

**Sistem Simulasi Kendali *Traffic Light* dengan Metode Pendeteksian
Gambar secara *Realtime* melalui Kamera Berbasis IoT**



TUGAS AKHIR

Disusun oleh :

Nama : Agus Maulana
Nomor Pokok : 111.142.0003
Peminatan : Teknik Elektronika Industri

**INSTITUT TEKNOLOGI INDONESIA
PROGRAM STUDI TEKNIK ELEKTRO
SERPONG**

2019

**INSTITUT TEKNOLOGI INDONESIA
PROGRAM STUDI TEKNIK ELEKTRO**

**Sistem Simulasi Kendali *Traffic Light* dengan Metode Pendeteksian
Gambar secara *Realtime* melalui Kamera Berbasis IoT**

TUGAS AKHIR

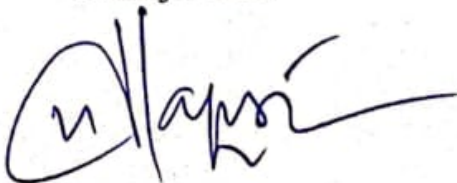
Disusun oleh :

Nama : Agus Maulana
Nomor Pokok : 111.142.0003
Peminatan : Teknik Elektronika Industri

Tugas Akhir diajukan untuk Memenuhi
Persyaratan Kurikulum Sarjana Strata Satu (S-1)
Program Studi Teknik Elektro
Institut Teknologi Indonesia
Serpong

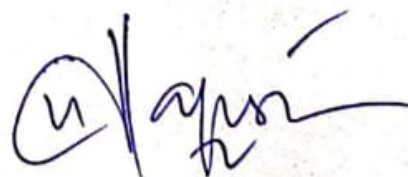
Serpong,

Disetujui Oleh:



Novy Hapsari, ST, M.Sc
Dosen Pembimbing

Diketahui Oleh:



Novy Hapsari, ST, M.Sc
Ketua Program Studi Teknik Elektro

ABSTRAK

Sistem Simulasi Kendali Traffic Light dengan Metode Pendeteksian Gambar secara Realtime melalui Kamera Berbasis IoT merupakan suatu *system control* perkotaan terpadu yang tengah dikembangkan dalam ruang lingkup *smart city*. Berbekal sebuah kendali traffic light yang masih menggunakan metode manual, sistem ini dikembangkan menjadi sebuah sistem kendali yang dapat dikontrol dari jarak jauh untuk dapat memonitor dan mengontrol sebuah *traffic light* dengan berbasis IOT (*Internet Of Things*). Dalam sistem ini, setiap *Traffic Light* menggunakan komponen modul mikrokontroler NodeMCU Lua ESP8266 Wifi dimana modul ini dapat dikendalikan dari pusat *control* untuk menerima data informasi yang telah diolah oleh Open CV sebagai perintah untuk mengatur kondisi lampu lalu lintas. Data tersebut didapat dari sebuah kamera pengontrol Traffic Light yang berbasis modul ESP8266 Wifi (mini) yang kemudian mengirimkan data informasi kepada ruang control berbasis web. Data informasi tersebut diolah sedemikian rupa yang dimana menggambarkan satu kondisi lalu lintas (macet/lancar/normal) agar dapat diolah oleh *software* dan dikirim pada tiap-tiap cabang *Traffic Light*.

Kata kunci : *Internet of Things, Traffic Lights, Orange Pi Zero, NodeMCU Lua ESP8266 Wifi, Smart City*

KATA PENGANTAR

Puji dan syukur penulis panjatkan kepada Allah SWT yang telah melimpahkan karunia, rahmat dan hidayah-Nya sehingga penulis dapat menyusun dan menyelesaikan makalah Tugas Akhir yang berjudul “Sistem Simulasi Kendali *Traffic Light* dengan Metode Pendeteksian Gambar secara *Realtime* melalui Kamera Berbasis IoT”.

Selama proses penyusunan dan pembuatan makalah Tugas Akhir ini, penulis banyak mengalami kesulitan dalam pembuatannya. Oleh karena itu, penulis ingin berterima kasih kepada:

1. Kedua orangtua tercinta, yang selalu mendoakan, memberikan semangat dan dukungan baik materi maupun non – materi.
2. Ibu Novy Hapsari, ST, M.Sc selaku Ketua Prodi Teknik Elektro, Institut Teknologi Indonesia dan dosen pembimbing yang banyak memberikan ide-ide dan materi yang sangat baik dalam pembuatan makalah ini.
3. Ibu Ir. Tita Aisyah, MT selaku Koordinator Tugas Akhir dan Engineering Project.
4. Ibu Tris Dewi Indraswati, ST, MT sebagai Penasihat Akademik angkatan 2014 atas segala motivasi yang telah diberikan sehingga penulis mampu menyelesaikan makalah ini.
5. Teman-teman seperjuangan dalam penyusunan makalah Tugas Akhir yang sering berbagi ilmu dan pendapat.
6. Semua pihak terkait yang telah membantu dalam penyusunan makalah Tugas Akhir ini.

Penulis menyadari bahwa dalam penulisan makalah ini masih banyak kekurangan, maka dari itu penulis sangat mengharapkan adanya kritik dan saran yang membangun dari semua pihak. Semoga makalah ini dapat memberikan pengetahuan tambahan dan dapat diimplementasikan secara nyata untuk kepentingan bersama.

Tangerang Selatan, September 2019

Penulis

DAFTAR ISI

COVER	i
HALAMAN PENGESAHAN	ii
ABSTRAK.....	iii
KATA PENGANTAR.....	iii
DAFTAR ISI.....	v
DAFTAR GAMBAR	ix
DAFTAR TABEL.....	xi
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Tujuan.....	2
1.3 Rumusan Masalah.....	2
1.4 Batasan Masalah	3
1.5 Metode Pengumpulan Data	3
1.6 Sistematika Penulisan.....	3
BAB II TINJAUAN PUSTAKA	5
2.1 IoT (Internet of Things).....	5
2.2 Smart City.....	6
2.3 Lampu Lalu Lintas	6
2.3.1 Karakteristik Sinyal Lalu Lintas	7
2.3.2 Perhitungan Siklus Lampu Lalu Lintas	9
2.4 Apache Web Server.....	10
2.5 PHP	11
2.6 Python.....	11
2.7 OpenCV.....	12
2.8 Arduino IDE	13
2.9 Orange Pi Zero.....	14
2.10 NodeMCU.....	16
2.11 Dot Matrix Display	17
2.12 Logitech C270.....	17
2.13 Konsep Pendeteksian Gambar dengan Kamera	18

4.2.1	Pengujian Tegangan Output.....	50
4.2.1.1	Tujuan.....	50
4.2.1.2	Cara Pengujian.....	50
4.2.1.3	Hasil Data.....	51
4.2.1.4	Analisa dan Pembahasan.....	52
4.2.2	Pengujian Tegangan Regulator.....	53
4.2.2.1	Tujuan.....	53
4.2.2.2	Cara Pengujian.....	53
4.2.2.3	Hasil Data.....	53
4.2.2.4	Analisa dan Pembahasan.....	54
4.2.3	Pengujian Tegangan LED.....	54
4.2.3.1	Tujuan.....	54
4.2.3.2	Cara Pengujian.....	55
4.2.3.3	Hasil Data.....	56
4.2.3.4	Analisa dan Pembahasan.....	57
4.2.4	Pengujian Tegangan NodeMCU.....	57
4.2.4.1	Tujuan.....	57
4.2.4.2	Cara Pengujian.....	57
4.2.4.3	Hasil Data.....	58
4.2.4.4	Analisa dan Pembahasan.....	59
4.2.5	Pengujian Software.....	59
4.2.5.1	Tujuan.....	59
4.2.5.2	Cara Pengujian.....	59
4.2.5.3	Hasil Pengujian :.....	59
4.2.5.4	Analisa dan Pembahasan.....	60
4.3	Pengujian Modul Kendali Utama.....	60
4.3.1	Pengujian Perantara WEB.....	60
4.3.1.1	Tujuan.....	60
4.3.1.2	Cara Pengujian.....	60
4.3.1.3	Hasil Pengujian.....	61
4.3.1.4	Analisa dan Pembahasan.....	63
4.3.2	Pengujian Sudut Kemiringan Kamera.....	63

4.3.2.1	Tujuan.....	63
4.3.2.2	Cara Pengujian	63
4.3.2.3	Hasil Pengujian	63
4.3.2.4	Analisa dan Pembahasan	67
4.3.3	Perhitungan Sudut Efisien Objek Kamera	67
4.3.3.1	Tujuan.....	67
4.3.3.2	Cara Pengukuran	68
4.3.3.3	Hasil Pengukuran	69
4.3.3.4	Analisa dan Pembahasan	69
4.4	Pengujian Keseluruhan Sistem	69
4.4.1	Pengujian Durasi Kondisi <i>Traffic Light</i>	69
4.4.1.1	Tujuan.....	69
4.4.1.2	Cara Pengujian	69
4.4.1.3	Hasil Pengujian	70
4.4.2	Pengujian Implementasi Sistem <i>Traffic Light</i>	73
4.4.2.1	Tujuan.....	73
4.4.2.2	Cara Pengujian	73
4.4.2.3	Hasil Pengujian	74
4.4.3	Analisa dan Pembahasan.....	75
4.5	Kesimpulan	75
BAB V	PENUTUP.....	77
5.1	Kesimpulan	77
5.2	Saran.....	78
DAFTAR PUSTAKA.....		79
LAMPIRAN.....		82

DAFTAR GAMBAR

Gambar 2.1	Konflik-konflik utama dan kedua pada simpang bersinyal dengan empat lengan[1].....	8
Gambar 2.2	Urutan waktu pada pengaturan sinyal dengan dua-fase[1].....	9
Gambar 2.3	Arduino IDE[10].....	14
Gambar 2.4	Orange Pi Zero tampak atas[9].....	15
Gambar 2.5	Orange Pi Zero tampak bawah[9].....	16
Gambar 2.6	NodeMCU 1.0 ESP8266-12E (AMICA)[10].....	17
Gambar 2.7	Dot Matrix Display 32x8[11].....	17
Gambar 2.8	Logitech C270.....	18
Gambar 2.9	Pengaruh nilai ambang, (a) citra referensi, (b) citra gray-scale, (c) citra dengan TV=60. (d) citra dengan TV=100, (e) citra dengan TV=200 dan (f) citra dengan TV=250.[15].....	20
Gambar 2.10	(a) Sebelum smoothing, (b) hasil smoothing[15].....	21
Gambar 2.11	Dilasi[15].....	22
Gambar 2.12	Erosi[15].....	23
Gambar 2.13	Blob detection[15].....	24
Gambar 3.1	Diagram blok sistem.....	25
Gambar 3.2	Sistem simulasi kendali traffic light dengan metode pendeteksian gambar secara realtime melalui kamera berbasis IoT.....	26
Gambar 3.3	Modul relay 4 chanel[1].....	28
Gambar 3.4	Diagram blok skema modul lampu lalu lintas.....	30
Gambar 3.5	Perangkat keras modul lampu lalu lintas.....	30
Gambar 3.6	Perangkat keras modul lampu lalu lintas.....	31
Gambar 3.7	Diagram blok skema modul display informasi lalu lintas.....	32
Gambar 3.8	Perangkat keras modul display informasi lalu lintas.....	33
Gambar 3.9	Flowchart kerja pemrograman NodeMCU modul lampu lalu lintas.....	35
Gambar 3.10	Flowchart kerja pemrograman NodeMCU modul display.....	36
Gambar 3.11	Flowchart kerja pemrograman OrangPi modul kendali utama.....	38
Gambar 3.12	Siklus pewaktuan simpang bersinyal 4 fase.....	41
Gambar 3.13	Tampilan antarmuka sistem kendali traffic light berbasis web.....	42

Gambar 3.14	Tampilan panel pengaturan lampu lalu lintas	42
Gambar 4.1	Pengukuran tegangan pada modul display informasi lalu lintas.....	45
Gambar 4.2	Respon modul display informasi lalu lintas setelah menerima data secara wireless	46
Gambar 4.3	Gambar teks yang ditampilkan pada pengujian modul display informasi lalu lintas	46
Gambar 4.4	Modul display kondisi lengang.....	48
Gambar 4.5	Modul display kondisi lancar.....	48
Gambar 4.6	Modul display kondisi ramai	49
Gambar 4.7	Modul display kondisi padat.....	49
Gambar 4.8	Pengukuran tegangan pada modul lampu lalu lintas.....	51
Gambar 4.9	Grafik tegangan power supply	52
Gambar 4.10	Grafik tegangan regulator	54
Gambar 4.11	Hasil pengujian modul lampu lalu lintas secara wireless	60
Gambar 4.12	Tampilan parameter kendali lampu lalu lintas pada perantara web	62
Gambar 4.13	Grafik hasil pengujian sudut kemiringan kamera 40 °.....	65
Gambar 4.14	Grafik hasil pengujian sudut kemiringan kamera 45 °.....	65
Gambar 4.15	Grafik hasil pengujian sudut kemiringan kamera 50 °.....	66
Gambar 4.16	Grafik hasil pengujian sudut kemiringan kamera 55o.....	66
Gambar 4.17	Grafik hasil pengujian sudut kemiringan kamera 60 °.....	67
Gambar 4.18	Input 7 kendaraan tersusun secara seri.....	71
Gambar 4.19	Proses ‘Background subtractor’ pada pendeteksian 7 kendaraan.....	71
Gambar 4.20	Output dari pendeteksian 7 kendaraan	71
Gambar 4.21	Input 2 kendaraan tersusun secara paralel.....	71
Gambar 4.22	Proses ‘Background subtractor’ pada pendeteksian 2 kendaraan.....	72
Gambar 4.23	Output dari pendeteksian 2 kendaraan	72
Gambar 4.24	Grafik regresi dari sistem pendeteksian kendaraan pada pengujian empat kondisi sistem kendali traffic light.....	73

DAFTAR TABEL

Tabel 2.1	Pendekatan ekuivalen kendaraan penumpang (EMP).....	9
Tabel 2.2	Spesifikasi teknis Orange Pi Zero	15
Tabel 2.3	Spesifikasi teknis NodeMCU 1.0 ESP-12E.....	16
Tabel 3.1	Pin pada NodeMCU yang digunakan.....	29
Tabel 3.2	Koneksi antara NodeMCU dengan LED dot matrix	32
Tabel 3.3	IP address modul.....	43
Tabel 4.1	Pengukuran tegangan pada dot matrix display	45
Tabel 4.2	Pengukuran tegangan power supply modul traffic light	52
Tabel 4.3	Pengukuran tegangan regulator modul traffic light	53
Tabel 4.4	Pengukuran tegangan pin LED	56
Tabel 4.5	Tegangan NodeMCU	58
Tabel 4.6	Pengujian sudut kemiringan kamera	64
Tabel 4.7	Data hasil pengujian keseluruhan sistem kendali traffic light	72
Tabel 4.8	Implementasi sistem traffic light keseluruhan.....	74

BAB I

PENDAHULUAN

Pada bab I ini akan dibahas mengenai beberapa sub bab yaitu latar belakang, tujuan, rumusan masalah, batasan masalah, metode pengumpulan data, dan sistematika penulisan dalam pembuatan Sistem Simulasi Kendali *Traffic Light* dengan Metode Pendeteksian Gambar secara *Realtime* melalui Kamera Berbasis IoT.

1.1 Latar Belakang

Lampu lalu lintas (menurut UU no. 22/2009 tentang Lalu lintas dan Angkutan Jalan: alat pemberi isyarat lalu lintas atau APILL) adalah lampu yang mengendalikan arus lalu lintas yang terpasang di persimpangan jalan, tempat penyeberangan pejalan kaki (zebra cross), dan tempat arus lalu lintas lainnya. Lampu ini yang menandakan kapan kendaraan harus berjalan dan berhenti secara bergantian dari berbagai arah. Pengaturan lalu lintas di persimpangan jalan dimaksudkan untuk mengatur pergerakan kendaraan pada masing-masing kelompok pergerakan kendaraan agar dapat bergerak secara bergantian sehingga tidak saling mengganggu antar arus yang ada.

Lampu lalu lintas telah diadopsi di hampir semua kota di dunia ini. Lampu ini menggunakan warna yang diakui secara universal untuk menandakan berhenti adalah warna merah, hati-hati yang ditandai dengan warna kuning dan hijau yang berarti dapat berjalan. Lampu ini awalnya diperkenalkan di Inggris dan dipergunakan untuk mengatur lalu lintas bagi pejalan kaki, atau pengendara sepeda, yang mana pada saat itu, sistem pencahayaannya menggunakan lentera dan terdiri dari 2 warna, yakni merah dan hijau. Dimana merah artinya berhenti dan hijau hati-hati.

Pada masa sekarang ini, sistem lampu lalu lintas kebanyakan diatur oleh ATCS (*Automatic control Light System*). Sistem ini otomatis mengontrol lalu lintas dengan menggunakan bantuan kamera berbasis mikrokontroler. Kamera tersebut biasanya terhubung dengan sistem yang ada sistem lampu lalu lintas yang bertugas untuk mengamati kepadatan kendaraan pada persimpangan jalan, yang kemudian hasil pengamatan tersebut diolah oleh komputer/monitor konflik yang untuk di rekam dan di transfer oleh mikrokontroler menggunakan transfer flash relay.

Mikrokontroler kemudian bekerja menyalakan lampu lalu lintas secara otomatis searah jarum jam, yang berarti ketika komputer terhubung dengan mikrokontroler, maka,

mikrokontroler akan otomatis mengirim informasi lampu mana yang sedang hijau/merah/kuning.

Yang mana kemudian komputer mengolah hal tersebut sesuai gambar yang di terima monitor konflik untuk membuat persentase kepadatan yang terjadi pada jalan tersebut, yang kemudian bisa mengambil keputusan melalui panel yang lain, berapa lama kira kira lampu harus menyala dan berganti. Dengan kemudian, lampu lalu lintas akan bekerja sesuai waktu yang telah ditentukan.

Dalam perkembangan teknologi informasi dan komunikasi saat ini, ada suatu basis *system* yang dapat digunakan sebagai basis pengendali atau kontrol, dari suatu objek yang terhubung ke satu jaringan yang sama. *Internet of Thing* (IoT) merupakan sebuah konsep dimana suatu objek yang memiliki kemampuan untuk mentransfer data melalui jaringan tanpa memerlukan interaksi manusia ke manusia atau manusia ke komputer. IoT telah berkembang dari konvergensi teknologi nirkabel, *micro-electromechanical systems* (MEMS), dan Internet.

Sistem kontrol pada *Traffic Light* berbasis Internet (IoT) ini dapat diterapkan untuk mengatur lampu lalu lintas secara otomatis dengan menggunakan mikrokontroler NodeMCU Lua ESP8266 yang dimana tiap-tiap perangkat memiliki alamat IP (*Internet Protocol*) yang berbeda untuk memberikan informasi dari sensor yang akan diolah oleh *system* menjadi perintah untuk mengatur durasi dari *Traffic Light* tersebut.

1.2 Tujuan

Adapun tujuan penulisan Tugas Akhir ini adalah :

1. Mengimplementasikan sistem kerja *Traffic Light* berbasis IoT.
2. Mengimplementasikan metode sensor pada kamera pemantau *Traffic Light*.
3. Dapat menerapkan otomatisasi pada *Traffic Light* dengan Iot.

