

**PERANCANGAN ALAT PERANGKAT KERAS
PADA *ADAPTIVE TRAFFIC LIGHT***

TUGAS AKHIR

**Diajukan untuk melengkapi salah satu syarat kelulusan
Program Diploma Tiga**



Oleh :

Hangga Adji Novianto

NPM. 213304015

**PROGRAM STUDI TEKNIK KOMPUTER KONTROL
JURUSAN TEKNIK
POLITEKNIK NEGERI MADIUN
JULI 2024**

**PERANCANGAN ALAT PERANGKAT KERAS
PADA *ADAPTIVE TRAFFIC LIGHT***

TUGAS AKHIR

**Diajukan untuk melengkapi salah satu syarat kelulusan
Program Diploma Tiga**



**Oleh :
Hangga Adji Novianto
NPM. 213304015**

**PROGRAM STUDI TEKNIK KOMPUTER KONTROL
JURUSAN TEKNIK
POLITEKNIK NEGERI MADIUN
JULI 2024**



**HALAMAN PERSETUJUAN
TUGAS AKHIR**

Judul TA : Perancangan Alat Perangkat Keras Pada *Adaptive Traffic Light*
Nama Penulis : Hangga Adji Novianto
NPM : 213304015
Program Studi : Teknik Komputer Kontrol
Jurusan : Teknik

Telah memenuhi segala persyaratan dan **DISETUJUI** untuk diajukan pada Sidang Tugas Akhir.

Madiun, 08 Juli 2024

Pembimbing I

Pembimbing II

Dahris Shahab, S.Pd., M.Pd.

NIDN. 0717016902

Imam Junaedi, S.T., M.T.

NIDN. 0025099009

Mengetahui,

Koordinator Program Studi

Teknik Komputer Kontrol

Aulia El Hakim, S.ST., M.T

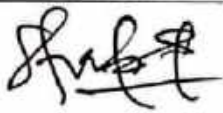
NIP. 198509122015041003



HALAMAN PENGESAHAN
TUGAS AKHIR

Judul TA : Perancangan Alat Perangkat Keras Pada *Adaptive Traffic Light*
Nama Mahasiswa : Hangga Adji Novianto
NPM : 213304015
Program Studi : Teknik Komputer Kontrol
Jurusan : Teknik
Telah diseminarkan pada hari kamis tanggal 18 Juli 2024 dan dinyatakan Lulus.

Madiun, 18 Juli 2024
Menyetujui,

No	Nama	Penguji	Tanda Tangan
1	<u>Dahris Shahab, S.Pd., M.Pd.</u> NIDN. 0717016902	I	
2	<u>Dirvi Eko Juliando Sudirman, S.Pd., M.T</u> NIDN. 0027078306	II	
3	<u>Rendi Pambudi Wicaksono, S.ST., M.Tr.T</u> NIDN. 0003069307	III	

Mengetahui,
Ketua Jurusan Teknik

Achmad Aminudin, S.Pd., M.T
NIP. 198704082015041003



KEMENTERIAN PENDIDIKAN, KEBUDAYAAN,
RISET, DAN TEKNOLOGI

POLITEKNIK NEGERI MADIUN

Jalan Serayu Nomor 84 Madiun Kode Pos 63133

Telepon (0351) 452970, Faksimile (0351) 492960

Laman : www.pnm.ac.id / Email : sekretariat@pnm.ac.id

SURAT PERNYATAAN ORISINALITAS

Yang bertanda tangan di bawah ini:

Nama : Hangga Adji Novianto
NPM : 213304015
Program Studi : Teknik Komputer Kontrol
Jurusan : Teknik

Dengan ini menyatakan bahwa Tugas Akhir saya dengan judul :

Perancangan Alat Perangkat Keras Pada *Adaptive Traffic Light*

Adalah orisinal dan merupakan hasil pemikiran saya sendiri, bukan hasil saduran dan/atau jiplakan dari karya orang lain.

Bilamana dikemudian hari ditemukan ketidaksesuaian dengan pernyataan ini, maka saya bersedia menerima segala bentuk sanksi sesuai ketentuan yang berlaku.

Demikian pernyataan ini dibuat dengan sesungguhnya dan sebenar – benarnya.

Madiun, 08 Juli 2024

Yang Menyatakan

Hangga Adji Novianto
NPM. 213304015

ABSTRAK

Banyaknya jumlah persimpangan jalan dan semakin banyaknya jumlah kendaraan di Indonesia, lampu lalu lintas menjadi salah satu hal yang paling penting untuk mengatur kenyamanan dan keamanan berlalu lintas. Pengaplikasian sistem perhitungan lamanya waktu pada lampu lalu lintas dilakukan secara manual melalui sistem prediksi atau perkiraan kepadatan kendaraan yang biasanya menggunakan metode estimasi statistik kepadatan kendaraan. Oleh karena itu, diperlukan pengaturan waktu lampu lalu lintas yang optimal, sehingga kemacetan yang disebabkan oleh ketidaksesuaian pengaturan lampu lalu lintas bisa dikurangi bahkan bisa diatasi. Penulis mengusulkan penelitian Tugas Akhir yang berjudul “Perancangan Alat Perangkat Keras Pada *Adaptive Traffic Light*”. Penelitian ini bertujuan untuk mengembangkan suatu sistem yang *responsif* terhadap perubahan kondisi lalu lintas dengan memanfaatkan teknologi sensor dan algoritma *adaptive* sesuai dengan kepadatan kendaraan. Pentingnya kemampuan sistem untuk merespons perubahan lalu lintas dengan cepat ditekankan, serta adanya fitur komunikasi antar-persimpangan untuk meningkatkan koordinasi. Penelitian ini juga bertujuan untuk membuat perangkat keras yang dapat dimonitoring di *website*. Perancangan dan pembuatan perangkat keras pada *adaptive traffic light* menggunakan *raspberry pi* sebagai *broker* digunakan untuk menjadi perantara antara komunikasi data dari *jetson nano* maupun *monitoring web*.

Kata kunci: Perangkat keras, *Traffic light*, *Raspberry Pi*

ABSTRACT

With the large number of road intersections and the increasing number of vehicles in Indonesia, traffic lights have become one of the most important things to regulate traffic comfort and safety. The application of the system for calculating the length of time at traffic lights is carried out manually through a prediction or vehicle density estimation system which usually uses a statistical vehicle density estimation method. Therefore, it is necessary to set optimal traffic light timings so that congestion caused by inappropriate traffic light settings can be reduced and even overcome. The author proposes a final assignment research entitled "Designing Hardware Tools for Adaptive Traffic Light". This research aims to develop a system that is responsive to changes in traffic conditions by utilizing sensor technology and adaptive algorithms according to vehicle density. The importance of the system's ability to respond quickly to traffic changes is emphasized, as well as inter-junction communication features to improve coordination. This research also aims to create hardware that can be monitored on websites. Design and manufacture of hardware for adaptive traffic light using Raspberry Pi as a broker used to act as an intermediary between data communication from Jetson Nano and web monitoring.

Keywords: Hardware, Traffic light, Raspberry Pi

KATA PENGANTAR

Dengan mengucapkan syukur atas kehadiran Allah Subhanahu Wa Ta'Ala atas limpahan Rahmat dan karunia-Nya, sehingga penulis dapat menyelesaikan Proposal Tugas Akhir dengan judul “Perancangan Alat Perangkat Keras Pada *Adaptive Traffic Light*”. Proposal ini disusun untuk memenuhi persyaratan tugas akhir pada Program Studi Teknik Komputer Kontrol Politeknik Negeri Madiun. Tidak lupa penulis mengucapkan terima kasih kepada semua pihak yang telah membantu dalam proses penyusunan Proposal Tugas Akhir ini, kepada :

1. Allah Subhanahu Wa Ta'Ala yang selalu memberikan Rahmat, karunia, serta hidayah yang tak terhingga besarnya.
2. Kedua orang tua dan saudara kandung yang telah memberi semangat serta doa yang tiada henti.
3. Bapak Dahris Shahab, S.Pd., M.Pd. selaku Dosen Pembimbing I Tugas Akhir.
4. Bapak Imam Junaedi, S.T., M.T. selaku Dosen Pembimbing II Tugas Akhir.
5. Bapak Aulia El Hakim, S.ST., M.T., selaku Koordinator Program Studi Teknik Komputer Kontrol Politeknik Negeri Madiun.
6. Bapak Achmad Aminudin, S.Pd., M.T., selaku Ketua Jurusan Teknik Politeknik Negeri Madiun.
7. Bapak Muhamad Fajar Subkhan, S.T., M.T. selaku Direktur Politeknik Negeri Madiun.
8. Tim seperjuangan saya, Alvindra Putra Ardhana dan Renata Tristiana yang telah membantu serta berjuang bersama.

Demikian Proposal Tugas Akhir ini disusun, semoga dapat bermanfaat bagi pihak-pihak yang berkepentingan. Oleh karena itu, kritik dan saran sangat diharapkan demi perbaikan Proposal Tugas Akhir.

Madiun, 08 Juli 2024

Penulis

DAFTAR ISI

HALAMAN JUDUL (<i>COVER</i>)	i
HALAMAN PERSETUJUAN TUGAS AKHIR.. Error! Bookmark not defined.	
HALAMAN PENGESAHAN TUGAS AKHIR.. Error! Bookmark not defined.	
SURAT PERNYATAAN ORISINALITAS	iv
ABSTRAK	v
<i>ABSTRACT</i>	vi
KATA PENGANTAR	vii
DAFTAR ISI	viii
DAFTAR TABEL	x
DAFTAR GAMBAR	xi
DAFTAR LAMPIRAN	xiii
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	3
1.3 Tujuan	3
1.4 Manfaat Tugas Akhir	3
1.5 Batasan Masalah.....	3
BAB II TINJAUAN PUSTAKA.....	4
2.1 Penelitian Terdahulu	4
2.2 Tinjauan Teoritis	7
2.3 Kerangka Pemikiran.....	15
BAB III METODE RANCANGAN DAN PENELITIAN	17
3.1 Metode.....	17
3.2 Lokasi Penelitian.....	17
3.3 Tahap Penelitian.....	17
3.4 Rancangan Penelitian	19
3.5 Perancangan Mekanika	26
3.6 Perancangan Elektrikal.....	28
3.7 Proses Pembuatan.....	32
3.8 Proses Pengujian	35
BAB IV HASIL DAN PEMBAHASAN	48

4.1	Merancang dan Membuat Perangkat Keras Pada <i>Adaptive Traffic Light</i> Berdasarkan Kepadatan Kendaraan Pada Persimpangan Jalan	48
4.2	Mengirim dan Menerima Sinyal Dari Sistem Pendeteksi dan Penghitung Kepadatan Kendaraan	49
4.3	Perangkat Keras <i>Adaptive Traffic Light</i> Dapat Mengirim Data Ke Sistem Monitoring <i>Adaptive Traffic Light</i>	51
BAB V KESIMPULAN DAN SARAN.....		57
5.1	Kesimpulan	57
5.2	Saran.....	58
DAFTAR PUSTAKA		59
LAMPIRAN.....		60

DAFTAR TABEL

Tabel 3. 1	Alat dan Bahan.....	20
Tabel 3. 2	Hasil Pengujian Sensor Tegangan <i>DC</i> Kondisi Menyala	41
Tabel 3. 3	Hasil Pengujian Sensor tegangan <i>DC</i> Kondisi Padam.....	42
Tabel 3. 4	Hasil Pengujian Pengiriman Kecepatan Data	45
Tabel 4. 1	Hasil Pengukuran Kecepatan Pengiriman Data <i>Capture</i> gambar	50
Tabel 4. 2	Hasil Pengujian Sensor Tegangan DC Ketika Kondisi Menyala.....	54
Tabel 4. 3	Hasil Pengujian Sensor Tegangan DC Ketika Kondisi Padam.....	55

DAFTAR GAMBAR

Gambar 2. 1 Foto Wawancara Dengan Dinas Perhubungan Kota Madiun.....	4
Gambar 2. 2 Lampu Lalu Lintas	8
Gambar 2. 3 <i>Raspberry Pi</i>	9
Gambar 2. 4 <i>ESP32</i>	10
Gambar 2. 5 Konfigurasi <i>ESP32</i>	11
Gambar 2. 6 <i>Relay Module 8 Channel</i>	12
Gambar 2. 7 <i>Miniature Circuit Breaker (MCB)</i>	13
Gambar 2. 8 Sensor tegangan <i>DC</i>	13
Gambar 2. 9 <i>Switch Power over Ethernet (PoE)</i>	14
Gambar 2. 10 <i>Router</i>	15
Gambar 3. 1 Tahap Penelitian.....	18
Gambar 3. 2 Diagram Blok Sistem	21
Gambar 3. 3 <i>FlowChart</i> Kontrol Otomatis	22
Gambar 3. 4 <i>FlowChart</i> Kontrol Manual.....	23
Gambar 3. 5 <i>FlowChart</i> Monitoring Web	24
Gambar 3. 6 <i>FlowChart</i> Sensor Tegangan.....	25
Gambar 3. 7 Desain <i>Box</i> Panel.....	26
Gambar 3. 8 Skematik <i>PCB Control</i> lampu lalu lintas	29
Gambar 3. 9 Board <i>PCB Control</i> lampu lalu lintas	29
Gambar 3. 10 Skematik <i>PCB</i> deteksi kerusakan lampu lalu lintas	30
Gambar 3. 11 Board <i>PCB</i> deteksi kerusakan lampu lalu lintas	30
Gambar 3. 12 Wiring Diagram Kontrol Lampu Lalu Lintas	31
Gambar 3. 13 Wiring Diagram Sensor Tegangan.....	32
Gambar 3. 14 Pembuatan <i>PCB Control</i>	33
Gambar 3. 15 Pembuatan <i>PCB</i> Sensor Tegangan.....	33
Gambar 3. 16 Proses Pengkabelan Pada Komponen	34
Gambar 3. 17 Tampilan Rangkaian Keseluruhan Sistem	35
Gambar 3. 18 Pengukuran Arus Voltase Yang Masuk <i>Box</i> Panel.....	36
Gambar 3. 19 Pengukuran Tegangan <i>Output Power Supply 12V</i>	36
Gambar 3. 20 Pengukuran <i>Output</i> Dari <i>Stepdown</i>	37
Gambar 3. 21 Pengukuran Tegangan <i>Input ESP32</i> Kontrol	37

Gambar 3. 22 Pengukuran <i>Input Relay Module</i>	38
Gambar 3. 23 Pengukuran <i>Output ESP32</i>	38
Gambar 3. 24 Pengujian <i>Input Sensor</i> tegangan Ketika Lampu Menyala	39
Gambar 3. 25 Hasil <i>Output Sensor</i> Tegangan ketika Lampu Menyala.....	39
Gambar 3. 26 Pengujian <i>Input Sensor</i> tegangan Ketika Lampu Padam	40
Gambar 3. 27 Hasil <i>Output Sensor</i> Tegangan Ketika Lampu Padam.....	40
Gambar 3. 28 Hasil Koneksi Dengan <i>Jetson Nano & ESP32</i> Kontrol	44
Gambar 3. 29 Hasil Pengujian Kecepatan Pengiriman Data.....	45
Gambar 4. 1 <i>Box Panel Adaptive Traffic light</i>	48
Gambar 4. 2 Hasil <i>Integrasi Jetson Nano</i> Dengan <i>ESP32</i>	49
Gambar 4. 3 Tampilan <i>Realtime</i> Lampu Merah Semua Fase	52
Gambar 4. 4 Tampilan <i>Realtime</i> Lampu Hijau dan Kuning Fase 1	52
Gambar 4. 5 Tampilan <i>Realtime</i> Lampu Hijau dan Kuning Fase 2	52
Gambar 4. 6 Tampilan <i>Realtime</i> Lampu Hijau dan Kuning Fase 3	52
Gambar 4. 7 Hasil Pengujian <i>Input Sensor</i> Tegangan Ketika Lampu Menyala ...	53
Gambar 4. 8 Hasil Pengujian <i>Input Sensor</i> tegangan Ketika Lampu Padam.....	53

DAFTAR LAMPIRAN

Lampiran 1. Lembar Monitoring Bimbingan Tugas Akhir

Lampiran 2. *Form* Revisi Sidang Tugas Akhir

Lampiran 3. Lembar Pernyataan Progres Pengerjaan Tugas Akhir

Lampiran 4. Lampiran Program

Lampiran 5. Biodata Peneliti Tugas Akhir

Lampiran 6. Motto dan Persembahan

BAB I

PENDAHULUAN

1.1 Latar Belakang

Kemacetan lalu lintas semakin hari semakin meningkat dan banyak permasalahan yang harus kita hadapi. Penyebab utama kemacetan lalu lintas adalah tingginya volume kendaraan, infrastruktur yang belum memadai, dan distribusi pembangunan yang tidak rasional. Kemacetan lalu lintas ditandai dengan waktu perjalanan yang lebih lama, kecepatan yang lebih lambat, dan antrian kendaraan yang meningkat. Ketika permintaan lalu lintas tinggi, sehingga interaksi antar kendaraan memperlambat kecepatan arus lalu lintas, hal ini juga mengakibatkan kemacetan lalu lintas. Sistem konvensional didasarkan pada pengatur waktu yang menyebabkan keterbatasan seperti pemborosan waktu, kemacetan parah, meskipun tidak ada kendaraan di jalan, sinyal tetap berwarna merah karena pengaturan waktu statis dari pengatur waktu sinyal. (Basil & Sawant, 2017)

Banyaknya jumlah persimpangan jalan dan semakin banyaknya jumlah kendaraan di Indonesia, lampu lalu lintas menjadi salah satu hal yang paling penting untuk mengatur kenyamanan dan keamanan berlalu lintas. Pengaplikasian sistem perhitungan lamanya waktu pada lampu lalu lintas dilakukan secara manual melalui sistem prediksi atau perkiraan kepadatan kendaraan yang biasanya menggunakan metode estimasi statistik kepadatan kendaraan. Namun pada kenyatannya di lapangan kepadatan lalu lintas tidak selalu sesuai dengan prediksi. Sehingga tidak jarang lampu lalu lintas justru menjadi sumber utama penyebab kemacetan. (Ramadhan et al., n.d.)

Oleh karena itu, diperlukan pengaturan waktu lampu lalu lintas yang optimal, sehingga kemacetan yang disebabkan oleh ketidaksesuaian pengaturan lampu lalu lintas bisa dikurangi bahkan bisa diatasi. Di Indonesia, sistem lalu lintas otomatis sudah mulai diterapkan sejak beberapa tahun yang lalu. Namun, masih terdapat beberapa masalah dalam penggunaan sistem tersebut, seperti kurangnya efektivitas dan efisiensi dalam pengaturan lalu lintas. Oleh karena itu, perlu dilakukan pengembangan sistem lalu lintas otomatis yang lebih canggih dan efektif untuk mengatasi masalah tersebut. (Rahmad et al., 2023). Dalam era teknologi dan

informasi saat ini, terdapat suatu basis sistem yang digunakan untuk mengontrol objek dari satu jaringan ke jaringan lainnya. Konsep *Internet of Things (IoT)* merupakan cara di mana objek-objek tersebut memiliki kemampuan untuk mentransfer data melalui jaringan tanpa campur tangan manusia, baik dari manusia ke manusia maupun manusia ke komputer. Sistem kontrol pada lampu lalu lintas berbasis *Internet of Things (IoT)* dapat diimplementasikan untuk mengatur lalu lintas secara otomatis. (Bayu Kusumo, 2023)

Berdasarkan permasalahan yang ada mengenai sistem kontrol kepadatan kendaraan pada persimpangan jalan, perlu adanya upaya untuk mengatasi permasalahan tersebut. Terdapat beberapa penelitian yang telah dilakukan sebelumnya untuk menyelesaikan permasalahan mengenai rancangan sistem lalu lintas otomatis, salah satunya penelitian yang dilakukan oleh (Hidayatullah et al., n.d.) yang berjudul “*Adaptive System Traffic Light Using Raspberry Pi 3*”. Pada penelitian tersebut membahas mengenai tipe kepadatan antrian kendaraan pada suatu lajur di persimpangan jalan yang dideteksi oleh sensor kamera. Tipe kepadatan pada suatu lajur akan menentukan lama pewaktuan lampu hijau pada lajur tersebut. Dengan metode *Adaptive* akan bergantung pada kondisi kepadatan yang dihitung dari jumlah kendaraan yang berada di jalur persimpangan. Pada saat kondisi normal durasi lampu khususnya lampu hijau akan berjalan secara normal yaitu selama 10 detik. Jika terdeteksi kepadatan di suatu jalur, maka sistem *adaptive* akan memberi durasi tambahan pada lampu hijau di jalur tersebut tergantung seberapa parah kepadatannya. (Fitria, 2021)

Berdasarkan latar belakang permasalahan yang ada, maka penulis tertarik untuk mengusulkan penelitian yang berjudul “*Perancangan Alat Perangkat Keras Pada Adaptive Traffic Light*”. Proses kontrol dilakukan dengan menggunakan mikrokontroler *ESP32* untuk mengatur waktu yang telah didapatkan dari data hasil pemrosesan *Jetson Nano* menggunakan kamera CCTV. Serta menggunakan mikroprosesor *raspberry pi 4* untuk menerima dan mengirim data dari kontrol *adaptive traffic Light* ke sistem monitoring *adaptive traffic Light* yang digunakan untuk menampilkan hasil data secara *web*. Penelitian ini diharapkan dapat membantu pihak dinas khususnya Dinas Perhubungan Kota Madiun untuk mempermudah dan mempercepat proses kontrol pada *adaptive traffic Light*.

