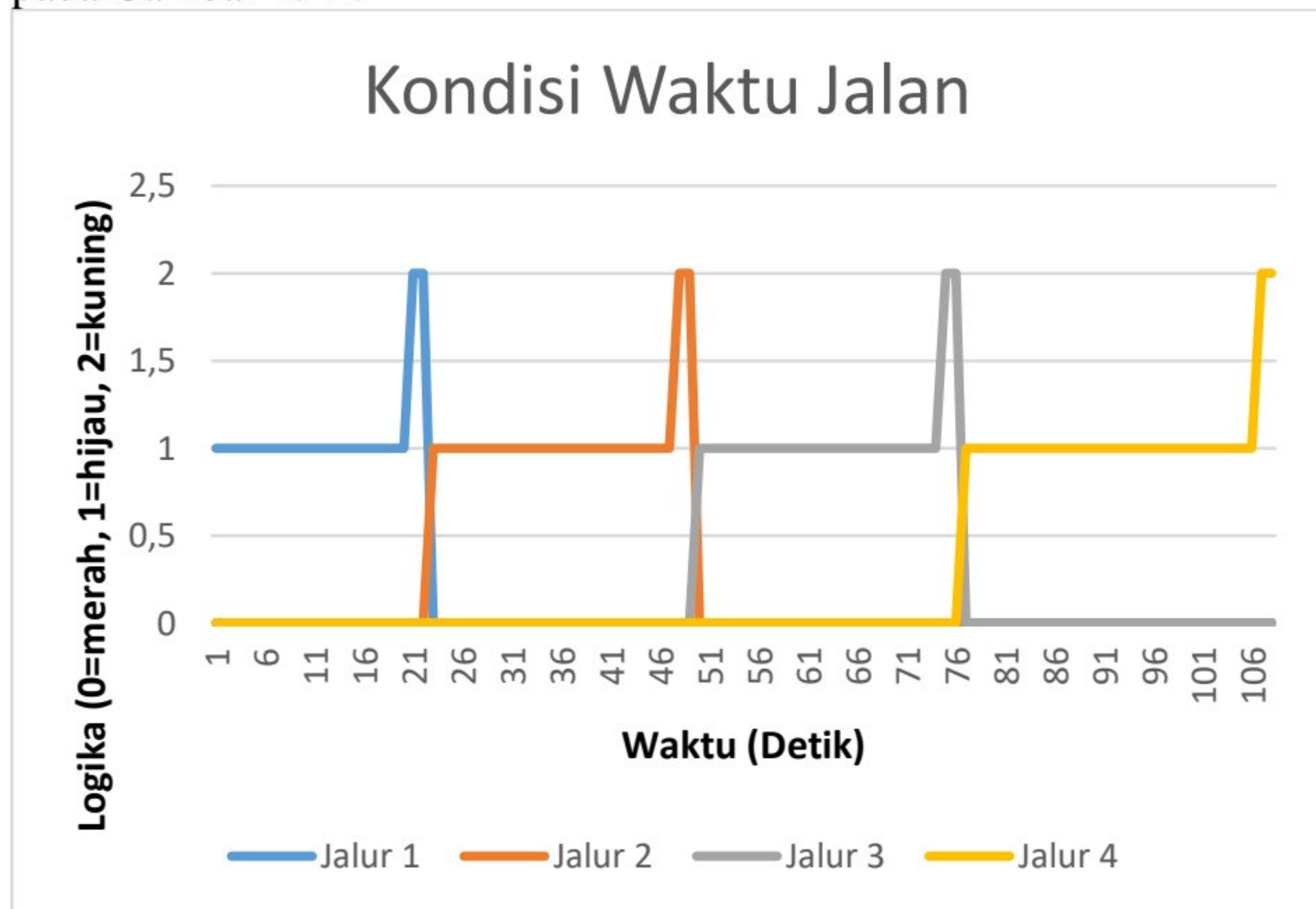
Gambar 4.21 Menunjukkan bahwa jalur 2 dan 3 terjadi kemacetan tingkat 1, jalur 4 terjadi kemacetan tingkat kedua, jalur 1 dalam keadaan sepi, terdapat dua jalur yaitu jalur 2 dan 3 yang diamana ada kendaraan berhenti dan menunggu pada sensor kemacetan tingkat 1, sekaligus terdapat juga kendaraan yang berhenti dan menunggu pada sensor kemacetan tingkat kedua pada jalur 4. Hasil lampu merah dalam skenario ini dapat dilihat pada Tabel 4.6.

Tabel 4.7 Hasil Pengujian Skenario Kelima

Jalur	Lalu Lintas	Lampu hijau (s)	Lampu merah (s)
Jalur 1	Sepi	20	80
Jalur 2	Macet Tingkat I	25	75
Jalur 3	Macet Tingkat I	25	75

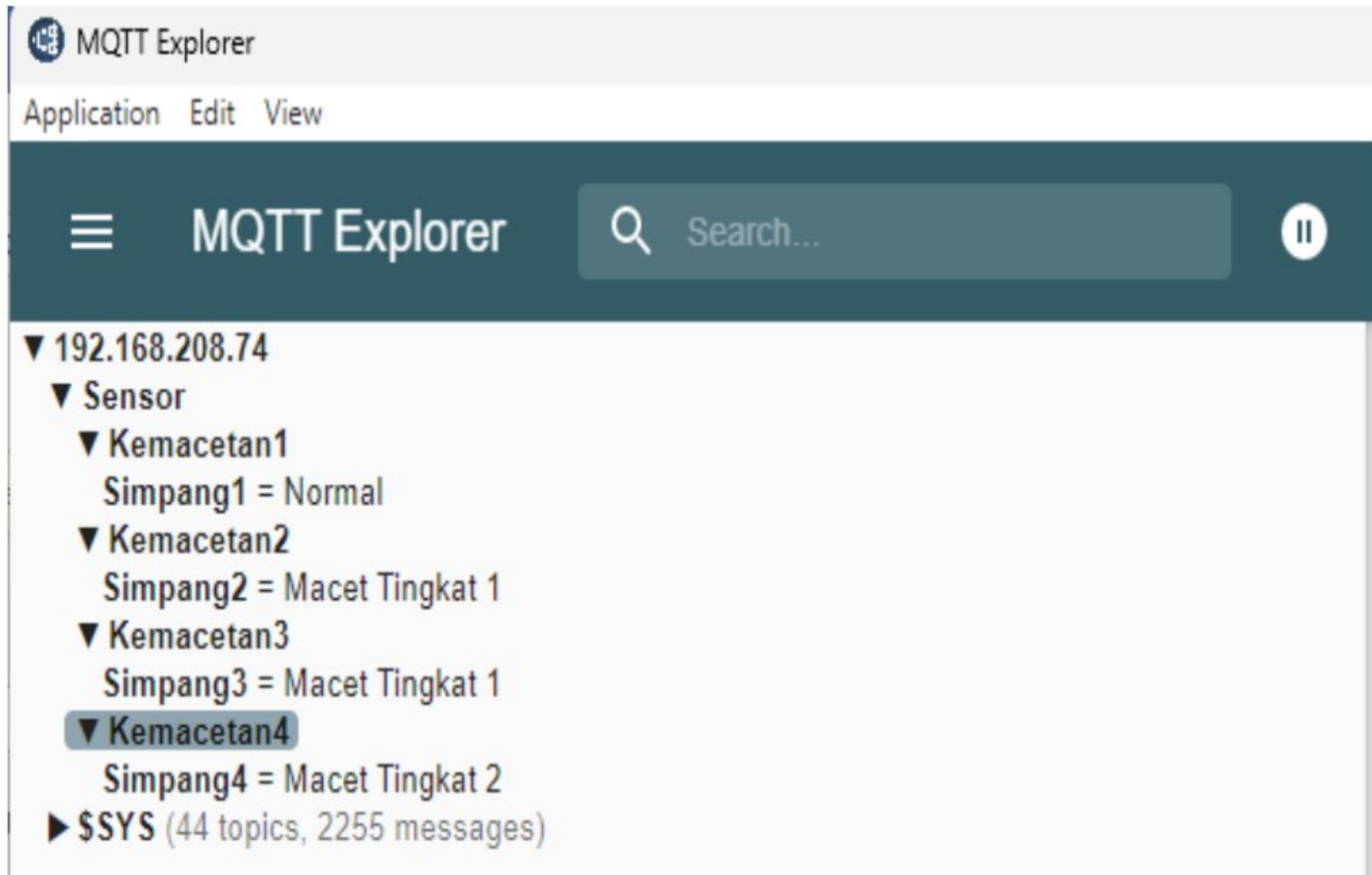
Jalur 4	Macet Tingkat II	30	70
---------	------------------	----	----

Tabel 4.8 menjelaskan bahwa jalur 2 dan 3 terjadi kemacetan tingkat pertama sekaligus terjadi kemacetan tingkat kedua pada jalur 4, jalur 1 dalam keadaan sepi. Sehingga lampu hijau pada jalur 2 dan 3 menjadi lebih lama dengan total lampu hijau bernilai 25 detik kemudian lampu hijau pada jalur 4 lebih lama lagi dengan lampu hijau senilai 30 detik. Grafik hasil pengujian skenario kelima dapat dilihat pada Gambar 4.22.



Gambar 4.22 Grafik Hasil Pengujian Skenario Kelima

Gambar 4.22 Menunjukkan bahwa jalur 2 dan 3 dalam kondisi terjadi kemacetan tingkat pertama sekaligus jalur 4 dalam kondisi terjadi kemacetan tingkat kedua sementara itu jalur 1 masih dalam kondisi sepi. Hal ini membuat lampu hijau jalur 2 dan 3 menjadi lebih lama 5 detik dari jalur 1, kemudian jalur 4 lebih lama 10 detik dari jalur 1. Sehingga kemacetan tingkat pertama yang terjadi pada jalur 2 dan 3 serta kemacetan tingkat kedua pada jalur 4 tidak bertambah parah.