1.3 Rumusan Masalah

Untuk menulis Tugas Akhir ini, diperlukan sebuah rumusan masalah sehingga permasalahan menjadi lebih jelas. Adapun rumusan masalah yang dimaksudkan adalah :

1. Bagaimana *Traffic Light* dapat dikontrol melalui jaringan internet.
2. Bagaimana metode pendeteksian gambar sebagai sumber informasi pada *system control* untuk diolah menjadi perintah kerja pada *Traffic Light*.

3. Bagaimana otomatisasi dan cara manual dalam pengoperasian *Traffic Light*.
4. Seberapa besar pengaruh akses internet antara jaringan *system control* dan *Traffic Light*.

1.4 Batasan Masalah

Mengacu pada hal diatas maka penulis membuat sistematika pembahasan dalam Sistem Simulasi Kendali *Traffic Light* dengan Metode Pendeteksian Gambar secara *Realtime* melalui Kamera Berbasis IoT dengan batasan :

1. System keamanan jaringan hanya menggunakan user login sederhana pada mikrokontroller.
2. Simpang bersinyal / lampu lalu lintas yang dibahas adalah simpang bersinyal 4 fase.
3. Pembahasan *Software*/aplikasi yang digunakan selain yang dibuat penulis.

1.5 Metode Pengumpulan Data

Metode penulisan Tugas Akhir ini adalah sebagai berikut :

1. Melakukan studi ke berbagai sumber dari perpustakaan hingga ke lapangan terkait teori-teori yang ada dalam Tugas Akhir.
2. Mengumpulkan berbagai sumber referensi tentang alat yang digunakan.
3. Melakukan desain, perancangan, dan pemrograman *project* terkait.
4. Melakukan konsultasi dengan dosen pembimbing, dan dosen pengajar lainnya serta rekanan diluar yang berkaitan dengan bidang masing-masing.
5. Melakukan pengujian alat dan kinerja *system* kendali rangkaian.

1.6 Sistematika Penulisan

Adapun sistematika penulisan dalam penulisan laporan penelitian Tugas Akhir ini adalah sebagai berikut :

BAB I PENDAHULUAN

Berisikan mengenai latar belakang, tujuan, rumusan masalah, batasan masalah, metode pengumpulan data dan sistematika penulisan dalam penulisan laporan penelitian Tugas Akhir.

BAB II TINJAUAN PUSTAKA

Berisikan mengenai teori-teori penunjang yang berhubungan dengan pembuatan alat ini. Teori tersebut mengenai system kerja *Traffic Light*, konekeksi jaringan yang digunakan, *system control*, NodeMCU Lua ESP8266, Orange Pi Zero, Relay 4 chanel serta *software* yang digunakan seperti Arduino IDE v.1.8.5, Apache v.2.4, PHP v.7, Open CV, dan Phyton v.2.7.

BAB III METODOLOGI RANCANG BANGUN

Berisikan mengenai pendekatan struktural dan fungsional sistem, perancangan dan pembuatan alat, sistem kendali berbasis IoT yang akan dibuat.

BAB IV PENGUJIAN DAN ANALISA

Berisikan mengenai hasil pengujian serta analisa masing-masing komponen dan sistem secara keseluruhan.

BAB V PENUTUP

Berisikan mengenai kesimpulan dari tujuan dan hasil pengujian serta saran.

BAB II

TINJAUAN PUSTAKA

Pada bab II ini akan membahas tentang teori – teori penunjang yang digunakan dalam perancangan sistem kendali *Traffict Light* secara *realtime* berbasis IoT, dimana didalamnya memuat penjelasan mengenai apa itu teknologi berbasis IoT, traffic light on smartcity, teori deteksi gambar, teori pengolahan gambar oleh OpenCV, komponen yang digunakan serta perangkat lunak yang digunakan seperti php, python, dan Apache.

2.1 IoT (Internet of Things)

Internet of Thing (IoT) adalah sebuah konsep dimana suatu objek yang memiliki kemampuan untuk mentransfer data melalui jaringan tanpa memerlukan interaksi manusia ke manusia atau manusia ke komputer. IoT telah berkembang dari konvergensi teknologi nirkabel, micro-electromechanical systems (MEMS), dan Internet. Penelitian pada IoT masih dalam tahap perkembangan. Oleh karena itu, tidak ada definisi dari Internet of Things. Berikut adalah beberapa definisi alternatif dikemukakan untuk memahami Internet of Things (IoT), antara lain :

- Casagras (Coordination and support action for global RFID-related activities and standardisation)

Mendefinisikan Internet of Things, sebagai sebuah infrastruktur jaringan global, yang menghubungkan benda-benda fisik dan virtual melalui eksploitasi data capture dan kemampuan komunikasi. Infrastruktur terdiri dari jaringan yang telah ada dan internet berikut pengembangan jaringannya. Semua ini akan menawarkan identifikasi objek, sensor dan kemampuan koneksi sebagai dasar untuk pengembangan layanan dan aplikasi ko-operatif yang independen. Ia juga ditandai dengan tingkat otonom data capture yang tinggi, event transfer, konektivitas jaringan dan interoperabilitas.

- SAP (Systeme, Anwendungen und Produkte)

Mendefinisikannya bahwa Dunia di mana benda-benda fisik diintegrasikan ke dalam jaringan informasi secara berkesinambungan, dan di mana benda-benda fisik tersebut berperan aktif dalam proses bisnis. Layanan yang tersedia berinteraksi dengan ‘objek pintar’ melalui Internet, mencari dan mengubah status mereka sesuai dengan setiap informasi yang dikaitkan, disamping memperhatikan masalah privasi dan keamanan.

- **CORDIS**

Rencana aksi untuk Uni Eropa untuk memperkenalkan pemerintahan berdasarkan Internet of Things.

- **ETP EPOSS**

Jaringan yang dibentuk oleh hal-hal atau benda yang memiliki identitas, pada dunia maya yang beroperasi di ruang itu dengan menggunakan kecerdasan antarmuka untuk terhubung dan berkomunikasi dengan pengguna, konteks sosial dan lingkungan.

2.2 Smart City

Kota cerdas merupakan sebuah visi pengembangan perkotaan untuk mengintegrasikan teknologi informasi dan komunikasi (TIK) dan teknologi *Internet of things* (IoT) dengan cara yang aman untuk mengelola aset kota. Aset ini meliputi sistem informasi instansi pemerintahan lokal, sekolah, perpustakaan, sistem transportasi, rumah sakit, pembangkit listrik, jaringan penyediaan air, pengelolaan limbah, penegakan hukum, dan pelayanan masyarakat lainnya. *Smart city* ditujukan dalam hal penggunaan informatika dan teknologi perkotaan untuk meningkatkan efisiensi pelayanan. TIK memungkinkan para pejabat kota berinteraksi langsung dengan masyarakat dan infrastruktur kota serta memantau apa yang terjadi di kota, bagaimana kota berkembang, dan bagaimana menciptakan kualitas hidup yang lebih baik. Melalui penggunaan sensor yang terintegrasi dengan *real-time monitoring system*, data yang dikumpulkan dari warga dan perangkat – kemudian diolah dan dianalisis. Informasi dan pengetahuan yang dikumpulkan adalah kunci untuk mengatasi inefisiensi.

2.3 Lampu Lalu Lintas

Simpang bersinyal merupakan bagian dari sistem kendali waktu tetap yang dirangkai atau sinyal aktuasi kendaraan terisolir, biasanya memerlukan metode dan perangkat lunak khusus dalam analisisnya. Pada umumnya sinyal lalu-lintas dipergunakan untuk satu atau lebih dari alasan berikut :

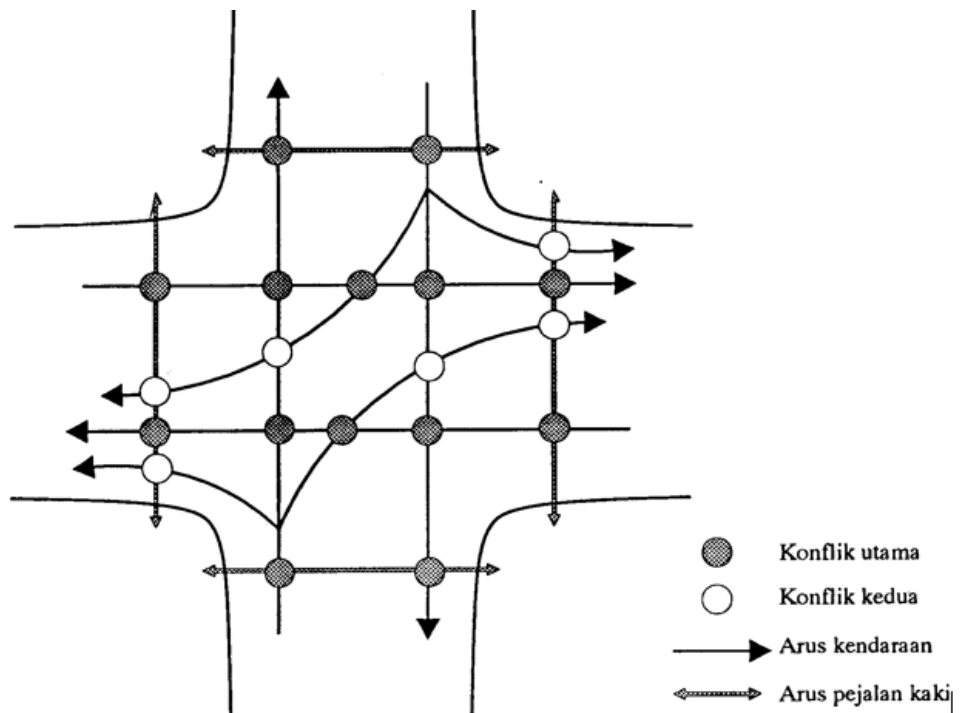
- Untuk menghindari kemacetan simpang akibat adanya konflik arus lalu-lintas, sehingga terjamin bahwa suatu kapasitas tertentu dapat dipertahankan, bahkan selama kondisi lalu-lintas jam puncak.

- Untuk memberi kesempatan kepada kendaraan dan/atau pejalan kaki dari jalan simpang (kecil) untuk/memotong jalan utama.
- Untuk mengurangi jumlah kecelakaan lalu-lintas akibat tabrakan antara kendaraan-kendaraan dari arah yang bertentangan.

2.3.1 Karakteristik Sinyal Lalu Lintas

Untuk sebagian besar fasilitas jalan, kapasitas dan perilaku lalu-lintas terutama adalah fungsi dari keadaan geometrik dan tuntutan lalu-lintas. Dengan menggunakan sinyal, kita dapat mendistribusikan kapasitas kepada berbagai pendekat melalui pengalokasian waktu hijau pada masing-masing pendekat. Maka dari itu untuk menghitung kapasitas dan perilaku lalu-lintas, pertama-tama perlu ditentukan fase dan waktu sinyal yang paling sesuai untuk kondisi yang ditinjau.

Penggunaan sinyal dengan lampu tiga-warna (hijau, kuning, merah) diterapkan untuk memisahkan lintasan dari gerakan-gerakan lalu-lintas yang saling bertentangan dalam dimensi waktu. Hal ini adalah keperluan yang mutlak bagi gerakan-gerakan lalu-lintas yang datang dari jalan-jalan yang saling berpotongan = konflik-konflik utama. Sinyal-sinyal dapat juga digunakan untuk memisahkan gerakan membelok dari lalu-lintas lurus melawan, atau untuk memisahkan gerakan lalu-lintas membelok dari pejalan-kaki yang menyeberang = konflik-konflik kedua.

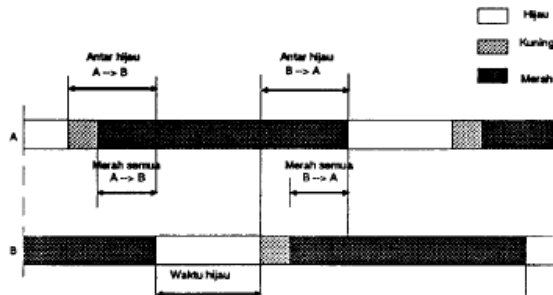
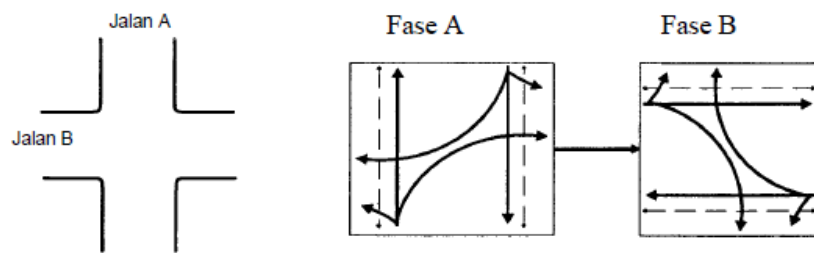


Gambar 2.1 Konflik-konflik utama dan kedua pada simpang bersinyal dengan empat lengan[1]

Gambar 2.1 memberikan penjelasan tentang urutan perubahan sinyal dengan sistem dua fase, termasuk definisi dari waktu siklus, waktu hijau dan periode antar hijau. Maksud dari periode antar hijau (IG = kuning + merah semua) di antara dua fase yang berurutan adalah untuk:

1. Memperingatkan lalu-lintas yang sedang bergerak bahwa fase sudah berakhir.
2. Menjamin agar kendaraan terakhir pada fase hijau yang baru saja diakhiri memperoleh waktu yang cukup untuk ke luar dari daerah konflik sebelum kendaraan pertama dari fase berikutnya memasuki daerah yang sama.

Fungsi yang pertama dipenuhi oleh waktu kuning, sedangkan yang kedua dipenuhi oleh waktu merah semua yang berguna sebagai waktu pengosongan antara dua fase.



Gambar 2.2 Urutan waktu pada pengaturan sinyal dengan dua-fase[1]

2.3.2 Perhitungan Siklus Lampu Lalu Lintas

Perhitungan dilakukan per satuan jam untuk satu atau lebih periode, misalnya didasarkan pada kondisi arus lalu-lintas rencana jam puncak pagi, siang dan sore. Arus lalu-lintas (Q) untuk setiap gerakan (belok-kiri QLT, lurus QST dan belokkanan QRT) dikonversi dari kendaraan per-jam menjadi satuan mobil penumpang (smp) per-jam dengan menggunakan ekuivalen kendaraan penumpang (emp) untuk masing-masing pendekatan terlindung dan terlawan.

Tabel 2.1 Pendekatan ekuivalen kendaraan penumpang (EMP)

Jenis Kendaraan	Emp Untuk Tipe Pendekatan	
	Terlindungi	Terlawan
Kendaraan Ringan (LV)	1,0	1,0
Kendaraan Berat (HV)	1,3	1,3
Sepeda Motor (MC)	0,2	0,4

Rumus Pendekatan Arus Lalu Lintas:

$$Q = Q_{LV} + Q_{HV} \times emp_{HV} + Q_{MC} \times emp_{MC} \dots\dots\dots(1)$$

Rumus Kapasitas Pendekatan Simpang Bersinyal:

$$C = S \times g/c \dots\dots\dots(2)$$

Definisi:

emp (Ekivalen Mobil Penumpang)	Faktor dari berbagai tipe kendaraan sehubungan dengan keperluan waktu hijau untuk keluar dari antrian apabila dibandingkan dengan sebuah kendaraan ringan (untuk mobil penumpang dan kendaraan ringan yang sasisnya sama, emp=1,0).
smp (Satuan Mobil Penumpang)	Satuan arus lalu-lintas dari berbagai tipe kendaraan.
Q (Arus Lalu Lintas)	Jumlah unsur lalu-lintas yang melalui titik tak terganggu di hulu, pendekat per satuan waktu (contoh: kebutuhan lalu-lintas kend/jam smp/jam).
C (Kapasitas)	Arus lalu-lintas maksimum yang dapat dipertahankan.
S (Arus Jenuh)	Besarnya keberangkatan antrian didalam suatu pendekat selama kondisi yang ditentukan (smp/jam hijau).
g (Waktu Hijau)	fuse untuk kendali lalu-lintas aktuasi kendaraan (detik).
c (Waktu Siklus)	Waktu untuk urutan lengkap dari indikasi sinyal (sbg. contoh, diantara dua saat permulaan hijau yang berurutan di dalam pendekat yang sama; detik).

2.4 Apache Web Server

Server HTTP Apache atau Server Web/WWW Apache adalah server web yang dapat dijalankan di banyak sistem operasi (Unix, BSD, Linux, Microsoft Windows dan Novell Netware serta platform lainnya) yang berguna untuk melayani dan memfungsikan situs web. Protokol yang digunakan untuk melayani fasilitas web/www ini menggunakan HTTP.

Apache memiliki fitur-fitur canggih seperti pesan kesalahan yang dapat dikonfigur, autentikasi berbasis basis data dan lain-lain. Apache juga didukung oleh sejumlah antarmuka pengguna berbasis grafik (GUI) yang memungkinkan penanganan server menjadi mudah. Apache HTTP server digunakan sebagai antarmuka pengguna berbasis grafik dengan program kendali utama. Beberapa modul bersama yang dipergunakan pada Tugas Akhir ini seperti modul PHP dan CGI (Common Gateway Interface).

2.5 PHP

PHP adalah bahasa server-side scripting yang dirancang untuk pengembangan web tetapi juga digunakan sebagai bahasa pemrograman tujuan umum. Yang Server-side scripting adalah skrip yang hanya dapat dieksekusi langsung pada server, dan skrip itu tidak dapat diunduh oleh klien/pengujung.

Kode PHP dapat dimasukkan ke dalam kode HTML, atau dapat digunakan dalam kombinasi dengan berbagai sistem template web, sistem manajemen konten web, dan kerangka web. Kode PHP biasanya diolah oleh seorang juru bahasa PHP yang diimplementasikan sebagai modul di web server atau sebagai Common Gateway Interface (CGI) yang bisa dieksekusi.

Server web menggabungkan hasil kode PHP yang diinterpretasikan dan dieksekusi, yang mungkin merupakan jenis data, termasuk gambar, dengan halaman web yang dihasilkan. Kode PHP juga bisa dijalankan dengan command-line interface (CLI) dan bisa digunakan untuk mengimplementasikan aplikasi grafis mandiri.

2.6 Python

Python adalah bahasa pemrograman interpretatif multiguna dengan filosofi perancangan yang berfokus pada tingkat keterbacaan kode. Python diklaim sebagai bahasa yang menggabungkan kapabilitas, kemampuan, dengan sintaksis kode yang sangat jelas, dan dilengkapi dengan fungsionalitas pustaka standar yang besar serta komprehensif.

Python dimaksudkan untuk menjadi bahasa yang mudah dibaca. Formatnya secara visual tidak berantakan, dan sering menggunakan kata kunci bahasa Inggris dimana bahasa lain menggunakan tanda baca. Tidak seperti banyak bahasa lainnya, tidak menggunakan kurung kurawal untuk membatasi blok, dan titik koma setelah pernyataan bersifat opsional. Ini memiliki lebih sedikit pengecualian sintaksis dan kasus khusus daripada C atau Pascal.

Sebuah studi empiris menemukan bahwa bahasa scripting, seperti Python, lebih produktif daripada bahasa konvensional, seperti C dan Java, untuk masalah pemrograman yang melibatkan manipulasi string dan pencarian di kamus, dan menentukan bahwa konsumsi memori sering kali "lebih baik daripada Java dan tidak jauh lebih buruk dari C atau C++".