1.2 Rumusan Masalah

Berdasarkan latar belakang tersebut, maka didapatkan rumusan masalah permasalahan sebagai berikut:

1. Bagaimana merancang dan membuat perangkat keras pada *Adaptive Traffic Light* berdasarkan kepadatan kendaraan pada persimpangan jalan?
2. Bagaimana perangkat keras pada *Adaptive Traffic Light* dapat mengirim dan menerima sinyal dari sistem pendeteksi dan penghitung kepadatan kendaraan?
3. Bagaimana perangkat keras pada *Adaptive Traffic Light* dapat mengirim data ke sistem monitoring *Adaptive Traffic Light*?

1.3 Tujuan

Berdasarkan identifikasi masalah yang telah dijabarkan, maka dapat dijabarkan tujuan dari pembuatan alat adalah sebagai berikut:

1. Dapat merancang dan membuat Perangkat Keras Pada *Adaptive Traffic Light* berdasarkan kepadatan kendaraan pada persimpangan jalan.
2. Dapat mengirim dan menerima sinyal dari sistem pendeteksi dan penghitung kepadatan kendaraan.
3. Dapat mengirim data ke sistem monitoring *Adaptive Traffic Light*.

1.4 Manfaat Tugas Akhir

Hasil akhir dari penelitian alat ini adalah diharapkan dapat membantu pihak dinas khususnya Dinas Perhubungan Kota Madiun untuk mempermudah dan mempercepat proses pendeteksian dan penghitungan kepadatan kendaraan yang dapat digunakan untuk menyesuaikan waktu lampu persimpangan lalu lintas. Sehingga pengendara yang melintasi persimpangan jalan tidak terburu-buru karena singkatnya nyala lampu persimpangan jalan dan menghindari terjadinya kecelakaan lalu lintas

1.5 Batasan Masalah

Untuk menghindari cakupan masalah yang meluas, maka pembahasan menitik beratkan pada:

1. Sistem ini hanya difokuskan pada sistem kontrol *Adaptive Traffic Light*.
2. Tidak membahas mendeteksi dan menghitung kepadatan kendaraan.
3. Tidak membahas monitoring jarak jauh *Adaptive Traffic Light*.

BAB II

TINJAUAN PUSTAKA

Dalam tinjauan pustaka menjelaskan mengenai penelitian terkait dan dasar teori dari “Perancangan Alat Perangkat Keras Pada *Adaptive Traffic Light*”. Serta tanggapan Bapak Achmad Lutfhi Prasetyo, A.Md., S.ST. selaku Kepala Subbagian Perencanaan dan Kepegawaian Dinas Perhubungan Kota Madiun mengenai sistem ini sebagai berikut, “Lampu lalu lintas merupakan hal penting yang perlu diperhatikan sebagai infrastruktur lalu lintas karena telah menjadi wewenang dan tanggung-jawab Dinas Perhubungan. Pada saat ini pengontrolan dan pemeriksaan lampu lalu lintas dilakukan setiap hari dengan datang langsung ke lapangan. Kami berharap kedepannya akan ada sebuah alat monitoring yang dapat mengontrol sekaligus memantau lampu lalu lintas hanya melalui kantor Dinas Perhubungan Kota Madiun.” Jum’at,(01/12/2023 09.30 WIB).



Gambar 2. 1
Foto Wawancara Dengan Dinas Perhubungan Kota Madiun

2.1 Penelitian Terdahulu

Beberapa penelitian, jurnal dan teori terdahulu yang digunakan sebagai referensi dalam pembuatan “Perancangan Alat Perangkat Keras Pada *Adaptive Traffic Light*” adalah sebagai berikut :

2.1.1 *Adaptive System Traffic Light Using Raspberry Pi 3* (Hidayatullahet.al.,n.d.)

Penelitian ini membahas tentang pengembangan sistem lampu lalu lintas *adaptif* dengan memanfaatkan *Raspberry Pi 3*. Penelitian ini bertujuan meningkatkan efisiensi pengaturan lampu lalu lintas dengan menggunakan teknologi canggih. *Raspberry Pi 3* digunakan sebagai otak utama sistem,

memproses data lalu lintas secara *real-time* dan membuat keputusan berdasarkan informasi yang diperoleh. Sistem ini *responsif* terhadap perubahan kondisi lalu lintas dan menggunakan sensor untuk mengumpulkan data. Penelitian ini juga membahas implementasi sensor, algoritma *adaptive*, serta hasil pengujian sistem, menyumbang pada pengembangan solusi untuk mengoptimalkan aliran lalu lintas dan meningkatkan efisiensi transportasi perkotaan.

2.1.2 *Internet of Things (IoT) Based Traffic Light Control System Using Raspberry Pi* (Basil & Sawant, 2017)

Penelitian ini membahas tentang penerapan sistem kontrol lampu lalu lintas berbasis *Internet of Things (IoT)* dengan memanfaatkan *Raspberry Pi*. Dalam penelitian ini, *Raspberry Pi* digunakan sebagai pusat kontrol yang terkoneksi internet, memungkinkan pengawasan dan pengendalian efisien terhadap lampu lalu lintas. Konsep *Internet of Things (IoT)* memungkinkan integrasi antara perangkat keras *Raspberry Pi* dan perangkat lunak untuk mengelola informasi dari berbagai sensor, seperti deteksi volume lalu lintas dan kondisi cuaca. Tujuan utama penelitian ini adalah meningkatkan efisiensi lampu lalu lintas, mengurangi kemacetan, dan meningkatkan pengalaman berlalu lintas secara keseluruhan. Dengan menggunakan *Raspberry Pi* dan *Internet of Things (IoT)*, penelitian ini memberikan kontribusi pada pengembangan sistem transportasi yang lebih cerdas dan efisien.

2.1.3 Rancang Bangun Sistem Kontrol dan Alat Monitoring *Traffic Light* Menggunakan *ESP8266* Berbasis *Internet of Things (IoT)* (Bayu Kusumo, 2023)

Penelitian ini membahas tentang pengembangan sistem kontrol dan pemantauan *traffic Light* menggunakan *ESP8266* berbasis *Internet of Things (IoT)*. Tujuan utamanya adalah meningkatkan efisiensi dan keamanan lalu lintas dengan menerapkan teknologi *Internet of Things (IoT)* pada sistem *traffic Light*. *ESP8266*, sebagai mikrokontroler nirkabel, memungkinkan integrasi yang mulus antara kontrol dan pemantauan. Sistem ini menawarkan pengaturan waktu yang fleksibel dan penyesuaian *real-time* terhadap kondisi lalu lintas. Kelebihan utama melibatkan adaptabilitas terhadap perubahan dalam lalu lintas dan kemampuan analisis kinerja jaringan transportasi. Meskipun demikian, penelitian ini menyoroti tantangan

teknis dan keamanan yang perlu diatasi, termasuk aspek keamanan jaringan dan ketahanan terhadap serangan siber. Secara keseluruhan, jurnal ini memberikan kontribusi positif untuk pengembangan *traffic Light* yang cerdas dan terhubung di lingkungan perkotaan.

2.1.4 Rancang Bangun Lampu Lalu Lintas Otomatis Berdasarkan Panjang Antrian Kendaraan Berbasis Pengolahan Citra Digital (Utama & Aulia, 2017)

Penelitian ini membahas tentang pengembangan sistem lampu lalu lintas otomatis dengan menggunakan teknologi pengolahan citra digital untuk merespons panjang antrian kendaraan. Melalui metode ini, sistem dapat mengidentifikasi dan mengukur panjang antrian dengan akurat, memungkinkan penyesuaian dinamis durasi lampu lalu lintas untuk mengoptimalkan aliran kendaraan. Pendekatan ini diharapkan dapat mengurangi kemacetan dan meningkatkan efisiensi lalu lintas perkotaan. Selain itu, potensi integrasi dengan teknologi transportasi pintar dan kendaraan otonom juga diperkenalkan, menciptakan peluang untuk kota pintar yang lebih berkelanjutan. Meskipun perlu uji coba lebih lanjut.

2.1.5 Rancang Bangun Sistem Kontrol Lampu Lalu Lintas Berbasis *Artificial Intelligence (AI)* (AJI, 2021)

Penelitian ini membahas tentang implementasi teknologi kecerdasan buatan dalam pengaturan lampu lalu lintas. Penelitian ini bertujuan untuk meningkatkan efisiensi dan keamanan lalu lintas melalui penerapan algoritma kecerdasan buatan yang mampu mengoptimalkan pengaturan lampu lalu lintas secara *adaptif*. Metode yang digunakan mencakup pengumpulan data lalu lintas, analisis pola pergerakan kendaraan, dan pengembangan algoritma kontrol berbasis kecerdasan buatan, seperti *machine learning* atau *neural networks*. Penelitian ini diharapkan dapat memberikan solusi inovatif untuk mengurangi kemacetan, memperbaiki aliran lalu lintas, dan mengoptimalkan penggunaan energi melalui penyesuaian dinamis durasi lampu lalu lintas berdasarkan kondisi lalu lintas aktual. Selain itu, jurnal ini juga membahas hasil eksperimen dan evaluasi kinerja sistem, serta potensi aplikasi luas dari implementasi teknologi ini dalam meningkatkan efisiensi sistem transportasi perkotaan secara keseluruhan. Dengan demikian, penelitian ini memberikan kontribusi signifikan dalam menghadirkan solusi cerdas untuk meningkatkan

kualitas lalu lintas perkotaan melalui pemanfaatan kecerdasan buatan dalam sistem kontrol lampu lalu lintas.

2.2 Tinjauan Teoritis

Berikut adalah beberapa tinjauan teoritis yang digunakan dalam pembuatan “Perancangan Alat Perangkat Keras Pada *Adaptive Traffic Light*”.

2.2.1 Pengertian Lalu Lintas

Lalu lintas adalah sebagai gerak kendaraan dan orang di ruang lalu lintas jalan, sebagai prasarana yang diperuntukkan bagi gerak pindah kendaraan, orang, dan/atau barang yang berupa jalan dengan fasilitas pendukungnya. (UU No. 22/2009, Pasal 1). Sedangkan yang dimaksud dengan Ruang Lalu Lintas Jalan adalah prasarana yang diperuntukkan bagi gerak pindah Kendaraan, orang, dan/atau barang yang berupa Jalan dan fasilitas pendukung lalu lintas ialah setiap hal yang berhubungan dengan sarana jalan umum sebagai sarana utama untuk tujuan yang ingin dicapai. Lalu lintas juga dapat diartikan sebagai hubungan antara manusia dengan atau tanpa disertai alat penggerak dari suatu tempat ke tempat lain dengan menggunakan jalan sebagai ruang geraknya. (Engel, 2019)

2.2.2 Pengertian Lampu Lalu Lintas

Lampu lalu lintas adalah lampu yang mengendalikan arus lalu lintas yang terpasang di persimpangan jalan dan tempat arus lalu lintas lainnya. Lampu ini yang menandakan kendaraan harus berjalan dan berhenti secara bergantian dari berbagai arah. Pengaturan lalu lintas di persimpangan jalan dimaksudkan untuk mengatur pergerakan kendaraan pada masing-masing kelompok pergerakan kendaraan agar dapat bergerak secara bergantian, sehingga tidak saling mengganggu arus yang berlawanan.

Lampu lalu lintas merupakan salah satu bagian penting dari arus lalu lintas yang menyebabkan tingginya penumpukan kendaraan. Apabila tidak dikelola dengan tepat, penumpukan kendaraan akan menjadi masalah utama yang harus dihadapi, terutama di persimpangan lampu lalu lintas. Masalah ini dapat diatasi dengan menyediakan sistem kendali lampu lalu lintas yang efisien pada persimpangan untuk pergerakan kendaraan yang melintasi persimpangan. (Veda, 2021)



Gambar 2. 2
Lampu Lalu Lintas

(https://1.bp.blogspot.com/_Aw7I3wlo8/VyBIsSZDnxI/AAAAAAAAUNA/I7VdeqcAgGUflberxcqEOp2WSWYvcVhrwCLcB/s1600/FOT-201211-004604.jpg)

2.2.3 Metode *Adaptive*

Sistem kendali *adaptive* merupakan sistem kendali yang mempunyai parameter-parameter kendali yang dapat beradaptasi. Parameter-parameter kendali tersebut beradaptasi terhadap perubahan kondisi lingkungan di sekitarnya, seperti adanya gangguan, serta perubahan karakter *internal* dari sistem yang dikendalikan. Penggunaan sistem kendali *adaptive* menunjukkan peningkatan kinerja sistem karena suatu sistem umumnya berada dalam situasi yang mengandung derau dan gangguan serta kondisi *internal* dan eksternalnya mengandung ketidakpastian. Sistem kendali *adaptive* telah banyak diaplikasikan dalam berbagai bidang industri pengolahan bahan kimia, sistem penerbangan, serta sistem otomotif.

Metode kendali *adaptive* adalah teknik kendali yang dilengkapi dengan algoritma pembelajaran. metode kendali *adaptive* didefinisikan juga sebagai sistem kendali dimana parameter-parameternya dapat diatur dan juga memiliki mekanisme untuk mengatur parameter-parameter tersebut. Pada penelitian ini, metode *adaptive* akan bergantung pada kondisi kepadatan yang dihitung dari jumlah kendaraan yang berada di jalur persimpangan. Pada saat kondisi normal durasi lampu khususnya lampu hijau akan berjalan secara normal yaitu selama 10 detik. Jika terdeteksi kepadatan di suatu jalur, maka sistem *adaptive* akan memberi durasi tambahan pada lampu hijau di jalur tersebut tergantung seberapa parah kepadatannya. Apabila keadaan padat 1 terpenuhi, akan ada penambahan durasi pada lampu hijau selama 2 detik. Jika Keadaan padat 2 terpenuhi, maka akan ada penambahan durasi pada

lampu hijau selama 4 detik. Dan apabila keadaan padat 3 terpenuhi, maka akan ada penambahan durasi lampu hijau selama 6 detik. (Fitria, 2021)

2.2.4 *Raspberry Pi*

Raspberry Pi adalah modul mikrokomputer dengan ukuran kartu kredit dan *port input/output* digital seperti pada mikrokontroler. *Raspberry Pi* dapat menjalankan berbagai program, termasuk perkantoran, permainan komputer, dan pemutar media hingga video beresolusi tinggi. Kelebihan *Raspberry pi* melibatkan kemampuan untuk terhubung dengan TV atau monitor PC melalui *port* koneksi, serta memiliki *port* USB untuk *keyboard* dan *mouse*. Tujuan utama *Raspberry Pi* adalah menyediakan perangkat murah untuk meningkatkan kemampuan pemrograman dan pemahaman *hardware*, terutama di tingkat pra-universitas. Dengan ukuran kecil dan harga terjangkau, *Raspberry Pi* cepat diadopsi oleh penggemar elektronik untuk berbagai proyek yang membutuhkan lebih dari mikrokontroler dasar.



Gambar 2. 3

Raspberry Pi

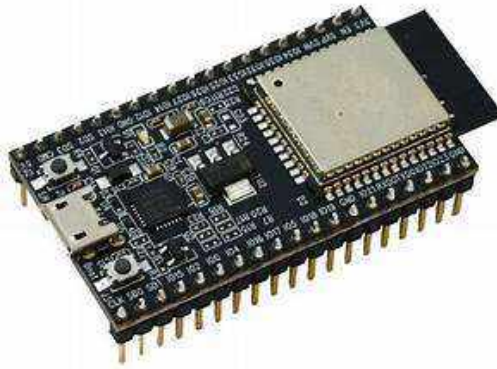
(<https://www.retro-kit.co.uk/user/custom/RaspPi/images/RaspPi-Hi-Res.jpg>)

Raspberry Pi bersifat *open hardware*, kecuali *chip* utama Broadcomm SoC, yang menggabungkan CPU, grafis, memori, USB *controller*, dan lainnya. Banyak proyek menggunakan *Raspberry Pi*, dengan dokumentasi yang baik untuk memudahkan pengguna membuat dan memodifikasi proyek mereka. Ada lima model *Raspberry Pi*, mulai dari A, A+, B, B+, hingga *Raspberry Pi Zero*, dengan perkembangan dari *Raspberry Pi* 1 hingga *Raspberry Pi* 3 dan *Raspberry Pi Zero*. *Raspberry Pi* didesain untuk sistem operasi *Linux*, tetapi sekarang juga mendukung

sistem operasi *Windows* dalam bentuk *Internet of Things (IoT)*. Sistem operasi populer untuk *Raspberry Pi* termasuk *Raspbian* dan *Pidora*. (Lesmana et al., 2019)

2.2.5 ESP 32

ESP32 dibuat oleh *Espressif Systems*, *ESP32* adalah sistem dengan biaya yang rendah, berdaya rendah pada seri *chip* dengan Wi-Fi & kemampuan *Bluetooth* dua mode. Keluarga *ESP32* termasuk *chip ESP32-D0WDQ6* (dan *ESP32-D0WD*), *ESP32-D2WD*, *ESP32-S0WD*, dan sistem dalam paket (SiP) *ESP32-PICO-D4*. Pada intinya, ada mikroprosesor *Tensilica Xtensa LX6 dualcore* atau *single-core* dengan *clock rate* hingga 240 MHz. *ESP32* sudah terintegrasi dengan *built-in antenna switches*, *RF balun*, *power amplifier*, *lownoise receive amplifier*, *filters*, and *power management modules*.



Gambar 2. 4
ESP32

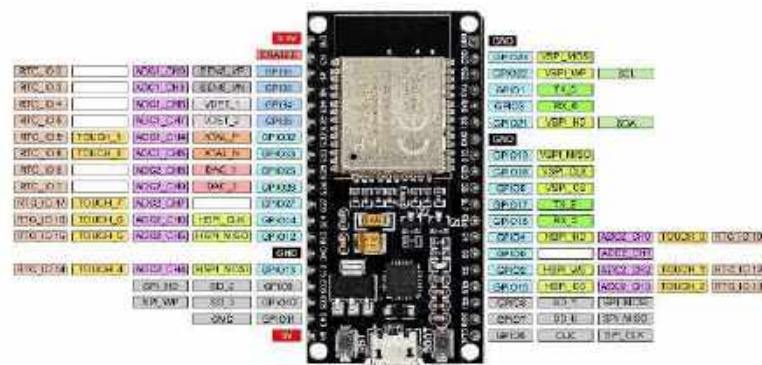
(<https://cdn.sos.sk/productdata/90/d5/9dcaac3b/esp32-devkitc.jpg>)

Didesain untuk perangkat seluler, perangkat elektronik yang dapat dipakai, dan aplikasi *Internet of Things (IoT)*, *ESP32* juga bekerja dengan konsumsi daya sangat rendah melalui fitur hemat daya termasuk *fine resolution clock gating*, *multiple power modes*, and *dynamic power scaling*. *Module ESP32* merupakan penerus dari *module ESP8266* yang cukup populer untuk Aplikasi *Internet of Things (IoT)*. Pada *ESP32* terdapat inti CPU serta Wi-Fi yang lebih cepat, GPIO yang lebih, dan mendukung *Bluetooth Low Energy*. (NUGROHO, 2021)

- *Microprosesor Xtensa Dual-Core 32 Bit LX6*
- *Freq Clock up to 240 MHz*
- *SRAM 520 kB • Flash memori 4 MB*

- 11b/g/n WiFi transceiver
- Bluetooth 4.2/BLE
- 48 pin GPIO
- 15 pin channel *Analog to Digital Converter (ADC)*
- 25 pin *Pulse Width Modulation (PWM)*
- 2 pin channel *Digital to Analog Converter (DAC)*

2.2.5.1 Konfigurasi Pin



Gambar 2. 5
Konfigurasi ESP32

(https://wolles-elektronikkiste.de/wp-content/uploads/2021/06/ESP32_dev_board_pinout.jpg)

2.2.6 Relay Module 8 Channel

Modul ini merupakan modul *relay* yang dirancang untuk *switching* satu perangkat memiliki *power* tinggi dari *Arduino*. Cara kerja *relay* yaitu dapat memutus dan menyambung aliran listrik dalam rangkaian. *Relay* menggunakan prinsip *elektromagnetik* untuk menggerakkan kontak saklar, sehingga dengan arus listrik yang kecil (*low power*) dapat menghantarkan listrik yang bertegangan lebih tinggi. Sebagai contoh, dengan *relay* yang menggunakan *elektromagnet 5V dan 50 mA* mampu menggerakkan *armature relay* (yang berfungsi sebagai saklarnya) untuk menghantarkan listrik 220V 2A. Kendali *ON / OFF switch (relay)*, sepenuhnya

ditentukan oleh nilai *output* sensor, yang setelah diproses mikrokontroler akan menghasilkan perintah kepada *relay* untuk melakukan fungsi *ON / OFF*.



Gambar 2. 6
Relay Module 8 Channel

2.2.7 Miniature Circuit Breaker (MCB)

Miniature Circuit Breaker (MCB) merupakan salah satu komponen penting dalam instalasi listrik rumah maupun perkantoran. Fungsi dari komponen ini adalah sebagai sistem proteksi apabila terjadi beban berlebih ataupun hubung singkat arus listrik (*short circuit*). Jika terjadi kegagalan fungsi *Miniature Circuit Breaker (MCB)*, akan berpotensi menimbulkan panas pada kabel penghantar dan percikan api dan bisa menyebabkan kebakaran. Pada instalasi listrik rumah, *Miniature Circuit Breaker (MCB)* dipasang pada *kWh* meter listrik PLN dan juga dipasang pada kotak *Miniature Circuit Breaker (MCB)* yang berada di dalam rumah. Jika di rumah terjadi *trip* disebabkan beban lebih atau hubung singkat, maka *Miniature Circuit Breaker (MCB)* akan panas kemudian *off* (saklar *MCB* turun). Untuk menyalakan listrik kembali *MCB* harus di-on-kan (saklar *Miniature Circuit Breaker (MCB)* dinaikkan) secara manual dan biasanya menunggu beberapa saat supaya *Miniature Circuit Breaker (MCB)* dingin. Terdapat tiga macam jenis *Miniature Circuit Breaker (MCB)* jika dilihat dari jenis fasanya, yaitu *Miniature Circuit Breaker (MCB)* 1 fasa, *Miniature Circuit Breaker (MCB)* 2 fasa, dan *Miniature Circuit Breaker (MCB)* 3 fasa seperti ditunjukkan dalam *Miniature Circuit Breaker (MCB)* 1 fasa mempunyai satu kutub listrik, *Miniature Circuit Breaker (MCB)* 2 fasa mempunyai dua kutub listrik, dan *Miniature Circuit Breaker (MCB)* 3 fasa mempunyai tiga kutub listrik. Pemakaian *Miniature Circuit Breaker (MCB)* 1 fasa, 2 fasa, dan 3 fasa tergantung dari daya listrik yang digunakan di dalam rumah/gedung.