Gambar 4.23 Hasil MQTT Skenario Kelima

4.4 Analisis Skenario

Sistem smart *Traffic Light* berbasis *IoT* menunjukkan adaptasi yang baik terhadap kondisi lalu lintas yang berbeda. Dengan adanya skenario mulai dari kondisi seluruh jalur sepi hingga kemacetan tingkat pertama dan kedua di beberapa jalur, sistem mampu mengalokasikan lampu hijau (green light) yang lebih lama untuk jalur-jalur dengan tingkat kemacetan lebih tinggi. Hal ini menunjukkan bahwa sistem berhasil mendeteksi kemacetan dan mengoptimalkan alokasi waktu.

Skenario 1 (seluruh jalur sepi): Semua jalur mendapatkan lampu hijau dan lampu merah yang sama (20 detik untuk lampu hijau dan 60 detik untuk lampu merah). Hal ini mencerminkan skenario dasar tanpa adanya beban lalu lintas.

Skenario 2-6: Ketika kemacetan terjadi pada satu atau lebih jalur, lampu hijau jalur-jalur yang macet bertambah (25-30 detik), sementara lampu merah pada jalur sepi meningkat. Sebagai contoh: Pada skenario 2, lampu hijau di jalur 4 meningkat menjadi 25 detik karena adanya kemacetan tingkat pertama. Pada skenario 6, jalur 3 dan 4 (kemacetan tingkat kedua) mendapat lampu hijau terpanjang (30

detik), sementara jalur 1 yang sepi harus menunggu lebih lama (85 detik). Pola ini menunjukkan bahwa sistem memberikan prioritas kepada jalur-jalur yang macet untuk mencegah penumpukan kendaraan lebih lanjut.

Dengan memanfaatkan sensor *IoT*, sistem mampu mendeteksi tingkat kemacetan (tingkat pertama atau kedua) secara akurat di setiap jalur. Hal ini ditunjukkan oleh perubahan lampu hijau yang disesuaikan dengan tingkat kemacetan yaitu : Kemacetan tingkat pertama, tambahan lampu hijau 5 detik dibanding jalur normal. Kemacetan tingkat kedua, tambahan lampu hijau hingga 10 detik dibanding jalur normal. Respons ini mencerminkan efisiensi sistem *IoT* dalam menangkap data real-time dan menyesuaikan durasi lampu lalu lintas.

Dalam setiap skenario, jalur yang sepi mengalami peningkatan lampu merah sebagai konsekuensi dari alokasi waktu lebih lama untuk jalur macet. Misalnya: Pada skenario 4 (kemacetan tingkat pertama di jalur 2, 3, dan 4), jalur 1 (sepi) memiliki lampu merah tertinggi (75 detik). Pada skenario 6 (kemacetan tingkat kedua di jalur 3 dan 4), jalur 1 harus menunggu hingga 85 detik. Hal ini menunjukkan adanya perubahan dalam pengelolaan waktu untuk menghindari kemacetan lebih parah di jalur sibuk.

Grafik hasil pengujian dan tampilan MQTT pada masing-masing skenario memberikan representasi visual yang jelas mengenai pola lampu hijau dan tunggu di setiap jalur. Grafik membantu memahami bagaimana sistem beradaptasi terhadap skenario tertentu, sedangkan MQTT menunjukkan data real-time yang digunakan untuk pengambilan keputusan.

Sistem smart *Traffic Light* berbasis *IoT* ini menunjukkan keberhasilan dalam mendeteksi dan menangani berbagai tingkat kemacetan di jalur lalu lintas. Penggunaan lampu hijau yang adaptif berdasarkan tingkat kemacetan memberikan solusi efektif untuk mengurangi risiko penumpukan kendaraan. Namun, ada dampak berupa peningkatan lampu merah di jalur sepi yang perlu diantisipasi untuk mencegah ketidakpuasan pengguna jalan.

Pada sub bab ini dilakukan analisa mendalam terhadap hasil implementasi dan pengujian sistem Smart *Traffic Light* (STL) berbasis Internet of Things (IoT) yang menggunakan panjang antrian sebagai

parameter utama untuk mengatur waktu lampu hijau. Analisa ini mencakup keakuratan sistem dalam mendeteksi kondisi lalu lintas, efektivitas algoritma pengambilan keputusan, keandalan komunikasi perangkat, serta relevansi sistem terhadap kebutuhan aktual berdasarkan hasil wawancara dengan Dinas Perhubungan.

4.4.1 Analisa Respon Sistem terhadap Kepadatan Lalu Lintas

Sistem ini menggunakan dua buah sensor inframerah (IR) yang diposisikan secara strategis di jalur pendekat simpang. Deteksi kendaraan oleh sensor menjadi indikator panjang antrian yang kemudian dikategorikan menjadi tiga kondisi: normal, Macet Tingkat I, dan Macet Tingkat II.

1. **Kondisi Normal (Kedua Sensor Tidak Aktif)**

Dalam kondisi ini, tidak ada kendaraan yang melampaui titik deteksi sensor. Waktu lampu hijau tetap berada di angka dasar, yaitu 20 detik. Berdasarkan observasi selama simulasi, sistem secara konsisten mempertahankan durasi ini jika lalu lintas sepi atau terputus-putus, yang mengindikasikan kestabilan logika dasar sistem.

2. **Kondisi Macet Tingkat I (Sensor 1 Aktif, Sensor 2 Tidak Aktif)**

Ketika kendaraan hanya melewati sensor pertama, sistem menafsirkan ini sebagai antrian sepanjang kurang dari 30 meter. Lampu hijau ditambahkan 5 detik menjadi 25 detik. Penambahan ini terbukti cukup untuk mengurangi panjang antrian secara bertahap selama beberapa siklus lampu lalu lintas. Pengujian menunjukkan bahwa dengan 25 detik.

3. **Kondisi Macet Tingkat II (Kedua Sensor Aktif)**

Ketika antrian melebihi 30 meter dan terdeteksi oleh kedua sensor, waktu lampu hijau diperpanjang menjadi 30 detik. Dalam simulasi, kondisi ini sering terjadi saat arus kendaraan padat secara terus menerus. Peningkatan durasi lampu hijau memberikan efek signifikan dalam memperpendek antrian dan mengembalikan arus ke kondisi semi-normal dalam 3–4 siklus.

Kesimpulan awal: sistem responsif terhadap variasi kondisi lalu lintas, dan penyesuaian waktu lampu hijau secara otomatis mampu mengurangi kemacetan tanpa intervensi manual.

4.4.2 Analisa Akurasi dan Efisiensi Sensor IR

Sensor IR digunakan karena kemudahannya dalam implementasi dan biayanya yang relatif rendah. Namun terdapat beberapa hal yang menjadi perhatian:

1. Akurasi dalam Kondisi Terang dan Gelap: Sensor IR menunjukkan konsistensi yang cukup baik dalam mendeteksi kendaraan siang maupun malam hari.
2. Gangguan Lingkungan: Potensi gangguan terjadi jika sensor terkena sinar matahari langsung atau refleksi dari bodi kendaraan. Untuk implementasi nyata, sistem perlu dilengkapi dengan housing pelindung dan pengkondisian sinyal.
3. Jarak Deteksi: Sensor IR hanya mendeteksi objek secara langsung di depannya (posisi tetap). Oleh karena itu, panjang antrian tidak dihitung secara numerik (dalam meter), melainkan berbasis titik logis.

Evaluasi: sistem berbasis sensor IR cocok untuk prototipe awal dan skenario sederhana, namun untuk akurasi yang lebih tinggi diperlukan sensor alternatif seperti LIDAR atau kamera AI.

4.4.3 Analisa Komunikasi dan Kendali IoT

Sistem menggunakan ESP32 yang terhubung melalui WiFi untuk mengontrol waktu nyala lampu berdasarkan input dari sensor IR. Pengujian komunikasi antar perangkat menunjukkan:

1. Latency komunikasi rata-rata < 100 ms dalam kondisi jaringan lokal.
2. Stabilitas sistem bergantung pada kestabilan koneksi WiFi dan suplai daya. Dalam beberapa percobaan, saat terjadi gangguan sinyal atau voltage drop, ESP32 mengalami reset otomatis.
3. Kemungkinan ekspansi sangat terbuka, termasuk integrasi dengan dashboard pemantauan pusat menggunakan protokol MQTT.

4.4.4 Kesesuaian Sistem terhadap Kebutuhan Dishub

Pengembangan prototipe *Smart Traffic Light* (STL) berbasis panjang antrian ini didasari oleh hasil wawancara langsung dengan petugas Dinas Perhubungan Kota Pekanbaru, yang mengungkapkan

adanya tantangan nyata di lapangan terkait manajemen lalu lintas, khususnya di simpang empat padat seperti Jalan Durian. Sistem lampu lalu lintas konvensional saat ini masih menggunakan pendekatan waktu tetap (*fixed-time scheduling*), di mana durasi lampu hijau, kuning, dan merah sudah ditentukan dan tidak berubah meskipun kondisi arus lalu lintas berbeda-beda setiap waktu. Kebutuhan Dishub saat ini dapat dikelompokkan ke dalam tiga aspek utama:

1. Adaptasi Real-Time terhadap Kepadatan Lalu Lintas
2. Efisiensi Operasional dan Pengurangan Beban Manual
3. Kesiapan Terhadap Integrasi Smart City

Dishub Kota telah menyatakan ketertarikannya untuk bergerak menuju implementasi sistem transportasi dalam kerangka Smart City, di mana seluruh infrastruktur lalu lintas akan terhubung, berkomunikasi secara digital, dan dapat dianalisis secara agregat.