1.1

2.7 OpenCV

OpenCV (*Open Source Computer Vision*) adalah perpustakaan fungsi pemrograman yang ditujukan untuk penglihatan komputer secara real time. Awalnya dikembangkan oleh Intel, kemudian didukung oleh Willow Garage dan sekarang dikelola oleh Itseez. Perpustakaan ini cross-platform dan gratis untuk digunakan di bawah lisensi *open-source* BSD.

OpenCV ditulis dalam bahasa C++ dan antarmuka utamanya ada di C++, namun masih mempertahankan antarmuka C yang lebih tua namun luas. Ada bindings di Python, Java dan MATLAB / OCTAVE. API untuk antarmuka ini dapat ditemukan di dokumentasi online. Wrappers dalam bahasa lain seperti C#, Perl, Ch, Haskell dan Ruby telah dikembangkan untuk mendorong adopsi oleh khalayak yang lebih luas.

Fitur seperti Haar adalah fitur gambar digital yang digunakan dalam pengenalan objek. Mereka berutang nama mereka pada kesamaan intuitif mereka dengan wavelet Haar dan digunakan pada detektor wajah real-time pertama.

Secara historis, bekerja dengan hanya intensitas gambar (yaitu, nilai pixel RGB pada setiap piksel gambar) membuat perhitungan fitur perhitungan secara komputasi mahal. Publikasi oleh Papageorgiou et al. dibahas bekerja dengan seperangkat fitur alternatif berdasarkan wavelet Haar dan bukan intensitas gambar yang biasa. Viola dan Jones mengadaptasi gagasan menggunakan wavelet Haar dan mengembangkan fitur yang disebut Haar.

Fitur Haar-Like mempertimbangkan daerah persegi panjang yang berdekatan di lokasi tertentu di jendela deteksi, meringkas intensitas piksel di setiap wilayah dan menghitung perbedaan antara jumlah ini.

Perbedaan ini kemudian digunakan untuk mengkategorikan subbagian gambar. Sebagai contoh, katakanlah kita memiliki database gambar dengan wajah manusia. Ini adalah pengamatan umum bahwa di antara semua wajah daerah mata lebih gelap dari daerah

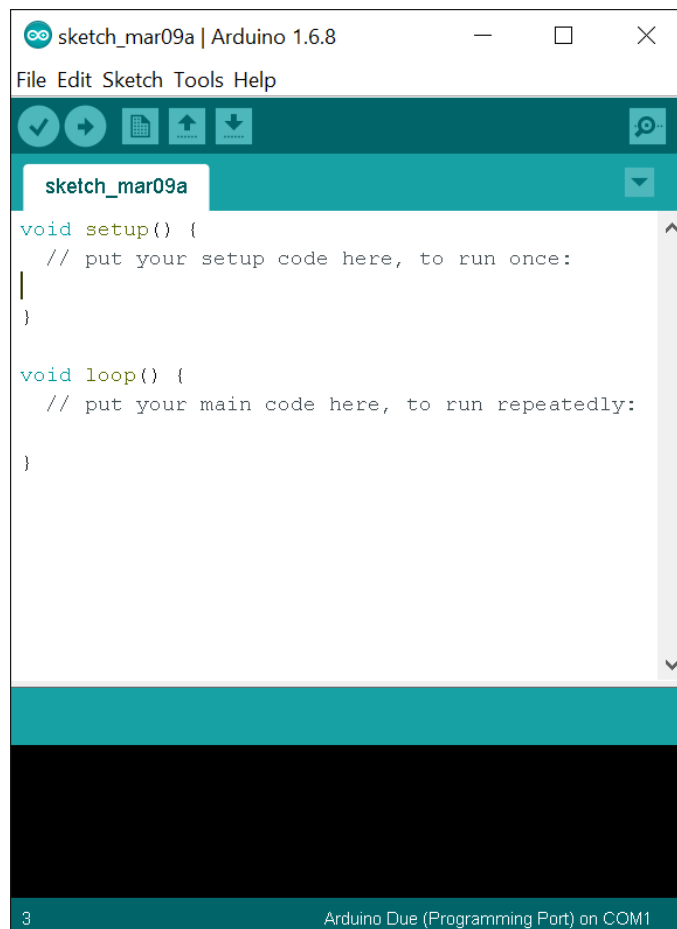
pipi. Oleh karena itu fitur Haar yang umum untuk deteksi wajah adalah sekumpulan dua persegi panjang yang berdekatan yang berada di atas mata dan daerah pipi. Posisi persegi panjang ini didefinisikan relatif terhadap jendela deteksi yang bertindak seperti kotak pembatas pada objek target (wajah dalam kasus ini) ^[8].

2.8 Arduino IDE

Arduino IDE (Integrated Development Environment) adalah software lingkungan pengembangan terpadu yang memungkinkan kita untuk menulis kode, membuat, mengupload dan mengatur berbagai setting pada hampir semua jenis mikrokontroler. Penampilan Arduino IDE seperti yang terlihat pada Gambar 2.3.

Hal ini dikarenakan Arduino IDE bersifat Open Source, yang memungkinkan semua pengembang perangkat lunak berkontribusi dalam bermacam-macam pustaka untuk berbagai mikrokontroler dan modul atau komponen lainnya.

Arduino IDE ditulis dalam bahasa Java, C dan C++. Program ataupun kode yang ditulis dalam Arduino IDE disebut “sketch”, memiliki struktur bahasa C/C++. Program tersebut memiliki 2 fungsi minimal, yaitu *setup()* dan *loop()*. Interface Arduino IDE dapat dilihat pada gambar 2.3.



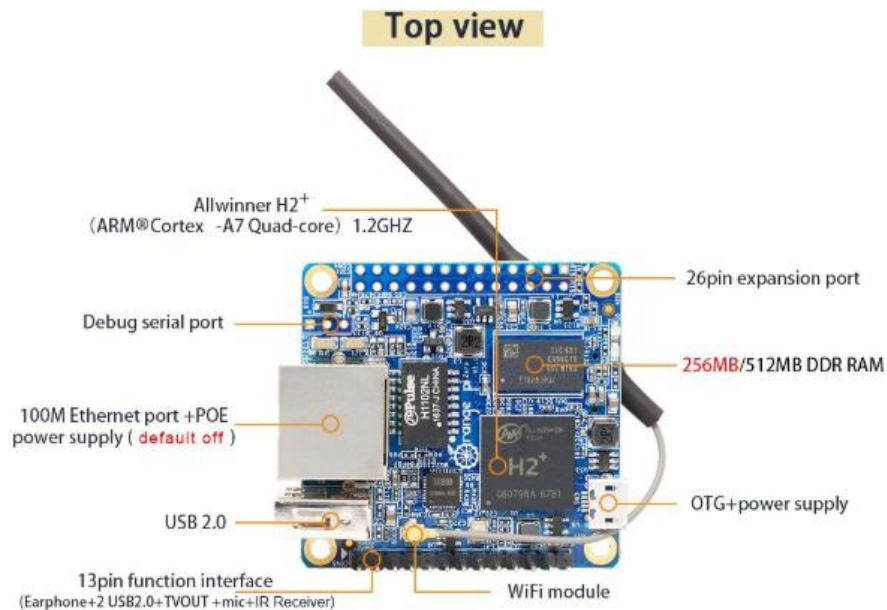
Gambar 2.3 Arduino IDE[10]

2.9 Orange Pi Zero

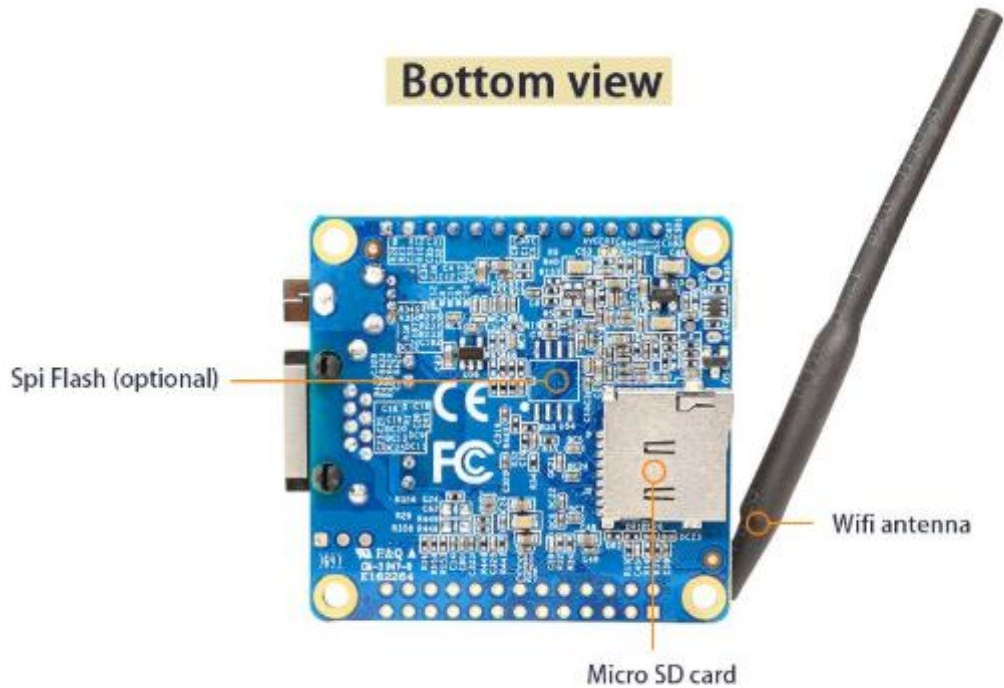
Orange Pi Zero adalah papan komputer tunggal yang bersifat opensource. Dapat menjalankan Sistem Operasi seperti Android 4.4, Ubuntu, Debian. Orange Pi Zero menggunakan Mikroprocessor jenis ARM Cortex-A7 AllWinner H2 SoC, dan memiliki 512MB DDR3 SDRAM. Spesifikasi Orange Pi Zero dapat dilihat pada tabel 2.2 dan bentuk perangkatnya seperti yang terlihat pada gambar 2.4 dan gambar 2.5

Tabel 2.2 Spesifikasi teknis Orange Pi Zero

Spesifikasi	
CPU	H2 Quad-core <u>Cortex-A7</u> H.265/HEVC 1080P.
GPU	·Mali400MP2 GPU @600MHz ·Supports OpenGL ES 2.0
Memory	256MB/512MB DDR3 SDRAM(Share with GPU)(256MB version is Standard version)
Onboard Storage	TF card (Max. 32GB)/ Spi Flash
Onboard Network	10/100M Ethernet RJ45 POE is default off.
Onboard WIFI	XR819, IEEE 802.11 b/g/n
Audio Input	MIC
Video Outputs	Supports external board via 13pins
Power Source	USB OTG, 5 VDC
USB 2.0 Ports	Only One USB 2.0 HOST, one USB 2.0 OTG
Buttons	Power Button
Low-level peripherals	26 Pins Header, compatible with Raspberry Pi B+ 13 Pins Header, with 2x USB, IR pin, AUDIO(MIC, AV)
LED	Power led & Status led
Supported OS	Android, Lubuntu, Debian, Raspbian



Gambar 2.4 Orange Pi Zero tampak atas [9]



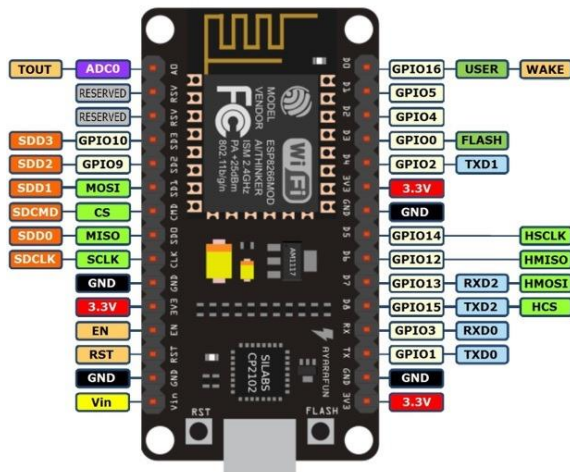
Gambar 2.5 Orange Pi Zero tampak bawah [9]

2.10 NodeMCU

NodeMCU adalah platform IoT open source. Ini mencakup *firmware* yang berjalan di ESP8266 Wi-Fi SoC dari Espressif Systems, dan perangkat keras yang berbasis pada modul ESP-12. Spesifikasi dapat dilihat pada tabel 2.3 dan bentuk NodeMCU nya seperti yang terlihat pada gambar 2.6. Istilah "NodeMCU" secara default mengacu pada firmware daripada perangkat pengembangan. Firmware menggunakan bahasa scripting Lua. Hal ini didasarkan pada proyek eLua, dan dibangun di atas SDK Non-OS Espresso untuk ESP8266.

Tabel 2.3 Spesifikasi teknis NodeMCU 1.0 ESP-12E

Microcontroller	ESP8266
Operating Voltage	3.3V
Input Voltage (USB)	5V
Flash Memory	4 MB
SRAM	128 kB
Clock Speed	80 MHz

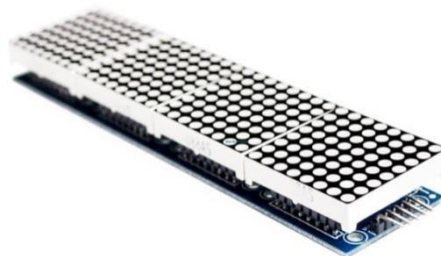


Gambar 2.6 NodeMCU 1.0 ESP8266-12E (AMICA)[10]

2.11 Dot Matrix Display

Dot-Matrix Display adalah perangkat layar yang digunakan untuk menampilkan informasi tentang mesin, jam, indikator keberangkatan kereta api dan banyak perangkat lain yang membutuhkan perangkat tampilan sederhana dengan resolusi terbatas. Bentuk display Dot-Matrix seperti terlihat pada gambar 2.7.

Layar terdiri dari dot matrix lampu atau indikator mekanis yang diatur dalam konfigurasi empat persegi panjang (bentuk lainnya juga mungkin, meski tidak umum) sehingga dengan menyalakan atau mematikan lampu, teks atau grafis yang dipilih dapat ditampilkan. Pengontrol dot matrix mengubah instruksi dari prosesor menjadi sinyal yang menghidupkan atau mematikan lampu dalam matriks sehingga tampilan yang dibutuhkan dihasilkan.



Gambar 2.7 Dot matrix display 32x8[11]

2.12 Logitech C270

Logitech C270 merupakan kamera jenis WebCam HD yang memiliki resolusi 3 Mega Pixel (1280 x 720 pixels), WebCam ini didesain untuk Video Call layar lebar (Wide

Screen) dengan resolusi 720 pixel. Kamera ini dilengkapi dengan koreksi level cahaya secara otomatis. Kamera ini dilengkapi juga dengan teknologi Logitech Fluid Crystal™ yang membuat kamera ini dapat menangkap video dan gambar dengan tajam dan ketajaman warna yang tinggi. WebCam tersebut dapat dilihat pada gambar 2.8.



Gambar 2.8 Logitech C270

Spesifikasi teknis:

- HD video calling (1280 x 720 pixels) with recommended system
- Video capture: Up to 1280 x 720 pixels
- Logitech Fluid Crystal™ Technology
- Photos: Up to 3.0 megapixels (software enhanced)
- Built-in mic with noise reduction
- Hi-Speed USB 2.0 certified (recommended)
- Universal clip fits laptops, LCD or CRT monitors
- FoV: 60°

2.13 Konsep Pendeteksian Gambar Dengan Kamera

Sistem penghitungan kendaraan otomatis ini menggunakan data video yang diperoleh dari kamera lalu lintas. Metode blob detection digunakan untuk mengolah serangkaian frame yang diperoleh dari video untuk menghitung jumlah kendaraan pada setiap jalur. Pada setiap frame ditentukan region of interest (ROI) dengan melacak objek yang terdeteksi di dalam ROI tertentu dan kemudian melakukan perhitungan. Beberapa teori yang digunakan pada deteksi kendaraan adalah sebagai berikut.

A. Region of Interest (ROI)

ROI merupakan subcitra dari suatu region. Hal ini merefleksikan kenyataan bahwa suatu citra seringkali memiliki suatu koleksi objek yang dapat menjadi suatu region tersendiri. ROI memungkinkan dilakukannya pengkodean secara berbeda pada area tertentu

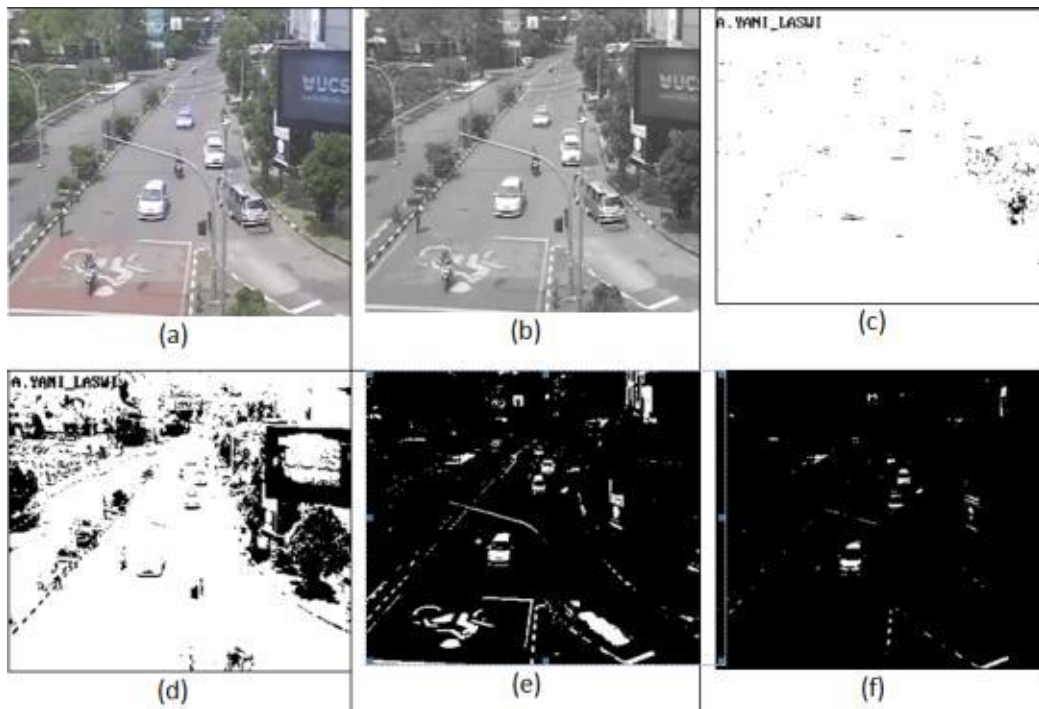
dari citra digital, sehingga mempunyai kualitas yang lebih baik dari area sekitarnya (background). Fitur ini menjadi sangat penting bila terdapat bagian tertentu dari citra digital yang dianggap lebih penting dari bagian lainnya. Metode seleksi ROI yang digunakan adalah dengan piksel karena model jalur yang digunakan lebih memungkinkan jika menggunakan metode piksel.