Gambar 2. 7
Miniature Circuit Breaker (MCB)
 (https://imagesna.sslimagesamazon.com/images/I/51A73MkhJQL._AC_SL1145_.jpg)

Sistem kerja *Miniature Circuit Breaker (MCB)* pada umumnya menggunakan prinsip *thermal tripping* dan magnetik *tripping*. *Thermal tripping* adalah pemutusan arus listrik pada *Miniature Circuit Breaker (MCB)* dikarenakan suhu tinggi. Di dalam *Miniature Circuit Breaker (MCB)* terdapat *bimetal* yang bekerja dipengaruhi temperatur. Jika arus yang mengalir pada *bimetal* besar, maka *bimetal* akan panas dan melengkung, sehingga *trip* yang menghubungkan kutub *Miniature Circuit Breaker (MCB)* akan putus, dengan demikian arus listrik tidak akan mengalir. Magnetik *tripping* adalah pemutusan arus listrik pada *Miniature Circuit Breaker (MCB)* dikarenakan adanya medan magnet. Jika arus listrik yang mengalir besar dan melebihi beban kapasitas *Miniature Circuit Breaker (MCB)*, maka timbul medan magnet di pangkal *Miniature Circuit Breaker (MCB)* dan membuatnya tertarik. Dengan demikian *trip* yang menghubungkan kutub *Miniature Circuit Breaker (MCB)* akan putus dan arus tidak akan mengalir.

2.2.8 Sensor Tegangan DC



Gambar 2. 8
 Sensor tegangan DC

Sensor tegangan *DC* ini digunakan untuk memonitor, menghitung dan menentukan suplai tegangan. Sensor ini dapat menentukan level tegangan *DC*. Masukan dari sensor ini dapat berupa tegangan sedangkan keluarannya berupa sakelar, sinyal tegangan analog, sinyal arus, sinyal suara, dll. Beberapa sensor memberikan bentuk gelombang sinus atau gelombang pulsa seperti keluaran & lainnya dapat menghasilkan keluaran seperti *Amplitude Modulation (AM)*, *Pulse Width Modulation (PWM)* atau *Frequency Modulation (FM)*. Pengukuran sensor ini dapat bergantung pada pembagi tegangan Sensor ini mencakup *input* dan *output*. Sisi *input* terutama mencakup dua pin yaitu pin positif dan negatif. Kedua pin perangkat dapat dihubungkan ke pin positif & negatif sensor. Pin positif & negatif perangkat dapat dihubungkan ke pin positif & negatif sensor. *Output* dari sensor ini terutama mencakup tegangan suplai (*Vcc*), *ground ground (GND)*, data *output* daya analog.

2.2.9 Switch Power Over Ethernet (PoE)



Gambar 2. 9
Switch Power over Ethernet (PoE)

Power over Ethernet (PoE) merupakan perangkat yang digunakan untuk memberikan pasokan tegangan pada perangkat jaringan internet dengan menggunakan kabel jaringan, kabel *Local Area Network (LAN)*, atau kabel *ethernet*. Dimana listrik yang selanjutnya dialirkan dapat memanfaatkan kabel *Unshielded Twisted Pair (UTP) Local Area Network (LAN)* sebagai media transmisi daya. *Power over Ethernet (PoE)* memungkinkan sebuah kabel *Unshielded Twisted Pair (UTP)* menghantarkan data sekaligus memberi pasokan daya listrik untuk perangkat jaringan yang sudah mendukung *Power over Ethernet (PoE)*, sehingga perangkat tersebut tidak membutuhkan tersedianya colokan listrik di dekatnya. Ada beberapa teknik umum untuk mentransmisikan daya melalui kabel *Ethernet*.

2.2.10 Router



Gambar 2. 10
Router

Router merupakan sebuah perangkat yang berfungsi untuk mentransmisikan paket data dari jaringan internet ke perangkat lain melalui proses *routing*. *Router* berfungsi untuk mengelola lalu lintas antar jaringan dengan meneruskan paket data ke alamat IP yang dituju. Nantinya, *Router* memungkinkan beberapa perangkat untuk menggunakan koneksi internet yang sama. Selain mengelola lalu lintas antar jaringan dan membagikan koneksi internet ke beberapa perangkat lain.

2.3 Kerangka Pemikiran

Pada pembuatan tugas akhir diperlukan suatu kerangka pemikiran sebagai alur penelitian dan dasar pemikiran yang jelas supaya dapat memperkuat latar belakang dari penelitian ini. Berikut kerangka pemikiran yang digunakan pada pembuatan “Perancangan Alat Perangkat Keras Pada *Adaptive Traffic Light*”.

2.3.1 Mengapa penelitian dilakukan:

1. Perangkat keras lampu lalu lintas saat ini belum dilengkapi dengan alat untuk sistem *adaptive*.
2. Pengontrolan *timer* lampu lalu lintas masih dilakukan secara manual melalui panel di titik lokasi.

2.3.2 Bagaimana proses penelitian dilakukan:

1. Observasi dan studi literatur.
2. Analisis kebutuhan alat dan bahan yang diperlukan.
3. Pembuatan perangkat keras.

4. Perancangan perangkat keras.
5. Pengujian alat.
6. Analisis, evaluasi, dan menarik kesimpulan hasil pengujian.
7. Membuat laporan.

2.3.3 Apa yang akan diperoleh dari penelitian tersebut:

1. Mengetahui cara pembuatan Perancangan Alat Perangkat Keras pada *Adaptive Traffic Light*
2. Membantu dinas terkait dalam perangkat keras lampu lalu lintas pada persimpangan jalan

2.3.4 Untuk apa hasil atau luaran penelitian tersebut:

Hasil atau *output* dari penelitian ini berupa Perancangan Alat Perangkat Keras pada *Adaptive Traffic Light* yang dapat dimanfaatkan oleh pihak Dinas Perhubungan khususnya Dinas Perhubungan Kota Madiun.

BAB III

METODE RANCANGAN DAN PENELITIAN

Metode penelitian atau perancangan merupakan metode yang digunakan mengerjakan tugas akhir dengan judul “Perancangan Alat Perangkat Keras Pada *Adaptive Traffic Light*” guna mencapai tujuan yang diinginkan penulis. Berikut paparan metode serta rancangan penelitian yang digunakan dalam pembuatan “Perancangan Alat Perangkat Keras Pada *Adaptive Traffic Light*” sebagai berikut:

3.1 Metode

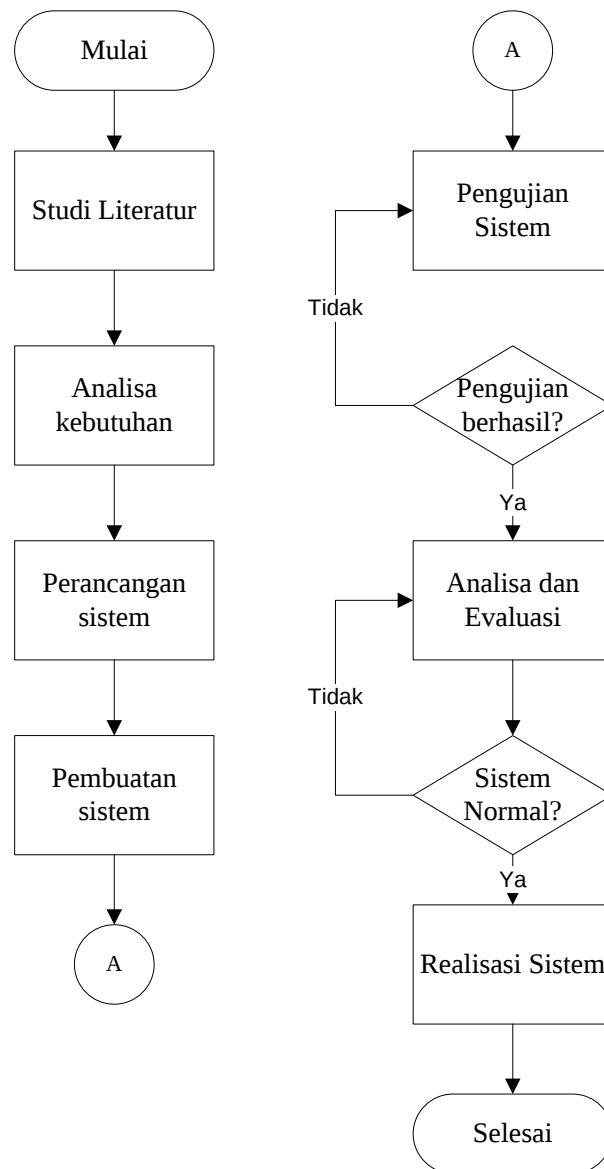
Metode yang digunakan dalam pembuatan “Perancangan Alat Perangkat Keras Pada *Adaptive Traffic Light*” adalah metode Eksperimen Perancangan. Penelitian eksperimen adalah suatu cara untuk mencari hubungan sebab akibat (hubungan *kausal*) antara dua faktor yang sengaja ditimbulkan oleh peneliti dengan mengeliminasi atau mengurangi atau menyisihkan faktor-faktor lain yang mengganggu (Thabroni, 2021). Terkadang penelitian ini juga disebut dengan metode penelitian eksperimental. Penelitian eksperimental merupakan penelitian yang bersifat *prediktif*, yaitu meramalkan akibat dari suatu manipulasi terhadap variabel terikatnya. Artinya penelitian ini biasa dilakukan dengan asumsi dasar atau hipotesis yang telah ditentukan sebelumnya, untuk kemudian dibuktikan kebenarannya melalui tindakan atau kondisi yang terkendalikan (Latipun, 2015)

3.2 Lokasi Penelitian

Tempat pelaksanaan penelitian Tugas Akhir dilakukan di Laboratorium Teknik Komputer Kontrol, dengan waktu pelaksanaan Penyusunan Tugas Akhir dilaksanakan dalam beberapa tahapan yang akan dijelaskan pada Tahapan Penelitian. Kegiatan wawancara dan mencari data-data tentang kebutuhan perangkat keras lampu lalu lintas dilaksanakan di lingkungan Dinas Perhubungan Kota Madiun.

3.3 Tahap Penelitian

Berikut ini adalah *diagram* alir tahapan penelitian perancangan “Perancangan Alat Perangkat Keras Pada *Adaptive Traffic Light*”. Terdapat pada **Error! Reference source not found.**



Gambar 3. 1
Tahap Penelitian

3.3.1 Studi Literatur

Tahap studi literatur ini bertujuan untuk mengumpulkan data dari beberapa sumber seperti jurnal, *website* internet, artikel, laporan tugas akhir terdahulu dan melakukan observasi langsung ke Dinas Perhubungan Kota Madiun.

3.3.2 Analisa Kebutuhan

Tahap Analisa kebutuhan ini bertujuan untuk menganalisa kebutuhan apa saja yang diperlukan sebelum merancang sistem. Analisa kebutuhan dilakukan dengan

wawancara ke pihak yang terkait dengan penelitian penulis yaitu Dinas Perhubungan Kota Madiun.

3.3.3 Perancangan Sistem

Tahap perancangan sistem ini bertujuan untuk merancang konsep dan kerangka sistem sebelum melakukan pembuatan perangkat keras lampu lalu lintas.

3.3.4 Pembuatan Sistem

Tahap pembuatan sistem ini bertujuan untuk membuat program untuk menerima hasil perhitungan kepadatan kendaraan dari sistem pendeteksi kendaraan serta menerima dan mengirim hasil data untuk sistem monitoring. Hal ini bertujuan agar sistem dapat mengolah data dan bekerja sesuai dengan konsep yang direncanakan.

3.3.5 Pengujian Sistem

Tahap pengujian sistem ini bertujuan agar dapat menguji ketepatan sistem dalam menerima, mengirim, mengontrol lampu lalu lintas sesuai dengan kepadatan kendaraan serta mengetahui perhitungan error pada sistem.

3.3.6 Analisis dan Evaluasi

Tahap ini dilakukan untuk menganalisis dan mengevaluasi perangkat keras apakah sudah sesuai dengan cara kerja sistem dengan baik atau belum, selain untuk analisis dan evaluasi tahap ini digunakan untuk mengetahui komponen yang dipakai sudah sesuai atau belum dan mengetahui kendala atau error terhadap perangkat keras yang telah dibuat.

3.4 Rancangan Penelitian

Berikut ini merupakan rancangan penelitian dalam pembuatan “Perancangan Alat Perangkat Keras Pada *Adaptive Traffic Light*”.

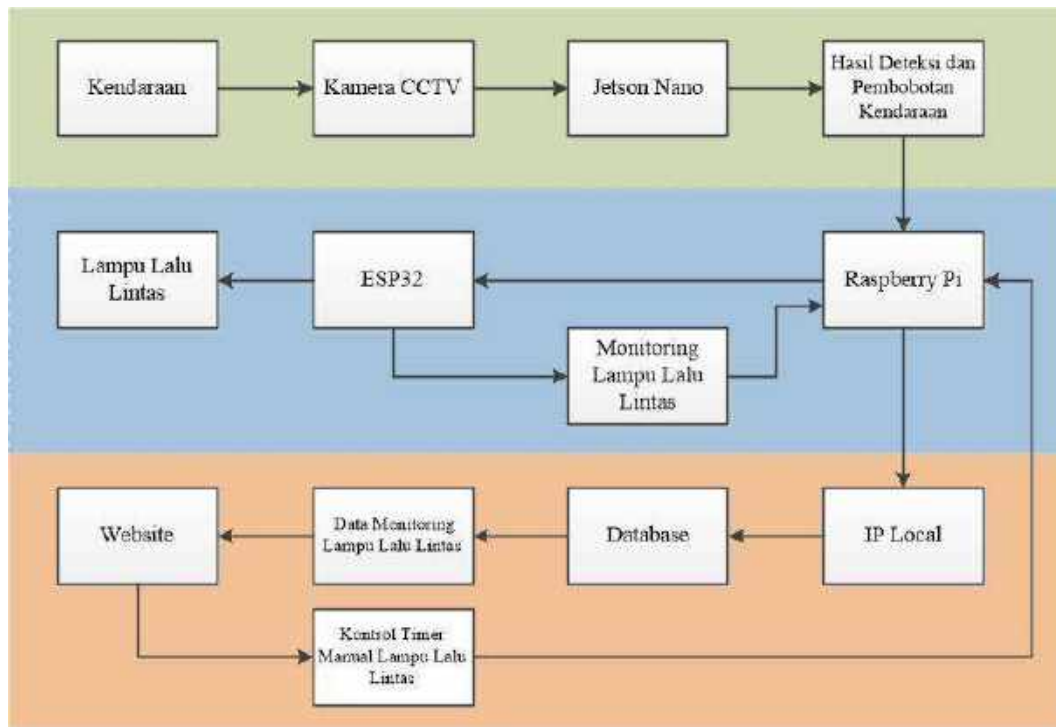
3.4.1 Spesifikasi Perancangan

Berikut ini merupakan tabel spesifikasi perancangan sistem Tugas Akhir dengan judul “Perancangan Alat Perangkat Keras Pada *Adaptive Traffic Light*”. Terdapat pada Tabel 3. 1

Tabel 3. 1
Alat dan Bahan

No	Alat dan Bahan	Keterangan
1	<i>Raspberry Pi</i>	Sebagai <i>Broker</i> menerima dan mengirim hasil perhitungan waktu serta <i>monitoring web</i>
2	<i>NVIDIA Jetson nano</i>	Alat untuk pemrosesan hasil kepadatan kendaraan
3	<i>Node MCU ESP32 (a)</i>	Perangkat pada sistem yang digunakan untuk kontroler
4	<i>Node MCU ESP32 (b)</i>	Perangkat pada sistem yang digunakan sebagai <i>output</i> dari sensor tegangan
5	<i>Relay Module 8 Channel</i>	Alat untuk memberikan <i>trigger</i> ke lampu lalu lintas
6	<i>Miniature Circuit Breaker (MCB) 4A</i>	Alat untuk menyalakan lampu serta meminimalisir terjadinya konsleting pada lampu
7	<i>Miniature Circuit Breaker (MCB) 10A</i>	Digunakan untuk menyalakan seluruh alat dalam <i>box panel</i>
8	<i>Router</i>	Alat untuk memberikan jaringan lokal pada sistem
9	<i>Adaptor Micro 5v</i>	Alat untuk memberikan sumber pada <i>raspberry pi</i>
10	Sensor Tegangan <i>5VDC/220VAC</i>	Alat untuk mengukur tegangan keluaran lampu lalu lintas
11	<i>Switch Power over Ethernet (PoE)</i>	Alat untuk menyediakan daya ke perangkat yang terhubung melalui kabel Ethernet.
12	<i>Power Suplay 5V 10A</i>	Alat untuk memberikan sumber daya 5v pada sistem
13	<i>CCTV HKVISION</i>	Alat untuk mendeteksi kemacetan pada persimpangan jalan
14	<i>Box Panel 60x80x20</i>	Digunakan untuk menempatkan seluruh komponen

3.4.2 Diagram Blok Sistem



Gambar 3. 2
Diagram Blok Sistem

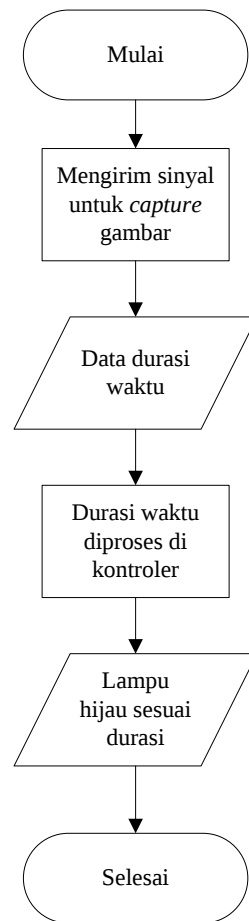
Dari diagram blok sistem pada Gambar 3.2, penulis mengerjakan blok berwarna biru, yaitu “Perancangan Alat Perangkat Keras Pada *Adaptive Traffic Light*”. Berikut penjelasan dari bagian yang dikerjakan oleh penulis:

1. *Raspberry Pi*, berfungsi sebagai perantara antara hasil data pembobotan kendaraan dari Jetson ke *ESP32*, memfasilitasi *transfer* data kontrol lampu lalu lintas dari *ESP32* ke *monitoring web*.
2. *ESP32*, berfungsi sebagai kontrol lampu lalu lintas yang telah menerima hasil data pembobotan kendaraan yang diolah menggunakan *jetson nano*. *ESP32* juga mengirim hasil kontrol melalui *Raspberry Pi* yang selanjutnya akan digunakan di *monitoring web*.
3. Data hasil kontrol yang telah diterima dari *jetson nano* ke *ESP32* diteruskan ke komponen *relay module* untuk menjalankan lampu lalu lintas sesuai program.

3.4.3 FlowChart

Pada bagian ini, akan dijelaskan *FlowChart* dalam pembuatan tugas akhir dengan judul “Perancangan Alat Perangkat Keras Pada *Adaptive Traffic Light*”.

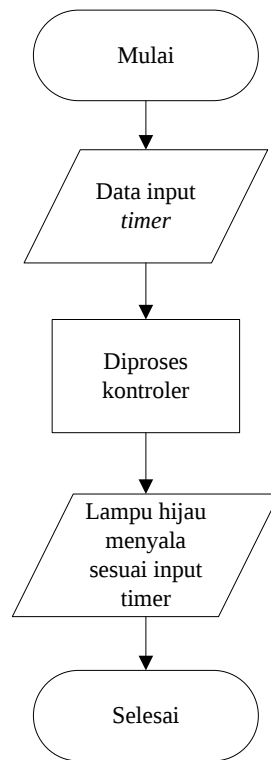
3.4.3.1 *FlowChart* Kontrol Otomatis



Gambar 3. 3
FlowChart Kontrol Otomatis

Pada Gambar 3.3 merupakan *FlowChart* kontrol otomatis, perangkat keras mengirim sinyal untuk *capture* gambar yang kemudian menghasilkan durasi waktu. Perangkat keras menerima *input* dari data durasi waktu yang kemudian diproses pada kontroler, setelah data diproses pada kontroler menghasilkan *output* berupa lampu hijau menyala sesuai durasi waktu.