Prototipe ini memiliki potensi untuk menjadi bagian dari sistem tersebut karena telah menggunakan perangkat keras IoT-ready (ESP32), sehingga dapat:

- a. Mengirimkan data lalu lintas ke server atau cloud secara berkala
- b. Menerima konfigurasi parameter dari pusat control
- c. Merekam riwayat kepadatan di setiap simpang yang dapat digunakan sebagai big data untuk evaluasi dan pengambilan kebijakan jangka panjang, seperti:
 - Penentuan jam sibuk berdasarkan data aktual
 - Perencanaan pembangunan flyover atau penambahan jalur
 - Evaluasi efektivitas rekayasa lalu lintas sebelumnya

Tabel 4. 8 Analisa Perbandingan Kondisi Eksisting

Aspek Kebutuhan Dishub	Kondisi Saat Ini	Fitur STL Prototipe	Kesesuaian
Adaptasi terhadap antrian	Tidak ada	Penyesuaian otomatis waktu lampu hijau berdasarkan sensor IR	☑

Otomatisasi kontrol lampu	Manual di banyak titik	Kontrol berbasis kondisi sensor dan logika pemrograman ESP32	☑
Konektivitas sistem	Terbatas/manual	Potensi koneksi ke WiFi, cloud, atau dashboard Dishub	☑
Pengumpulan data historis	Tidak dilakukan	Dapat ditambahkan fitur logging dan pengiriman data real-time	⚠ Butuh ekspansi
Fleksibilitas konfigurasi	Bergantung teknisi di lapangan	Sistem dapat diprogram ulang secara jarak jauh	☑
Kompatibilitas Smart City	Rendah	Arsitektur berbasis IoT (IoT-ready)	☑

Prototipe *Smart Traffic Light* (STL) yang dikembangkan telah mampu menjawab sebagian besar kebutuhan teknis dan strategis Dinas Perhubungan, khususnya dalam hal adaptasi real-time terhadap kondisi lalu lintas dan otomatisasi sistem lampu. Dengan tambahan pengembangan pada fitur konektivitas cloud dan integrasi dengan sistem monitoring Dishub, sistem ini berpotensi besar untuk diimplementasikan sebagai bagian dari proyek *Smart Traffic Management System* dalam mendukung visi Smart City Kota Pekanbaru.

4.4.5 Kelemahan dan Pengembangan Sistem

Meskipun prototipe *Smart Traffic Light* (STL) yang dikembangkan telah menunjukkan performa yang fungsional dan sesuai kebutuhan dasar di lapangan, namun sistem ini masih memiliki sejumlah keterbatasan baik dari segi teknis maupun struktural. Identifikasi kelemahan dilakukan berdasarkan hasil uji coba lapangan, simulasi logika sistem, dan umpan balik dari wawancara dengan pihak Dinas Perhubungan. Berikut adalah beberapa kelemahan utama:

1. Deteksi Panjang Antrian yang Terbatas
 - a. Sistem saat ini hanya menggunakan dua buah sensor IR yang diletakkan pada titik tetap. Dengan demikian, panjang antrian tidak diukur secara aktual dalam satuan meter, melainkan dibagi ke dalam kategori berdasarkan titik deteksi aktif.

- b. Kelemahannya: sistem tidak mampu mengukur jumlah kendaraan secara presisi atau memberikan data numerik mengenai panjang antrian.
 - c. Dampaknya: respon sistem terbatas pada 3 level (normal, padat I, padat II), padahal kondisi lalu lintas dapat lebih kompleks dan dinamis.
2. Ketergantungan pada Sensor IR
 - a. Sensor IR memiliki keterbatasan dalam kondisi lingkungan tertentu, seperti:
 - Sinar matahari langsung
 - Hujan deras atau kabut
 - Permukaan kendaraan yang tidak memantulkan sinyal secara optimal
 - b. Kelemahannya: menurunkan akurasi deteksi, sehingga ada risiko sistem salah mengklasifikasikan kepadatan lalu lintas.
3. Belum Terintegrasi Secara Real-Time dengan Backend Monitoring
 - a. Sistem belum dikembangkan untuk mengirimkan data ke cloud server atau dashboard pemantauan.
 - b. Kelemahannya: keterbatasan untuk dimonitor oleh pihak Dishub secara pusat; seluruh keputusan dibuat oleh ESP32 secara lokal (standalone).
 - c. Dampaknya: sulit melakukan evaluasi jangka panjang atau pengawasan terpusat.
4. Tidak Mengakomodasi Semua Arah Jalur
 - a. Prototipe hanya mendeteksi kondisi dari satu jalur pendekat.
 - b. Kelemahannya: sistem tidak mempertimbangkan kepadatan dari arah lain (jalur menyilang), padahal setiap simpang memiliki banyak pendekat.
 - c. Dampaknya: distribusi waktu lampu hijau berpotensi tidak adil dan menyebabkan bottleneck di jalur lain.
5. Tidak Adaptif terhadap Situasi Khusus
 - a. Sistem tidak mengenali kendaraan prioritas seperti ambulans, pemadam kebakaran, atau kendaraan umum (busway).

- b. Sistem juga belum mempertimbangkan kondisi darurat, kecelakaan, atau jalan ditutup.
- c. Kelemahannya: sistem belum memiliki *situational awareness* yang kompleks seperti dalam sistem *Adaptive Traffic Control* berbasis AI.

Berdasarkan kelemahan-kelemahan yang teridentifikasi, pengembangan sistem diarahkan pada tiga domain: teknologi deteksi, logika kontrol, dan sistem integrasi. Tujuan pengembangan adalah meningkatkan akurasi, skalabilitas, dan interoperabilitas sistem dengan infrastruktur Dishub serta potensi perluasan ke dalam ekosistem Smart City.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil eksperimen dan analisis dapat diambil kesimpulan bahwa:

1. Prototipe sistem lampu lalu lintas berbasis mikrokontroler ESP32 berhasil dibuat dan mampu mendeteksi panjang antrian kendaraan secara real-time. Sistem ini dapat menjadi dasar untuk pengembangan lebih lanjut dalam implementasi *Smart Traffic Light*.
2. Sistem yang dikembangkan mampu mendeteksi panjang antrian kendaraan pada persimpangan lampu lalu lintas, sistem menggunakan sensor IR sebagai pendeteksi panjang antrian, sehingga dapat memberikan informasi kondisi lalu lintas secara akurat. Hal ini memungkinkan pengelolaan lalu lintas yang lebih efisien dengan menyesuaikan durasi lampu hijau berdasarkan kepadatan kendaraan.
3. Dari sisi Dishub, keberadaan sistem ini memiliki relevansi yang tinggi dengan kebutuhan di lapangan, khususnya dalam mendukung strategi manajemen lalu lintas berbasis teknologi. Dengan sistem yang mampu melakukan monitoring real-time, Dishub berpotensi memperoleh data akurat terkait pola kepadatan kendaraan di setiap persimpangan. Data tersebut dapat menjadi dasar dalam pengambilan keputusan, evaluasi kebijakan lalu lintas, serta perencanaan infrastruktur ke depan.

5.2 Saran

Berikut adalah saran atau masukan yang dapat diberikan berdasarkan hasil penelitian yang telah dilakukan:

1. Diharapkan penelitian yang akan datang dapat menambahkan beberapa sensor lagi untuk dapat mengklasifikasikan tingkat kemacetan dan memvariasikan lampu merah yang lebih bervariasi lagi.

2. Diharapkan melakukan skenario lebih bervariasi lagi sehingga hasil yang didapatkan lebih kompleks untuk dianalisis.

DAFTAR PUSTAKA

- Agung K., 2014. “Penjualan Mobil di Indonesia 1.22 Juta Unit” <https://otomotif.kompas.com/read/xml/2014/01/03/1008443/2013.Penjualan.Mobil.di.Indonesia.1.22.Juta.Unit>”, diakses pada agustus 2023.
- Alfith, A., & Kartiria, K. (2019). Pengembangan Perancangan *Smart Traffic Light* Berbasis LDR Sensor Dan Timer Delay System. *Jurnal Teknik Elektro*, 8(1), 35-39.
- Aria and R. Faizal, 2017, “Sistem Lalu Lintas Terpadu Embedded Traffic System,” *Telekontran*, vol. 5, no. 2, pp. 83–93.
- Aryaviocholda, F., Ichsan, M. H. H., & Budi, A. S. (2020). Rancangan Sistem Pendeteksi Pencurian Helm Menggunakan Protokol MQTT Dan Bluetooth HC-05 Berbasis Arduino. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, 4(2), 517-525.
- Bagus, I. (2022). Pengembangan *Smart Traffic Light* Berbasis *IoT* Dengan Mobile Backend As A Service Sebagai Wujud *Smart City* Bidang Transportasi. *Jurnal Portal Data*, 2(10).
- Hozanna, G., Nur, D., & Kasim, K. (2021, October). Sistem Monitoring Dan Controlling Lampu Lalu Lintas Berbasis Wireless Sensor Network Menggunakan Lora. In *Seminar Nasional Teknik Elektro dan Informatika (SNTEI)* (pp. 223-228).
- Indonesia. (2009). *Undang-Undang Nomor 22 Tahun 2009 tentang Lalu Lintas dan Angkutan Jalan*. Lembaran Negara Republik Indonesia Tahun 2009 Nomor 96.
- Intan, J., & Gunawan, I. R. (2019). *Traffic Light* Automation Berbasis Arduino Menggunakan RF Transceiver Untuk Kendaraan Prioritas. *Seminastika*, 2, 122-29.
- Jamal, J., & Thamrin, T. (2021). Sistem Kontrol Kandang Ayam Closed House Berbasis *Internet Of Things*. *Voteteknika (Vocational Teknik Elektronika dan Informatika)*, 9(3), 79-90.
- Jatmika, S., & Andiko, I. (2014). Simulasi pengaturan lampu lalu lintas berdasarkan data image processing kepadatan