B. Thresholding

Thresholding digunakan untuk mengatur gray-level yang ada pada gambar. Misalnya pada sebuah gambar, $f(x,y)$ tersusun dari objek yang terang pada sebuah background yang gelap. Gray-level milik objek dan milik background terkumpul menjadi dua grup yang dominan. Salah satu cara untuk mengambil objek dari background-nya adalah dengan memilih sebuah nilai ambang (threshold) T yang memisahkan grup yang satu dengan grup yang lain. Maka, semua piksel yang memiliki nilai $> T$ disebut titik objek, sedangkan yang lain disebut titik background. Proses ini disebut thresholding. Sebuah gambar yang telah dikenai proses thresholding $g(x,y)$ dapat didefinisikan sebagai berikut [6].

$$g(x,y) = \begin{cases} 1 & \text{jika } f(x,y) > T \\ 0 & \text{jika } f(x,y) \leq T \end{cases} \dots\dots\dots(3)$$

Nilai T dapat ditentukan melalui perhitungan rata-rata dari keseluruhan nilai karna yang ada pada gambar. Pada perhitungan ini, nilai T yang diperoleh tetap disimpan dalam bilangan real. Nilai T yang diperoleh untuk gambar yang memiliki histogram yang telah mengalami proses penyamaan (equalization) adalah berkisar antara 127 dan 128. Nilai maksimum T adalah nilai tertinggi dari sistem warna yang digunakan dan nilai minimum T adalah nilai terendah dari sistem warna yang digunakan. Untuk 256-graylevel maka nilai tertinggi T adalah 255 dan nilai terendahnya adalah 0 [6]. Pengaruh nilai ambang ditunjukkan pada gambar 2.9.



Gambar 2.9 Pengaruh nilai ambang, (a) citra referensi, (b) citra gray-scale, (c) citra dengan TV=60. (d) citra dengan TV=100, (e) citra dengan TV=200 dan (f) citra dengan TV=250.[15]

C. Smoothing Filter

Smoothing, juga disebut blurring/pengaburan adalah operasi pengolahan citra sederhana dan sering digunakan. Ada banyak alasan dilakukannya smoothing. Pada dasarnya, smoothing digunakan untuk mengurangi derau. Untuk melakukan operasi smoothing, digunakan filter. Jenis filter yang paling umum adalah dari filter yang linear, dengan nilai keluaran piksel ($g(i, j)$) ditentukan sebagai jumlah bobot nilai masukan piksel yaitu, $f(i+k, j+l)$ [7].

$$g(i, j) = \sum_{k, l} f(i+k, j+l)h(k, l) \dots\dots\dots(4)$$

dengan $h(k, l)$ disebut kernel, yang tidak lebih dari koefisien filter. Ini membantu untuk memvisualisasikan filter sebagai jendela koefisien meluncur di gambar. Ada banyak jenis filter, di antaranya adalah Gaussian dan filter median [8], seperti yang ditunjukkan pada gambar 2.10.



Gambar 2.10 (a) Sebelum smoothing, (b) hasil smoothing[15]

D. Image Morphology

Image morphology merupakan teknik atau proses yang digunakan untuk mengolah citra berdasarkan prinsip morfologi matematika. Dalam pemrosesan citra, hasil yang diharapkan diperoleh berdasarkan bentuk atau struktur citra asal. Lebih lanjut dikatakan bahwa morfologi senantiasa berkaitan erat dengan proses ketetanggaan (*neighborhood*) yang terbentuk dari blok nilai biner, satu dan nol [9]. Lebih lanjut lagi, proses morfologi suatu citra merupakan kumpulan operasi nonlinear yang berkaitan dengan bentuk atau morfologi dalam suatu citra [10].

Hasil operasi morfologi dimanfaatkan untuk pengambilan keputusan dengan analisis lebih lanjut. Operasi ini antara lain meliputi pencarian batas/kontur, dilasi, erosi, penutupan (*closing*), pembukaan (*opening*), pengisian (*filling*), pelabelan, dan pengerangkaan (*skeletonization*) [11].

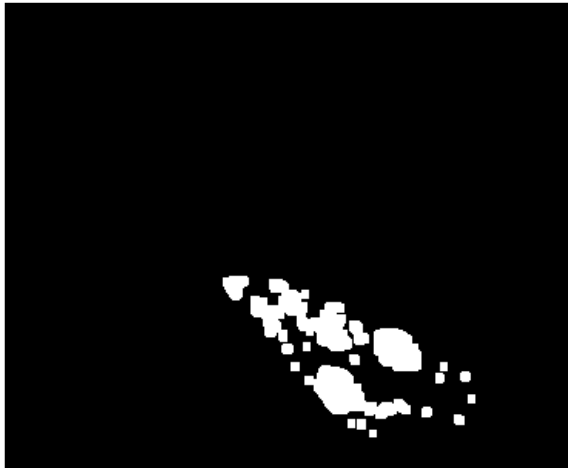
E. Dilasi

Dilasi adalah transformasi morfologi yang menggabungkan dua himpunan dengan menggunakan penjumlahan vektor elemen himpunan. Dilasi merupakan proses penggabungan titik-titik latar menjadi bagian dari objek, berdasarkan *structuring element* yang digunakan, dengan contoh seperti pada gambar 2.11. Proses ini adalah kebalikan dari erosi, yaitu mengubah latar di sekeliling objek menjadi bagian dari objek tersebut [12]. Ada dua cara untuk melakukan operasi ini, yaitu sebagai berikut.

1. Dengan mengubah semua titik latar yang bertetangga dengan titik batas menjadi titik objek.
2. Dengan mengubah semua titik di sekeliling titik batas menjadi titik objek.

Operasi morfologi dasar matematika dilasi dilakukan berdasarkan aljabar Minkowski, ditunjukkan oleh (3) [12].

$$D(A, B) = A + B = U_{\beta \in B} (A + \beta) \dots\dots\dots(5)$$



Gambar 2.11 Dilasi[15]

F. Erosi

Erosi dapat diperoleh dengan melebarkan komplemen dari piksel hitam dan kemudian mengambil komplemen dari set point yang dihasilkan. Erosi merupakan proses penghapusan titik-titik batas objek menjadi bagian dari latar, berdasarkan structuring element yang digunakan, dengan contoh yang ditunjukkan pada gambar 2.12. [12]. Pada operasi ini, ukuran objek diperkecil dengan mengikis sekeliling objek. Ada dua cara yang dapat dilakukan untuk melakukan proses erosi, yaitu sebagai berikut.

1. Dengan mengubah semua titik batas menjadi titik latar.
2. Dengan mengatur semua titik di sekeliling titik latar menjadi titik latar

Operasi morfologi dasar matematika erosi dilakukan berdasarkan aljabar Minkowski seperti pada (4) [12].

$$E(A,B) = A - (-B) = \bigcup_{\beta \in B} (A - \beta) \text{ dengan } -B = \{-B | \beta \in B\}. \dots\dots\dots(6)$$

Himpunan A atau B dapat dianggap sebagai "citra". A biasanya dianggap sebagai citra dan B disebut structuring element.

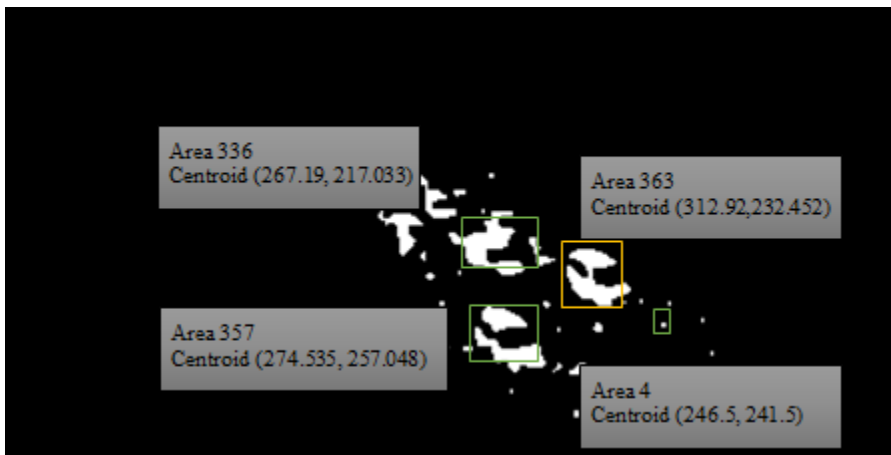


Gambar 2.12 Erosi[15]

G. Blob Detection

Blob detection adalah algoritma yang digunakan untuk menentukan suatu grup dari piksel saling berhubungan satu sama lain atau tidak. Metode ini sangat berguna untuk mengidentifikasi objek yang terpisah-pisah pada suatu citra atau menghitung jumlah dari suatu objek pada suatu citra. Pada metode blob detection, citra harus diproses dengan metode ambang terlebih dahulu, berdasarkan warna yang akan dideteksi. Setelah itu, citra dengan warna di atas nilai ambang dapat dikategorikan sesuai dengan aturan yang telah ditentukan terlebih dahulu. Misalnya untuk piksel yang memiliki nilai lebih kecil daripada nilai ambang akan dikelompokkan sebagai komponen dari objek yang dideteksi, sedangkan yang nilainya di atas ambang dikelompokkan sebagai informasi yang tidak penting. Sedangkan untuk mendapatkan titik berat massa adalah dengan merata-ratakan lokasi tiap piksel dengan warna tertentu.

Bagi banyak pengolahan citra, mendeteksi objek low-level dalam sebuah gambar merupakan hal yang sangat penting. Objek dalam bentuk dua atau tiga dimensi tersebut biasa disebut blob. Bentuk blob timbul dalam cara yang berbeda, bergantung pada ukuran, dan dapat dideteksi dengan menggunakan metode sederhana dalam sebuah representasi gambar [13]. Gambar 2.13 memperlihatkan *blob detection* dari video lalu lintas.



Gambar 2.13 Blob detection[15]

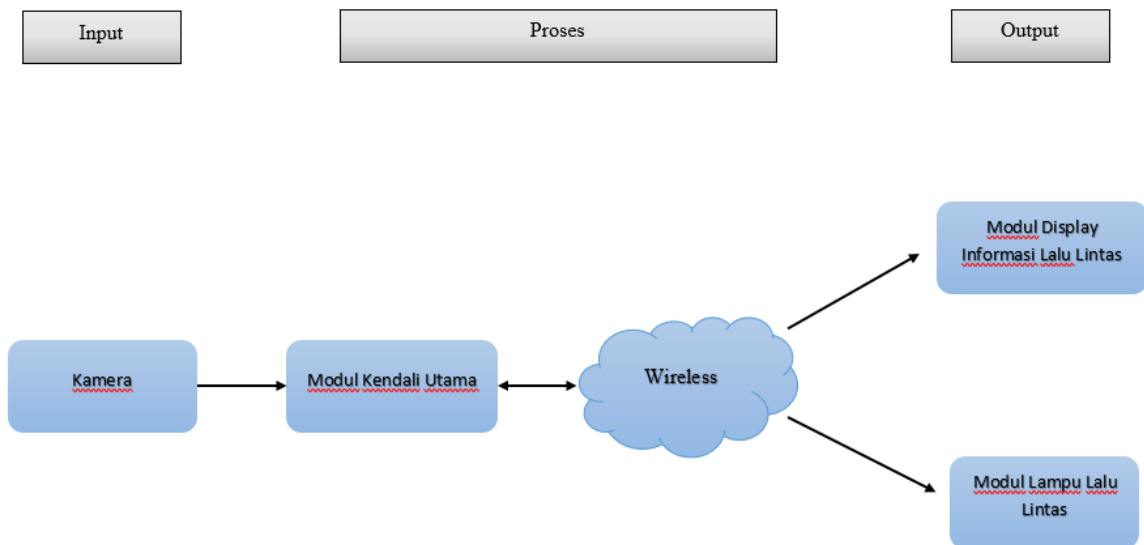
BAB III METODOLOGI RANCANG BANGUN

3.1 Pendekatan Fungsional Dan Prinsip Kerja Alat

Pada bab III ini akan membahas tentang fungsi dari peralatan yang digunakan, dimana di dalamnya memuat prinsip kerja dan cara kerja dari alat serta membahas tentang analisa sistem baik perangkat keras dan perangkat lunak. terdiri dari tiga bagian utama, yaitu rancang bangun struktural, rancang bangun fungsional dan perancangan alat prototype. Sebagai pengendali digunakan program bahasa pyhton.

Prinsip kerja dasar *Sistem Kendali Traffic Light Dengan Metode Pendeteksian Gambar Secara Realtime Melalui Kamera Berbasis IoT* adalah mengolah data gambar kondisi lalu lintas pada ruas jalan tertentu, sebagai sampel, memproses data tersebut menggunakan metode blob dari OpenCV untuk mendapatkan jumlah kendaraan yang terdapat pada ruas jalan tersebut yang kemudian membuat keputusan sesuai dengan parameter yang telah ditentukan.

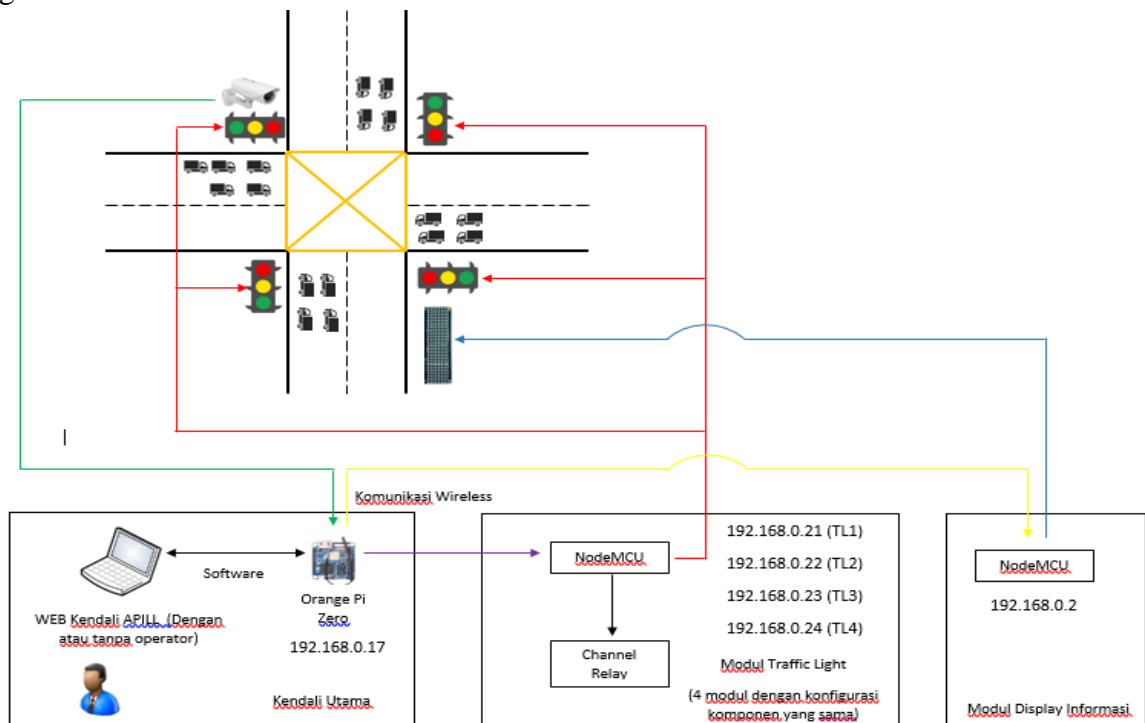
Pada system keseluruhan, dibagi menjadi tiga titik bagian, yaitu bagian *input*, bagian proses, dan bagian *output*. Setiap bagian mempunyai fungsi dan cara kerjanya masing-masing. Semua bagian tersebut dapat dilihat pada gambaran diagram blok keseluruhan system pada gambar 3.1 dibawah ini.



Gambar 3.1 Diagram blok sistem

Hasil dari proses blob tersebut akan dikalkulasikan menggunakan rumus pendekatan simpang bersinyal, parameter-parameter pendukung lainnya telah ditetapkan sebelumnya sebagai contoh dalam pengerjaan Tugas Akhir ini. Cara kerja alat secara global adalah modul kamera diposisikan pada salah satu jalur dari simpang bersinyal tersebut, modul utama mengirim perintah ke modul kamera untuk mengambil gambar, interval waktu pengambilan dan kondisi pengambilan gambar dapat diatur pada modul utama, modul kamera melakukan pengambilan gambar yang kemudian data gambar tersebut dikirim ke modul pusat.

Modul pusat memproses data tersebut menggunakan OpenCV untuk mendapatkan jumlah kendaraan yang terdapat pada ruas jalan tersebut yang kemudian membuat keputusan sesuai dengan parameter yang telah ditentukan. Keputusan tersebut berupa status kondisi arus lalu lintas saat itu, yang dikelompokkan menjadi “kondisi lengang”, “kondisi lancar”, “kondisi ramai” dan “kondisi padat”. Kemudian modul pusat mengirim data siklus pewaktuan ke modul lampu lalu lintas sesuai dengan siklus yang telah ditentukan pada parameter didalam modul utama seperti dapat dilihat pada Sistem Kendali Traffic Light Dengan Metode Pendeteksian Gambar Secara Realtime Melalui Kamera Berbasis IoT gambar 3.2 dibawah ini.



Gambar 3.2 Sistem simulasi kendali traffic light dengan metode pendeteksian gambar secara realtime melalui kamera berbasis IoT

3.2 Pendekatan Struktural

Pada bagian ini dijelaskan rancangan teknis dari pembuatan alat pada tugas akhir ini. Secara garis besar berdasarkan fungsi, sistem ini dapat dibagi menjadi 4 bagian yang akan disebut sebagai modul, seperti yang terlihat pada Gambar 3.1. Adapun bagian-bagian tersebut yaitu :

1. Modul Lampu Lalu Lintas
2. Modul Kamera
3. Modul Display Informasi Lalu Lintas
4. Modul Kendali Utama

Keseluruhan bagian-bagian tersebut dapat berkomunikasi dan dikendalikan secara Wireless, adapun bagian-bagian tersebut yaitu :

1. Modul Lampu Lalu Lintas

Berfungsi sebagai alat peraga sinyal lalu lintas dimana siklus waktu dan kondisinya dapat ditentukan secara real-time.

2. Modul Display Informasi Lalu Lintas

Berfungsi untuk menampilkan informasi lalu lintas yang dikirim dari modul utama untuk pengguna jalan.

3. Modul Kendali Utama

Modul ini merupakan bagian inti dari sistem ini, hampir keseluruhan proses berada pada modul ini, proses-proses tersebut dapat diuraikan sebagai berikut:

- Mengirim perintah secara rutin dengan interval tertentu ke modul kamera untuk mengambil gambar kondisi lalu lintas saat itu.
- Menerima dan mengolah data gambar kondisi lalu lintas yang kemudian akan menghasilkan suatu kesimpulan mengenai kondisi lalu lintas saat itu.
- Kesimpulan tersebut akan dikonversikan menjadi siklus waktu sesuai dengan yang telah ditetapkan pada parameter sebelumnya, yang kemudian akan dikirimkan ke modul lampu lalu lintas.
- Sebagai antarmuka berbasis web untuk operator kendali.