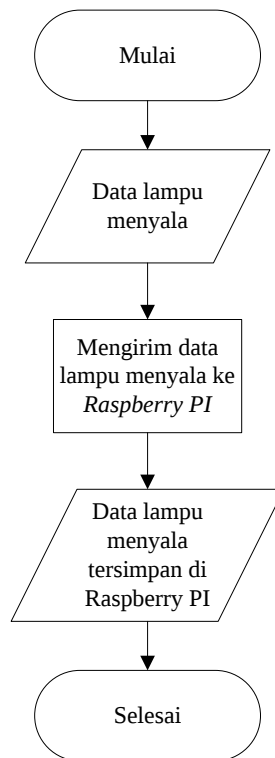
3.4.3.2 *FlowChart* Kontrol Manual



Gambar 3. 4
FlowChart Kontrol Manual

Pada Gambar 3.4 merupakan *FlowChart* kontrol manual, perangkat menerima data *input timer* dari sistem *monitoring web* yang kemudian *input timer* diproses pada kontroler, sehingga menghasilkan *output* berupa lampu hijau menyala sesuai *input timer*.

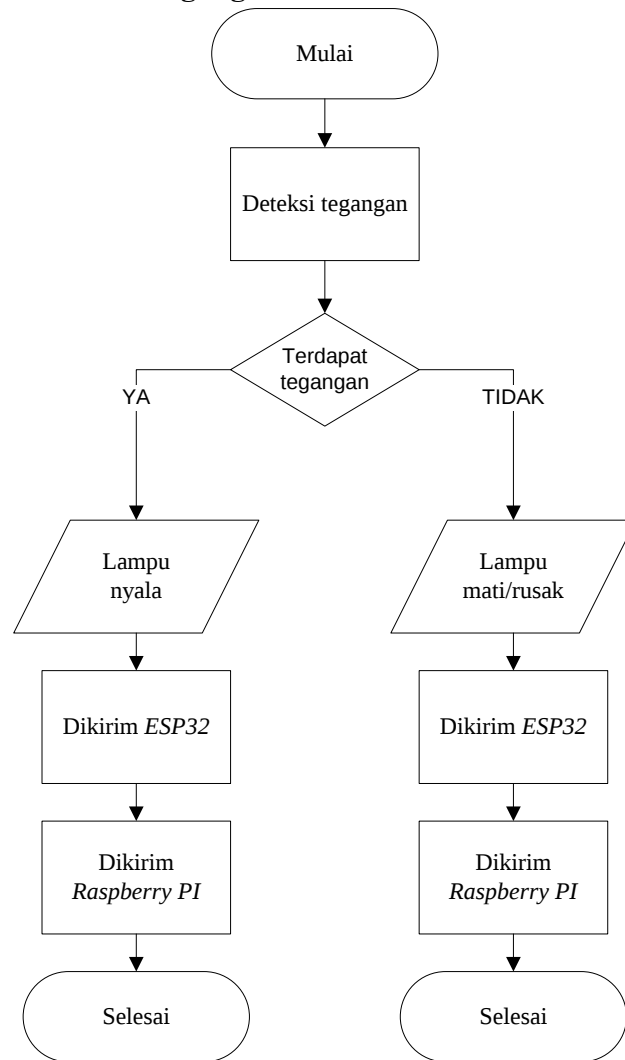
3.4.3.3 FlowChart Monitoring Web



Gambar 3. 5
FlowChart Monitoring Web

Pada Gambar 3.5 merupakan *FlowChart monitoring web*, dimana perangkat menghasilkan status data lampu menyala yang kemudian data nyala lampu dikirim ke *Raspberry Pi*. Data lampu menyala tersimpan di *Raspberry Pi*.

3.4.3.4 FlowChart Sensor Tegangan



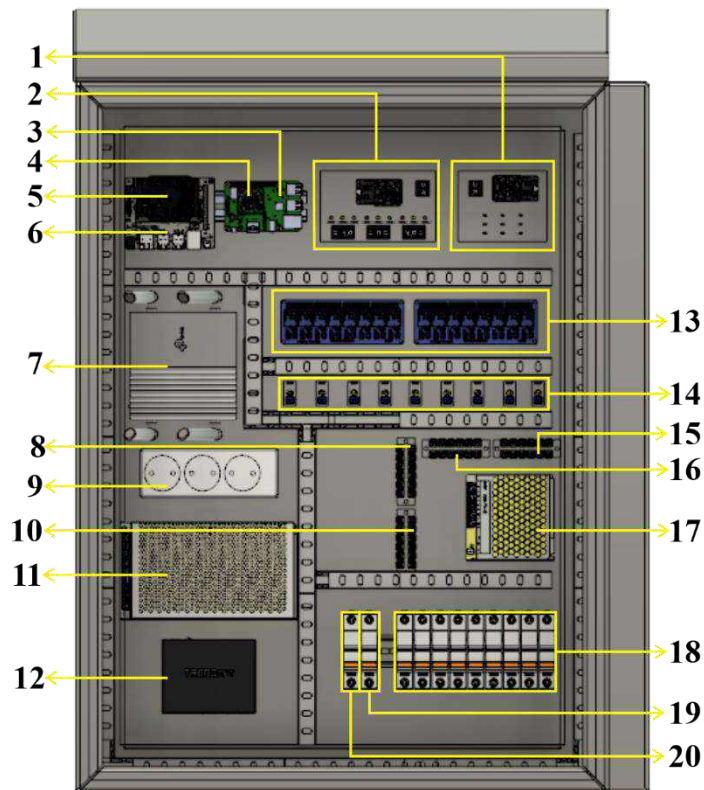
Gambar 3. 6
FlowChart Sensor Tegangan

Pada Gambar 3.6 merupakan *FlowChart* sensor tegangan, dimana sensor mendeteksi tegangan akan memberikan data. Jika menerima tegangan lampu akan menyala lalu akan mengirim data ke *ESP32* selanjutnya akan dikirim dan diproses pada *Raspberry Pi*. Jika tidak menerima tegangan lampu akan mati dan mendeteksi adanya kerusakan pada lampu lalu mengirim data ke *ESP32* selanjutnya akan dikirim dan diproses pada *Raspberry Pi*.

3.5 Perancangan Mekanika

Berikut ini merupakan perancangan mekanika dari “Perancangan Alat Perangkat Keras Pada *Adaptive Traffic Light*”. Terdapat beberapa sub-bab yang akan dibahas yaitu desain alat,

3.5.1 Desain *Box Panel*



Gambar 3. 7
Desain *Box Panel*

Pada Gambar 3.7 merupakan *box panel* pada perangkat keras *adaptive traffic Light*, dengan ukuran 60x80x100 cm. berikut penjelasannya :

1. *PCB Sensor Tegangan DC*, terdiri dari beberapa komponen yaitu : *ESP32*, *Fan dc 5v*, Konektor *JST*, Konektor sumber. Dari komponen tersebut saling terhubung, sehingga menghasilkan sebuah *PCB*. Berfungsi sebagai mengukur hasil *output* dari sensor tegangan dan akan diteruskan ke *monitoring web* sebagai *realtime* indikator lampu lalu lintas.
2. *PCB Control*, terdiri dari beberapa komponen yaitu : *ESP32*, *Fan dc 5v*, Konektor *output*, Konektor sumber. Berfungsi sebagai kontrol lampu lalu

lintas yang telah menerima hasil data pembobotan kendaraan yang diolah menggunakan *jetson nano*. *ESP32* juga mengirim hasil kontrol melalui *Raspberry Pi* yang selanjutnya akan digunakan pada *monitoring web*.

3. *Raspberry Pi*, berfungsi sebagai *server* perantara antara hasil data pembobotan kendaraan dari Jetson ke *ESP32*, memfasilitasi *transfer* data kontrol lampu lalu lintas dari *ESP32* ke *monitoring web*. Dengan menghubungkan kedua perangkat ke jaringan yang sama, melalui *Wi-Fi*.
4. *Fan dc 5v*, berfungsi sebagai pendingin supaya alat tidak mengalami panas berlebih, sehingga tidak mengganggu proses kerja alat.
5. *Fan dc 5v*, berfungsi sebagai pendingin supaya alat tidak mengalami panas berlebih, sehingga tidak mengganggu proses kerja alat.
6. *Nvidia Jetson Nano*, berfungsi sebagai pengolah gambar untuk mendeteksi kepadatan kendaraan serta memproses data hasil gambar, sehingga menghasilkan hasil durasi waktu lampu hijau.
7. *Router*, berfungsi sebagai penghubung jaringan antara *Jetson Nano*, *ESP32*, dan *Monitoring Web*. Memungkinkan untuk berkomunikasi satu sama lain dengan layanan tertentu mendapatkan bandwidth yang dibutuhkan.
8. *Terminal Ekstensi (AC)*, berfungsi sebagai sumber (*AC*) listrik. Untuk menyalakan alat yaitu : Kipas *AC*, *Power Supply 5v*, *Power Supply 12v*.
9. *Stop Kontak*, berfungsi sebagai sumber listrik untuk menyalakan *Raspberry Pi*, *Router*, *Switch PoE*.
10. *Terminal Ekstensi (DC 5v)*, berfungsi sebagai sumber (*DC 5v*). Untuk menyalakan komponen yaitu : *ESP32*, sensor tegangan, *relay module*, *jetson nano*.
11. *Power Supply 12v*, berfungsi sebagai sumber lampu lalu lintas.
12. *Switch PoE*, berfungsi untuk memberikan pasokan tegangan pada *CCTV* atau perangkat jaringan internet dengan menggunakan kabel jaringan, kabel *LAN*, atau kabel *ethernet*.
13. *Relay Module*, berfungsi untuk memberikan *trigger* ke lampu lalu lintas, sehingga lampu bisa menyala/padam sesuai dengan program yang telah dikirimkan dari *ESP32*.

14. Sensor Tegangan *DC*, berfungsi untuk mengukur tegangan lampu lalu lintas, sehingga hasil dari tegangan digunakan untuk *realtime* indikator di *monitoring web*.
15. *Terminal Ekstensi (DC+ 12v)*, berfungsi sebagai sumber (*DC+ 12v*) untuk diberikan ke sensor tegangan, sehingga sensor bisa membaca sumber tegangan dari lampu lalu lintas.
16. *Terminal Ekstensi (DC- 12v)*, berfungsi sebagai sumber (*DC- 12v*) *ground (GND)* dari lampu lalu lintas.
17. *Power Supply 5v*, berfungsi untuk memberikan sumber *DC 5v* pada komponen seperti, *ESP32*, *relay module*, *jetson nano*, sensor tegangan.
18. *Miniature Circuit Breaker (1)*, berfungsi untuk memutus aliran arus listrik secara otomatis pada saat terjadi beban arus listrik melebihi kapasitas sesuai kemampuan MCB tersebut, sehingga MCB *OFF* secara otomatis bisa karena *overload*, atau terjadi konsleting pada instalasi. Serta sebagai saklar *On/Off* pada lampu lalu lintas.
19. *Miniature Circuit Breaker (2)*, berfungsi untuk memutus aliran arus listrik secara otomatis pada saat terjadi beban arus listrik melebihi kapasitas sesuai kemampuan MCB tersebut, sehingga MCB *OFF* secara otomatis bisa karena *overload*, atau terjadi konsleting pada instalasi. Serta sebagai saklar *On/Off* pada *jetson nano*.
20. *Miniature Circuit Breaker (3)*, berfungsi untuk memutus aliran arus listrik secara otomatis pada saat terjadi beban arus listrik melebihi kapasitas sesuai kemampuan MCB tersebut, sehingga MCB *OFF* secara otomatis bisa karena *overload*, atau terjadi konsleting pada instalasi. Serta sebagai saklar keseluruhan alat *box panel*, sehingga alat bisa menyala/padam.

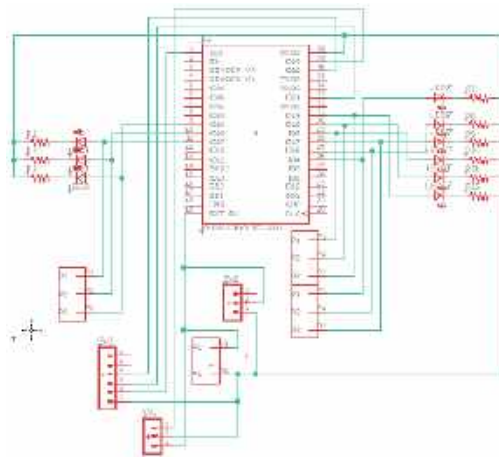
3.6 Perancangan Elektrikal

Berikut ini merupakan perancangan elektrikal dari “Perancangan Alat Perangkat Keras Pada *Adaptive Traffic Light*”. Terdapat beberapa sub-bab yang akan dibahas yaitu pembuatan skematik dan *PCB board*, *wiring diagram*.

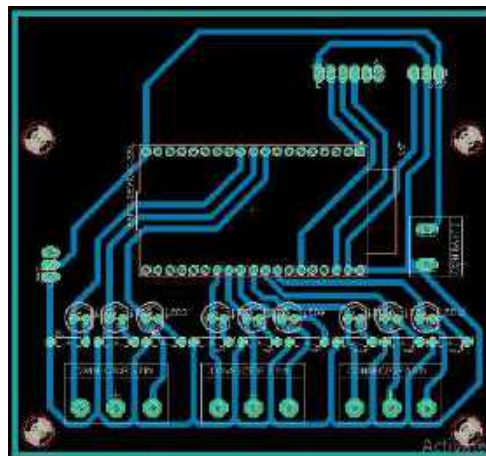
3.6.1 Pembuatan Skematik dan *PCB Board*

Membuat Desain dari *layout PCB* yang dibuat dengan *software Eagle*. Tahap awal melakukan desain yaitu membuat skematik rangkaian. Kemudian beralih ke

menu *board* di *software Eagle* lalu melakukan *wiring* antar komponen yang sudah ditentukan. Berikut hasil desain skematik dan *board* pada *PCB*:

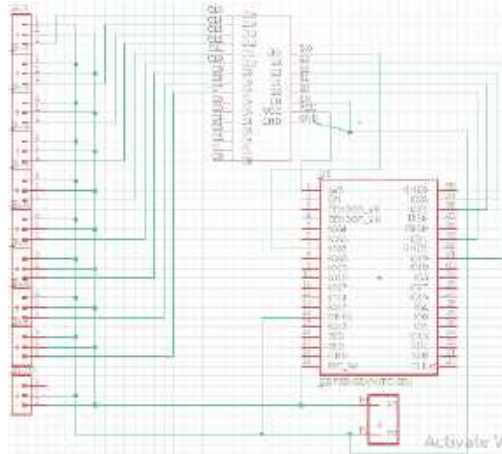


Gambar 3. 8
Skematik *PCB* Control lampu lalu lintas

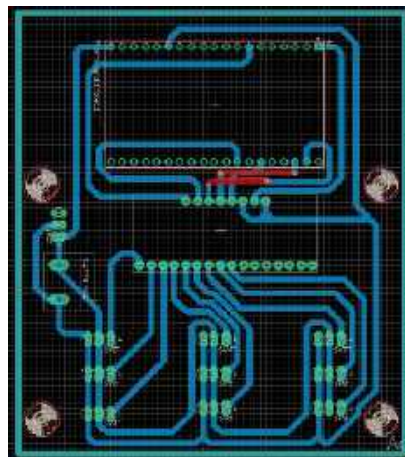


Gambar 3. 9
Board *PCB* Control lampu lalu lintas

Pada Gambar 3.8 dan Gambar 3.9 merupakan rangkaian skematik kontrol lampu lalu lintas yang digunakan untuk men-*trigger* *Relay Module* ke lampu lalu lintas, serta digunakan untuk mendeteksi waktu dan jam pada sistem *warning light*. Dimana gambar menunjukkan sebuah sistem bekerja dengan saling terhubung dan berinteraksi satu sama lain. Dalam rangkaian skematik ini terdapat komponen berupa *Node MCU ESP32*, *Led 3mm*, *Resistor 220ohm*, konektor 3 pin, konektor 2 pin, Modul *Real Time Clock (RTC)*, *Pin header*.



Gambar 3. 10
Skematik *PCB* deteksi kerusakan lampu lalu lintas



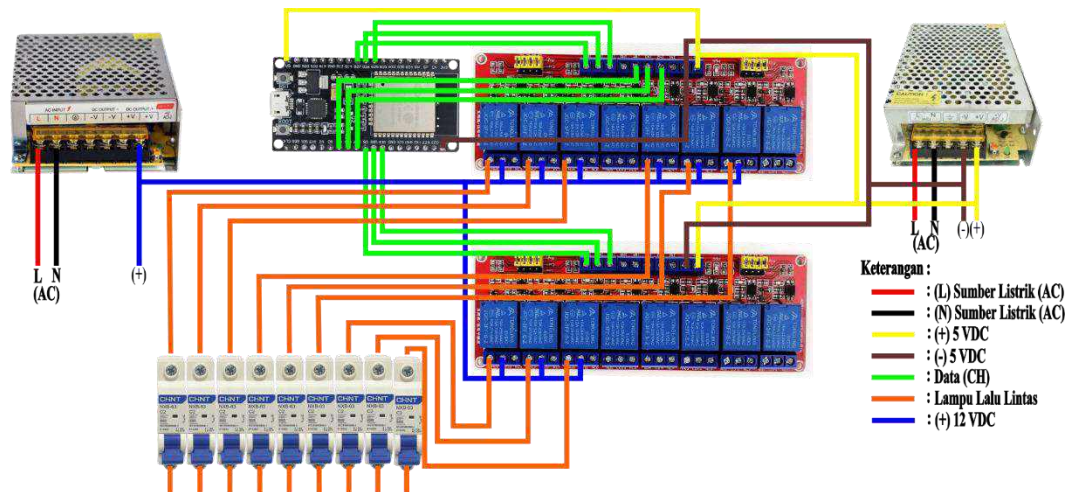
Gambar 3. 11
Board PCB deteksi kerusakan lampu lalu lintas

Pada Gambar 3.10 dan Gambar 3.11 merupakan rangkaian skematik dan *PCB* deteksi kerusakan lampu lalu lintas yang digunakan untuk mendeteksi kerusakan pada lampu lalu lintas. Dimana gambar menunjukkan bagaimana sebuah sistem bekerja dengan saling terhubung pada pin *ADC ESP32* dan berinteraksi satu sama lain. Dalam rangkaian ini terdapat *Node MCU ESP32*, konektor 2 pin, *Pin header*

3.6.2 *Wiring Diagram*

Pada bagian ini, akan dijelaskan *Wiring Diagram* dalam pembuatan tugas akhir dengan judul “Perancangan Alat Perangkat Keras Pada *Adaptive Traffic Light*”.

3.6.2.1 Wiring Kontrol Lampu Lalu Lintas

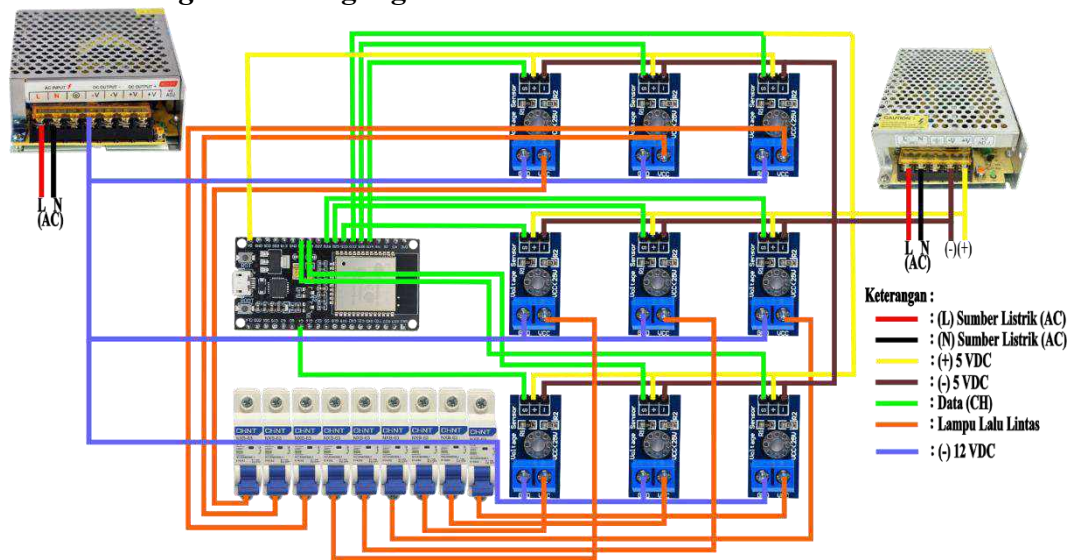


Gambar 3. 12
Wiring Diagram Kontrol Lampu Lalu Lintas

Pada Gambar 3.12 merupakan *wiring diagram* untuk kontrol lampu lalu lintas. Berikut Penejelasanannya:

1. Pada kaki *Normaly Open (NO) Relay Module* terhubung dengan sumber listrik dari lampu lalu lintas serta terhubung dengan *Miniature Circuit Breaker (MCB)*, yang nantinya akan digunakan sebagai nyala/padam lampu sesuai dengan kontrol
2. Pada kaki *Common (COM) Relay Module* terhubung dengan sumber (+) 12VDC, yang nantinya memberikan sumber tegangan ke kaki *Normaly Open (NO) Relay Module*
3. Pada kaki *Input (IN) Relay Module* menerima *output (5v)* dari ESP32 berupa data kontrol lampu lalu lintas sesuai dengan program, sehingga memberikan *trigger Relay Module* untuk menyala/padam.
4. Pada kaki *(DC-) Relay Module* merupakan *ground (GND)* dari *Relay Module* yang dihubungkan pada *ground (GND) ESP32*.
5. Pada kaki *(DC+) Relay Module* merupakan sumber 5v dari *Relay Module* yang dihubungkan pada 5v ESP32.