- kendaraan berbasis mikrokontroler atmega16. *Jurnal Ilmiah Teknologi dan Informasi ASIA*, 8(2).
- Kepolisian Republik Indonesia (2024) “Perkembangan Jumlah Kendaraan Bermotor Menurut Jenis (Unit), 2021-2022” <https://www.bps.go.id/indicator/17/57/1/jumlah-kendaraan-bermotor.html>” diakses pada agustus 2023.
- Khawas, F. A. (2022). *Analisis Kinerja Topologi Wireless Sensor Network Untuk Aplikasi Kelembapan Tanah Perkebunan Kentang* (Doctoral dissertation, Universitas Komputer Indonesia).
- Kurniawan, R. (2019). *Perancangan Sistem Penerangan Lampu Otomatis Dengan Menggunakan Sensor Proximity Berbasis Plc Omron Cplc E20sdr* (Doctoral dissertation, Politeknik Negeri Sriwijaya).
- Muda, I. (2013). *Elektronika Dasar*. Gunung Samudera [PT Book Mart Indonesia].
- Nirwanda, N., Yenni, H., Anam, M. K., & Lathifah, L. (2023). Prototype Smart Time Scheduler Lampu Lalu Lintas Menggunakan Algoritma Haar Cascade. *Jurnal Teknoinfo*, 17(1), 328-341.
- Romli, I., Hugo, K. L. N., & Afriantoro, I. (2021). Perancangan Dan Implementasi Smart Garden Berbasis *Internet of Things* (Iot) Pada Perumahan Central Park Cikarang. *Indonesian Journal of Business Intelligence (IJUBI)*, 4(2), 42-52.
- Saputro, A. D. (2020). Rancang bangun robot pendeteksi kadar gas sulfur dioksida (so₂) dan gas karbon monoksida (co) untuk ekplorasi kawah ijo objek wisata candi gedong songo berbasis *Internet Of Things*. Universitas Islam Negeri Walisongo Semarang.
- Siswaya, N. R. (2021). Prototype Smart Traffic Light Otomatis Berbasis Atmega-328 dengan Sensor Jarak. *Jurnal JE-UNISLA: Electronic Control, Telecommunication, Computer Information and Power System*, 4(2), 265-268.
- Taufiq, R. M., Sunanto, S., & Rizki, Y. (2020). Integrated Smart Traffic Control System Menuju Pekanbaru Sebagai Smart City. *JURTEKSI (Jurnal Teknologi dan Sistem Informasi)*, 7(1), 67-74.

- Zabar, A. A., & Novianto, F. (2015). Keamanan Http Dan Https Berbasis Web Menggunakan Sistem Operasi Kali Linux. *Komputa: Jurnal Ilmiah Komputer dan Informatika*, 4(2), 69-74.
- Zulianti, F., Munadi, R., & Santoso, I. H. (2021). Smart Traffic Light Berbasis *Internet Of Things* Pada Keselamatan Ambulans. *eProceedings of Engineering*, 8(5).

Lampiran 1. Hasil Wawancara



PEMERINTAH KOTA PEKANBARU DINAS PERHUBUNGAN

Jalan Abdul Rahman Hamid Komplek Perkantoran Tenayan Raya
Gedung B.9 Lt. I dan II Kec. Tenayan Raya
PEKANBARU



SURAT KETERANGAN

Nomor : 200.1.1/Dishub-PKB/2269/2025

Berdasarkan Surat Dari Politeknik Caltex Riau Nomor : 1005/AKA-SU/PCR/2025 Perihal Pelaksanaan Kegiatan Riset / Pra Riset dan Pengumpulan Data Untuk Bahan Tesis.

Kepala Rekayasa dan Fasilitas Lalu Lintas Jalan dan Perairan Dinas Perhubungan Kota Pekanbaru dengan ini menyatakan bahwa :

Nama : **JIHAN ATHIRA RAMADHANI**
NIM : 2256102004
Jurusan : Magister Terapan Teknik komputer
Jenjang : Strata 2 (S2)
Lokasi Penelitian : Dinas Perhubungan Kota Pekanbaru

Mahasiswa tersebut diatas telah selesai melakukan Tesis penelitian pada Seksi Rekayasa dan Fasilitas Lalu Lintas Jalan dan Perairan Judul : " **PROTOTYPE SMART TRAFFIC LIGHT BERBASIS IoT (INTERNET OF THINGS) UNTUK PENANGGULANGAN KEMACETAN** ".

Demikian Surat Keterangan ini diberikan agar dapat dipergunakan sebagaimana mestinya.

DIKELUARKAN DI : PEKANBARU
PADA TANGGAL : 19 September 2025
KEPALA SEKSI REKAYASA DAN
FASILITAS LALU LINTAS JALAN DAN
PERAIRAN



RABIUL MISQA HAGL SSTP, M.Si
NIP.19910912 201406 1 001



PEMERINTAH KOTA PEKANBARU DINAS PERHUBUNGAN

Jalan Abdul Rahman Hamid Komplek Perkantoran Tenayan Raya
Gedung B.9 Lt. I dan II Kec. Tenayan Raya
PEKANBARU



SURAT KETERANGAN

Nomor : 200.1.1/Dishub-PKB/2269/2025

Berdasarkan Surat Dari Politeknik Caltex Riau Nomor : 1005/AKA-SU/PCR/2025 Perihal Pelaksanaan Kegiatan Riset / Pra Riset dan Pengumpulan Data Untuk Bahan Tesis.

Kepala Rekayasa dan Fasilitas Lalu Lintas Jalan dan Perairan Dinas Perhubungan Kota Pekanbaru dengan ini menyatakan bahwa :

Nama : **JIHAN ATHIRA RAMADHANI**
NIM : 2256102004
Jurusan : Magister Terapan Teknik komputer
Jenjang : Strata 2 (S2)
Lokasi Penelitian : Dinas Perhubungan Kota Pekanbaru

Mahasiswa tersebut diatas telah selesai melakukan Tesis penelitian pada Seksi Rekayasa dan Fasilitas Lalu Lintas Jalan dan Perairan Judul : " **PROTOTYPE SMART TRAFFIC LIGHT BERBASIS IoT (INTERNET OF THINGS) UNTUK PENANGGULANGAN KEMACETAN** ".

Demikian Surat Keterangan ini diberikan agar dapat dipergunakan sebagaimana mestinya.

DIKELUARKAN DI : PEKANBARU
PADA TANGGAL : 19 September 2025
KEPALA SEKSI REKAYASA DAN
FASILITAS LALU LINTAS JALAN DAN
PERAIRAN



RABIUL MISQA HAGI, SSTP, M.Si
NIP.19910912 201406 1 001

Peneliti:

1. Pada simpang empat Jalan Durian, jika dipasang sensor panjang antrian, seberapa panjang antrian yang biasanya masih dianggap normal, dan pada titik mana kondisi tersebut sudah dikategorikan macet tingkat I maupun macet tingkat II?

Dishub:

Di simpang empat Jalan Durian, kami mengklasifikasikan kepadatan lalu lintas berdasarkan panjang antrian kendaraan. Normalnya, jika panjang antrian masih di bawah 10 meter, itu dianggap kondisi normal karena masih dalam batas toleransi tanpa menimbulkan gangguan signifikan pada persimpangan. Jika panjang antrian sudah mencapai sekitar 10–30 meter, kami kategorikan sebagai macet tingkat I, yaitu kepadatan mulai terasa dan ada potensi keterlambatan arus lalu lintas. Sedangkan jika panjang antrian melebihi 30 meter atau lebih, kondisi tersebut sudah masuk macet tingkat II, di mana terjadi penumpukan signifikan, bahkan bisa menyebabkan antrean menutup jalur di belakangnya.

Peneliti:

2. Jika lampu hijau diatur secara real time mengikuti kondisi antrian, berapa durasi ideal lampu hijau yang biasanya digunakan Dishub untuk kondisi normal, macet tingkat I, dan macet tingkat II?

Dishub:

Secara umum, pengaturan lampu hijau di simpang Jalan Durian menggunakan sistem waktu tetap dengan rata-rata 20 detik per siklus. Namun, jika menggunakan sistem adaptif berbasis sensor, maka kategorinya bisa seperti berikut:

- Normal (antrian < 10 m): durasi tetap, sekitar 20 detik, karena arus kendaraan masih lancar.
- Macet Tingkat I (antrian 10–30 m): lampu hijau idealnya diperpanjang menjadi 25–30 detik, untuk memberi kesempatan lebih banyak kendaraan lewat sehingga mengurangi kepadatan.

df

- Macet Tingkat II (antrian > 30 m): lampu hijau sebaiknya diperpanjang menjadi 30–35 detik, bahkan bisa sampai 40 detik bila diperlukan. Namun, harus tetap diperhitungkan agar jalur lain tidak terlalu lama menunggu.

Peneliti:

3. Bagaimana kondisi lampu lalu lintas saat ini di Pekanbaru, khususnya di simpang empat Jalan Durian?

Dishub:

Untuk saat ini, sebagian besar lampu lalu lintas di Pekanbaru masih menggunakan sistem waktu tetap (fixed time) dengan siklus yang sudah diprogram berdasarkan rata-rata kepadatan lalu lintas. Beberapa simpang besar memang sudah terhubung ke ATCS (Area Traffic Control System), tetapi penerapannya belum menyeluruh. Khusus di simpang empat Jalan Durian, pengaturan masih mengandalkan sistem waktu tetap, sehingga belum bisa menyesuaikan dengan kondisi real time. Akibatnya, pada jam-jam sibuk seperti pukul 07.00–08.30 dan 16.30–18.00, antrian kendaraan sering kali lebih panjang dari durasi hijau yang tersedia.