3.2.1 Perancangan Hardware Sistem Kendali Traffic Light

Pada sisi *hardware*, perancangan *Sistem Kendali Traffic Light Dengan Metode Pendeteksian Gambar Secara Realtime Melalui Kamera Berbasis IoT*

dipisahkan dalam beberapa bagian, bagian itu disebut modul. Dan modul-modul tersebut terdiri dari 3 modul, yaitu:

1. **Modul Lampu Lalu Lintas**
2. **Modul Kendali Utama**
3. **Modul Display Informasi Lalu Lintas**

3.2.1.1 Perancangan Hardware Modul Lampu Lalu Lintas

Pada modul ini, level tegangan dan daya lampu lalu lintas lebih tinggi dibandingkan dengan level tegangan operasi mikrokontroler pada umumnya. Untuk itu diperlukan sebuah relay sebagai saklar yang dapat dikendalikan oleh mikrokontroler. Modul relay yang akan digunakan dalam alat ini seperti yang terlihat pada Gambar 3.3



Gambar 3.3 Modul relay 4 chanel[1]

Pada modul relay tersebut relay digerakkan melalui rangkaian driver yang terdiri dari transistor, dioda, resistor dan optocoupler. Penggunaan optocoupler pada modul relai membuat antara rangkaian relai dengan mikrokontroler menjadi terisolasi dengan aman. Hal ini pun membuat mikrokontroler tidak terbebani dengan arus tinggi untuk menggerakkan relay. Selain isolasi antar bus, optocoupler tersebut dapat digerakkan pada level tegangan 3.3 v sesuai dengan level tegangan operasi NodeMCU, dengan kata lain, tidak lagi membutuhkan “Logic Level Converter”.

Pada modul ini akan dipasang 3 buah lampu LED 3 volt (merah, kuning dan hijau) sebagai indikator lampu APILL internal, sehingga dapat memantau secara langsung kondisi lampu lalin ataupun proses yang terjadi walaupun lampu lalin eksternal tidak disambungkan.

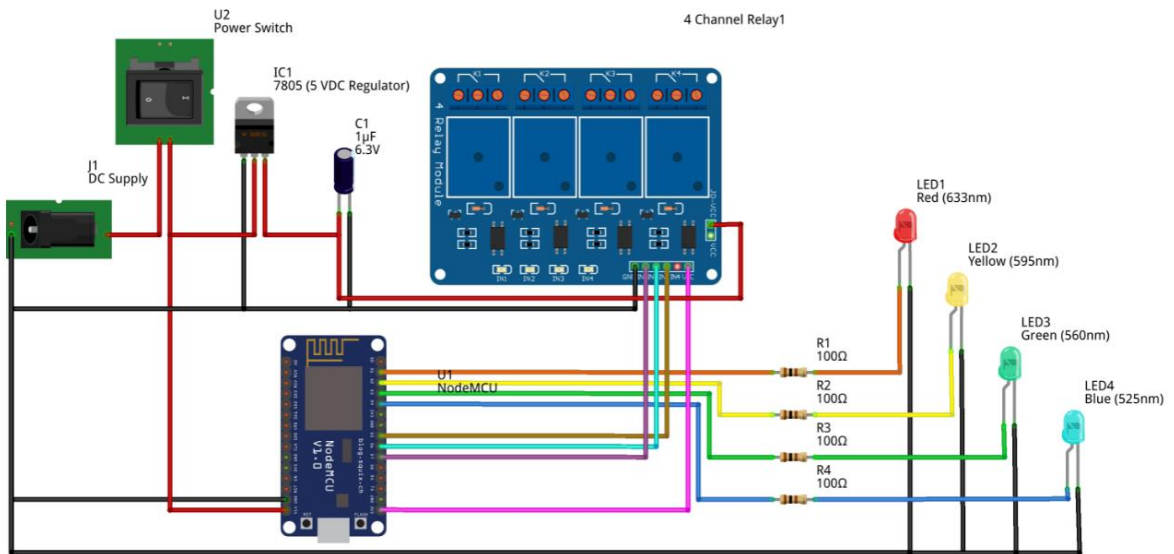
Agar modul lampu lalin tidak terbebani dengan arus yang dibutuhkan untuk menggerakkan relay pada saat lampu lalin eksternal tidak digunakan, mikrokontroller haruslah dapat ditentukan apakah akan menggerakkan relay atau tidak. Apabila mikrokontroller tidak menggerakkan relay, maka pin output yang digunakan untuk menggerakkan relay ditahan pada kondisi yang sesuai dengan modul relay pada saat terbuka.

Pada modul relay yang akan digunakan, relay dalam kondisi terbuka (tidak terhubung) apabila input dalam kondisi HIGH, untuk itu mikrokontroller haruslah dapat menahan pin output yang digunakan untuk menggerakkan relay dalam kondisi HIGH apabila lampu lalin eksternal tidak digunakan.

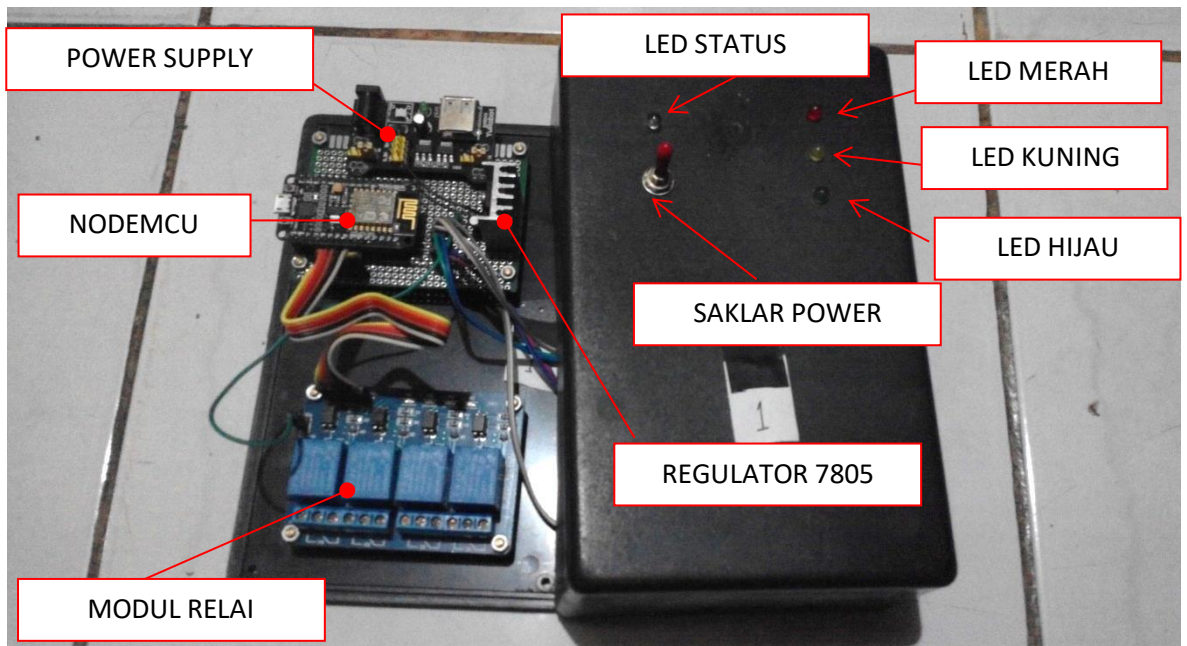
Modul mikrokontroller yang akan digunakan adalah NodeMCU versi 1.0 ESP-12E keluaran AMICA, spesifikasinya telah dijelaskan pada Bab 2. Terdapat perbedaan penomoran pin secara fisik pada NodeMCU dan secara software pada Arduino IDE, pin-pin yang akan digunakan akan diuraikan pada tabel 3.1 dan diagram blok modul lampu lalin seperti yang terlihat pada Gambar 3.4.

Tabel 3.1 Pin pada NodeMCU yang digunakan

Fungsi Pin	Nomor Pin	
	Arduino IDE	NodeMCU
Indikator Lampu Merah	0	GPIO D3
Indikator Lampu Kuning	4	GPIO D2
Indikator Lampu Hijau	5	GPIO D1
Relay Lampu Merah	14	GPIO D7
Relay Lampu Kuning	12	GPIO D6
Relay Lampu Hijau	13	GPIO D5



Gambar 3.4 Diagram blok skema modul lampu lalu lintas



Gambar 3.5 Perangkat keras modul lampu lalu lintas

3.2.1.2 Perancangan Hardware Kendali Utama

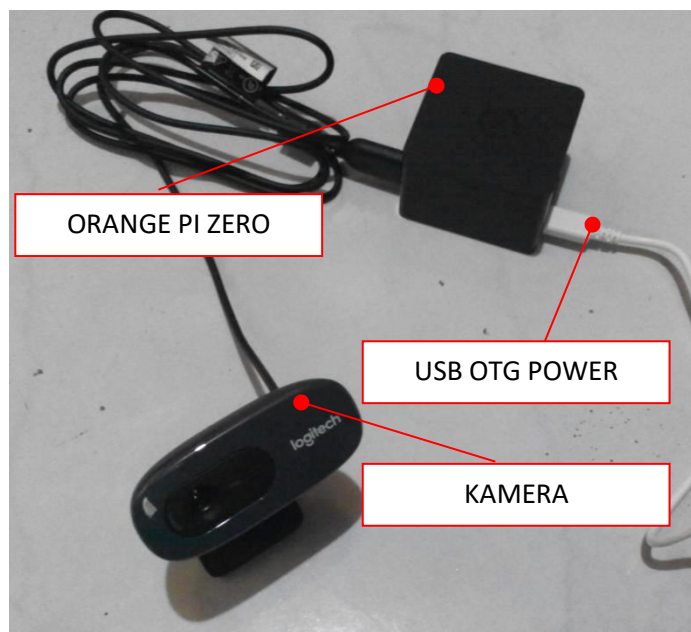
Modul kamera terintegrasi dengan modul kendali utama, modul ini menggunakan Orange Pi Zero dengan sistem operasi linux Armbian. Orange Pi Zero menggunakan Mikroprocessor jenis ARM Cortex-A7 AllWinner H2 SoC, dan memiliki 512MB DDR3 SDRAM. Mikroprocessor tersebut mendukung decode video dengan format H.265/HEVC 1080P secara hardware. Orange Pi Zero dilengkapi dengan koneksi jaringan wired maupun wireless.

Untuk dapat mengambil gambar, digunakan webcam Logitech tipe C270, kamera tersebut dapat menangkap gambar dengan resolusi hingga 3 Mega Pixel. Kamera tersebut terkoneksi ke Orange Pi melalui USB.

Modul ini juga digunakan sebagai modul kendali utama, maka modul ini haruslah dilengkapi / diinstall dengan Python dan OpenCV untuk mengolah gambar dan menghitung jumlah kendaraan, Apache HTTP Server dan PHP agar dapat menjadi web server untuk menampilkan informasi ke pengguna dan sebagai sistem utama pengaturan arus lalu lintas dengan cara mengatur durasi lampu lalu lintas sesuai dengan jumlah kendaraan yang ada yang disesuaikan dengan parameter yang telah ditentukan.

Informasi mengenai durasi lampu lalin akan dikirim secara otomatis kepada Modul Lampu Lalu Lintas, dan informasi mengenai kondisi lalu lintas akan dikirim kepada Modul Display Informasi Lalu Lintas.

Modul ini juga di setting sebagai Wireless Hot Spot dengan tujuan agar modul lainnya (Modul Lampu Lalu Lintas dan Modul Display Informasi Lalu Lintas) dapat terkoneksi secara wireless ataupun melalui VPN (internet).



Gambar 3.6 Perangkat keras kodul lampu lalu lintas

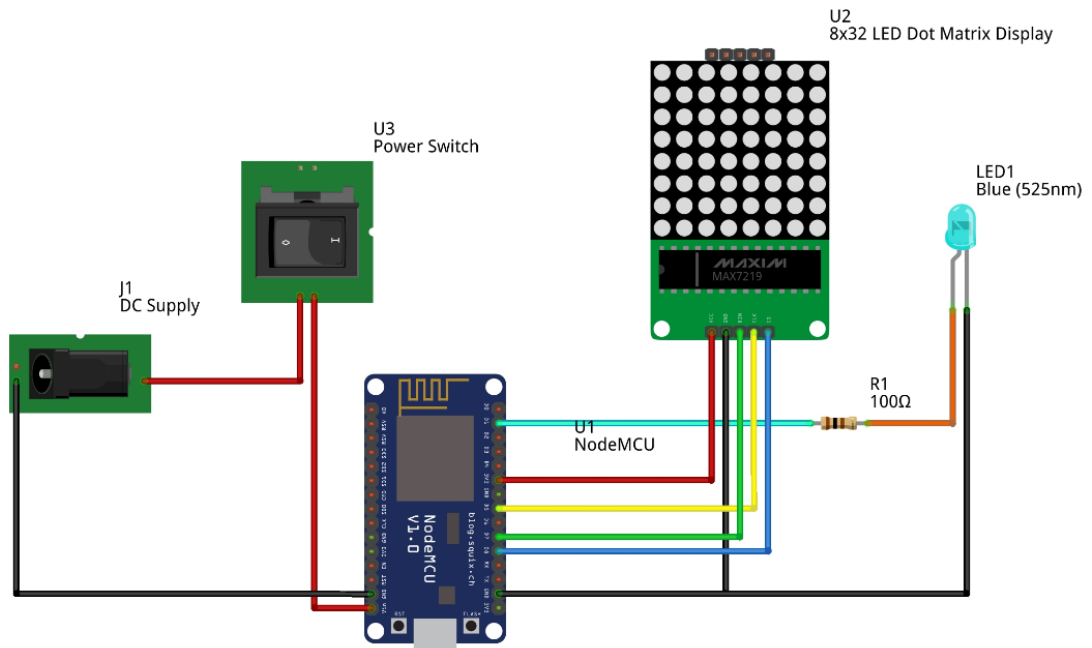
3.2.1.3 Perancangan Hardware Modul Display Informasi Lalu Lintas

Modul ini dipergunakan untuk menampilkan informasi yang dikirim dari modul kendali utama. Alat peraganya menggunakan LED Dot Matrix 32x8 pixel. Komunikasi antarmuka-nya menggunakan metode SPI.

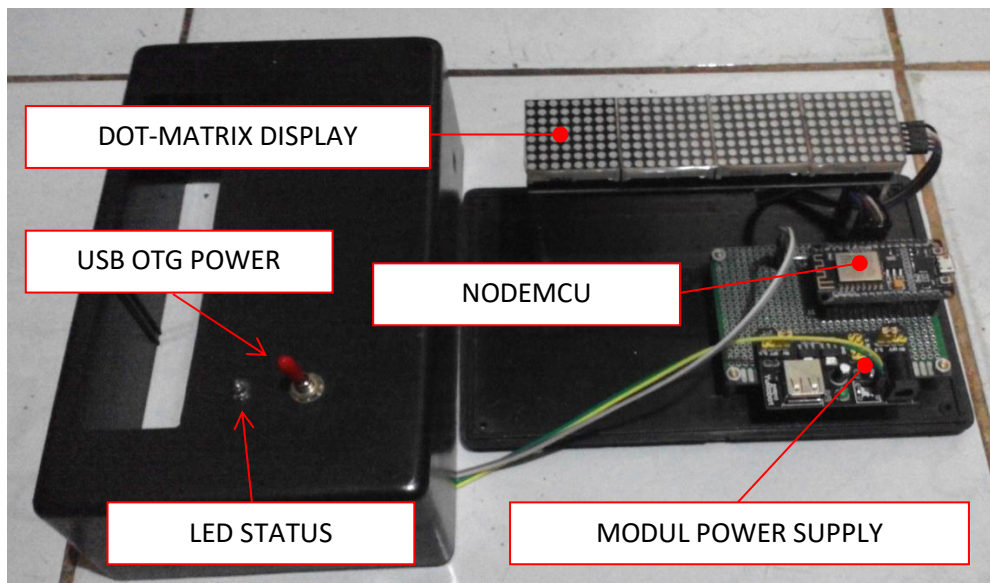
LED Dot Matrix yang akan digunakan dapat beroperasi pada tegangan 3.3v, sehingga level tegangan operasi LED Dot Matrix sama dengan level tegangan operasi NodeMCU. Susunan pin yang dikoneksikan antara NodeMCU dengan LED Dot Matrix dapat dilihat pada Tabel 3.2. Dan diagram blok modul display terdapat pada Gambar 3.7.

Tabel 3.2 Koneksi antara NodeMCU dengan LED dot matrix

Nomor Pin	
NodeMCU	LED Dot Matrix
D8	CS
D7	DIN
D5	CLK
VCC	VCC
GND	GND



Gambar 3.7 Diagram blok skema modul display informasi lalu lintas



Gambar 3.8 Perangkat keras modul display informasi lalu lintas

3.2.2 Perancangan Software Sistem Kendali Traffic Light

Setelah membahas perancangan prototipe dari sisi hardware dan telah ditentukan pin-pin yang akan digunakan. Kemudian kita dapat merancang software yang disesuaikan dengan rancangan hardware.

Untuk memudahkan pembahasan, software dikelompokkan berdasarkan fungsi, maka software dapat dibagi menjadi 3 bagian, bagian tersebut sebagai berikut:

1. Software Modul Lampu Lalu Lintas

Software lampu lalu lintas berfungsi untuk mengatur penyalan lampu sesuai dengan urutan penyalan dan durasi masing-masing lampu yang sebelumnya telah diterima melalui WiFi ataupun Serial Port.

2. Software Modul Display Informasi Lalu Lintas

Software display informasi lalu lintas berfungsi untuk menampilkan teks berupa informasi kondisi lalu lintas yang dikirim dari modul kendali utama.

3. Software Modul Kendali Utama

Software kendali utama berfungsi untuk mengambil gambar melalui kamera sesuai dengan durasi pengambilan sampel, mengolah gambar tersebut agar dapat menghitung jumlah kendaraan yang tertangkap oleh kamera, menentukan kondisi lalu lintas berdasarkan jumlah kendaraan, mengirim data berupa

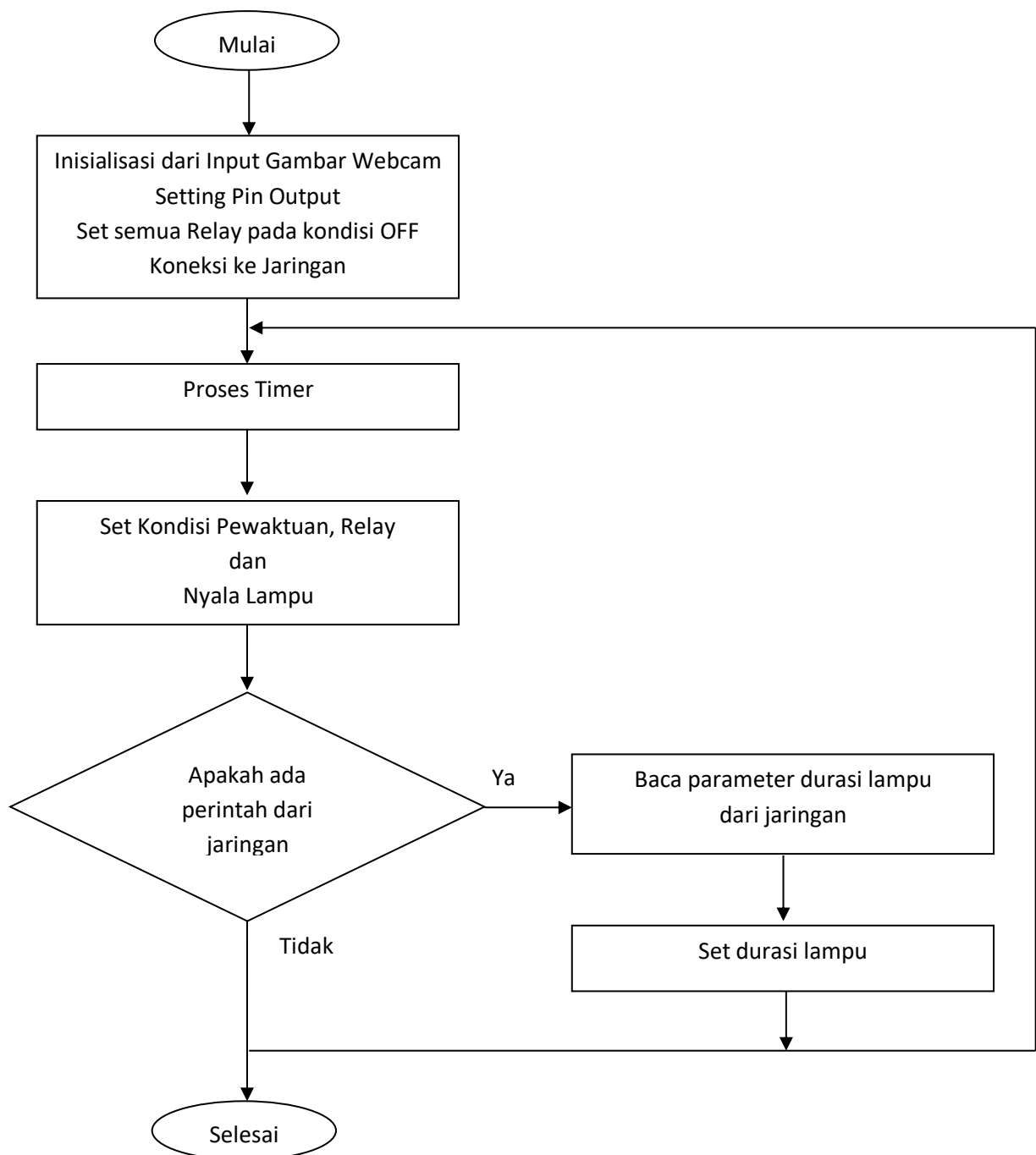
informasi kondisi lalu-lintas ke modul display dan durasi masing-masing lampu dan urutan pewaktuannya ke masing-masing modul lampu lalu-lintas, dan menyajikan panel kendali berbasis web kepada operator/client.