3.6.2.2 Wiring Sensor Tegangan



Gambar 3. 13
Wiring Diagram Sensor Tegangan

Pada Gambar 3.13 merupakan *wiring diagram* untuk sensor tegangan. Berikut penjelasannya:

1. Pada kaki *ground (GND)* sensor tegangan terhubung ke sumber (-) 12VDC *power supply*, dan terhubung dengan sumber (-) lampu lalu lintas.
2. Pada kaki (*VCC*) sensor tegangan terhubung ke *output Miniature Circuit Breaker (MCB)*, serta nantinya terhubung dengan sumber (+) lampu lalu lintas.
3. Pada kaki (*S*) sensor tegangan menerima *output (5v)* dari pin *ADC ESP32* berupa data sensor sesuai dengan program, sehingga memberikan *output* menyala/padam sesuai dengan lampu lalu lintas.
4. Pada kaki (+) sensor tegangan merupakan sumber 5v terhubung dengan sumber 5v *ESP32*, digunakan untuk sumber sensor tegangan, sehingga sensor menyala.
5. Pada kaki (-) sensor tegangan merupakan *ground (GND)* dari sensor tegangan terhubung dengan *ground (GND) ESP32*.

3.7 Proses Pembuatan

Proses pembuatan dan perakitan alat dilakukan secara bertahap dan berkala. Perakitan dilakukan di Laboratorium Prodi Teknik Komputer Kontrol Politeknik Negeri Madiun. Adapun yang penulis lakukan selama proses pembuatan sebagai berikut :



Gambar 3. 14
Pembuatan *PCB Control*

Gambar 3.14 merupakan hasil dari proses pembuatan *PCB Control* yang digunakan untuk mengontrol lampu lalu lintas setelah menerima *input* dari sistem deteksi kepadatan lalu lintas maupun *input* dari sistem *monitoring web*. Dimana gambar menunjukkan sebuah sistem bekerja dengan saling terhubung dan berinteraksi satu sama lain. Dalam rangkaian ini terdapat *ESP32* sebagai otak atau pengendali dari rangkaian untuk memberikan *output* ke *input relay module*, konektor 3 pin sebanyak 3 buah sebagai *output* dari program kontrol yang nantinya diteruskan oleh *relay module* untuk menyala/padam kan lampu lalu lintas, konektor 2 pin sebagai sumber tegangan 5v untuk menyalakan *ESP32*, *LED 3mm* warna merah, kuning, hijau masing-masing sebanyak 3 buah sebagai indikator *output* lampu lalu lintas 3 *fase*.



Gambar 3. 15
Pembuatan *PCB Sensor Tegangan*

Gambar 3.15 merupakan hasil dari proses pembuatan *PCB* sensor tegangan yang digunakan untuk mengukur tegangan dari lampu lalu lintas menggunakan sensor tegangan *DC*, serta digunakan sebagai indikator *realtime* pada *monitoring web*. Dimana gambar menunjukkan sebuah sistem berkerja dengan saling terhubung dan berinteraksi satu sama lain. Dalam rangkaian ini terdapat *ESP32* sebagai otak atau pengendali dari rangkaian untuk menerima *input* dari sensor tegangan *DC* lalu mengeluarkan *output* pada *serial monitor* dan pada indikator di *monitoring web*, konektor 2 pin sebagai sumber tegangan 5v untuk menyalakan *ESP32*, *Japan Solderless Terminal (JST)* 3 pin sebanyak 9 buah sebagai penerima dari *output* sensor tegangan *DC* yang nantinya akan diproses pada *ESP32*.

3.7.1 Proses Perakitan

Proses perakitan dan pemasangan komponen pada *box panel*. Komponen yang terpasang pada *box panel* terdiri dari *miniature circuit breaker (MCB)*, *Power Supply 12v 10a*, *Power Supply 5v 10a*, *Switch PoE*, *Router*, *Relay Module*, *Sensor Tegangan*, *Jetson Nano*, *Raspberry Pi*, *PCB Control*, *PCB Sensor Tegangan*, *Stop Kontak*, *Terminal Ekstensi*, *Kipas AC*.

3.7.2 Pengkabelan

Pengkabelan pada komponen yang terdapat di dalam *box panel*, sesuai dengan skema perancangan yang telah ditentukan. Kabel yang digunakan dalam proses pengkabelan antara lain, kabel *NYAF 1,5 mm*, kabel *AWG 26*.



Gambar 3. 16
Proses Pengkabelan Pada Komponen

3.8 Proses Pengujian

Proses pengujian dilakukan untuk memastikan sistem bekerja sesuai dengan perintah dan tidak ada *problem*, sehingga bisa dilakukan *investigasi* ketika terdapat kesalahan. Adapun proses pengujian dimulai dari mengecek koneksi antar *hardware*, pengujian *trigger output* dari *ESP32*, pengujian sensor tegangan *DC*.

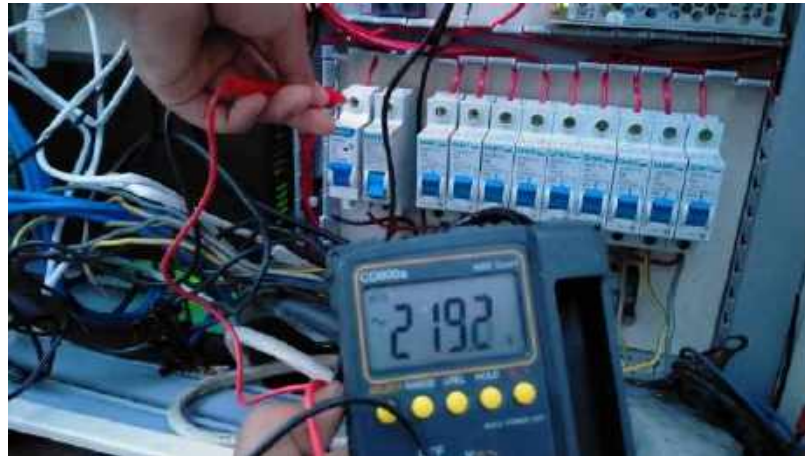
3.8.1 Koneksi Antar *Hardware*

Hal ini diperlukan untuk memastikan semua koneksi komponen terhubung dengan perangkat lainnya dengan benar dan berfungsi secara optimal. Pada Gambar 3.17 merupakan rangkaian dari keseluruhan sistem yang sudah terpasang pada *box panel*.



Gambar 3. 17
Tampilan Rangkaian Keseluruhan Sistem

Pada Gambar 3.18 merupakan hasil pengukuran arus tegangan yang masuk ke dalam *box panel*. Menunjukkan voltase yang masuk atau *input* tegangan PLN sebesar $219VAC$. Arus tersebut masuk ke terminal blok, untuk dialirkan selanjutnya ke perangkat lainnya.



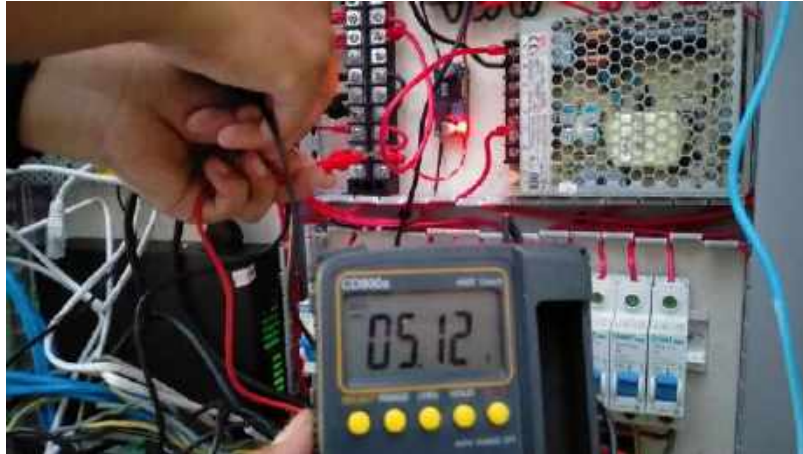
Gambar 3. 18
Pengukuran Arus *Voltase* Yang Masuk Box Panel

Pada Gambar 3.19 merupakan hasil pengukuran *output* tegangan dari *power supply* 12V. Hasil *output* dari *power supply* yang digunakan bernilai 11,73V DC. Arus DC ini akan menjadi sumber tegangan untuk perangkat atau komponen yang membutuhkan arus 12V DC seperti *ground (GND)* lampu lalu lintas, *input* sensor tegangan yang akan membaca sumber 12V DC.



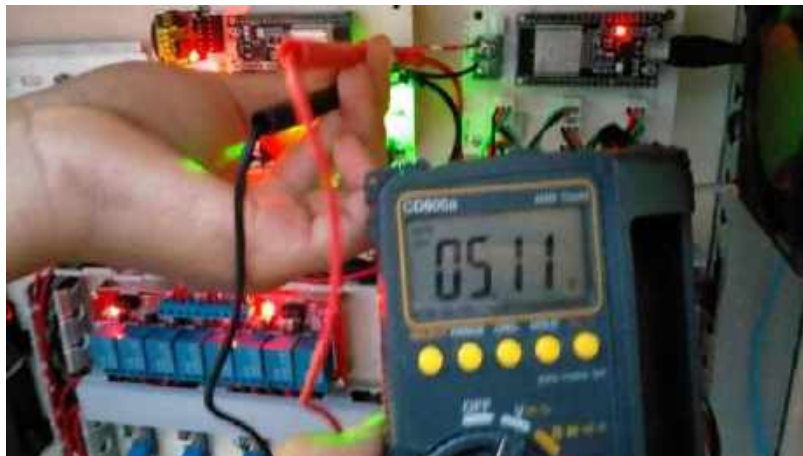
Gambar 3. 19
Pengukuran Tegangan *Output Power Supply* 12V

Pada Gambar 3.20 merupakan hasil pengukuran *output* dari *stepdown* yang berasal dari sumber 12V DC yang nantinya akan dirubah menjadi sumber 5V DC. Keluaran pada *stepdown* ini bernilai 5,12V DC. Arus DC ini akan menjadi sumber tegangan untuk perangkat atau komponen yang membutuhkan arus 5V DC seperti *ESP32* kontrol, *ESP32* tegangan, *relay module*, *sensor tegangan*.



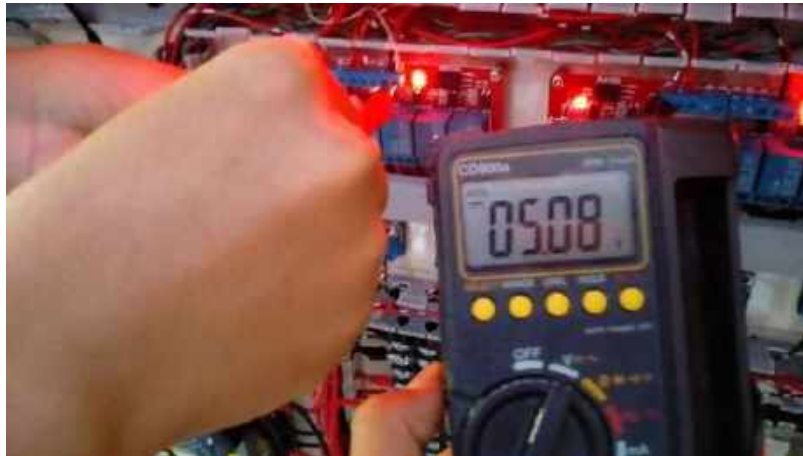
Gambar 3. 20
Pengukuran *Output* Dari *Stepdown*

Pada Gambar 3.21 Merupakan hasil pengukuran tegangan *input* dengan sumber $5,12V DC$ yang digunakan untuk menyalakan *ESP32*. Sehingga *ESP32* bisa menjalankan instruksinya sesuai dengan program yang telah dibuat.



Gambar 3. 21
Pengukuran Tegangan *Input ESP32* Kontrol

Gambar 3.22 merupakan hasil pengukuran tegangan *input* dengan sumber $5,08V DC$ yang digunakan untuk menyalakan *relay module*. Sehingga *relay module* bisa menjalankan instruksinya sesuai dengan *output* dari program *ESP32* kontrol yang telah dibuat.



Gambar 3. 22
Pengukuran *Input Relay Module*

Pada Gambar 3.23 merupakan hasil pengukuran *output* dari *ESP32* kontrol sebesar $3,12V\ DC$, dari *output* tersebut digunakan untuk *men-trigger relay module* sesuai program yang telah dibuat, sehingga lampu lalu lintas bisa menyala/padam.



Gambar 3. 23
Pengukuran *Output ESP32*

3.8.2 Pengujian Sensor Tegangan *DC*

Pengujian sensor tegangan *DC* dilakukan bertujuan untuk mengetahui tingkat akurasi dan *error* pendeteksian tegangan *DC*, serta menguji apakah *output* dari sensor tegangan *DC* ini sebesar $12V\ DC$ atau tidak. *Input* dari sensor tegangan ini berupa tegangan *DC* $12V\ DC$ yang didapatkan dari tegangan nyala/padam lampu lalu lintas.

3.8.2.1 Pengujian Sensor Tegangan DC Ketika Lampu Menyala



Gambar 3. 24
Pengujian *Input* Sensor tegangan Ketika Lampu Menyala

Gambar 3.24 merupakan hasil pengukuran *input* dari sensor tegangan ketika lampu menyala dengan menghasilkan tegangan sebesar $11,92V$ DC. dengan diidentifikasi bahwa lampu lalu lintas bernilai 1. Hasil dari pengukuran tersebut kemudian masuk ke *ESP32* yang kemudian dilanjutkan menuju *server raspberry pi* menggunakan protokol komunikasi *Message Queue Telemetry Protocol (MQTT)* untuk menampilkan indikator *realtime* nyala/padam lampu lalu lintas.

```
Red 1: 12.00V (ON)
```

Gambar 3. 25
Hasil *Output* Sensor Tegangan ketika Lampu Menyala

Gambar 3.25 merupakan hasil pengukuran *output* dari sensor tegangan ketika lampu menyala. Hasil pengukuran tersebut terbaca di *serial monitor software Arduino IDE*, sebelum ditampilkan hasil pengukuran tersebut sensor tegangan di kalibrasi terlebih dahulu dengan rumus perhitungan konversi *ADC to Volt* $4096 * 15 * 32350 / 27000$, sehingga menghasilkan nilai tegangan 12V. Kemudian hasil pengukuran itu juga dikirim ke *server raspberry pi* menggunakan protokol komunikasi *Message Queue Telemetry Protocol (MQTT)*.



Gambar 3. 26
Pengujian *Input* Sensor tegangan Ketika Lampu Padam

Gambar 3.26 merupakan hasil pengukuran *input* dari sensor tegangan ketika lampu padam dengan menghasilkan tegangan sebesar $0,3mV$ DC. dengan diidentifikasi bahwa lampu lalu lintas bernilai 0. Hasil dari pengukuran tersebut kemudian masuk ke *ESP32* yang kemudian dilanjutkan menuju *server raspberry pi* menggunakan protokol komunikasi *Message Queue Telemetry Protocol (MQTT)* untuk menampilkan indikator *realtime* nyala/padam lampu lalu lintas.

Yellow 1: 0.00V (OFF)

Gambar 3. 27
Hasil *Output* Sensor Tegangan Ketika Lampu Padam

Gambar 3.27 merupakan hasil pengukuran *output* dari sensor tegangan ketika lampu menyala. Hasil pengukuran tersebut terbaca di *serial monitor software Arduino IDE*, sebelum ditampilkan hasil pengukuran tersebut sensor tegangan di kalibrasi terlebih dahulu dengan rumus perhitungan konversi *ADC to Volt* $4096 * 15 * 32350 / 27000$, sehingga menghasilkan nilai tegangan 0V. Kemudian hasil pengukuran itu juga dikirim ke *server raspberry pi* menggunakan protokol komunikasi *Message Queue Telemetry Protocol (MQTT)*.

Setelah melakukan pengujian, maka didapatkan hasil pengujian sensor tegangan *DC* yang dapat dilihat pada tabel 3.2.

Tabel 3. 2
Hasil Pengujian Sensor Tegangan *DC* Kondisi Menyala

No	Tegangan <i>Input</i> Sensor Tegangan <i>DC</i>	Tegangan <i>Output</i> Sensor Tegangan <i>DC</i>	Nilai <i>Error</i>
1	11,92 <i>V</i>	11,94 <i>V</i>	0,02 <i>V</i>
2	11,96 <i>V</i>	11,97 <i>V</i>	0,01 <i>V</i>
3	11,99 <i>V</i>	12 <i>V</i>	0,01 <i>V</i>
4	12,01 <i>V</i>	12,03 <i>V</i>	0,02 <i>V</i>
5	11,89 <i>V</i>	11,92 <i>V</i>	0,03 <i>V</i>
6	12,03 <i>V</i>	12,05 <i>V</i>	0,02 <i>V</i>
7	11,98 <i>V</i>	12,03 <i>V</i>	0,05 <i>V</i>
8	12,01 <i>V</i>	12,04 <i>V</i>	0,03 <i>V</i>
9	12,02 <i>V</i>	12,06 <i>V</i>	0,04 <i>V</i>
Rata-rata <i>Input</i> Tegangan		11,97 <i>V</i>	
Rata-rata <i>Output</i> Tegangan		12,00 <i>V</i>	
Total Rata-rata Nilai <i>Error</i>		0,02 <i>V</i>	
Total Rata-rata Tegangan		1,02 <i>V</i>	

Berikut perhitungan perbandingan nilai rata-rata tegangan di setiap lampu dengan kondisi menyala pada persamaan berikut:

Input

$$\begin{aligned}
 &= \frac{11,92 + 11,96 + 11,99 + 12,01 + 11,89 + 12,03 + 11,98 + 12,01 + 12,02}{9} \\
 &= \frac{107,81}{9} = 11,97 \text{ V}
 \end{aligned}$$

Output

$$= \frac{11,94 + 11,97 + 12 + 12,03 + 11,92 + 12,05 + 12,03 + 12,04 + 12,06}{9}$$

$$= \frac{108,04}{9} = 12,0 \text{ V}$$

$$V_{total} = \frac{Output}{Input} = \frac{12,0}{11,97} = 1,02V$$

Berikut perhitungan nilai rata-rata *error* tegangan di setiap lampu dengan kondisi padam sebagai berikut:

$$Error = \frac{0,2 + 0,1 + 0,1 + 0,2 + 0,3 + 0,2 + 0,5 + 0,3 + 0,4}{9}$$

$$= \frac{0,23}{9} = 0,02 \text{ V}$$

Berdasarkan hasil perhitungan perbandingan nilai rata-rata tegangan di setiap lampu. Pada *input* tegangan mendapatkan nilai rata-rata sebesar 11,97 *V*. Pada *output* tegangan mendapatkan nilai rata-rata sebesar 12,0 *V*. Total nilai rata-rata perhitungan tegangan ini menghasilkan sebesar 1,02 *V*. Dan total nilai rata-rata hasil perhitungan *error* menghasilkan 0,02 *V*

Tabel 3. 3
Hasil Pengujian Sensor tegangan *DC* Kondisi Padam

No	Tegangan <i>Input</i> Sensor Tegangan <i>DC</i>	Tegangan <i>Output</i> Sensor Tegangan <i>DC</i>	Nilai <i>Error</i>
1	0,3 <i>V</i>	0 <i>V</i>	0,3 <i>V</i>
2	0,2 <i>V</i>	0 <i>V</i>	0,2 <i>V</i>
3	0,1 <i>V</i>	0 <i>V</i>	0,1 <i>V</i>
4	0,1 <i>V</i>	0 <i>V</i>	0,1 <i>V</i>
5	0,3 <i>V</i>	0 <i>V</i>	0,3 <i>V</i>
6	0,4 <i>V</i>	0 <i>V</i>	0,4 <i>V</i>
7	0,2 <i>V</i>	0 <i>V</i>	0,2 <i>V</i>

8	0,1 <i>V</i>	0 <i>V</i>	0,1 <i>V</i>
9	0,2 <i>V</i>	0 <i>V</i>	0,2 <i>V</i>
Rata-rata <i>Input</i> Tegangan		0,21 <i>V</i>	
Rata-rata <i>Output</i> Tegangan		0 <i>V</i>	
Total Rata-rata Nilai Error		0,21 <i>V</i>	
Total Rata-rata Tegangan		0 <i>V</i>	

Berikut perhitungan perbandingan nilai rata-rata tegangan di setiap lampu dengan kondisi padam sebagai berikut:

$$Input = \frac{0,3 + 0,2 + 0,1 + 0,1 + 0,3 + 0,4 + 0,2 + 0,1 + 0,2}{9}$$

$$= \frac{1,9}{9} = 0,21 \text{ V}$$

$$Output = \frac{0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0}{9}$$

$$= \frac{0}{9} = 0 \text{ V}$$

$$V_{total} = \frac{Output}{Input} = \frac{0}{0,21} = 0 \text{ V}$$

Berikut perhitungan nilai rata-rata *error* tegangan di setiap lampu dengan kondisi padam sebagai berikut:

$$Error = \frac{0,3 + 0,2 + 0,1 + 0,1 + 0,3 + 0,4 + 0,2 + 0,1 + 0,2}{9}$$

$$= \frac{1,9}{9} = 0,21 \text{ V}$$

Berdasarkan hasil perhitungan perbandingan nilai rata-rata tegangan di setiap lampu. Pada *input* tegangan mendapatkan nilai rata-rata sebesar 0,21 *V*. Pada Pada *output* tegangan mendapatkan nilai rata-rata sebesar 0 *V*. Total nilai rata-rata perhitungan tegangan ini menghasilkan sebesar 0 *V*. Dan total nilai rata-rata hasil perhitungan *error* menghasilkan 0,21 *V*

3.8.3 Koneksi Antar *Jetson Nano*

Hal ini diperlukan untuk memastikan semua koneksi antara *jetson nano* terhubung dengan perangkat keras pada *adaptive traffic light* yang dijalankan.