Peneliti:

4. Menurut Dishub, bagaimana potensi peningkatan teknologi untuk mengatasi permasalahan kemacetan di Pekanbaru, terutama dengan konsep berbasis sensor dan IoT?

Dishub:

Potensinya sangat besar. Dengan teknologi berbasis sensor dan IoT, lampu lalu lintas dapat beradaptasi secara dinamis terhadap kondisi nyata di lapangan. Data sensor dapat dikirimkan ke pusat kontrol Dishub secara real time, sehingga kami bisa melakukan pemantauan sekaligus evaluasi pola kepadatan lalu lintas harian. Jika sistem ini dipadukan dengan ATCS, maka perintah pengaturan lampu bisa dilakukan otomatis tanpa menunggu intervensi manual. Selain itu, teknologi ini juga membuka peluang integrasi dengan CCTV berbasis AI untuk analisis pola lalu lintas, deteksi pelanggaran, hingga prediksi kepadatan.

7

Peneliti:

5. Apa kendala utama yang selama ini dihadapi Dishub dalam meningkatkan teknologi pada lampu lalu lintas di Pekanbaru?

Dishub:

Ada beberapa kendala utama, antara lain:

1. Keterbatasan anggaran. Pengadaan sensor, jaringan IoT, dan perangkat kontrol cerdas memerlukan biaya investasi yang cukup tinggi.
2. Infrastruktur pendukung. Tidak semua simpang memiliki jaringan listrik dan internet yang stabil untuk menunjang sistem real time.
3. Pemeliharaan. Alat sensor rentan terhadap kerusakan akibat cuaca dan kondisi jalan. Dibutuhkan tim khusus untuk pemeliharaan rutin.
4. Integrasi sistem. Saat ini ATCS belum sepenuhnya terintegrasi di semua simpang, sehingga sulit mengendalikan secara menyeluruh.
5. Sumber daya manusia. Dibutuhkan tenaga yang terlatih dalam bidang IoT dan manajemen lalu lintas digital agar sistem benar-benar optimal.

Hasil Wawancara dengan Dinas Perhubungan (Bagian 2)

Peneliti:

6. Bagaimana Dishub biasanya menentukan durasi lampu hijau pada sistem konvensional tanpa sensor?

Dishub:

Penentuan durasi lampu hijau pada sistem konvensional biasanya didasarkan pada studi lalu lintas manual. Petugas Dishub melakukan survei di lapangan untuk menghitung volume kendaraan pada jam-jam tertentu, kemudian dibuatkan rata-rata per siklus. Misalnya, jika di jalur A rata-rata kendaraan 400 unit/jam, sedangkan di jalur B hanya 200 unit/jam, maka jalur A akan diberi lampu hijau lebih lama. Namun, metode ini memiliki kelemahan karena tidak fleksibel

71

terhadap fluktuasi lalu lintas harian. Jika tiba-tiba ada lonjakan kendaraan, sistem tetap menggunakan waktu tetap yang sudah diprogram.

Peneliti:

7. Bagaimana respons masyarakat terhadap lampu lalu lintas yang saat ini menggunakan sistem waktu tetap?

Dishub:

Masyarakat sering menyampaikan keluhan, terutama pada jam sibuk. Banyak pengendara merasa waktu menunggu terlalu lama di jalur yang sepi, sementara di jalur padat justru durasi hijau terasa singkat. Hal ini menimbulkan ketidakpuasan dan terkadang membuat pengendara melakukan pelanggaran seperti menerobos lampu merah. Dari sisi kami, keluhan ini menjadi salah satu alasan mengapa teknologi adaptif berbasis sensor perlu dipertimbangkan untuk masa depan.

Peneliti:

8. Apakah Dishub pernah mencoba atau menguji sistem lampu lalu lintas adaptif di Pekanbaru?

Dishub:

Sejauh ini, di Pekanbaru belum ada implementasi penuh sistem adaptif berbasis sensor. Namun, kami sudah melakukan uji coba terbatas melalui ATCS di beberapa simpang utama, di mana operator di ruang kontrol dapat menyesuaikan waktu lampu hijau berdasarkan pantauan CCTV. Sayangnya, sistem ini masih semi-manual, artinya bergantung pada pengawasan operator. Dengan adanya IoT, proses tersebut dapat diotomatisasi sehingga tidak perlu lagi campur tangan manusia secara terus-menerus.

71

Peneliti:

9. Menurut Dishub, apa kelebihan utama dari konsep Smart Traffic Light berbasis IoT dibandingkan sistem yang ada sekarang?

Dishub:

Ada beberapa kelebihan utama:

1. Adaptif real time. Sistem mampu menyesuaikan durasi lampu hijau secara otomatis berdasarkan panjang antrian, bukan angka rata-rata.
2. Efisiensi waktu. Mengurangi waktu tunggu yang tidak perlu bagi pengendara, khususnya pada jalur yang sepi.
3. Data lalu lintas akurat. Sensor dapat mengirimkan data kepadatan ke server Dishub, yang bermanfaat untuk analisis jangka panjang.
4. Integrasi. Dapat dihubungkan dengan sistem ATCS dan CCTV, sehingga menciptakan manajemen lalu lintas yang lebih modern.
5. Keselamatan. Dengan arus kendaraan yang lebih lancar, risiko kecelakaan akibat antrean panjang atau saling serobot berkurang.

Peneliti:

10. Jika sistem ini diimplementasikan, bagaimana Dishub melihat tantangan integrasi dengan infrastruktur yang sudah ada?

Dishub:

Tantangan integrasi pasti ada. Pertama, lampu lalu lintas lama harus dimodifikasi agar kompatibel dengan modul kontrol berbasis IoT. Kedua, dibutuhkan server pusat yang bisa menampung data dari banyak simpang sekaligus. Ketiga, dibutuhkan jaringan komunikasi yang andal, misalnya fiber optic atau 4G/5G, agar data sensor bisa dikirimkan dengan latensi rendah. Kami juga harus memastikan sistem baru tidak mengganggu operasional ATCS yang sudah berjalan. Jadi, integrasi memerlukan perencanaan matang, standar protokol komunikasi yang jelas, serta uji coba bertahap di beberapa simpang sebelum diperluas ke seluruh kota.

11

Peneliti:

9. Menurut Dishub, apa kelebihan utama dari konsep Smart Traffic Light berbasis IoT dibandingkan sistem yang ada sekarang?

Dishub:

Ada beberapa kelebihan utama:

1. Adaptif real time. Sistem mampu menyesuaikan durasi lampu hijau secara otomatis berdasarkan panjang antrian, bukan angka rata-rata.
2. Efisiensi waktu. Mengurangi waktu tunggu yang tidak perlu bagi pengendara, khususnya pada jalur yang sepi.
3. Data lalu lintas akurat. Sensor dapat mengirimkan data kepadatan ke server Dishub, yang bermanfaat untuk analisis jangka panjang.
4. Integrasi. Dapat dihubungkan dengan sistem ATCS dan CCTV, sehingga menciptakan manajemen lalu lintas yang lebih modern.
5. Keselamatan. Dengan arus kendaraan yang lebih lancar, risiko kecelakaan akibat antrean panjang atau saling serobot berkurang.

Peneliti:

10. Jika sistem ini diimplementasikan, bagaimana Dishub melihat tantangan integrasi dengan infrastruktur yang sudah ada?

Dishub:

Tantangan integrasi pasti ada. Pertama, lampu lalu lintas lama harus dimodifikasi agar kompatibel dengan modul kontrol berbasis IoT. Kedua, dibutuhkan server pusat yang bisa menampung data dari banyak simpang sekaligus. Ketiga, dibutuhkan jaringan komunikasi yang andal, misalnya fiber optic atau 4G/5G, agar data sensor bisa dikirimkan dengan latensi rendah. Kami juga harus memastikan sistem baru tidak mengganggu operasional ATCS yang sudah berjalan. Jadi, integrasi memerlukan perencanaan matang, standar protokol komunikasi yang jelas, serta uji coba bertahap di beberapa simpang sebelum diperluas ke seluruh kota.

71

Peneliti:

11. Bagaimana cara Dishub memantau kondisi lalu lintas saat ini tanpa adanya sensor IoT?

Dishub:

Pemantauan lalu lintas saat ini masih mengandalkan dua metode utama: CCTV ATCS dan laporan manual petugas lapangan. CCTV dipasang di beberapa simpang besar dan terhubung ke ruang kontrol Dishub. Operator bertugas mengamati kondisi kepadatan secara visual, lalu menyesuaikan pengaturan lampu jika diperlukan. Namun, keterbatasannya adalah kamera hanya mencakup area tertentu, dan penilaian kepadatan masih bersifat subjektif. Selain itu, untuk jalan-jalan yang tidak dilengkapi CCTV, kami bergantung pada laporan petugas lapangan. Hal ini membuat respon terhadap perubahan arus lalu lintas tidak selalu cepat dan konsisten.

Peneliti:

12. Bagaimana menurut Dishub mengenai penggunaan sensor IR untuk mendeteksi panjang antrian?

Dishub:

Sensor IR sebenarnya bisa menjadi salah satu opsi sederhana untuk mendeteksi kendaraan. Kelebihannya adalah biaya relatif murah dan pemasangan mudah. Namun, dari sisi kami, ada beberapa catatan: IR sensitif terhadap cuaca ekstrem, seperti hujan deras atau debu jalanan, yang bisa mengurangi akurasi deteksi. Selain itu, IR bekerja lebih optimal pada jarak tertentu, sehingga jika digunakan untuk mengukur antrian panjang, perlu beberapa titik sensor yang dipasang berurutan. Meski begitu, sebagai prototipe, IR dapat dijadikan dasar pembuktian konsep. Jika ingin implementasi skala kota, mungkin akan lebih baik dipadukan dengan sensor berbasis kamera AI atau radar.