3.2.2.1 Perancangan Software Modul Lampu Lalu Lintas

Modul lampu lalu lintas terdiri dari NodeMCU, modul relay 4 chanel untuk menggerakkan lampu lalu lintas eksternal, 3 buah lampu led (merah, kuning dan hijau) sebagai indikator internal pada modul dan modul power supply. Fungsi yang disediakan pada modul lampu lalu lintas antara lain:

- Menerima perintah dan mengolah perintah dari modul kendali utama secara wireless maupun serial port.
- Dapat mengirim informasi parameter durasi yang tersimpan dan sedang diaplikasikan pada modul tersebut.
- Mengatur siklus nyala lampu dan urutan nyala secara realtime.
- Dapat mengendalikan lampu lalu lintas eksternal melalui relay.

Flowchart kerja pemrograman NodeMCU pada modul lalu lintas ini dapat dilihat pada gambar 3.9.



Gambar 3.9 Flowchart kerja pemrograman NodeMCU modul lampu lalu

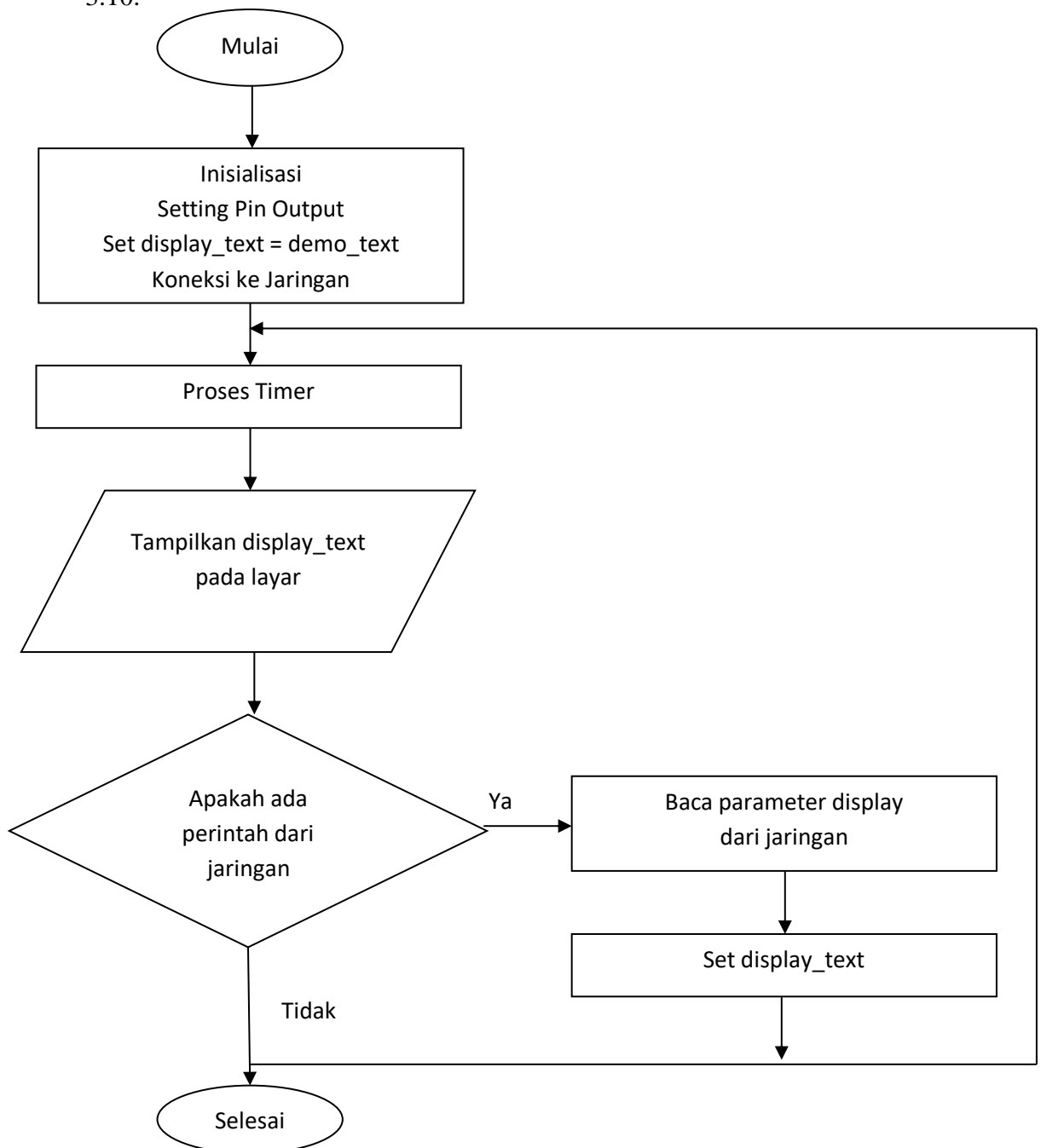
3.2.2.2 Perancangan Software Modul Display Informasi Lalu Lintas

Modul Display Informasi Lalu Lintas terdiri dari NodeMCU dan modul LED Display Matrix 32x8 pixel SPI. Fungsi yang akan disediakan pada modul ini antara lain:

- Menerima perintah dan mengolah perintah dari modul kendali utama secara wireless.

- Dapat mengirim informasi settingan yang tersimpan dan sedang diaplikasikan pada modul tersebut.
- Menampilkan informasi yang diterima dari modul utama ke alat peraga/display.

Pada modul ini, untuk menggerakkan LED Display Matrix menggunakan driver dari link <https://github.com/markruys/arduino-Max72xxPanel>. Flowchart kerja pemrograman NodeMCU pada modul display ini dapat dilihat pada gambar 3.10.



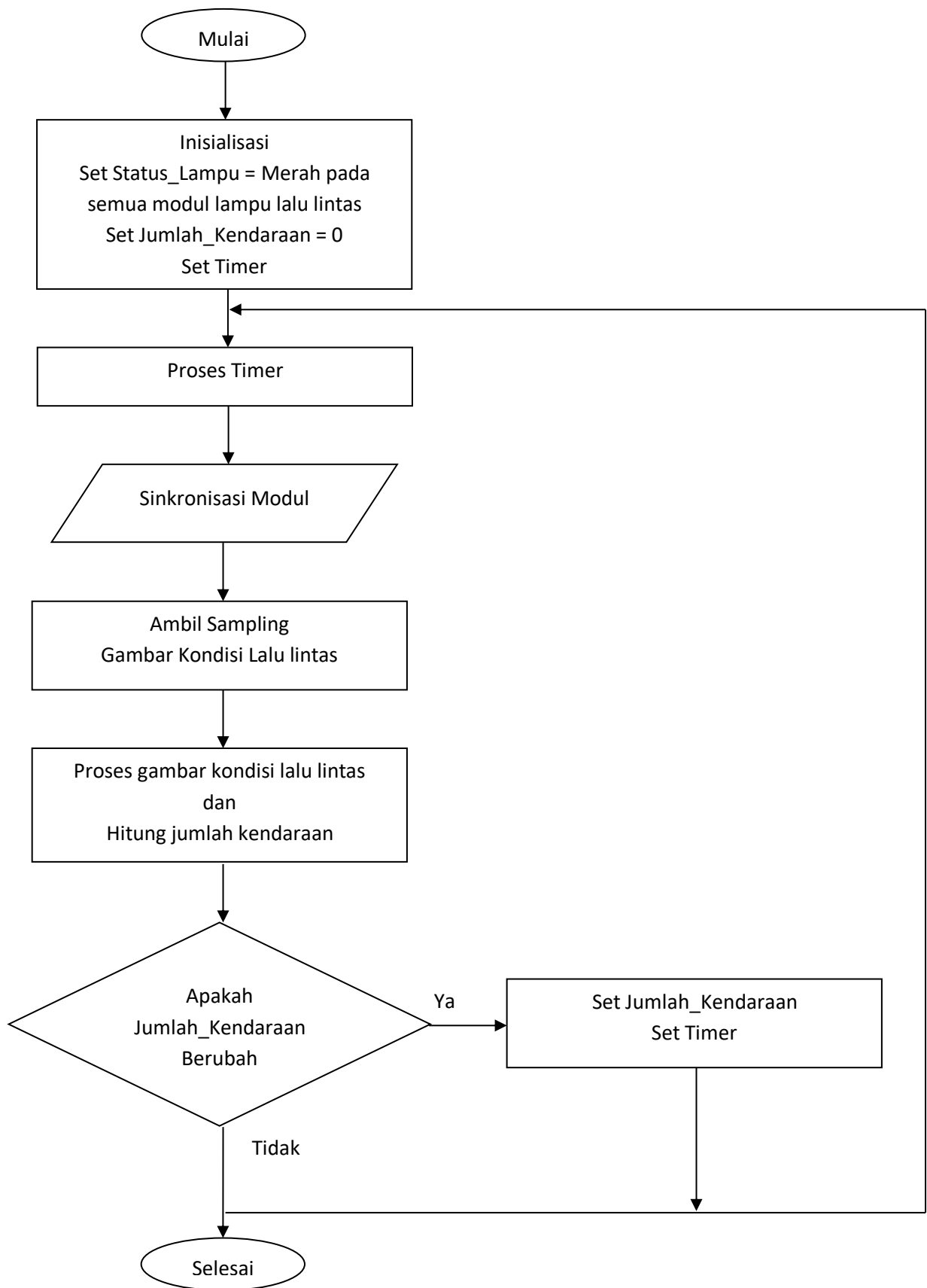
Gambar 3.10 Flowchart kerja pemrograman NodeMCU modul display informasi lalu lintas

3.2.2.3 Perancangan Software Modul Kendali Utama

Modul Kendali utama terdiri dari kumpulan skrip yang ditulis dalam PHP dan Python, modul ini memiliki antarmuka berbasis web untuk menampilkan informasi kondisi lalu lintas dan parameter lampu lalu lintas pada setiap kategori kondisi lalu lintas. Fungsi yang akan disediakan pada modul ini antara lain:

- Melayani dan menampilkan antarmuka berbasis web kepada operator.
- Mengirim data perintah ke modul-modul cabang seperti modul lampu lalu lintas dan modul display informasi lalu lintas.
- Menyimpan, menampilkan dan mengirim parameter setting untuk tiap-tiap modul.
- Mengambil dan memproses gambar untuk menghitung / memperoleh jumlah mobil yang terrekam oleh kamera.
- Membuat keputusan yang sesuai dengan parameter yang telah ditentukan, kemudian mengirim data tersebut ke modul lampu lalu lintas dan modul display informasi lalu lintas.

Flowchart kerja pemrograman OrangePi dari Sistem Kendali Traffic Light ini dapat dilihat pada gambar 3.11.



Gambar 3.11 Flowchart kerja pemrograman OrangePi modul kendali utama informasi lalu lintas

Teknologi yang digunakan pada skrip yang terdapat didalam modul kendali utama, dikelompokkan menjadi dua bagian. Teknologi PHP digunakan untuk menulis skrip antarmuka berbasis web yang berfungsi untuk menampilkan dan menyimpan parameter Sistem Kendali Traffic Light ini dan menghitung durasi merah menggunakan rumus pendekatan untuk menghitung siklus simpang bersinyal.

Rumus pendekatan durasi simpang bersinyal:

$$DM = ((DH + DK + (DT \times 2)) \times JL) - (DH+DK+DT).....(7)$$

Definisi:

DM = Durasi Merah

DK = Durasi Kuning

DT = Durasi Transisi (Lampu merah pada setiap modul lampu lalu lintas menyala bersamaan dengan maksud menandakan bahwa periode hijau telah berakhir)

JL = Jumlah Modul Lampu yang menyala berbeda siklus pewaktuan (fase siklus simpang bersinyal)

Teknologi Python dan OpenCV digunakan untuk menulis skrip sistem pendeteksian kendaraan melalui gambar yang ditangkap oleh kamera dengan metode pendeteksian blob, menentukan kategori kondisi lalu lintas berdasarkan jumlah kendaraan yang terdeteksi sesuai dengan parameter yang telah ditentukan, menghitung siklus pewaktuan menggunakan rumus pendekatan untuk menghitung siklus pewaktuan simpang bersinyal, dan mengirim parameter kepada masing-masing modul secara wireless.

Rumus pendekatan siklus pewaktuan simpang bersinyal 4 fase:

$$TD \text{ (Total Durasi)} = DM + DK + DH + DT(8)$$

$$DW = DH + DK + (DT \times 2)(9)$$

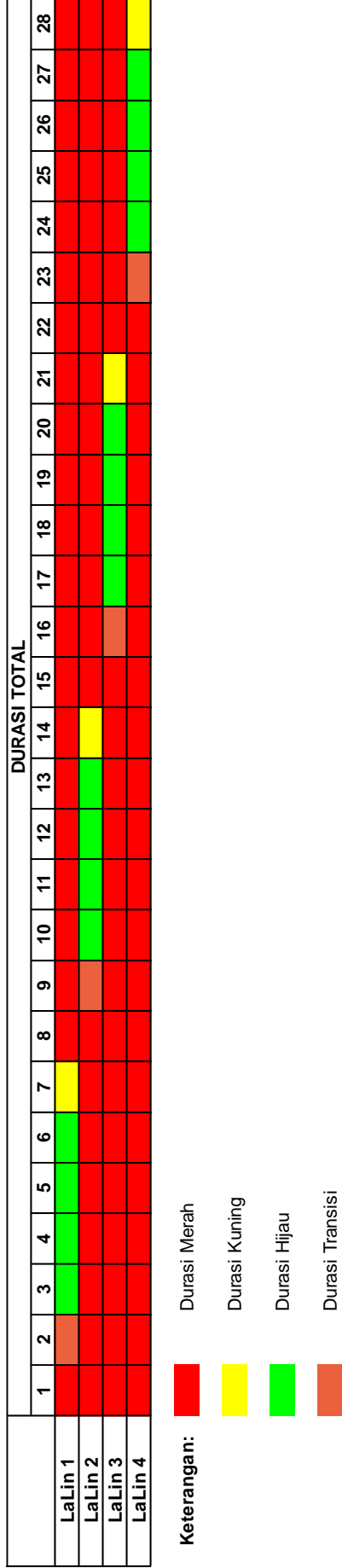
Posisi Pewaktuan Lampu 1 = DT

Posisi Pewaktuan Lampu 2 = Posisi Pewaktuan Lampu 1 + DW

Posisi Pewaktuan Lampu 3 = (DW x 2) + DT

Posisi Pewaktuan Lampu 4 = DM

Contoh hasil perhitungan siklus pewaktuan simpang bersinyal 4 fase dapat dilihat pada gambar 3.12. Sedangkan tampilan antarmuka berbasis web seperti yang terdapat pada gambar 3.13 dan 3.14.



Gambar 3.12 Siklus pewaktuan simpang bersinyal 4 fase

Parameter Kendali Lampu Lalu Lintas						
Kondisi Lalu Lintas	Jumlah Kendaraan Per Sampling	Status Lampu	Durasi			
			Merah	Kuning	Hijau	Tra
Padat		1 - Aktif	113	3	30	2
	Padat					
Ramai	7	1 - Aktif	83	3	20	2
	Ramai					
Lancar	5	1 - Aktif	53	3	10	2
	Lancar					
Lengang	3	0 - Non Aktif (Kuning Berkedip)	38	3	5	2
	Lengang					
Sistem Kendali	1 - Aktif	Durasi Kuning Berkedip	On	?	Off	1

Gambar 3.13 Tampilan antarmuka sistem kendali traffic light berbasis web

Traffic Management Berbasis IoT

Kamera Lalu Lintas	
	
Kondisi Lalu Lintas	
Jumlah Kendaraan	0
Status Lalu Lintas	Kondisi Lalu Lintas Lengang

Gambar 3.14 Tampilan panel pengaturan lampu lalu lintas

3.3 Manufaktur Prototipe Sistem Kendali Traffic Light

Pembuatan prototipe Sistem Kendali Traffic Light menggunakan 4 Modul Lampu Lalu Lintas, 1 Modul Display Informasi Lalu Lintas dan 1 Modul Kendali utama.

Maksud dari pembuatan prototipe tersebut adalah untuk mensimulasikan sebuah Sistem Kendali Traffic Light pada simpang bersinyal 4 jalur menggunakan pendeteksian

kondisi lalu lintas pada salah satu jalur simpang bersinyal tersebut. Pada IP Address yang digunakan pada setiap modul Sistem Kendali Traffic Light dapat dilihat pada tabel 3.3.

Tabel 3.3 IP address modul

MODUL	IP ADDRESS
Modul Kendali Utama	192.168.0.17
Modul Display Informasi Lalu Lintas	192.168.0.20
Modul Lampu Lalu Lintas 1	192.168.0.21
Modul Lampu Lalu Lintas 2	192.168.0.22
Modul Lampu Lalu Lintas 3	192.168.0.23
Modul Lampu Lalu Lintas 4	192.168.0.24

Pengoperasikan Sistem Kendali Traffic Light:

1. Memastikan ketinggian kamera sekitar 30 – 50 cm.
2. Mengaktifkan modul kendali utama dan menunggu hingga WiFi HotSpot ‘kendaliAPILL’ tersedia.
3. Mengaktifkan program ‘TrafficManager’ melalui terminal SSH pada modul kendali utama, cara mengaktifkannya sebagai berikut:
 - a. Login ke SSH pada ip address 192.168.0.17
 - b. User name: root Password: OrangePiZero
4. Pada terminal SSH, ketik: `python /var/www/html/TrafficManager.py`
5. Aktifkan modul-modul lainnya.