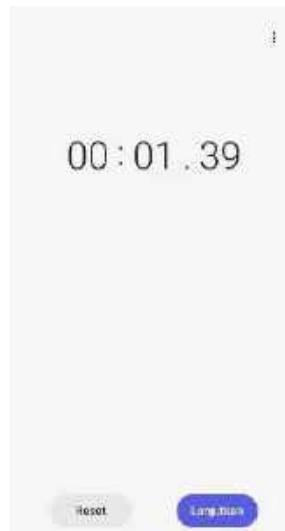
```

Durasi lampu hijau 1 :15
Durasi lampu Merah 1 :25
MQTT message for Capture Phase 2 sent successfully
Message arrived [greentime] jalan2: 15 detik
Updated green time for jalan2 to 15 seconds
Durasi lampu hijau 2 :15
Durasi lampu Merah 2 :25
MQTT message for Capture Phase 3 sent successfully
Message arrived [greentime] jalan3: 15 detik
Updated green time for jalan3 to 15 seconds
Durasi lampu hijau 3 :15
Durasi lampu Merah 3 :25
MQTT message for Capture Phase 1 sent successfully
Message arrived [greentime] jalan1: 15 detik
Updated green time for jalan1 to 15 seconds
Durasi lampu hijau 1 :15
Durasi lampu Merah 1 :25
MQTT message for Capture Phase 2 sent successfully
Message arrived [greentime] jalan2: 15 detik
Updated green time for jalan2 to 15 seconds
Durasi lampu hijau 2 :15
Durasi lampu Merah 2 :25

```

Gambar 3. 28
Hasil Koneksi Dengan *Jetson Nano* & *ESP32* Kontrol

Pada Gambar 3.28 merupakan hasil koneksi dengan *jetson nano* dan *ESP32*. Hasil dari perhitungan *jetson nano* dikirim menggunakan komunikasi *Message Queue Telemetry Protocol (MQTT)* pada *server raspberry pi*. Kemudian data tersebut terbaca pada *serial monitor* di *software Arduino IDE*. Ketika data berhasil dikirim akan menampilkan pesan berupa (*MQTT message for capture phase 1 sent succesfully*) bahwa *ESP32* mampu mengirimkan sinyal berupa lampu hijau dilanjutkan dengan pesan (*update green time for jalan 1 to 15 seconds*) bahwa waktu lampu hijau dan lampu merah telah berubah sesuai hasil *capture* dan perhitungan dari *jetson nano*.



Gambar 3. 29
Hasil Pengujian Kecepatan Pengiriman Data

Pada Gambar 3.29 merupakan hasil pengujian kecepatan pengiriman data berupa hasil perhitungan dari *jetson nano* melalui *MQTT* pada *server raspberry pi*. Hasil perhitungan kecepatan pengiriman kali ini diuji menggunakan *stopwatch* dengan satuan kecepatan berupa detik.

Setelah melakukan pengujian, maka mendapatkan hasil berupa waktu kecepatan pengiriman data yang dapat dilihat pada tabel 3.4.

Tabel 3. 4
Hasil Pengujian Pengiriman Kecepatan Data

No	Nama Pengujian	Hasil Pengujian
1.	Kecepatan Pengiriman Data Fase 1	2,45 detik
		2,15 detik
		2,08 detik
		2,31 detik
		1,48 detik
	Rata-rata Kecepatan Pengiriman Data Fase 1	2.09 detik
		1,35 detik

2.	Kecepatan Pengiriman Data Fase 2	1,52 detik
		2,27 detik
		2,35 detik
		1,20 detik
	Rata-rata Kecepatan Pengiriman Data Fase 2	2,13 detik
3.	Kecepatan Pengiriman Data Fase 3	2,33 detik
		2,12 detik
		1,36 detik
		1,17 detik
		2,23 detik
	Rata-rata Kecepatan Pengiriman Data Fase 3	2,24 detik
	Total Rata-rata Kecepatan Pengiriman Data	2,15 detik

Berikut perhitungan nilai rata-rata kecepatan pengiriman data di setiap fase pada persamaan (4.1) sampai (4.4) berikut:

$$F1 = \frac{2,45 + 2,15 + 2,08 + 2,31 + 1,48}{10} = \frac{10,47}{5} = 2,09 \text{ detik} \quad (4.1)$$

$$F2 = \frac{1,35 + 1,52 + 2,27 + 2,35 + 1,20}{10} = \frac{1,73}{5} = 1,73 / 2,13 \text{ detik} \quad (4.2)$$

$$F3 = \frac{2,33 + 2,12 + 1,36 + 1,17 + 2,23}{10} = \frac{9,21}{5} = 1,84 / 2,24 \text{ detik} \quad (4.3)$$

$$R_{total} = \frac{2,09 + 2,13 + 2,24}{3} = \frac{6,46}{3} = 2,15 \text{ detik} \quad (4.4)$$

Berdasarkan hasil pengujian pengukuran kecepatan pengiriman data. Pada kecepatan pengiriman data pada fase 1 mendapatkan nilai rata-rata sebesar 2,09 detik. Pada kecepatan pengiriman data pada fase 2 mendapatkan nilai rata-rata sebesar 2,13 detik. Serta pada kecepatan pengiriman data pada fase 3 mendapatkan nilai rata-rata sebesar 2,24 detik. Total rata-rata kecepatan pengiriman ini menghasilkan sebesar 2,15 detik.

BAB IV

HASIL DAN PEMBAHASAN

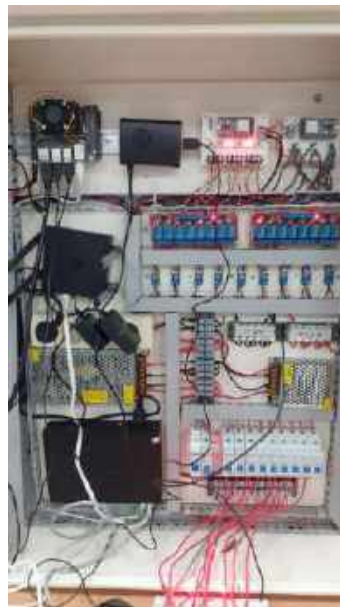
Berdasarkan serangkaian pengujian yang sudah penulis lakukan, untuk memaparkan hasil dan pembahasan sesuai dengan rumusan masalah yaitu, Bagaimana merancang dan membuat perangkat keras pada *Adaptive Traffic Light* berdasarkan kepadatan kendaraan pada persimpangan jalan, Bagaimana perangkat keras pada *Adaptive Traffic Light* dapat mengirim dan menerima sinyal dari sistem pendeteksi dan penghitung kepadatan kendaraan, serta Bagaimana perangkat keras pada *Adaptive Traffic Light* dapat mengirim data ke sistem monitoring *Adaptive Traffic Light*. Berikut merupakan pembahasan dari ketiga rumusan masalah sebagai berikut:

4.1 Merancang dan Membuat Perangkat Keras Pada *Adaptive Traffic Light* Berdasarkan Kepadatan Kendaraan Pada Persimpangan Jalan

Untuk merancang dan membuat perangkat keras pada *adaptive traffic light* ini akan dipaparkan tentang perancangan sistem pada perangkat keras.

4.1.1 Perancangan Sistem Perangkat Keras *Adaptive Traffic Light*

Pada perancangan sistem perangkat keras *adaptive traffic light* ini, akan dipaparkan tentang tampilan *box panel adaptive traffic light*.



Gambar 4. 1
Box Panel Adaptive Traffic Light

Pada Gambar 4.1 merupakan tampilan *box panel* perangkat keras pada *adaptive traffic light*. *Box panel Adaptive Traffic Light* memiliki dimensi ukuran *box panel* dengan ukuran Panjang 80cm, lebar 60cm, tinggi 20cm. Memiliki topi pada atas *box panel* berfungsi sebagai *penapis* air hujan ketika hujan turun, sehingga air tidak bisa masuk ke dalam *box panel* dan seluruh komponen aman dari *korsleting* listrik.

4.2 Mengirim dan Menerima Sinyal Dari Sistem Pendeteksi dan Penghitung Kepadatan Kendaraan

Setelah melakukan pengiriman dan menerima sinyal dari sistem deteksi dan penghitung kendaraan menggunakan protokol komunikasi *Message Queue Telemetry Protocol (MQTT)*, maka dilakukan pengujian tentang komunikasi pengiriman dan dan penerimaan data hasil *capture* dan data kontrol hasil perhitungan.

4.2.1 Integrasi Perangkat Keras Dengan *Jetson Nano*

Hasil dari menerima dan mengirim sinyal dari sistem pendeteksi dan penghitung kepadatan kendaraan dapat diketahui dari serial monitor pada *software Arduino IDE*.

```

Durasi lampu hijau 1 :15
Durasi lampu Merah 1 :25
MQTT message for Capture Phase 2 sent successfully
Message arrived [greentime] jalan2: 15 detik
Updated green time for jalan2 to 15 seconds
Durasi lampu hijau 3 :15
Durasi lampu Merah 2 :25
MQTT message for Capture Phase 3 sent successfully
Message arrived [greentime] jalan3: 15 detik
Updated green time for jalan3 to 15 seconds
Durasi lampu hijau 3 :15
Durasi lampu Merah 3 :25
MQTT message for Capture Phase 1 sent successfully
Message arrived [greentime] jalan1: 15 detik
Updated green time for jalan1 to 15 seconds
Durasi lampu hijau 1 :15
Durasi lampu Merah 1 :25
MQTT message for Capture Phase 2 sent successfully
Message arrived [greentime] jalan2: 15 detik
Updated green time for jalan2 to 15 seconds
Durasi lampu hijau 2 :15
Durasi lampu Merah 2 :25

```

Gambar 4. 2
Hasil Integrasi *Jetson Nano* Dengan *ESP32*

Gambar 4.2 merupakan hasil *integrasi* antara *jetson nano* dengan *ESP32*. Terdapat pesan berupa (*MQTT message for capture phase 1 sent succesfully*) bahwa *ESP32* mampu mengirimkan sinyal berupa lampu sebelum hijau, maka akan memberikan sinyal ke *jetson nano* untuk saatnya *capture* gambar. Selanjutnya akan menerima pesan kembali pada *serial monitor* berupa (*updated green time for jalan1*

to 15 second) kemudian pesan (durasi lampu hijau 1: 15) dan (durasi lampu merah 1: 25) bahwa durasi lampu hijau dan lampu merah berhasil diubah sesuai dengan hasil perhitungan yang telah dihitung pada *jetson nano*.

Setelah melakukan pengujian tentang komunikasi antara *jetson nano* dan perangkat keras, maka melanjutkan pengujian tentang seberapa cepat proses pengiriman data hasil *capture* gambar. Berikut merupakan hasil pengujian kecepatan data hasil *capture* gambar dapat dilihat pada tabel 4.1

Tabel 4. 1
Hasil Pengukuran Kecepatan Pengiriman Data *Capture* gambar

No	Nama Pengujian	Hasil Pengujian
1.	Kecepatan Pengiriman Data Fase 1	2,45 detik
		2,15 detik
		2,08 detik
		2,31 detik
		1,48 detik
	Rata-rata Kecepatan Pengiriman Data Fase 1	2.09 detik
2.	Kecepatan Pengiriman Data Fase 2	1,35 detik
		1,52 detik
		2,27 detik
		2,35 detik
		1,20 detik
	Rata-rata Kecepatan Pengiriman Data Fase 2	2,13 detik
3.	Kecepatan Pengiriman Data Fase 3	2,33 detik
		2,12 detik
		1,36 detik

		1,17 detik
		2,23 detik
	Rata-rata Kecepatan Pengiriman Data Fase 3	2,24 detik
	Total Rata-rata Kecepatan Pengiriman Data	2,15 detik

Berikut perhitungan nilai rata-rata kecepatan pengiriman data di setiap fase pada persamaan (4.1) sampai (4.4) berikut:

$$F1 = \frac{2,45 + 2,15 + 2,08 + 2,31 + 1,48}{10} \quad (4.1)$$

$$= \frac{10,47}{5} = 2,09 \text{ detik}$$

$$F2 = \frac{1,35 + 1,52 + 2,27 + 2,35 + 1,20}{10} \quad (4.2)$$

$$= \frac{1,73}{5} = 1,73 / 2,13 \text{ detik}$$

$$F3 = \frac{2,33 + 2,12 + 1,36 + 1,17 + 2,23}{10} \quad (4.3)$$

$$= \frac{9,21}{5} = 1,84 / 2,24 \text{ detik}$$

$$R_{total} = \frac{2,09 + 2,13 + 2,24}{3} = \frac{6,46}{3} = 2,15 \text{ detik} \quad (4.4)$$

Berdasarkan hasil pengujian pengukuran kecepatan pengiriman data. Pada kecepatan pengiriman data pada fase 1 mendapatkan nilai rata-rata sebesar 2,09 detik. Pada kecepatan pengiriman data pada fase 2 mendapatkan nilai rata-rata sebesar 2,13 detik. Serta pada kecepatan pengiriman data pada fase 3 mendapatkan nilai rata-rata sebesar 2,24 detik. Total rata-rata kecepatan pengiriman ini menghasilkan sebesar 2,15 detik.

4.3 Perangkat Keras *Adaptive Traffic Light* Dapat Mengirim Data Ke Sistem Monitoring *Adaptive Traffic Light*.

Setelah melakukan pengiriman data nyala/padam lampu lalu lintas yang telah diukur serta dideteksi oleh sensor tegangan menggunakan protokol komunikasi *Message Queue Telemetry Protocol (MQTT)*, maka dilakukan pengujian tentang komunikasi pengiriman data hasil *output* sensor tegangan *DC*.

4.3.2 Integrasi Perangkat Keras Dengan *Monitoring Web*

Hasil pengiriman sinyal dari perangkat keras sistem pendeteksi dan penghitung kepadatan kendaraan dapat diketahui dari *monitoring web*.



Gambar 4. 3
Tampilan *Realtime* Lampu Merah Semua Fase



Gambar 4. 4
Tampilan *Realtime* Lampu Hijau dan Kuning Fase 1



Gambar 4. 5
Tampilan *Realtime* Lampu Hijau dan Kuning Fase 2



Gambar 4. 6
Tampilan *Realtime* Lampu Hijau dan Kuning Fase 3

Pada Gambar 4.3, Gambar 4.4, Gambar 4.5, Gambar 4.6 merupakan hasil *realtime* dari *monitoring web* data tersebut sesuai dengan hasil pembacaan dari sensor tegangan *DC* yang kemudian diproses oleh *PCB* tegangan yang kemudian dikirim ke *raspberry pi* menggunakan protokol komunikasi *Message Queue Telemetry Protocol (MQTT)*.



Gambar 4. 7
Hasil Pengujian *Input* Sensor Tegangan Ketika Lampu Menyala



Gambar 4. 8
Hasil Pengujian *Input* Sensor tegangan Ketika Lampu Padam

Pada Gambar 4.7 dan Gambar 4.8 merupakan hasil pengukuran tegangan *input* dan tegangan *output* yang diukur menggunakan *Avometer*. Dilakukan pengujian ketika menerima tegangan yang menghasilkan tegangan sekitar 11,92V DC ditandai dengan nilai lampu bernilai 1 yang nantinya akan menampilkan lampu menyala pada *realtime monitoring web*. Serta dilakukan pengujian ketika tidak menerima tegangan yang menghasilkan tegangan sekitar 0,3mV DC ditandai dengan nilai lampu bernilai 0 yang nantinya akan menampilkan lampu padam pada *realtime monitoring web*.

Setelah melakukan pengujian tentang komunikasi antara perangkat keras dan sistem *monitoring web*, maka melanjutkan pengujian tentang berapa tegangan *input* yang terbaca pada *Avometer* dan berapa tegangan *output* yang terbaca pada serial

monitor, sehingga mendapatkan hasil nyala/padam lampu lalu lintas sesuai dengan *output* dari kontrol *ESP32*. Berikut merupakan hasil pengujian tegangan *input* dan tegangan *output* dari sensor tegangan, gambar dapat dilihat pada tabel 4.2

Tabel 4. 2
Hasil Pengujian Sensor Tegangan *DC* Ketika Kondisi Menyala

No	Tegangan <i>Input</i> Sensor Tegangan <i>DC</i>	Tegangan <i>Output</i> Sensor Tegangan <i>DC</i>	Nilai <i>Error</i>
1	11,92 <i>V</i>	11,94 <i>V</i>	0,02 <i>V</i>
2	11,96 <i>V</i>	11,97 <i>V</i>	0,01 <i>V</i>
3	11,99 <i>V</i>	12 <i>V</i>	0,01 <i>V</i>
4	12,01 <i>V</i>	12,03 <i>V</i>	0,02 <i>V</i>
5	11,89 <i>V</i>	11,92 <i>V</i>	0,03 <i>V</i>
6	12,03 <i>V</i>	12,05 <i>V</i>	0,02 <i>V</i>
7	11,98 <i>V</i>	12,03 <i>V</i>	0,05 <i>V</i>
8	12,01 <i>V</i>	12,04 <i>V</i>	0,03 <i>V</i>
9	12,02 <i>V</i>	12,06 <i>V</i>	0,04 <i>V</i>
Rata-rata <i>Input</i> Tegangan		11,97 <i>V</i>	
Rata-rata <i>Output</i> Tegangan		12,00 <i>V</i>	
Total Rata-rata Nilai <i>Error</i>		0,02 <i>V</i>	
Total Rata-rata Tegangan		1,02 <i>V</i>	

Berikut perhitungan perbandingan nilai rata-rata tegangan di setiap lampu dengan kondisi menyala pada persamaan berikut:

Input

$$= \frac{11,92 + 11,96 + 11,99 + 12,01 + 11,89 + 12,03 + 11,98 + 12,01 + 12,02}{9}$$

$$= \frac{107,81}{9} = 11,97 \text{ V}$$

Output

$$= \frac{11,94 + 11,97 + 12 + 12,03 + 11,92 + 12,05 + 12,03 + 12,04 + 12,06}{9}$$

$$= \frac{108,04}{9} = 12,0 \text{ V}$$

$$V_{total} = \frac{Output}{Input} = \frac{12,0}{11,97} = 1,002V$$

Berikut perhitungan nilai rata-rata *error* tegangan di setiap lampu dengan kondisi padam sebagai berikut:

$$Error = \frac{0,2 + 0,1 + 0,1 + 0,2 + 0,3 + 0,2 + 0,5 + 0,3 + 0,4}{9}$$

$$= \frac{0,23}{9} = 0,02 \text{ V}$$

Berdasarkan hasil perhitungan perbandingan nilai rata-rata tegangan di setiap lampu. Pada *input* tegangan mendapatkan nilai rata-rata sebesar 11,97 V. Pada *output* tegangan mendapatkan nilai rata-rata sebesar 12,0 V. Total nilai rata-rata perhitungan tegangan ini menghasilkan sebesar 1,02 V. Dan total nilai rata-rata hasil perhitungan *error* menghasilkan 0,02 V

Tabel 4. 3
Hasil Pengujian Sensor Tegangan DC Ketika Kondisi Padam

No	Tegangan <i>Input</i> Sensor Tegangan DC	Tegangan <i>Output</i> Sensor Tegangan DC	Nilai <i>Error</i>
1	0,3 V	0 V	0,3 V
2	0,2 V	0 V	0,2 V
3	0,1 V	0 V	0,1 V
4	0,1 V	0 V	0,1 V
5	0,3 V	0 V	0,3 V
6	0,4 V	0 V	0,4 V

7	0,2 <i>V</i>	0 <i>V</i>	0,2 <i>V</i>
8	0,1 <i>V</i>	0 <i>V</i>	0,1 <i>V</i>
9	0,2 <i>V</i>	0 <i>V</i>	0,2 <i>V</i>
Rata-rata <i>Input</i> Tegangan		0,21 <i>V</i>	
Rata-rata <i>Output</i> Tegangan		0 <i>V</i>	
Total Rata-rata Nilai Error		0,21 <i>V</i>	
Total Rata-rata Tegangan		0 <i>V</i>	

Berikut perhitungan perbandingan nilai rata-rata tegangan di setiap lampu dengan kondisi padam pada persamaan berikut:

$$Input = \frac{0,3 + 0,2 + 0,1 + 0,1 + 0,3 + 0,4 + 0,2 + 0,1 + 0,2}{9}$$

$$= \frac{1,9}{9} = 0,21 \text{ V}$$

$$Output = \frac{0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0}{9}$$

$$= \frac{0}{9} = 0 \text{ V}$$

$$V_{total} = \frac{Output}{Input} = \frac{0}{0,21} = 0 \text{ V}$$

Berikut perhitungan nilai rata-rata *error* tegangan di setiap lampu dengan kondisi padam sebagai berikut:

$$Error = \frac{0,3 + 0,2 + 0,1 + 0,1 + 0,3 + 0,4 + 0,2 + 0,1 + 0,2}{9}$$

$$= \frac{1,9}{9} = 0,21 \text{ V}$$

Berdasarkan hasil perhitungan perbandingan nilai rata-rata tegangan di setiap lampu. Pada *input* tegangan mendapatkan nilai rata-rata sebesar 0,21 *V*. Pada Pada *output* tegangan mendapatkan nilai rata-rata sebesar 0 *V*. Total nilai rata-rata perhitungan tegangan ini menghasilkan sebesar 0 *V*. Dan total nilai rata-rata hasil perhitungan *error* menghasilkan 0,21 *V*

BAB V KESIMPULAN DAN SARAN

Pada bab ini akan dijelaskan mengenai kesimpulan dan saran dari seluruh perancangan, pembuatan, dan pengujian sistem alat dari tugas akhir yang berjudul “Perancangan Alat Perangkat Keras Pada *Adaptive Traffic Light*”. Dengan adanya pembahasan kesimpulan dan saran dari penulis pada tugas akhir ini diharapkan untuk kedepannya alat yang telah dibuat bisa dikembangkan lebih baik lagi.