71

Peneliti:

13. Jika sistem ini diintegrasikan dengan ATCS, bagaimana prosedur pengelolaan datanya di Dishub?

Dishub:

Jika sistem IoT ini terhubung ke ATCS, maka data sensor akan masuk ke server pusat Dishub. Dari situ, data dapat digunakan untuk dua tujuan:

1. Pengendalian operasional: ATCS bisa otomatis menyesuaikan durasi lampu hijau sesuai data sensor. Operator hanya bertindak sebagai pengawas.
2. Analisis jangka panjang: Data kepadatan dapat disimpan sebagai basis untuk membuat heatmap lalu lintas, evaluasi jam sibuk, dan penyusunan kebijakan transportasi. Misalnya, jika data menunjukkan simpang tertentu selalu padat di jam tertentu, maka Dishub bisa mempertimbangkan pelebaran jalan atau perubahan arus lalu lintas. Jadi, data dari IoT bukan hanya untuk kontrol real time, tapi juga sebagai alat pengambilan keputusan berbasis evidence.

Peneliti:

14. Apa yang akan menjadi indikator keberhasilan jika sistem Smart Traffic Light ini diterapkan?

Dishub:

Indikator keberhasilan bisa diukur dari beberapa aspek:

1. Waktu tunggu rata-rata kendaraan. Jika waktu tunggu berkurang signifikan dibandingkan sistem konvensional, berarti sistem bekerja baik.
2. Panjang antrian kendaraan. Jika panjang antrian di jam sibuk bisa dipangkas, berarti arus lalu lintas lebih efisien.
3. Jumlah keluhan masyarakat. Jika pengendara merasa lebih puas karena waktu lampu lebih adil sesuai kondisi, itu menjadi tanda positif.

71

4. Efisiensi energi dan emisi. Kendaraan yang tidak terlalu lama idle akan mengurangi konsumsi bahan bakar dan polusi udara.
5. Integrasi sistem. Jika sistem bisa terhubung dengan ATCS tanpa gangguan, berarti teknologi kompatibel untuk skala lebih besar.

Peneliti:

15. Apakah ada rencana Dishub untuk mulai mengadopsi teknologi berbasis IoT dalam 5 tahun ke depan?

Dishub:

Ya, ada. Dishub sudah memasukkan program digitalisasi lalu lintas dalam rencana jangka menengah daerah. Fokus awal masih pada perluasan ATCS ke lebih banyak simpang. Namun, teknologi berbasis IoT, termasuk sensor untuk smart traffic light, mulai dipertimbangkan dalam skala uji coba. Kami melihat tren kota-kota besar di Indonesia yang mulai beralih ke sistem berbasis IoT, seperti Jakarta dan Surabaya. Dengan adanya prototipe dari penelitian ini, sangat mungkin Pekanbaru juga mengambil langkah serupa, meski bertahap sesuai ketersediaan anggaran dan kesiapan infrastruktur.

Peneliti:

16. Menurut Dishub, bagaimana peran penelitian akademik seperti prototipe Smart Traffic Light ini dalam mendukung tugas Dishub di lapangan?

Dishub:

Penelitian akademik sangat penting karena memberikan gambaran solusi baru yang mungkin belum bisa langsung diimplementasikan Dishub karena keterbatasan anggaran dan regulasi. Prototipe semacam ini berfungsi sebagai proof of concept, menunjukkan bahwa teknologi dapat diaplikasikan dalam konteks lokal Pekanbaru. Dengan adanya hasil penelitian ini, Dishub dapat menjadikannya referensi dalam menyusun proposal anggaran atau proyek kerja sama dengan pihak swasta. Selain itu, penelitian akademik juga membantu kami

21

memetakan potensi risiko serta menilai kelayakan teknologi sebelum benar-benar diterapkan di lapangan.

Peneliti:

17. Bagaimana pandangan Dishub terhadap integrasi Smart Traffic Light dengan teknologi kendaraan pintar (smart vehicle) di masa depan?

Dishub:

Arah transportasi ke depan memang menuju vehicle-to-infrastructure (V2I), di mana kendaraan pintar bisa berkomunikasi dengan lampu lalu lintas. Jika Smart Traffic Light sudah berbasis IoT, maka integrasi dengan kendaraan pintar tinggal masalah protokol komunikasi. Kendaraan bisa menerima informasi kapan lampu akan berubah, atau bahkan memberikan data jumlah kendaraan langsung ke sistem Dishub. Walaupun di Pekanbaru kendaraan pintar belum banyak, persiapan infrastruktur harus dilakukan sejak dini agar tidak tertinggal.

Peneliti:

18. Menurut Dishub, apakah masyarakat Pekanbaru siap menerima teknologi baru ini?

Dishub:

Kesiapan masyarakat menjadi faktor penting. Dari pengalaman kami, masyarakat biasanya cepat beradaptasi dengan teknologi yang jelas manfaatnya, misalnya pengaturan lalu lintas berbasis ATCS yang mengurangi kemacetan. Namun, tantangan ada pada edukasi publik, karena masyarakat harus diberi pemahaman bahwa sistem ini dirancang untuk kepentingan bersama, bukan membatasi mobilitas mereka. Jika hasilnya nyata, yaitu waktu tunggu lebih singkat dan perjalanan lebih lancar, kami yakin masyarakat Pekanbaru akan mendukung penuh penerapan teknologi ini.

Peneliti:

19. Apa yang menjadi prioritas Dishub jika harus memilih antara menambah

21

Lampiran 2. Program Lengkap

```
#ifdef ESP8266
  #include <ESP8266WiFi.h>    /* WiFi library for ESP8266 */
#else
  #include <WiFi.h>           /* WiFi library for ESP32 */
#endif
#include <Wire.h>
#include <PubSubClient.h>

#define wifi_ssid "GID"
#define wifi_password "nanafisyah18"
#define mqtt_server "192.168.1.1"

#define kemacetan_simpang1 "Sensor/Kemacetan1/Simpang1"
#define kemacetan_simpang2 "Sensor/Kemacetan2/Simpang2"
#define kemacetan_simpang3 "Sensor/Kemacetan3/Simpang3"
#define kemacetan_simpang4 "Sensor/Kemacetan4/Simpang4"

WiFiClient espClient;
PubSubClient client(espClient);

#include <Wire.h>
#include <LiquidCrystal_I2C.h>

LiquidCrystal_I2C lcd(0x27, 2, 16);

int hijau1 = 13;
int hijau2 = 12;
int hijau3 = 14;
int hijau4 = 27;
int merah1 = 26;
int merah2 = 25;
int merah3 = 33;
int merah4 = 32;
int kuning1 = 35;
int kuning2 = 34;
int kuning3 = 39;
```



```

int kuning4 = 36;
int hasil1 = 0;
int hasil2 = 0;
int hasil3 = 0;
int hasil4 = 0;
int hasil5 = 0;
int hasil6 = 0;
int hasil7 = 0;
int hasil8 = 0;
const int SensorProximty1 = 19;
const int SensorProximty2 = 18;
const int SensorProximty3 = 5;
const int SensorProximty4 = 17;
const int SensorProximty5 = 16;
const int SensorProximty6 = 4;
const int SensorProximty7 = 2;
const int SensorProximty8 = 15;

void setup() {
  Serial.begin(115200);

  pinMode(SensorProximty1, INPUT);
  pinMode(SensorProximty2, INPUT);
  pinMode(SensorProximty3, INPUT);
  pinMode(SensorProximty4, INPUT);
  pinMode(SensorProximty5, INPUT);
  pinMode(SensorProximty6, INPUT);
  pinMode(SensorProximty7, INPUT);
  pinMode(SensorProximty8, INPUT);
  pinMode(hijau1, OUTPUT);
  pinMode(hijau2, OUTPUT);
  pinMode(hijau3, OUTPUT);
  pinMode(hijau4, OUTPUT);
  pinMode(merah1, OUTPUT);
  pinMode(merah2, OUTPUT);
  pinMode(merah3, OUTPUT);
  pinMode(merah4, OUTPUT);

```



```

digitalWrite(hijau1, HIGH);
digitalWrite(hijau2, HIGH);
digitalWrite(hijau3, HIGH);
digitalWrite(hijau4, HIGH);
digitalWrite(merah1, HIGH);
digitalWrite(merah2, HIGH);
digitalWrite(merah3, HIGH);
digitalWrite(merah4, HIGH);
pinMode(kuning1, OUTPUT);
pinMode(kuning2, OUTPUT);
pinMode(kuning3, OUTPUT);
pinMode(kuning4, OUTPUT);
digitalWrite(kuning1, LOW);
digitalWrite(kuning2, LOW);
digitalWrite(kuning3, LOW);
digitalWrite(kuning4, LOW);
lcd.begin();
lcd.backlight();
lcd.print("KAK JIHAN");
lcd.setCursor(0, 1);
lcd.print("OPTIMASI");
delay(2000);
lcd.clear();

setup_wifi();
client.setServer(mqtt_server, 1883);
}

void setup_wifi() {
  delay(10);
  // We start by connecting to a WiFi network
  Serial.println();
  Serial.print("Connecting to ");
  Serial.println(wifi_ssid);

  WiFi.begin(wifi_ssid, wifi_password);