BAB IV

PENGUJIAN DAN ANALISIS

Pada bab ini akan dilakukan proses pengujian dan analisa alat yang telah selesai dibuat serta penjelasan mengenai hasil pengujian dan pengambilan data serta cara pengukurannya. Hal ini bertujuan untuk mengetahui tiap-tiap komponen dari alat dapat bekerja dengan baik sesuai dengan parameter yang telah ditentukan. Sehingga apabila terdapat system atau komponen yang belum berjalan dengan sesuai maka akan dilakukan perbaikan dan analisa.

Perancangan Sistem Simulasi Kendali *Traffic Light* dengan Metode Pendeteksian Gambar secara *Realtime* melalui Kamera Berbasis IoT ini membutuhkan akurasi pembacaan jumlah objek kendaraan yang akurat serta berjalan dengan baiknya tiap-tiap komponen. Adapun komponen dan system yang dilakukan pengukuran serta pengujian antara lain :

1. Pengujian Modul Display Informasi Lalu Lintas
2. Pengujian Modul Lampu Lalu Lintas
3. Pengujian Modul Kendali Utama dan Kamera
4. Pengujian Keseluruhan Sistem

4.1 Pengujian Modul Display Informasi Lalu Lintas

4.1.1. Pengujian Tegangan Output

4.1.1.1 Tujuan Pengujian

Pada system simulasi kendali *traffic light* ini terdapat modul *display* informasi lalu lintas untuk menampilkan informasi kondisi traffic light. Modul ini bekerja dengan menerima dan menampilkan data perintah dari kendali utama atau prosesor melalui jaringan wireless berupa kode string yang dapat dilihat pada Putty. Tujuan pengujian ini untuk melihat apakah modul ini dapat menjalankan perintah dan berfungsi dengan baik. Pengujian pada modul ini dapat dilihat pada gambar 4.1.

4.1.1.2 Cara Pengujian

1. Persiapkan modul display dan multimeter.
2. Modul display dinyalakan pada kondisi ON
3. Ukur tegangan menggunakan jarum multimeter pada pin VCC, CS, dan DIN pada display dot matrix. Cara pengukuran dapat dilihat pada gambar 4.1



Gambar 4.1 Pengukuran tegangan pada modul display informasi lalu lintas

4.1.1.3 Hasil Data :

Tabel 4.1 menunjukkan hasil pengukuran tegangan yang terdapat pada pin-pin Dot Matrix Display dapat dilihat pada tabel 4.1.

Tabel 4.1 Pengukuran tegangan pada dot matrix display

Pin	Tegangan Terukur (V)	Tegangan Referensi (V)	Status Hasil
VCC	3,16	-0.3 ~ 6	Normal
CS	3,22	-0.3 ~ 6	Normal
DIN	0	-0.3 ~ 6	Normal

4.1.1.4 Analisa dan Pembahasan

Dari hasil pengujian tegangan pada dot matrix display dapat diketahui bahwa tegangan tersebut masih sesuai dengan spesifikasi yang terdapat pada datasheet 8-Digit LED Display Drivers yaitu antara -0.3 volt sampai dengan 6 volt.

Berdasarkan pada hasil pengukuran tegangan seperti yang ditampilkan pada tabel 4.1, didapat bahwa modul display informasi lalu lintas ini berfungsi dengan baik dari hasil pengukuran tegangan.

4.1.2. Pengujian Software

4.1.2.1 Tujuan:

Untuk mengetahui apakah modul tersebut dapat menerima, memproses dan menampilkan teks, *level brightness*, kecepatan teks *scrolling* pada dot matrix display sesuai dengan data yang dikirim ke modul tersebut secara wireless.

4.1.2.2 Cara Pengujian:

Pengujian perangkat lunak dengan metode wireless dilakukan dengan menghubungkan laptop ke *WiFi HotSpot* yang disediakan oleh modul kendali utama, kemudian melakukan pengujian tersebut melalui terminal Secure Shell (dalam hal ini dengan menggunakan software PuTTY).

4.1.2.3 Hasil Pengujian

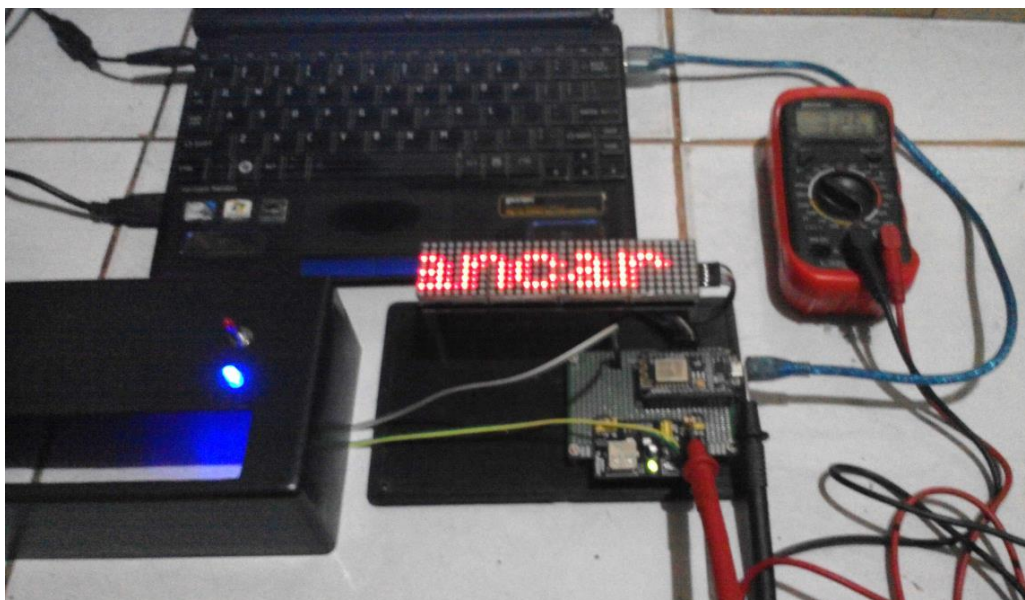
Gambar 4.2 memperlihatkan respon dari modul display sesaat setelah modul tersebut menerima data yang dikirim melalui *WiFi/Wireless* dan gambar 4.3 menunjukkan tampilan teks pada modul display informasi lalu lintas.

```
root@kendaliapill1:/var/www/html/pengujian# python test_modul_display.py
URL:
{'content': 'Content-Length: 447\r\n\r\n<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd"> \r\n<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en"> \r\n<head> \r\n <title>Terminal_ID</title> \r\n <meta http-equiv="content-type" content="text/html; charset=utf-8" /> \r\n</head> \r\n<body> \r\n[ssid=KendaliAPILL] [Status_Display=2] [Display_Text=Kondisi Padat] [Display_Brightness=0] [Display_Speed=150] </body> \r\n</html> \r\n\r\n', 'headers': <httplib.HTTPMessage instance at 0xb673b440>, 'httpcode': 200}

Respon Yang Diterima:
Content-Length: 447

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
<head>
  <title>Terminal_ID</title>
  <meta http-equiv="content-type" content="text/html; charset=utf-8" />
</head>
<body>
[ssid=KendaliAPILL] [Status_Display=2] [Display_Text=Kondisi Padat] [Display_Brightness=0] [Display_Speed=150] </body>
</html>
```

Gambar 4.2 Respon modul display informasi lalu lintas setelah menerima data secara wireless



Gambar 4.3 Gambar teks yang ditampilkan pada pengujian modul display informasi lalu lintas

4.1.2.4 Analisa dan Pembahasan

Dari hasil pengujian perangkat lunak seperti yang terlihat pada gambar 4.2 dan gambar 4.3 menunjukkan bahwa modul tersebut dapat bekerja secara baik, menerima data melalui wireless dan memprosesnya sampai menampilkan teks sesuai dengan kondisi lalu lintas yang telah di program.

Berdasarkan data dan hasil pengujian yang telah dilakukan pada modul display informasi lalu lintas tersebut, dapat disimpulkan bahwa modul dapat bekerja dengan baik sesuai perintah dan fungsi yang telah diatur. Sehingga modul dapat digunakan untuk percobaan system secara keseluruhan nantinya dan *real* di lapangan.

4.1.3 Pengujian Hardware

4.1.3.1 Tujuan :

Untuk menampilkan dan membuktikan teks yang berjalan apakah input yang diterima sesuai dengan output 4 kondisi lalu lintas sebagaimana mestinya. Dimana teks 4 kondisi lampu lalu lintas yang adalah “kondisi lengang”, “kondisi lancar”, “kondisi ramai”, dan “kondisi padat”.

4.1.3.2 Cara Pengujian

Pengujian modul display untuk menampilkan teks yang sesuai adalah sama dengan pengujian implementasi system simulasi *traffic light* secara keseluruhan, hanya saja dibedakan dengan input jumlah kendaraan yang ditangkap oleh kamera. Dimana dapat diambil contoh, dalam kondisi lengang sebanyak 2 kendaraan, kondisi lancar 4 kendaraan, kondisi ramai 7 kendaraan dan kondisi padat 10 kendaraan yang ditangkap kamera. Namun karena berupa teks berjalan, maka hanya dapat ditampilkan berupa tangkapan gambar saja dari modul display tersebut dalam hasil pengujian.

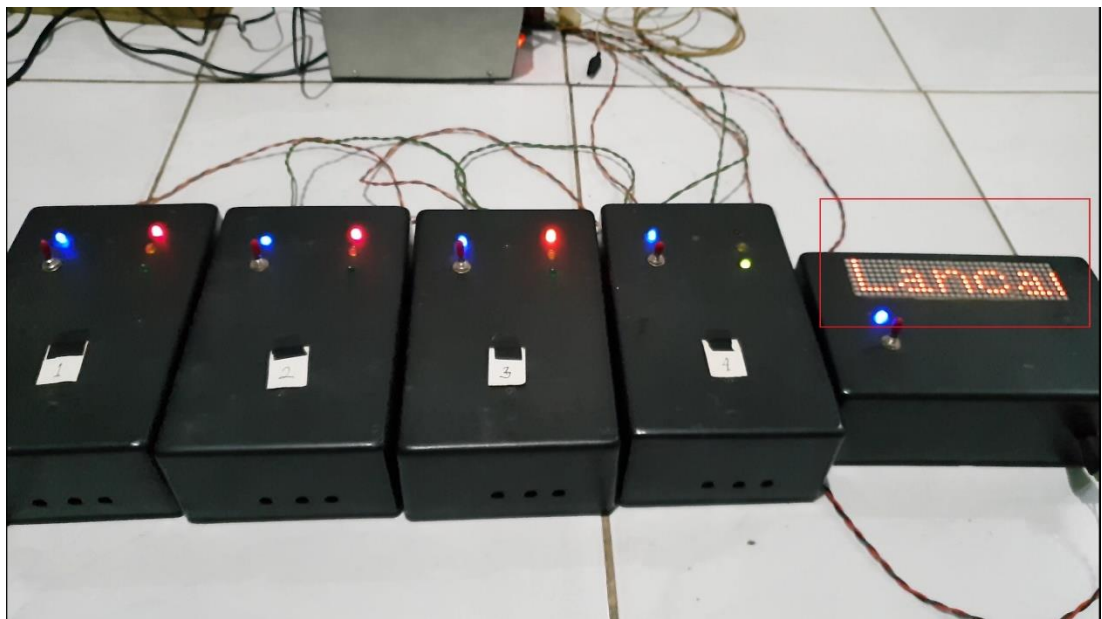
4.1.3.3 Hasil Pengujian

Gambar 4.4 memperlihatkan modul display dengan teks bertuliskan “kondisi lengang” yang mana sesuai dengan jumlah objek yang ditangkap oleh kamera sebanyak 2. Berikutnya pada gambar 4.5 memperlihatkan modul display dengan teks bertuliskan “kondisi lancar” yang mana pula sesuai dengan jumlah objek yang ditangkap kamera sebanyak 4. Kemudian gambar 4.6 memperlihatkan modul display dengan teks bertuliskan “kondisi ramai” yang mana sesuai dengan jumlah objek yang ditangkap kamera sebanyak 7. Dan terakhir gambar 4.7 memperlihatkan modul display dengan

teks bertuliskan “kondisi padat” yang mana sesuai dengan jumlah objek yang ditangkap kamera sebanyak 10.



Gambar 4.4 Modul display kondisi lengang



Gambar 4.5 Modul display kondisi lancar



Gambar 4.6 Modul display kondisi ramai



Gambar 4.7 Modul display kondisi padat

4.1.3.4 Analisa dan Pembahasan

Karena pengujian ini berupa teks atau gambar yang bergerak, maka pengujian ini hanya dapat ditampilkan berupa tangkapan gambar pada modul display tersebut. Dari hasil pengujian *hardware* modul display yang terlihat pada gambar 4.4, 4.5, 4.6 dan 4.7 menunjukkan bahwa modul display dapat menampilkan teks berjalan sesuai kondisi *traffic light*. Sehingga pengujian *hardware* modul ini dapat dikatakan berhasil dan modul *display* dapat diimplementasikan pada percobaan sesungguhnya.

Apabila terjadi ketidakcocokan teks yang ditampilkan dengan kondisi lalu lintas, terdapat beberapa factor yang mempengaruhi seperti pembacaan jumlah objek kendaraan yang error, akses jaringan *wireless* yang kurang baik, kode yang di kirim dari modul kendali utama salah, serta komponen led display yang bermasalah. Namun dalam pengujian ini modul display berfungsi dengan baik dan dapat menampilkan teks sesuai kondisi lalu lintas.

4.2 Pengujian Modul Lampu Lalu Lintas

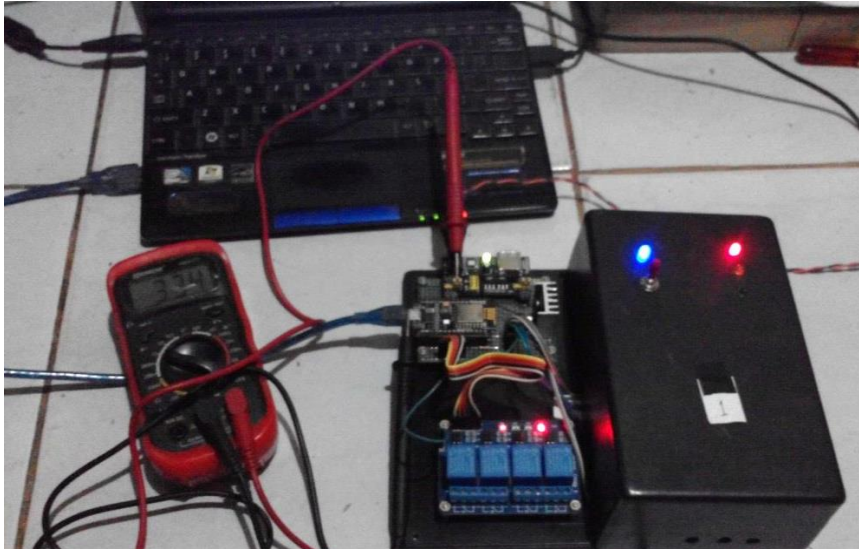
4.2.1 Pengujian Tegangan Output

4.2.1.1 Tujuan:

Untuk mengetahui tegangan yang bekerja pada jack DC yang berasal dari adaptor, apakah masih dalam kategori normal sesuai dengan spesifikasi komponen.

4.2.1.2 Cara Pengujian

1. Mempersiapkan modul lampu lalu lintas yang akan diuji, modul kendali utama dan multimeter
2. Modul lampu lalu lintas dan modul kendali utama dalam keadaan ON
3. Memastikan modul lampu lalu lintas terhubung ke Hot Spot modul kendali utama
4. Menjalankan software pengujian modul lampu lalu lintas dengan tujuan men-set modul lampu lalu lintas pada mode yang sesuai dengan pengujian (mode blink dan mode normal)
5. Ukur tegangan menggunakan jarum multimeter pada pin Jack DC pada modul power supply. Cara pengukuran dapat dilihat pada gambar 4.4



Gambar 4.8 Pengukuran tegangan pada modul lampu lalu lintas

4.2.1.3 Hasil Data :

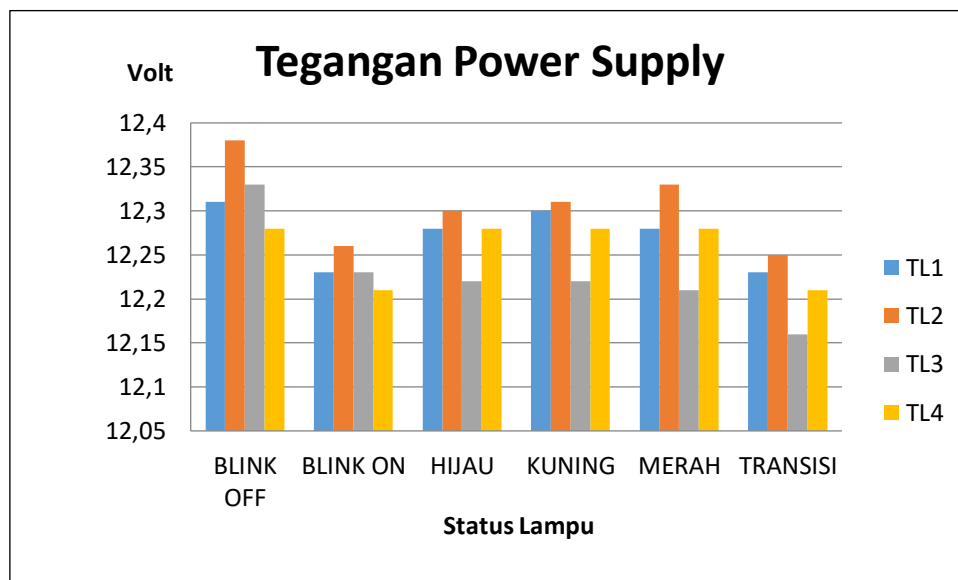
Pada modul lampu lalu lintas terdapat status lampu dimana status tersebut menunjukkan kondisi lalu lintas dan perintah terhadap kendaraan. Status ini memiliki indicator berupa lampu mana yang menyala dan apa arti dari tanda lampu tersebut. Yaitu sebagai berikut :

- Blink off (B.off) dan Blink on (B.on) yaitu kondisi dimana tidak ada kendaraan sampai dengan 2 atau dikategorikan lalu lintas sepi pada jalanan dan dengan indicator lampu kuning nyala berkedip.
- Hijau yaitu kondisi dimana lampu memberi perintah pada kendaraan di persimpangan tertentu untuk maju dengan indicator lampu hijau menyala selama durasi yang telah ditentukan.
- Kuning yaitu kondisi dimana lampu memberi perintah pada kendaraan di persimpangan tertentu untuk hati-hati atau persiapan awal mula jalan dan akan menuju pada kondisi berhenti dengan lampu kuning menyala selama durasi yang telah ditentukan.
- Merah yaitu kondisi dimana lampu memberi perintah pada kendaraan di persimpangan tertentu untuk berhenti dengan kondisi lampu merah menyala selama durasi yang telah ditentukan.
- Transisi yaitu kondisi dimana *traffic light* (TL) memberi jeda atau persiapan kepada kendaraan untuk perpindahan kondisi pada persimpangan satu ke persimpangan lainnya agar kendaraan tidak terjadi bentrok, dimana lampu merah dan kuning nyala secara bersamaan selama durasi yang telah ditentukan.

Hasil data pengukuran tegangan modul *traffic light* (TL) dapat dilihat pada table 4.2 sebagai berikut.