5.1 Kesimpulan

Berdasarkan perancangan dan pengujian yang dilakukan pada “Perancangan Alat Perangkat Keras Pada *Adaptive Traffic Light*”, maka dapat diambil kesimpulan sebagai berikut:

1. Perangkat keras pada *adaptive traffic light* telah berhasil dirancang, dibuat, dan dilakukan pengujian. Hasil pengujian yang didapatkan perangkat keras dapat dirancang sesuai dengan rancangan yang telah dibuat serta mampu menjalankan program sesuai dengan *input* yang diberikan dari sistem deteksi perhitungan kendaraan maupun dari *monitoring web*.
2. Sistem perangkat keras berhasil melakukan komunikasi data dengan *jetson nano* untuk mengirim sinyal *capture* gambar kepadatan dan sistem perangkat keras berhasil menerima data durasi waktu yang telah diproses sebelumnya oleh *jetson nano*. Data-data tersebut dikirim menggunakan protokol komunikasi *Message Queue Telemetry Protocol (MQTT)*.
3. Sistem perangkat keras berhasil mengirim data hasil pengukuran tegangan menggunakan sensor tegangan *DC*. Seluruh data tersebut dikirim menggunakan protokol komunikasi *Message Queue Telemetry Protocol (MQTT)* yang dikirim ke *raspberry pi* lalu sistem *monitoring web* mengambil data-data tersebut yang nantinya akan digunakan untuk *realtime* indikator nyala/padam lampu lalu lintas.

5.2 Saran

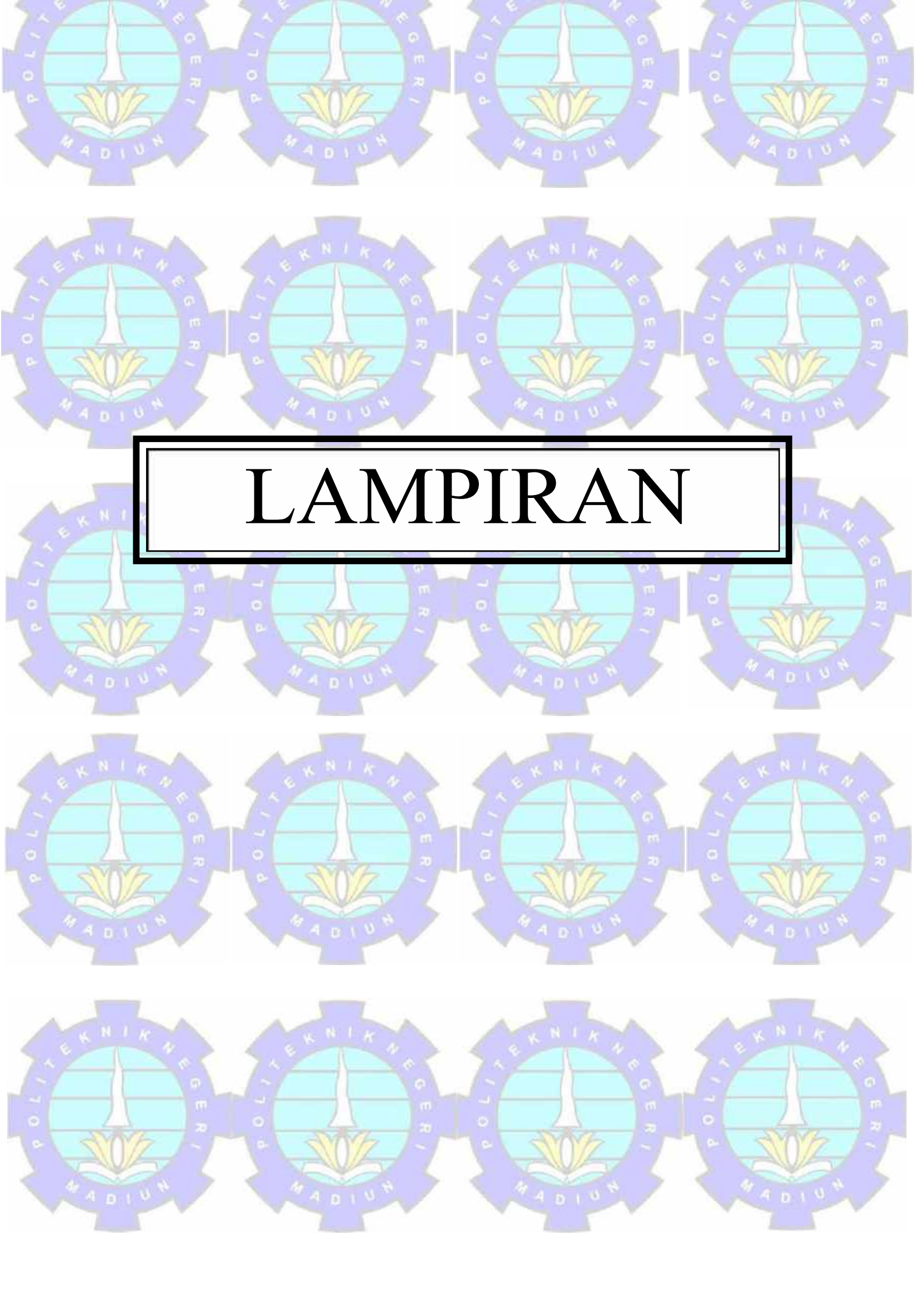
Berdasarkan dari kesimpulan yang telah didapatkan terdapat beberapa saran sebagai bagian evaluasi dari penelitian yang sudah dilakukan serta untuk memberikan referensi bagi mahasiswa atau penelitian yang sejenis.

1. Melakukan riset secara matang, sehingga komponen yang digunakan sesuai dengan apa yang diimplementasikan. Serta menggunakan komponen dengan tegangan *alternate current (AC)* supaya perangkat keras bisa tahan awet lebih lama.
2. Mengkaji lebih lanjut cara pengiriman hasil *capture* gambar, sehingga waktu pengiriman tidak membutuhkan waktu yang lama dan meminimalisir terjadinya *error*
3. Menambahkan sensor cahaya di setiap kotak lampu lalu lintas, sehingga ketika terjadi kerusakan pada setiap lampu dapat terbaca dengan akurat tidak hanya terbaca dengan sensor tegangan.

DAFTAR PUSTAKA

- AJI, R. S. G. (2021). Rancang Bangun Sistem Kontrol Lampu Lalu Lintas Berbasis *Artificial Intelligence*.
- Basil, E., & Sawant, S. D. (2017). *Internet of Things (IoT) based traffic Light control system using Raspberry Pi*. 2017 International Conference on Energy, Communication, Data Analytics and Soft Computing, ICECDS 2017, 1078–1081. <https://doi.org/10.1109/ICECDS.2017.8389604>
- Bayu Kusumo, F. salam. (2023). Rancang Bangun Sistem Kontrol dan Alat Monitoring Traffic Light Menggunakan ESP8266 Berbasis *Internet of Things (IoT)*. 12(2), 1–23.
- Engel. (2019). Upaya Penanganan Kecelakaan Lalu Lintas. *Paper Knowledge . Toward a Media History of Documents*.
- Fitria, M. A. (2021). Adaptive control pada traffic *Light* untuk kasus persimpangan luar kota. *Journal of Chemical Information and Modeling*, 53(9), 1–15.
- Hidayatullah, S., Kurniawan, Y., & Istiqphara, S. (n.d.). *Adaptive System Traffic Light Using Raspberry Pi 3*.
- Lesmana, C., Lim, R., & Santoso, L. W. (2019). Implementasi *Face Recognition* menggunakan *Raspberry pi* untuk akses Ruangan Pribadi. *Jurnal Infra Petra*, 7(1), 2–5. <https://publication.petra.ac.id/index.php/teknik-informatika/article/view/8046>
- NUGROHO, A. D. (2021). Rancang Sistem Alarm Asap Rokok Ruangan Yang Termonitor Berbasis *Internet of Things (IoT)* Menggunakan ESP-32 Dengan Notifikasi SMS. In *Journal of Physics A: Mathematical and Theoretical* (Vol. 44, Issue 8). <https://doi.org/10.1088/1751-8113/44/8/085201>
- Rahmad, I., Mirzal, A., Berliyan, A. M., & Indonesia, U. P. (2023). Perancangan Prototipe Sistem Lampu Lalu Lintas Otomatis DENGAN Penjejakan Kendaraan Menggunakan *Motion Detection-Blob* dan Pengontrolan Berbasis *Fuzzy Logic Control*. 3, 1025–1036.
- Ramadhan, A. H., Kurniawan, Y., Istiqphara, S., Elektro, T., Institut, J., Sumatera, T., & Selatan, L. (n.d.). Rancang Bangun Sistem Pendeteksian Jenis Kepadatan Antrian Lalu Lintas (Lengang, Sedang, dan Padat) Pada Perangkat *Adaptive System Traffic Light (ASTRAL)* Dengan Metode *Machine Learning*. 2–5.
- Utama, R. P., & Aulia, S. (2017). Rancang Bangun Lampu Lalu Lintas Otomatis Berdasarkan Panjang Antrian Kendaraan Berbasis Pengolahan Citra Digital. *July*. <https://doi.org/10.13140/RG.2.2.18672.61447>

Veda, P. A. (2021). *Prototype Pada Lampu Lalu Lintas Berbasis SMS Gateway*.
<https://repository.uir.ac.id/14549/%0Ahttps://repository.uir.ac.id/14549/1/153510509.pdf>

The background of the page is a repeating pattern of the Politeknik Negeri Madiun logo. The logo is a circular emblem with a purple outer ring containing the text 'POLITEKNIK NEGERI' at the top and 'MADIUN' at the bottom. Inside the ring is a light blue circle with horizontal lines, a white stylized torch or flame in the center, and a yellow lotus flower at the base.

LAMPIRAN

Lampiran 1. Lembar Monitoring Bimbingan Tugas Akhir



**KEMENTERIAN PENDIDIKAN, KEBUDAYAAN,
RISET, DAN TEKNOLOGI
POLITEKNIK NEGERI MADIUN**

Jl. Ring Road Barat, Manguharjo, Kota Madiun, Kode Pos 63162
Telepon (0351)-452970, Faksimile (0351)-492960
Laman: www.pnm.ac.id Surel: sekretariat@pnm.ac.id

LEMBAR MONITORING BIMBINGAN

TUGAS AKHIR

Judul TA : Perancangan Alat Perangkat Keras Pada *Adaptive Traffic Light*
Nama Mahasiswa : Hangga Adji Novianto
NPM : 213304015
Program Studi : Teknik Komputer Kontrol
Jurusan : Teknik
Dosen Pembimbing 1 : Dahris Shahab, S.Pd., M.Pd.
Dosen Pembimbing 2 : Imam Junaedi, S.T., M.T.

No	Tanggal	Materi Konsultasi	Catatan Pembimbing	Paraf Pembimbing
1	23 Januari 2024	Konsultasi Pengerjaan Tugas Akhir	Membuat sesuai dengan hasil seminar proposal	
2	19 Juni 2024	Konsultasi mengenai Pengerjaan Laporan Tugas Akhir	Menyerjakan sesuai dengan Buku Penduan	
3	20 Juni 2024	Konsultasi Laporan Bab 3 penduan	Revisi tulisan asing di ubah italic	
4	25 Juni 2024	Konsultasi Laporan Bab 3 Penambahan sub bab Perancangan Hardware	merapikan tata tulis kata kerja - di	
5	26 Juni 2024	Konsultasi Laporan revisi Perancangan Hardware	merapikan tata tulis kalimat asing di ubah italic	

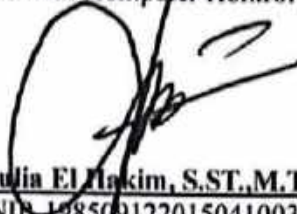
6	6 Mei 2024	Konsultasi Progres Pengerjaan tugas Akhir	membuat broker pada Raspberry Pi dan membuat database	Yj
7	02 Juli 2024	Konsultasi Laporan tugas Akhir Pembuatan Hardware	di akhir kalimat diberi tanda titik	Yj
8	28 Mei 2024	Konsultasi terkait Model perancangan Hardware	mengimplementasikan ke dalam box panel	Yj
9	03 Juli 2024	Konsultasi terkait Laporan tugas Akhir pengujian Bab 3	halaman Pengesahan diberikan tipis, bab 1 poin 1.1 ada bbk warna hitam, mengakh	Yj
10	14 Juni 2024	Mengampaikan Progres pengerjaan Hardware	Melakukan uji coba menyala pada komponen	Yj
11	19 Juni 2024	Konsultasi terkait Pengkabelan pada box panel	Merupakan pengkabelan mengetes ulang Hardware	Yj
12	28 Juni 2024	Konsultasi terkait komponen yang tidak bisa trigger arus AC	menyentuh komponen ssr dengan relay module supaya bisa men-trigger lampu	Yj
13	03 Juli 2024	Konsultasi terkait Laporan tugas Akhir Bab 4 Hasil dan pembahasan	Singkatan istilah huruf kapital	Yj
14	02 Juli 2024	Konsultasi terkait Laporan Bab 3	Menambahkan hasil Pengujian pada box panel	Yj

15	3 Juli 2024	Konsultasi terkait laporan	melanjutkan Bab 4	
16	03 Juli 2024	Konsultasi terkait Laporan tugas Akhir Bab 4 Hasil dan pembahasan.	Daftar isi error, dibenahi melanjutkan Bab 5	
17	09 Juli 2024	Konsultasi terkait Laporan tugas Akhir Bab 5	Memambahkan Hasil dan pembahasan	
18	10 Juli 2024	Konsultasi terkait abstrak dan tanda tangan lembar Persewaan	menambahkan abstrak dan daftar isi	
19				
20				

Pembimbing I

**Dahris Shahab, S.Pd., M.Pd.**
NIDN. 0717016902Madiun, 08... Juli 2024
Pembimbing II**Imam Junaedi, S.T., M.T.**
NIDN. 0025099009

Mengetahui,
Koordinator Program Studi
Teknik Komputer Kontrol



Aulia El Hakim, S.ST., M.T.
NIP. 198509122015041003

Lampiran 2. Form Revisi Sidang Tugas Akhir



KEMENTERIAN PENDIDIKAN, KEBUDAYAAN,
RISET, DAN TEKNOLOGI
POLITEKNIK NEGERI MADIUN
JURUSAN TEKNIK

Jl. Ring Road Barat, Manguharjo, Kota Madiun, Kode Pos 63162
Telepon (0351) 452970, Faksimile (0351) 452960
Laman: www.pnm.ac.id Surel: sekretariat@pnm.ac.id

FORM REVISI TUGAS AKHIR

Nama Mahasiswa : Hangga Adji Novianto
NPM / Kelas : 213304015 / 6A
Program Studi / Jurusan : Teknik Komputer Kontrol / Teknik
Penguji I :
Penguji II :
Penguji III :
Judul : Perancangan Alat Perangkat Keras Pada *Adaptive Traffic Light*

NO	Uraian Catatan / Revisi
1	Daftar Tabel 4.1 = awal baris-2 luas awal baris-1.
2.	Daftar gambar = istilah asing Italia.
3	Bab II 2.1.2 = setiap objek kata huruf kapital.
4	Tabel 4.1 = huruf super kecil, nomor-14 jadi halaman.
5	3.6.2 = poin-2 = spasi.
6	4.2.1 = cek ejaan.
7	4.3.2 = proses desain, dll. di atas/di luar.
8.	kecil, maka / padam, sehingga ...

Madiun, 18 Juli 2024

Penguji I

Dahrus Sholah, S.Pd., M.Pd.

NIDN. 0717016982



KEMENTERIAN PENDIDIKAN, KEBUDAYAAN,
RISET, DAN TEKNOLOGI
POLITEKNIK NEGERI MADIUN
JURUSAN TEKNIK

Jl. Ring Road Barat, Manguharjo, Kota Madiun, Kode Pos 63162
Telepon (0351) 452970, Faksimile (0351) 452960
Laman: www.pnm.ac.id Surel: sekretariat@pnm.ac.id

FORM REVISI TUGAS AKHIR

Nama Mahasiswa : Hangga Adji Novianto
NPM / Kelas : 213304015 / 6A
Program Studi / Jurusan : Teknik Komputer Kontrol / Teknik
Penguji I : Dahris Shahab, S.Pd., M.Pd.
Penguji II : Dirvi Eko Juliando Sudirman, S.Pd., M.Pd.
Penguji III : Rendi Pambudi Wicaksono, S.ST., M.Tr.T.
Judul : Perancangan Alat Perangkat Keras Pada *Adaptive Traffic Light*

NO	Uraian Catatan / Revisi
1.	Kalibrasi Sensor Tegangan
2.	Bandingkan nilai Pembacaan Sensor Tegangan
3.	Perbaiki delay Pengiriman data Sensor Tegangan

Madiun, 18 Juli 2024
Penguji 3

Rendi Pambudi Wicaksono, S.ST., M.Tr.T
NIDN. 0003069307

Lampiran 3. Lembar Pernyataan Progres Pengerjaan Tugas Akhir



KEMENTERIAN PENDIDIKAN, KEBUDAYAAN,
RISET, DAN TEKNOLOGI
POLITEKNIK NEGERI MADIUN
JURUSAN TEKNIK

Jl. Ring Road Barat, Manguharjo, Kota Madiun, Kode Pos 63162
Telepon (0351) 452970, Faksimile (0351) 452960
Laman: www.pnm.ac.id Surel: sekretariat@pnm.ac.id

SURAT PERNYATAAN

Yang bertanda tangan di bawah ini:

Nama : Dahris Shabab, S.Pd., M.Pd.
Jabatan : Dosen Pembimbing I
NIDN : 0717016902

Dengan ini menerangkan bahwa mahasiswa berikut ini :

Nama : Hangga Adji Novianto
NPM : 213304015
Judul Tugas Akhir : Perancangan Alat Perangkat Keras Pada *Adaptive Traffic Light*

Bahwasannya proses pengerjaan Tugas Akhir berupa alat, sistem dan Laporan Tugas Akhir mahasiswa tersebut telah mencapai 85%.

Demikian pernyataan ini dibuat, semoga dapat digunakan sebagai mestinya.

Madiun, 08 Juli 2024

Pembimbing I

Dahris Shabab, S.Pd., M.Pd.
NIDN. 0717016902



KEMENTERIAN PENDIDIKAN, KEBUDAYAAN,
RISET, DAN TEKNOLOGI
POLITEKNIK NEGERI MADIUN
JURUSAN TEKNIK

Jl. Ring Road Barat, Manguharjo, Kota Madiun, Kode Pos 63162
Telepon (0351) 452970, Faksimile (0351) 452960
Laman: www.pnm.ac.id Surel: sekretariat@pnm.ac.id

SURAT PERNYATAAN

Yang bertanda tangan di bawah ini:

Nama : Imam Junaedi, S.T., M.T.
Jabatan : Dosen Pembimbing 2
NIDN : 0025099009

Dengan ini menerangkan bahwa mahasiswa berikut ini :

Nama : Hangga Adji Novianto
NPM : 213304015
Judul Tugas Akhir : Perancangan Alat Perangkat Keras Pada *Adaptive Traffic Light*

Bahwasannya proses pengerjaan Tugas Akhir berupa alat, sistem dan Laporan Tugas Akhir mahasiswa tersebut telah mencapai 85%.

Demikian pernyataan ini dibuat, semoga dapat digunakan sebagai mestinya.

Madiun, 08 Juli 2024

Pembimbing 2

Imam Junaedi, S.T., M.T.