```



```

while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
}

Serial.println("");
Serial.println("WiFi connected");
Serial.println("IP address: ");
Serial.println(WiFi.localIP());
}

void reconnect() {
    // Loop until we're reconnected
    while (!client.connected()) {
        Serial.print("Attempting MQTT connection...");

        if (client.connect("ESP8266Client")) {
            Serial.println("connected");
        } else {
            Serial.print("failed, rc=");
            Serial.print(client.state());
            Serial.println(" try again in 5 seconds");
            delay(5000);
        }
    }
}

void loop() {
    if (!client.connected()) {
        reconnect();
    }
    client.loop();
    hasil1 = digitalRead(SensorProximty1);
    hasil2 = digitalRead(SensorProximty2);
    lampu1();
    digitalWrite(kuning1, HIGH);
    kuning();
}

```



```

digitalWrite(kuning1, LOW);
hasil3 = digitalRead(SensorProximty3);
hasil4 = digitalRead(SensorProximty4);
lampu2();
digitalWrite(kuning2, HIGH);
kuning();
digitalWrite(kuning2, LOW);
hasil5 = digitalRead(SensorProximty5);
hasil6 = digitalRead(SensorProximty6);
lampu3();
digitalWrite(kuning3, HIGH);
kuning();
digitalWrite(kuning3, LOW);
hasil7 = digitalRead(SensorProximty7);
hasil8 = digitalRead(SensorProximty8);
lampu4();
digitalWrite(kuning4, HIGH);
kuning();
digitalWrite(kuning4, LOW);
}

void Getdata() {
  hasil1 = digitalRead(SensorProximty1);
  hasil2 = digitalRead(SensorProximty2);
  hasil3 = digitalRead(SensorProximty3);
  hasil4 = digitalRead(SensorProximty4);
  hasil5 = digitalRead(SensorProximty5);
  hasil6 = digitalRead(SensorProximty6);
  hasil7 = digitalRead(SensorProximty7);
  hasil8 = digitalRead(SensorProximty8);
}

void kuning(){
  lcd.setCursor(0, 1);
  lcd.print("03 detik");
  delay(1000);
  lcd.setCursor(0, 1);

```



```
lcd.print("02 detik");  
delay(1000);  
lcd.setCursor(0, 1);  
lcd.print("01 detik");  
delay(1000);  
}
```

```
void dela5(){  
  lcd.setCursor(0, 1);  
  lcd.print("20 detik");  
  delay(1000);  
  lcd.setCursor(0, 1);  
  lcd.print("19 detik");  
  delay(1000);  
  lcd.setCursor(0, 1);  
  lcd.print("18 detik");  
  delay(1000);  
  lcd.setCursor(0, 1);  
  lcd.print("17 detik");  
  delay(1000);  
  lcd.setCursor(0, 1);  
  lcd.print("16 detik");  
  delay(1000);  
  lcd.setCursor(0, 1);  
  lcd.print("15 detik");  
  delay(1000);  
  lcd.setCursor(0, 1);  
  lcd.print("14 detik");  
  delay(1000);  
  lcd.setCursor(0, 1);  
  lcd.print("13 detik");  
  delay(1000);  
  lcd.setCursor(0, 1);  
  lcd.print("12 detik");  
  delay(1000);  
  lcd.setCursor(0, 1);  
  lcd.print("11 detik");  
}
```



```

delay(1000);
lcd.setCursor(0, 1);
lcd.print("10 detik");
delay(1000);
lcd.setCursor(0, 1);
lcd.print("09 detik");
delay(1000);
lcd.setCursor(0, 1);
lcd.print("08 detik");
delay(1000);
lcd.setCursor(0, 1);
lcd.print("07 detik");
delay(1000);
lcd.setCursor(0, 1);
lcd.print("06 detik");
delay(1000);
lcd.setCursor(0, 1);
lcd.print("05 detik");
delay(1000);
lcd.setCursor(0, 1);
lcd.print("04 detik");
delay(1000);
lcd.setCursor(0, 1);
lcd.print("03 detik");
delay(1000);
lcd.setCursor(0, 1);
lcd.print("02 detik");
delay(1000);
lcd.setCursor(0, 1);
lcd.print("01 detik");
delay(1000);
}
void dela8(){
  lcd.setCursor(0, 1);
  lcd.print("25 detik");
  delay(1000);
  lcd.setCursor(0, 1);

```



```
lcd.print("24 detik");  
delay(1000);  
lcd.setCursor(0, 1);  
lcd.print("23 detik");  
delay(1000);  
lcd.setCursor(0, 1);  
lcd.print("22 detik");  
delay(1000);  
lcd.setCursor(0, 1);  
lcd.print("21 detik");  
delay(1000);  
lcd.setCursor(0, 1);  
lcd.print("20 detik");  
delay(1000);  
lcd.setCursor(0, 1);  
lcd.print("19 detik");  
delay(1000);  
lcd.setCursor(0, 1);  
lcd.print("18 detik");  
delay(1000);  
lcd.setCursor(0, 1);  
lcd.print("17 detik");  
delay(1000);  
lcd.setCursor(0, 1);  
lcd.print("16 detik");  
delay(1000);  
lcd.setCursor(0, 1);  
lcd.print("15 detik");  
delay(1000);  
lcd.setCursor(0, 1);  
lcd.print("14 detik");  
delay(1000);  
lcd.setCursor(0, 1);  
lcd.print("13 detik");  
delay(1000);  
lcd.setCursor(0, 1);  
lcd.print("12 detik");
```



```

delay(1000);
lcd.setCursor(0, 1);
lcd.print("11 detik");
delay(1000);
lcd.setCursor(0, 1);
lcd.print("10 detik");
delay(1000);
lcd.setCursor(0, 1);
lcd.print("09 detik");
delay(1000);
lcd.setCursor(0, 1);
lcd.print("08 detik");
delay(1000);
lcd.setCursor(0, 1);
lcd.print("07 detik");
delay(1000);
lcd.setCursor(0, 1);
lcd.print("06 detik");
delay(1000);
lcd.setCursor(0, 1);
lcd.print("05 detik");
delay(1000);
lcd.setCursor(0, 1);
lcd.print("04 detik");
delay(1000);
lcd.setCursor(0, 1);
lcd.print("03 detik");
delay(1000);
lcd.setCursor(0, 1);
lcd.print("02 detik");
delay(1000);
lcd.setCursor(0, 1);
lcd.print("01 detik");
delay(1000);
}
void dela10(){
  lcd.setCursor(0, 1);

```



```
lcd.print("30 detik");  
delay(1000);  
lcd.setCursor(0, 1);  
lcd.print("29 detik");  
delay(1000);  
lcd.setCursor(0, 1);  
lcd.print("28 detik");  
delay(1000);  
lcd.setCursor(0, 1);  
lcd.print("27 detik");  
delay(1000);  
lcd.setCursor(0, 1);  
lcd.print("26 detik");  
delay(1000);  
lcd.setCursor(0, 1);  
lcd.print("25 detik");  
delay(1000);  
lcd.setCursor(0, 1);  
lcd.print("24 detik");  
delay(1000);  
lcd.setCursor(0, 1);  
lcd.print("23 detik");  
delay(1000);  
lcd.setCursor(0, 1);  
lcd.print("22 detik");  
delay(1000);  
lcd.setCursor(0, 1);  
lcd.print("21 detik");  
delay(1000);  
lcd.setCursor(0, 1);  
lcd.print("20 detik");  
delay(1000);  
lcd.setCursor(0, 1);  
lcd.print("19 detik");  
delay(1000);  
lcd.setCursor(0, 1);  
lcd.print("18 detik");
```



```
delay(1000);  
lcd.setCursor(0, 1);  
lcd.print("17 detik");  
delay(1000);  
lcd.setCursor(0, 1);  
lcd.print("16 detik");  
delay(1000);  
lcd.setCursor(0, 1);  
lcd.print("15 detik");  
delay(1000);  
lcd.setCursor(0, 1);  
lcd.print("14 detik");  
delay(1000);  
lcd.setCursor(0, 1);  
lcd.print("13 detik");  
delay(1000);  
lcd.setCursor(0, 1);  
lcd.print("12 detik");  
delay(1000);  
lcd.setCursor(0, 1);  
lcd.print("11 detik");  
delay(1000);  
lcd.setCursor(0, 1);  
lcd.print("10 detik");  
delay(1000);  
lcd.setCursor(0, 1);  
lcd.print("09 detik");  
delay(1000);  
lcd.setCursor(0, 1);  
lcd.print("08 detik");  
delay(1000);  
lcd.setCursor(0, 1);  
lcd.print("07 detik");  
delay(1000);  
lcd.setCursor(0, 1);  
lcd.print("06 detik");  
delay(1000);
```



```

    lcd.setCursor(0, 1);
    lcd.print("05 detik");
    delay(1000);
    lcd.setCursor(0, 1);
    lcd.print("04 detik");
    delay(1000);
    lcd.setCursor(0, 1);
    lcd.print("03 detik");
    delay(1000);
    lcd.setCursor(0, 1);
    lcd.print("02 detik");
    delay(1000);
    lcd.setCursor(0, 1);
    lcd.print("01 detik");
    delay(1000);
}

void lampu1(){
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("Simpang 1");
    digitalWrite(hijau1, HIGH);
    digitalWrite(hijau2, LOW);
    digitalWrite(hijau3, LOW);
    digitalWrite(hijau4, LOW);
    digitalWrite(merah1, LOW);
    digitalWrite(merah2, HIGH);
    digitalWrite(merah3, HIGH);
    digitalWrite(merah4, HIGH);
    if(hasil1 == LOW && hasil2 == HIGH)
    {
        lcd.setCursor(14, 1);
        lcd.print("M1");
        Serial.println("1dela8");
        client.publish(kemacetan_simpang1, String("Macet Tingkat
1").c_str(), true);
        dela8();
    }
}

```



```

}
if(hasil2 == LOW && hasil1 == LOW)
{
    lcd.setCursor(14, 1);
    lcd.print("M2");
    Serial.println("1dela10");
    client.publish(kemacetan_simpang1, String("Macet Tingkat
2").c_str(), true);
    dela10();
}
if(hasil1 == HIGH && hasil2 == HIGH)
{
    lcd.setCursor(14, 1);
    lcd.print("N");
    Serial.println("1dela5");
    client.publish(kemacetan_simpang1, String("Normal").c_str(),
true);
    dela5();
}
delay(250);
}

void lampu2(){
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("Simpang 2");
    digitalWrite(hijau1, LOW);
    digitalWrite(hijau2, HIGH);
    digitalWrite(hijau3, LOW);
    digitalWrite(hijau4, LOW);
    digitalWrite(merah1, HIGH);
    digitalWrite(merah2, LOW);
    digitalWrite(merah3, HIGH);
    digitalWrite(merah4, HIGH);
    if(hasil3 == LOW && hasil4 == HIGH)
    {
        lcd.setCursor(14, 1);

```



```

    lcd.print("M1");
    Serial.println("2dela8");
    client.publish(kemacetan_simpang2, String("Macet Tingkat
1").c_str(), true);
    dela8();
}
if(hasil4 == LOW && hasil3 == LOW)
{
    lcd.setCursor(14, 1);
    lcd.print("M2");
    Serial.println("2dela10");
    client.publish(kemacetan_simpang2, String("Macet Tingkat
2").c_str(), true);
    dela10();
}
if(hasil3 == HIGH && hasil4 == HIGH)
{
    lcd.setCursor(14, 1);
    lcd.print("N");
    Serial.println("2dela5");
    client.publish(kemacetan_simpang2, String("Normal").c_str(),
true);
    dela5();
}
delay(250);
}

void lampu3(){
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("Simpang 3");
    digitalWrite(hijau1, LOW);
    digitalWrite(hijau2, LOW);
    digitalWrite(hijau3, HIGH);
    digitalWrite(hijau4, LOW);
    digitalWrite(merah1, HIGH);
    digitalWrite(merah2, HIGH);
}

```



```

digitalWrite(merah3, LOW);
digitalWrite(merah4, HIGH);
if(hasil5 == LOW && hasil6 == HIGH)
{
    lcd.setCursor(14, 1);
    lcd.print("M1");
    Serial.println("3dela8");
    client.publish(kemacetan_simpang3, String("Macet Tingkat
1").c_str(), true);
    dela8();
}
if(hasil6 == LOW && hasil5 == LOW)
{
    lcd.setCursor(14, 1);
    lcd.print("M2");
    Serial.println("3dela10");
    client.publish(kemacetan_simpang3, String("Macet Tingkat
2").c_str(), true);
    dela10();
}
if(hasil5 == HIGH && hasil6 == HIGH)
{
    lcd.setCursor(14, 1);
    lcd.print("N");
    Serial.println("3dela5");
    client.publish(kemacetan_simpang3, String("Normal").c_str(),
true);
    dela5();
}
delay(250);
}

void lampu4(){
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("Simpang 4");
    digitalWrite(hijau1, LOW);

```



```

digitalWrite(hijau2, LOW);
digitalWrite(hijau3, LOW);
digitalWrite(hijau4, HIGH);
digitalWrite(merah1, HIGH);
digitalWrite(merah2, HIGH);
digitalWrite(merah3, HIGH);
digitalWrite(merah4, LOW);
if(hasil7 == LOW && hasil8 == HIGH)
{
    lcd.setCursor(14, 1);
    lcd.print("M1");
    Serial.println("4dela8");
    client.publish(kemacetan_simpang4, String("Macet Tingkat
1").c_str(), true);
    dela8();
}
if(hasil8 == LOW && hasil7 == LOW)
{
    lcd.setCursor(14, 1);
    lcd.print("M2");
    Serial.println("4dela10");
    client.publish(kemacetan_simpang4, String("Macet Tingkat
2").c_str(), true);
    dela10();
}
if(hasil7 == HIGH && hasil8 == HIGH)
{
    lcd.setCursor(14, 1);
    lcd.print("N");
    Serial.println("4dela5");
    client.publish(kemacetan_simpang4, String("Normal").c_str(),
true);
    dela5();
}
delay(250);
}

```


Lampiran 3. Datasheet IR



ADIY

IR Sensor Module with Pot



Description:

ADIY IR Sensor senses whether there is an object near it or not. The IR stands for Infrared sensor. Infrared is the light out of our visible spectrum. This module can be used in obstacle detection in autonomous robots and cars, in positional feedback for control system, industrial safety system, Wheel encoder etc.

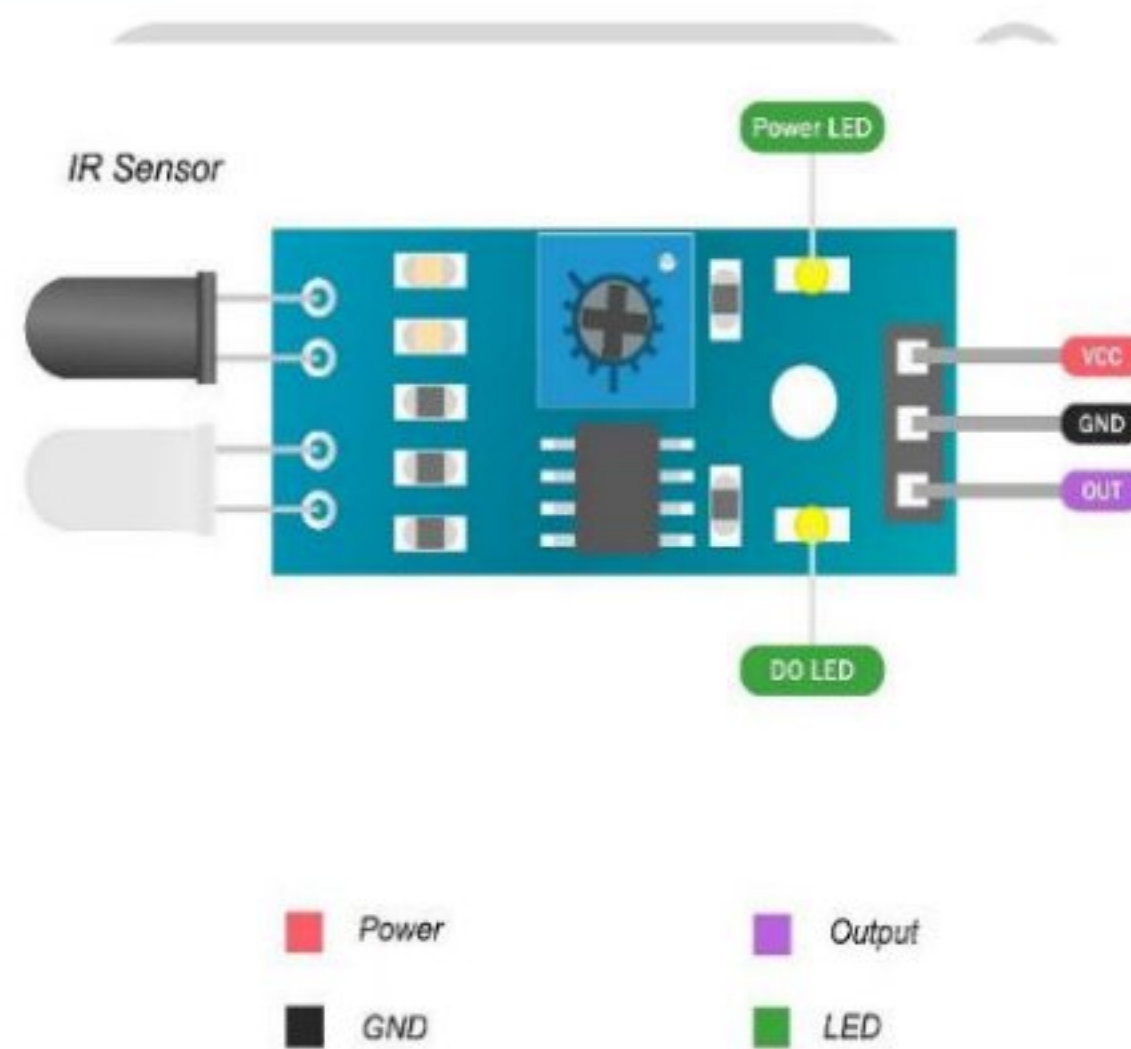
Features:

1. Potentiometer knob to adjust the detection distance
2. Obstacle detection
3. Onboard detection indication
4. Easy to assemble and use
5. Comparator using LM393

Specification:

- Operating Voltage: 3 to 5 VDC
- Measuring Range: 2 to 10 cm
- Detection Angle: 35°
- Current consumption: 20mA

Pin Configuration:



VCC: 5V external voltage

GND: Ground

OUT: Digital output interface (0 and 1)

How it works:

The white LED here is an IR LED which works as the transmitter and the component next to the IR LED is a photodiode that works as the receiver in the IR sensor. The IR transmitter continuously emits the IR light and the IR receiver keeps on checking for the reflected light. If the light gets



reflected back by hitting any object in front it, the IR receiver receives this light. This way the object is detected in the case of the IR sensor.

The blue knob here is a potentiometer. User can control the range i.e. from how far you want to detect the object by changing the value of the potentiometer.

An IR sensor has two small LED indicators – one for power, which is ON the entire time the sensor is ON; the other is the Signal LED which detects the object.

Application:

- Night Vision Devices
- Radiation Thermometers
- Flame Monitors
- Water Analysis
- Rail Safety