NIDN. 0025099009

Lampiran 4. Lampiran Program/Aplikasi/Rangkaian *Hardware*

1. Kode Program *ESP32* Kontrol Lampu Lalu Lintas

```
#include <WiFi.h>
#include <PubSubClient.h>
#include <SPI.h>
#include <Wire.h>
#include "RTCLib.h"
#include <AsyncTCP.h>
#include <SPIFFS.h>
#include <ESPAsyncWebServer.h>
#include <ArduinoJson.h>

// Pin Definisi untuk Program 1 (adaptive) dan Program 2 (default)
#define tl1_hijau 27
#define tl1_kuning 14
#define tl1_merah 12
#define tl2_hijau 26
#define tl2_kuning 25
#define tl2_merah 33
#define tl3_hijau 32
#define tl3_kuning 4
#define tl3_merah 2
#define lampuKedip 15

// Konstanta Waktu untuk Program 1
const int hijau_Default = 15;
const int kuning_Default = 5;
const int merah_Default = hijau_Default + kuning_Default;
int hijau_Durasi = hijau_Default;
int kuning_Durasi = kuning_Default;
int merah_Durasi = merah_Default;

// WiFi dan MQTT untuk Program 1 (adaptive)
const char* ssid = "YOUR_SSID";
const char* password = "YOUR_PASSWORD";
const char* mqtt_server = "YOUR_MQTT_SERVER";
WiFiClient espClient;
PubSubClient client(espClient);
long lastMsg = 0;
char msg[50];
int value = 0;

bool adaptiveMode = true;

void setup_wifi() {
    delay(10);
    Serial.println();
    Serial.print("Connecting to ");
    Serial.println(ssid);
    WiFi.begin(ssid, 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() {
        while (!client.connected()) {
            Serial.print("Attempting MQTT connection...");
            if (client.connect("ESP32Client")) {
                Serial.println("connected");
                client.subscribe("traffic/control");
            } else {
                Serial.print("failed, rc=");
                Serial.print(client.state());
                Serial.println(" try again in 5 seconds");
                delay(5000);
            }
        }
    }

    void callback(char* topic, byte* payload, unsigned int length) {
        Serial.print("Message arrived [");
        Serial.print(topic);
        Serial.print("] ");
        String message = "";
        for (unsigned int i = 0; i < length; i++) {
            message += (char)payload[i];
        }
        Serial.println(message);
        hijau_Durasi = message.toInt();
        merah_Durasi = hijau_Durasi + kuning_Durasi;
    }

    void smart_delay(int ms) {
        long start = millis();
        while (millis() - start < ms) {
            client.loop();
        }
    }

    void adaptive() {
        if (!client.connected()) {
            reconnect();
        }
        client.loop();

        // Siklus lampu lalu lintas untuk adaptive
        digitalWrite(tl1_hijau, HIGH);
        digitalWrite(tl2_merah, HIGH);
        digitalWrite(tl3_merah, HIGH);
        client.publish("traffic/status", "TL1 Hijau");
        smart_delay(hijau_Durasi * 1000);
        digitalWrite(tl1_hijau, LOW);

        digitalWrite(tl1_kuning, HIGH);
        client.publish("traffic/status", "TL1 Kuning");
        smart_delay(kuning_Durasi * 1000);
        digitalWrite(tl1_kuning, LOW);

        digitalWrite(tl1_merah, HIGH);
        digitalWrite(tl2_hijau, HIGH);
    }

```

```

    client.publish("traffic/status", "TL2 Hijau");
    smart_delay(hijau_Durasi * 1000);
    digitalWrite(tl2_hijau, LOW);

    digitalWrite(tl2_kuning, HIGH);
    client.publish("traffic/status", "TL2 Kuning");
    smart_delay(kuning_Durasi * 1000);
    digitalWrite(tl2_kuning, LOW);

    digitalWrite(tl2_merah, HIGH);
    digitalWrite(tl3_hijau, HIGH);
    client.publish("traffic/status", "TL3 Hijau");
    smart_delay(hijau_Durasi * 1000);
    digitalWrite(tl3_hijau, LOW);

    digitalWrite(tl3_kuning, HIGH);
    client.publish("traffic/status", "TL3 Kuning");
    smart_delay(kuning_Durasi * 1000);
    digitalWrite(tl3_kuning, LOW);
}

// Program 2 (default) Konfigurasi WiFi dan Web Server
const char* ssid2 = "YOUR_SSID";
const char* password2 = "YOUR_PASSWORD";
const char* http_username = "admin";
const char* http_password = "admin";

RTC_DS1307 rtc;
AsyncWebServer server(80);
AsyncWebSocket ws("/ws");

String readFile(fs::FS &fs, const char * path) {
    File file = fs.open(path, "r");
    if (!file || file.isDirectory()) {
        return String();
    }
    String fileContent;
    while (file.available()) {
        fileContent += String((char)file.read());
    }
    file.close();
    return fileContent;
}

void writeFile(fs::FS &fs, const char * path, const char *
message) {
    File file = fs.open(path, FILE_WRITE);
    if (!file) {
        return;
    }
    file.print(message);
    file.close();
}

String processor(const String& var) {
    if (var == "TL1H") return readFile(SPIFFS, "/tl1h.txt");
    if (var == "TL1K") return readFile(SPIFFS, "/tl1k.txt");
    if (var == "TL2H") return readFile(SPIFFS, "/tl2h.txt");
    if (var == "TL2K") return readFile(SPIFFS, "/tl2k.txt");
}

```

```

        if (var == "TL3H") return readFile(SPIFFS, "/tl3h.txt");
        if (var == "TL3K") return readFile(SPIFFS, "/tl3k.txt");
        return String();
    }

    void notFound(AsyncWebServerRequest *request) {
        request->send(404, "text/plain", "Not found");
    }

    void handleWebSocketMessage(void *arg, uint8_t *data, size_t len)
    {
        AwsFrameInfo info = (AwsFrameInfo)arg;
        if (info->final && info->index == 0 && info->len == len &&
            info->opcode == WS_TEXT) {
            data[len] = 0;
            String message = (char*)data;
            if (message == "ON") {
                adaptiveMode = true;
            } else if (message == "OFF") {
                adaptiveMode = false;
            }
        }
    }

    void onEvent(AsyncWebSocket *server, AsyncWebSocketClient
        *client, AwsEventType type, void *arg, uint8_t *data, size_t len)
    {
        if (type == WS_EVT_CONNECT) {
            Serial.println("WebSocket client connected");
        } else if (type == WS_EVT_DISCONNECT) {
            Serial.println("WebSocket client disconnected");
        } else if (type == WS_EVT_DATA) {
            handleWebSocketMessage(arg, data, len);
        }
    }

    void defaultMode() {
        int durasi_tl1h = readFile(SPIFFS, "/tl1h.txt").toInt();
        int durasi_tl1k = readFile(SPIFFS, "/tl1k.txt").toInt();
        int durasi_tl2h = readFile(SPIFFS, "/tl2h.txt").toInt();
        int durasi_tl2k = readFile(SPIFFS, "/tl2k.txt").toInt();
        int durasi_tl3h = readFile(SPIFFS, "/tl3h.txt").toInt();
        int durasi_tl3k = readFile(SPIFFS, "/tl3k.txt").toInt();

        DateTime now = rtc.now();
        Serial.printf("%02d-%02d-%02d %02d:%02d:%02d\n", now.year(),
            now.month(), now.day(), now.hour(), now.minute(), now.second());

        digitalWrite(tl1_hijau, HIGH);
        digitalWrite(tl2_merah, HIGH);
        digitalWrite(tl3_merah, HIGH);
        delay(durasi_tl1h * 1000);
        digitalWrite(tl1_hijau, LOW);
        digitalWrite(tl1_kuning, HIGH);
        delay(durasi_tl1k * 1000);
        digitalWrite(tl1_kuning, LOW);
        digitalWrite(tl1_merah, HIGH);

        digitalWrite(tl2_hijau, HIGH);

```

```

    delay(durasi_tl2h * 1000);
    digitalWrite(tl2_hijau, LOW);
    digitalWrite(tl2_kuning, HIGH);
    delay(durasi_tl2k * 1000);
    digitalWrite(tl2_kuning, LOW);
    digitalWrite(tl2_merah, HIGH);

    digitalWrite(tl3_hijau, HIGH);
    delay(durasi_tl3h * 1000);
    digitalWrite(tl3_hijau, LOW);
    digitalWrite(tl3_kuning, HIGH);
    delay(durasi_tl3k * 1000);
    digitalWrite(tl3_kuning, LOW);
}

void setup() {
    Serial.begin(115200);
    pinMode(tl1_hijau, OUTPUT);
    pinMode(tl1_kuning, OUTPUT);
    pinMode(tl1_merah, OUTPUT);
    pinMode(tl2_hijau, OUTPUT);
    pinMode(tl2_kuning, OUTPUT);
    pinMode(tl2_merah, OUTPUT);
    pinMode(tl3_hijau, OUTPUT);
    pinMode(tl3_kuning, OUTPUT);
    pinMode(tl3_merah, OUTPUT);

    setup_wifi();
    client.setServer(mqtt_server, 1883);
    client.setCallback(callback);

    if (!SPIFFS.begin(true)) {
        Serial.println("An Error has occurred while mounting
SPIFFS");
        return;
    }

    if (!rtc.begin()) {
        Serial.println("Couldn't find RTC");
        while (1);
    }

    if (!rtc.isrunning()) {
        Serial.println("RTC is NOT running!");
        rtc.adjust(DateTime(F(_DATE), F(_TIME)));
    }

    server.on("/", HTTP_GET, [] (AsyncWebServerRequest *request) {
        request->send(SPIFFS, "/index.html", String(), false,
processor);
    });

    server.on("/update", HTTP_POST, [] (AsyncWebServerRequest
*request) {
        int params = request->params();
        for (int i = 0; i < params; i++) {
            AsyncWebParameter* p = request->getParam(i);
            if (p->isPost()) {
                if (p->name() == "tl1h") {

```

```

        writeFile(SPIFFS,      "/tl1h.txt",      p-
>value().c_str());
        } else if (p->name() == "tl1k") {
        writeFile(SPIFFS,      "/tl1k.txt",      p-
>value().c_str());
        } else if (p->name() == "tl2h") {
        writeFile(SPIFFS,      "/tl2h.txt",      p-
>value().c_str());
        } else if (p->name() == "tl2k") {
        writeFile(SPIFFS,      "/tl2k.txt",      p-
>value().c_str());
        } else if (p->name() == "tl3h") {
        writeFile(SPIFFS,      "/tl3h.txt",      p-
>value().c_str());
        } else if (p->name() == "tl3k") {
        writeFile(SPIFFS,      "/tl3k.txt",      p-
>value().c_str());
        }
    }
    }
    request->send(200, "text/plain", "Updated");
});

server.onNotFound(notFound);

ws.onEvent(onEvent);
server.addHandler(&ws);

server.begin();
}

void loop() {
    if (adaptiveMode) {
        adaptive();
    } else {
        defaultMode();
    }
    delay(100);
}

```

2. Kode Program *ESP32* Sensor Tegangan

```

#include <WiFi.h>
#include <PubSubClient.h>
#include <HTTPClient.h>

const char* serverName = "http://10.80.0.1/save_data.php";

// Replace the next variables with your SSID/Password
combination
const char* ssid = "adaptive";
const char* password = "sidavta123";

// Add your MQTT Broker IP address, example:
const char* mqtt_server = "10.80.0.1";

WiFiClient espClient;

```

```

PubSubClient client(espClient);
long lastMsg = 0;
char msg[50];
int value = 0;

// Set your Static IP address
IPAddress local_IP(10, 80, 0, 113);
// Set your Gateway IP address
IPAddress gateway(10, 80, 0, 1);

IPAddress subnet(255, 255, 255, 0);
//IPAddress primaryDNS(8, 8, 8, 8); //optional
//IPAddress secondaryDNS(8, 8, 4, 4); //optional

// Definisikan pin-pin untuk ESP32
const int muxS0 = 21; // Pin ESP32 untuk S0
const int muxS1 = 22; // Pin ESP32 untuk S1
const int muxS2 = 23; // Pin ESP32 untuk S2
const int muxS3 = 19; // Pin ESP32 untuk S3
const int muxSig = 32; // Pin ESP32 untuk S16 (Output dari Multiplexer)

// Fungsi untuk mengatur alamat channel pada multiplexer
void setMuxChannel(int channel) {
    // Tentukan nilai biner untuk channel
    digitalWrite(muxS0, bitRead(channel, 0));
    digitalWrite(muxS1, bitRead(channel, 1));
    digitalWrite(muxS2, bitRead(channel, 2));
    digitalWrite(muxS3, bitRead(channel, 3));
}

float readChannel(int channel) {
    setMuxChannel(channel);
    delay(10); // Tunggu sejenak agar nilai stabil
    int adcValue = analogRead(muxSig);
    // Konversi nilai ADC ke volt, disesuaikan dengan rumus
    float result = (float)adcValue / 4096 * 15 * 32350 / 27000;
    return result;
}

void setup() {
    Serial.begin(115200);
    // Configures static IP address , primaryDNS, secondaryDNS
    if (!WiFi.config(local_IP, gateway, subnet)) {
        Serial.println("STA Failed to configure");
    }
    setup_wifi();
    client.setServer(mqtt_server, 1883);
    client.setCallback(callback);
    // Inisialisasi pin-pin sebagai output
    pinMode(muxS0, OUTPUT);
    pinMode(muxS1, OUTPUT);
    pinMode(muxS2, OUTPUT);
    pinMode(muxS3, OUTPUT);
    // Inisialisasi pin untuk input analog
    pinMode(muxSig, INPUT_PULLUP);
}

void setup_wifi() {
    delay(10);

```

```

// We start by connecting to a WiFi network
Serial.println();
Serial.print("Connecting to ");
Serial.println(ssid);

WiFi.begin(ssid, 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 callback(char* topic, byte* message, unsigned int length) {
    Serial.print("Message arrived on topic: ");
    Serial.print(topic);
    Serial.print(". Message: ");
    String messageTemp;

    for (int i = 0; i < length; i++) {
        Serial.print((char)message[i]);
        messageTemp += (char)message[i];
    }
    Serial.println();
}

void reconnect() {
    // Loop until we're reconnected
    while (!client.connected()) {
        Serial.print("Attempting MQTT connection...");
        // Attempt to connect
        if (client.connect("ESP32Client")) {
            Serial.println("connected");
            // Subscribe
            client.subscribe("esp32/output");
        } else {
            Serial.print("failed, rc=");
            Serial.print(client.state());
            Serial.println(" try again in 5 seconds");
            // Wait 5 seconds before retrying
            delay(5000);
        }
    }
}

void sendData(String data) {
    if(WiFi.status() == WL_CONNECTED) {
        HTTPClient http;

        // Buat URL lengkap dengan data
        String postData = "data=" + data;

        http.begin(serverName);

```

```

    http.addHeader("Content-Type", "application/x-www-form-
urlencoded");

    // Kirim permintaan POST
    int httpResponseCode = http.POST(postData);

    if (httpResponseCode > 0) {
        String response = http.getString();
        Serial.println(httpResponseCode);
        Serial.println(response);
    } else {
        Serial.print("Error on sending POST: ");
        Serial.println(httpResponseCode);
    }

    http.end();
} else {
    Serial.println("Error in WiFi connection");
}
}

void loop() {
    if (!client.connected()) {
        reconnect();
    }
    client.loop();

    float Merah1 = readChannel(15);
    float Kuning1 = readChannel(14);
    float Hijau1 = readChannel(13);
    float Merah2 = readChannel(12);
    float Kuning2 = readChannel(11);
    float Hijau2 = readChannel(10);
    float Merah3 = readChannel(9);
    float Kuning3 = readChannel(8);
    float Hijau3 = readChannel(6);

    Serial.print("Red 1: ");
    Serial.print(Merah1);
    if(Merah1 < 7){
        Serial.print("V (OFF) || ");
        client.publish("lalin/statuslampu/red1", "OFF");
        String data = "Lampu Merah fase 1 Rusak";
        sendData(data);
    } else {
        Serial.print("V (ON) || ");
        client.publish("lalin/statuslampu/red1", "ON");
    }

    Serial.print("Yellow 1: ");
    Serial.print(Kuning1);
    if(Kuning1 < 7){
        Serial.print("V (OFF) || ");
        client.publish("lalin/statuslampu/yellow1", "OFF");
        String data = "Lampu Kuning fase 1 Rusak";
        sendData(data);
    } else {
        Serial.print("V (ON) || ");
        client.publish("lalin/statuslampu/yellow1", "ON");
    }
}

```

```

}

Serial.print("Green 1: ");
Serial.print(Hijau1);
if(Hijau1 < 7){
    Serial.print("V (OFF) || ");
    client.publish("lalin/statuslampu/green1", "OFF");
    String data = "Lampu Hijau fase 1 Rusak";
    sendData(data);
} else {
    Serial.print("V (ON) || ");
    client.publish("lalin/statuslampu/green1", "ON");
}

Serial.print("Red 2: ");
Serial.print(Merah2);
if(Merah2 < 7){
    Serial.print("V (OFF) || ");
    client.publish("lalin/statuslampu/red2", "OFF");
    String data = "Lampu Merah fase 2 Rusak";
    sendData(data);
} else {
    Serial.print("V (ON) || ");
    client.publish("lalin/statuslampu/red2", "ON");
}

Serial.print("Yellow 2: ");
Serial.print(Kuning2);
if(Kuning2 < 7){
    Serial.print("V (OFF) || ");
    client.publish("lalin/statuslampu/yellow2", "OFF");
    String data = "Lampu Kuning fase 2 Rusak";
    sendData(data);
} else {
    Serial.print("V (ON) || ");
    client.publish("lalin/statuslampu/yellow2", "ON");
}

Serial.print("Green 2: ");
Serial.print(Hijau2);
if(Hijau2 < 7){
    Serial.print("V (OFF) || ");
    client.publish("lalin/statuslampu/green2", "OFF");
    String data = "Lampu Hijau fase 2 Rusak";
    sendData(data);
} else {
    Serial.print("V (ON) || ");
    client.publish("lalin/statuslampu/green2", "ON");
}

Serial.print("Red 3: ");
Serial.print(Merah3);
if(Merah3 < 7){
    Serial.print("V (OFF) || ");
    client.publish("lalin/statuslampu/red3", "OFF");
    String data = "Lampu Merah fase 3 Rusak";
    sendData(data);
} else {
    Serial.print("V (ON) || ");

```

```

        client.publish("lalin/statuslampu/red3", "ON");
    }

    Serial.print("Yellow 3: ");
    Serial.print(Kuning3);
    if(Kuning3 < 7){
        Serial.print("V (OFF) || ");
        client.publish("lalin/statuslampu/yellow3", "OFF");
        String data = "Lampu Kuning fase 3 Rusak";
        sendData(data);
    } else {
        Serial.print("V (ON) || ");
        client.publish("lalin/statuslampu/yellow3", "ON");
    }

    Serial.print("Green 3: ");
    Serial.print(Hijau3);
    if(Hijau3 < 7){
        Serial.println("V (OFF)");
        client.publish("lalin/statuslampu/green3", "OFF");
        String data = "Lampu Hijau fase 3 Rusak";
        sendData(data);
    } else {
        Serial.println("V (ON)");
        client.publish("lalin/statuslampu/green3", "ON");
    }

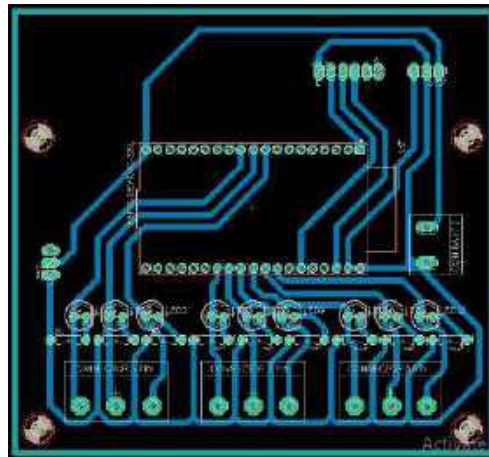
    delay(100); // Tunggu 1 detik sebelum memulai siklus berikutnya
}

```

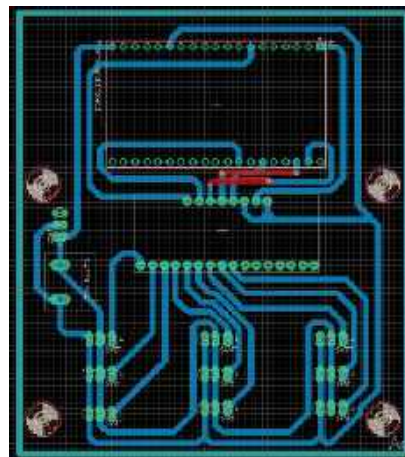
3. Desain Box Panel



4. Desain *PCB* Kontrol Lampu Lalu Lintas



5. Desain *PCB* Sensor Tegangan



Lampiran 5. Biodata Penulis

**BIODATA PENULIS**

Nama : Hangga Adji Novianto
Tempat, Tanggal Lahir : Magetan, 03 November 2002
NPM : 213304015
Jurusan : Teknik
Program Studi : Teknik Komputer Kontrol
Alamat : Jl.Branjangan No.21 Lanud Iswahjudi Rt.02, Rw.01,
Kel.Maospati, Kec.Maospati, Kab.Magetan
Email : hanggaadjinovianto@gmail.com
Nomor Handphone : 08817106408



Madiun, 08 Juli 2024
Penulis

Hangga Adji Novianto
NPM. 213304015

Lampiran 6 Motto dan Persembahan

MOTTO

“Only you can change your life, nobody else do it for you”

Orang lain gak akan bisa paham struggle dan masa sulitnya kita yang mereka ingin tau hanya bagian success stories. Berjuanglah untuk diri sendiri walaupun gak ada yang tepuk tangan. Kelak diri kita di masa depan akan sangat bangga dengan apa yang kita perjuangkan hari ini, tetap berjuang ya

PERSEMBAHAN

Dengan mengucap Alhamdulillahirabbil'alamiin, puji dan syukur kepada Allah Subhanahu Wa Ta'ala atas rahmat dan ridho-Nya saya dapat menyelesaikan Tugas Akhir ini dengan tepat waktu. Tugas akhir ini dapat terselesaikan berkat bantuan, bimbingan dan saran-saran serta masukan dari berbagai pihak. Oleh karena itu pada kesempatan ini, saya ingin mengucapkan terimakasih kepada :

1. Diri saya sendiri Hangga Adji Novianto yang selalu semangat, terus berusaha dan pantang menyerah untuk dapat menyelesaikan Tugas Akhir ini.
2. Ibu saya serta keluarga yang selalu memberikan motivasi, doa, serta semangat dalam menyelesaikan Laporan Tugas Akhir.
3. Kedua Dosen Pembimbing Bapak Dahris Shahab, S.Pd., M.Pd. selaku dosen pembimbing I dan Bapak Imam Junaedi, S.T.,M.T. selaku dosen pembimbing II yang telah meluangkan waktunya untuk membimbing dan memberikan saran selama pengerjaan Tugas Akhir.
4. Partner dalam Tugas Akhir, Renata Tristiana dan Alvindra Putra Ardhana, yang telah berjuang bersama dan saling membantu untuk dapat menyelesaikan Tugas Akhir.
5. Teman-teman Teknik Komputer Kontrol angkatan 2021 yang saling memberi dukungan dan semangat.
6. Dinda Aenun Fatkha yang telah memberikan doa, semangat, dan motivasi untuk dapat menyelesaikan Tugas Akhir
7. Dan terimakasih kepada pihak yang belum saya sebutkan, yang selalu memberikan motivasi, semangat, dukungan tanpa henti sehingga secara tidak langsung membantu saya dalam menyelesaikan Tugas Akhir ini