Tabel 4.2 Pengukuran tegangan power supply modul traffic light

Status_Lampu	Tegangan Power Supply(V)			
	TL 1	TL 2	TL 3	TL 4
Blink OFF	12,31	12,38	12,33	12,28
B.ON	12,23	12,26	12,23	12,21
HIJAU	12,28	12,30	12,22	12,28
KUNING	12,30	12,31	12,22	12,28
MERAH	12,28	12,33	12,21	12,28
TRANSISI	12,23	12,25	12,16	12,21
Rata-rata	12,27	12,31	12,23	12,26
Error (%)	2,16	2	2,72	2,32



Gambar 4.9 Grafik tegangan power supply

4.2.1.4 Analisa dan Pembahasan

Dari hasil pengukuran tegangan power supply didapat tegangan tanpa beban 12,5 volt, hasil tersebut sesuai dengan spesifikasi komponen. Pada pengukuran ini didapat tegangan power supply terendah 12,16 volt dan tertinggi 12,38 volt serta dengan tegangan rata-rata yaitu 12,2675 volt dimana angka tersebut masih dalam kategori baik. Dari rata-rata tegangan yang diukur memiliki persentase kesalahan rata-rata sebesar 2,405% dimana angka tersebut cukup kecil dan tidak mempengaruhi kinerja alat.

Penurunan tegangan terhadap tegangan tanpa beban terjadi dikarenakan terdapat beban pada komponen yang mendapatkan supply power seperti led, relay dan

NodeMCU. Namun penurunan tegangan tersebut masih dalam kategori normal sesuai spesifikasi perangkat.

4.2.2 Pengujian Tegangan Regulator

4.2.2.1 Tujuan

Untuk mengetahui tegangan output dari regulator 7805, apakah masih dalam kategori normal sesuai dengan spesifikasi masing komponen.

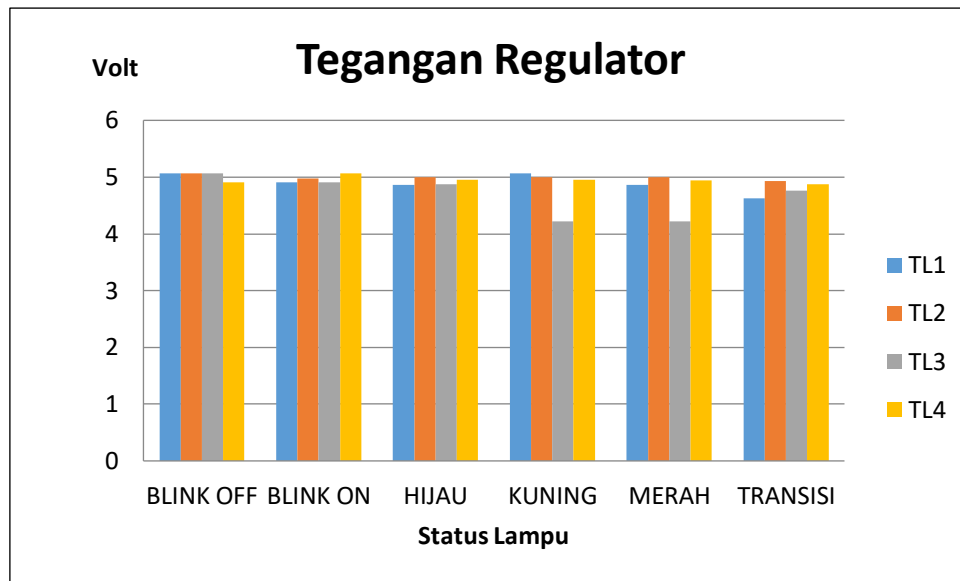
4.2.2.2 Cara Pengujian

1. Mempersiapkan modul lampu lalu lintas yang akan diuji, modul kendali utama dan multimeter
2. Modul lampu lalu lintas dan modul kendali utama dalam keadaan ON
3. Memastikan modul lampu lalu lintas terhubung ke Hot Spot modul kendali utama
4. Menjalankan software pengujian modul lampu lalu lintas dengan tujuan menset modul lampu lalu lintas pada mode yang sesuai dengan pengujian (mode blink dan mode normal)
5. Ukur tegangan menggunakan jarum multimeter pada pin Output dari IC Regulator 7805 pada modul power supply. Cara pengukuran dapat dilihat pada gambar 4.3

4.2.2.3 Hasil Data :

Tabel 4.3 Pengukuran tegangan regulator modul traffic light

Status_Lampu	Kondisi Lampu			Tegangan Regulator (V)			
	Hijau	Kuning	Merah	TL 1	TL 2	TL 3	TL 4
B.OFF	OFF	OFF	OFF	5,07	5,07	5,07	4,91
B.ON	OFF	ON	OFF	4,91	4,98	4,91	5,07
HIJAU	ON	OFF	OFF	4,86	5,00	4,87	4,95
KUNING	OFF	ON	OFF	5,07	5,00	4,22	4,95
MERAH	OFF	OFF	ON	4,86	5,00	4,22	4,94
TRANSISI	OFF	ON	ON	4,63	4,93	4,76	4,87
Rata-rata				4,90	5,00	4,68	4,95
Error (%)				7,4	1,4	15,6	2,6



Gambar 4.10 Grafik tegangan regulator

4.2.2.4 Analisa dan Pembahasan

Hasil pengukuran tegangan output regulator yang tertera pada tabel 4.3, dapat diketahui bahwa tegangan tertinggi ada pada 5,07 Volt dan terendah pada 4,22 Volt. Pada traffic light 3 memiliki persentase error sebesar 15,6%. Cukup tinggi dibanding yang lainnya. Namun hal tersebut masih dalam kategori normal karena beban pada tegangan dalam jumlah volt hanya penurunan kecil.

Dari turunnya hasil tegangan tersebut memiliki rata-rata persentase eror sebesar 6,75% dan rata-rata standar deviasi sebesar 0,2325. Nilai persentase error tersebut dalam komponen elektronika masih dibawah *Relative Limiting Error* pada alat ukur listrik yang sebesar 10%. Jadi dapat dikatakan rata-rata error pada tegangan regulator masih diambang batas normal.

Dari hasil pengukuran output regulator dan berdasarkan spesifikasi menurut datasheet UA7800 hal tersebut masih dalam kategori normal dan berfungsi dengan baik.

4.2.3 Pengujian Tegangan LED

4.2.3.1 Tujuan:

Untuk mengetahui tegangan pada pin anoda LED setelah ditempatkan resistor dengan tahanan 100 ohm. Resistor ini sendiri sebagai pembatas arus agar arus yang

melewati led tidak melebihi batas maksimum arus led itu sendiri antara 25-30mA, serta apakah sesuai dengan spesifikasi tiap komponen.

4.2.3.2 Cara Pengujian:

1. Mempersiapkan modul lampu lalu lintas yang akan diuji, modul kendali utama dan multimeter
2. Modul lampu lalu lintas dan modul kendali utama dalam keadaan ON
3. Memastikan modul lampu lalu lintas terhubung ke Hot Spot modul kendali utama
4. Menjalankan software pengujian modul lampu lalu lintas dengan tujuan menset modul lampu lalu lintas pada mode yang sesuai dengan pengujian (mode blink dan mode normal)
5. Ukur tegangan menggunakan jarum multimeter pada pin anoda dari LED

4.2.3.3 Hasil Data :

Hasil pengujian data dilihat pada tabel 4.4.

Tabel 4.4 Pengukuran tegangan pin LED

Status_Lampu	Tegangan Pada LED (V)											
	TL 1			TL 2			TL 3			TL 4		
	Hijau	Kuning	Merah	Hijau	Kuning	Merah	Hijau	Kuning	Merah	Hijau	Kuning	Merah
B.OFF	0	0	0,1	0	0	0,04	0	0	0,06	0,06	0	0
B.ON	0	1,99	0,1	0	2,11	0,04	0	2,11	0,06	0,06	2,05	0
HIJAU	2,01	0	0,1	1,96	0	0,04	1,97	0	0,06	2,05	0	0
KUNING	0	1,99	0,1	0	2,09	0,04	0	2,15	0,06	0,06	2,05	0
MERAH	0	0	1,99	0	0	1,81	0	0	1,82	0,06	0	1,93
TRANSISI	0	1,99	1,92	0	2,08	1,8	0	2,05	1,81	0,06	2,05	1,93

4.2.3.4 Analisa dan Pembahasan

Dari hasil pengukuran tegangan pada pin anoda dari LED seperti yang tertera pada tabel 4.4, dapat diketahui tegangan pada logika HIGH tertinggi ada pada 2,15 Volt dan terendah ada pada 1,80 Volt. Pengukuran ini dilakukan pada led dengan tahanan resistor untuk mendapatkan arus yang dilewati led antara 10-20 mA. Jika tanpa menggunakan resistor, arus yang dilewati akan lebih dari 20 mA. Jika itu terjadi lampu led akan menyala lebih terang dan akan terbakar jika menyala dalam waktu lama. Menurut tabel elektrikal karakteristik yang terdapat pada datasheet COM-09590-YSL R531R3D-D2, maka dapat dikatakan bahwa LED pada modul lampu lalu lintas bekerja dengan normal.

4.2.4 Pengujian Tegangan NodeMCU

4.2.4.1 Tujuan:

Untuk mengetahui tegangan pada beberapa pin nodemcu, apakah sesuai dengan spesifikasi masing komponen.

4.2.4.2 Cara Pengujian:

1. Mempersiapkan modul lampu lalu lintas yang akan diuji, modul kendali utama dan multimeter
2. Modul lampu lalu lintas dan modul kendali utama dalam keadaan ON
3. Memastikan modul lampu lalu lintas terhubung ke Hot Spot modul kendali utama
4. Menjalankan software pengujian modul lampu lalu lintas dengan tujuan menset modul lampu lalu lintas pada mode yang sesuai dengan pengujian (mode blink dan mode normal)
5. Ukur tegangan menggunakan jarum multimeter pada pin 3V3, D1, D2 dan D3 pada NodeMCU

4.2.4.3 Hasil Data :

Hasil pengujian data dilihat pada tabel 4.5.

Tabel 4.5 Tegangan NodeMCU

Status Lampu	Tegangan Pada NodeMCU (V)																			
	Traffic Light 1					Traffic Light 2					Traffic Light 3					Traffic Light 4				
	3V3	D1	D2	D3	3V3	D1	D2	D3	3V3	D1	D2	D3	3V3	D1	D2	D3	3V3	D1	D2	D3
B.OFF	3,3	0	0	0,10	3,27	0	0	0,04	3,28	0	0	0,06	3,25	0	0	0,06	3,25	0	0	0,06
B.ON	3,3	0	3,03	0,10	3,27	0	3,01	0,04	3,28	0	3,03	0,06	3,25	0	2,97	0,06	3,25	0	2,97	0,06
HIJAU	3,3	3,03	0	0,10	3,27	2,96	0	0,04	3,28	2,99	0	0,06	3,25	2,99	0	0,06	3,25	2,99	0	0,06
KUNING	3,3	0	3,02	0,10	3,27	0	3,00	0,04	3,28	0	3,06	0,06	3,25	0	2,97	0,06	3,25	0	2,97	0,06
MERAH	3,3	0	0	3,02	3,27	0	0	2,94	3,28	0	0	3,01	3,25	0	0	2,93	3,25	0	0	2,93
TRANSISI	3,3	0	3,01	3,02	3,26	0	3,00	2,93	3,28	0	3,06	3,01	3,25	0	2,96	3,01	3,25	0	2,96	2,92
Rata-rata	3,3	3,268					3,28					3,25								
Error (%)	0,00	1,21					0,61					1,52								

4.2.4.4 Analisa dan Pembahasan

Dari hasil pengukuran tegangan pada beberapa pin NodeMCU seperti yang tertera pada tabel 4.5, dapat diketahui tegangan I/O (Input dan Output) pada logika HIGH tertinggi ada pada 3,3 Volt dan terendah ada pada 3,25 Volt.

Menurut tabel elektrikal karakteristik yang terdapat pada datasheet ESP8266EX, tegangan I/O pada logika LOW ada pada level 3,6 Volt dan terendah ada pada 2,7 Volt. Sedangkan tegangan kerja pada pin 3V3 tertinggi ada pada 3,6 Volt dan terendah ada pada 3,0 Volt. Maka dapat dikatakan bahwa NodeMCU pada modul lampu lalu lintas bekerja dengan normal berdasarkan tegangan yang bekerja.

4.2.5 Pengujian Software

4.2.5.1 Tujuan

Untuk mengetahui apakah modul tersebut dapat menerima, memproses dan mengeksekusi durasi lampu, status lampu dan timer sesuai dengan data yang dikirim ke modul tersebut baik secara wireless.

4.2.5.2 Cara Pengujian

1. Mempersiapkan modul lampu lalu lintas yang akan diuji, modul kendali utama dan laptop
2. Modul lampu lalu lintas dan modul kendali utama dalam keadaan ON
3. Memastikan modul lampu lalu lintas terhubung ke Hot Spot modul kendali utama
4. Menjalankan software pengujian melalui terminal SSH (dalam hal ini penulis menggunakan bantuan software PuTTY).

4.2.5.3 Hasil Pengujian :

Pada gambar 4.7 adalah respon dari Modul Lampu Lalu Lintas yang berisikan parameter yang diterima/yang sedang diaplikasikan oleh modul tersebut.

```

root@kendaliapill:/var/www/html/pengujian# python test_modul_lalin_1.py
URL:
{'content': 'Content-Length: 540\r\n\r\n<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd"> \r\n<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en"> \r\n<head> \r\n <title>Terminal_ID</title> \r\n <meta http-equiv="content-type" content="text/html; charset=utf-8" /> \r\n</head> \r\n<body> \r\n[ssid=KendaliAPILL] [Status_Lampu=1] [Durasi_Merah=53] [Durasi_Kuning=3] [Durasi_Hijau=10] [Durasi_Transisi=2] [Durasi_Blink_On=2] [Durasi_Blink_Off=1] [Status_Blink=1] [Status_Relay=1] [Timer_Current=0] </body> \r\n</html> \r\n\r\n', 'headers': <httplib.HTTPMessage instance at 0xb6808440>, 'httpcode': 200}

Respon Yang Diterima:
Content-Length: 540

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
<head>
  <title>Terminal_ID</title>
  <meta http-equiv="content-type" content="text/html; charset=utf-8" />
</head>
<body>
[ssid=KendaliAPILL] [Status_Lampu=1] [Durasi_Merah=53] [Durasi_Kuning=3] [Durasi_Hijau=10] [Durasi_Transisi=2] [Durasi_Blink_On=2] [Durasi_Blink_Off=1] [Status_Blink=1] [Status_Relay=1] [Timer_Current=0] </body>
</html>

root@kendaliapill:/var/www/html/pengujian# █

```

Gambar 4.11 Hasil pengujian modul lampu lalu lintas secara wireless

4.2.5.4 Analisa dan Pembahasan

Seperti yang dapat dilihat pada gambar 4.7 yang merupakan respon setelah data parameter berupa string dikirim ke modul. Dari hasil pengujian dan respon dari modul lampu lalu lintas setelah dikirim data secara wireless, maka dapat disimpulkan bahwa software pada modul lampu lalu lintas bekerja normal.

4.3 Pengujian Modul Kendali Utama

Pengujian pada Modul Kendali Utama hanya dapat dilakukan secara software dengan beberapa cara, diantaranya:

- Menguji Perantara Web.
- Menguji Sudut Kemiringan Kamera.
- Perhitungan Sudut Efisien Objek Kamera

4.3.1 Pengujian Perantara WEB

4.3.1.1 Tujuan

Tujuan pengujian ini adalah untuk mengetahui apakah perantara web tersebut bekerja dengan baik sesuai dengan fungsinya yaitu menampilkan parameter dan menyimpan parameter dari Sistem Kendali Lampu Lalu Lintas ini.

4.3.1.2 Cara Pengujian

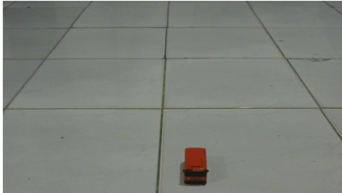
Cara pengujian ini adalah dengan mengakses perantara web tersebut menggunakan internet browser (dalam hal ini pengguna menggunakan program FireFox) dengan alamat URL 192.168.0.17 (ip address dari modul kendali utama), kemudian mengubah parameter yang tersedia pada perantara web tersebut.

4.3.1.3 Hasil Pengujian

Pada gambar 4.8 adalah tampilan dari perantara web yang menampilkan Parameter Kendali Lampu Lalu Lintas.

Traffic Management Berbasis IoT

Kamera Lalu Lintas



Kondisi Lalu Lintas

Jumlah Kendaraan	1
Status Lalu Lintas	Kondisi Lalu Lintas Lengah

Display Lalu Lintas

Status Display	1 - Teks Kondisi Lalin	ID Modul	Display-Lalin-1
Kecepatan Rotasi	150	Sandi Modul	●●●●●●●●
Brightness	0.5	SSID	KendaliAPILL
Status Modul	Offline	Password	●●●●●●●●
		IP Address	192.168.0.20

Parameter Kendali Lampu Lalu Lintas

Kondisi Lalu Lintas	Jumlah Kendaraan Per Sampel	Status Lampu	Durasi			
			Merah	Kuning	Hijau	Tran
Padat	113	1 - Aktif	113	3	30	2
Ramai	83	1 - Aktif	83	3	20	2
Lancar	53	1 - Aktif	53	3	10	2
Lengah	38	0 - Non Aktif (Kuning Berkedip)	38	3	5	2
Sistem Kendali Otomatis	1 - Aktif	Durasi Kuning Berkedip	On	2	Off	1
Durasi Pengambilan Sampel	15 detik					

Konfigurasi Lampu Lalu Lintas

Indikator	Lampu Lalin 1	Lampu Lalin 2	Lampu Lalin 3	Lampu Lalin 4
Lampu Lalin Eksternal	1 - Aktif	1 - Aktif	1 - Aktif	1 - Aktif
ID Modul	Lampu-Lalin-1	Lampu-Lalin-2	Lampu-Lalin-3	Lampu-Lalin-4
Sandi Modul	●●●●●●●●	●●●●●●●●	●●●●●●●●	●●●●●●●●
SSID	KendaliAPILL	KendaliAPILL	KendaliAPILL	KendaliAPILL
Password	●●●●●●●●	●●●●●●●●	●●●●●●●●	●●●●●●●●
IP Address	192.168.0.21	192.168.0.22	192.168.0.23	192.168.0.24
Status Modul	Offline	Offline	Offline	Offline

Gambar 4.12 Tampilan parameter kendali lampu lalu lintas pada perantara web

4.3.1.4 Analisa dan Pembahasan

Pada saat Parameter Durasi Hijau diset pada angka 10 dan kemudian menekan tombol simpan yang terdapat dibawah form tersebut, alat dapat menyimpan data tersebut dan kemudian menampilkan kembali nilai data tersebut. Hal ini menandakan bahwa perantara web telah bekerja sesuai dengan fungsinya.

4.3.2 Pengujian Sudut Kemiringan Kamera

4.3.2.1 Tujuan

Tujuan pengujian ini adalah untuk mendapatkan nilai akurasi dari pendeteksian kamera antara kendaraan yang diletakkan didepan kamera dan jumlah kendaraan yang terdeteksi terhadap sudut kemiringan tertentu, sehingga dapat mengetahui sudut kemiringan kamera yang terbaik.

4.3.2.2 Cara Pengujian

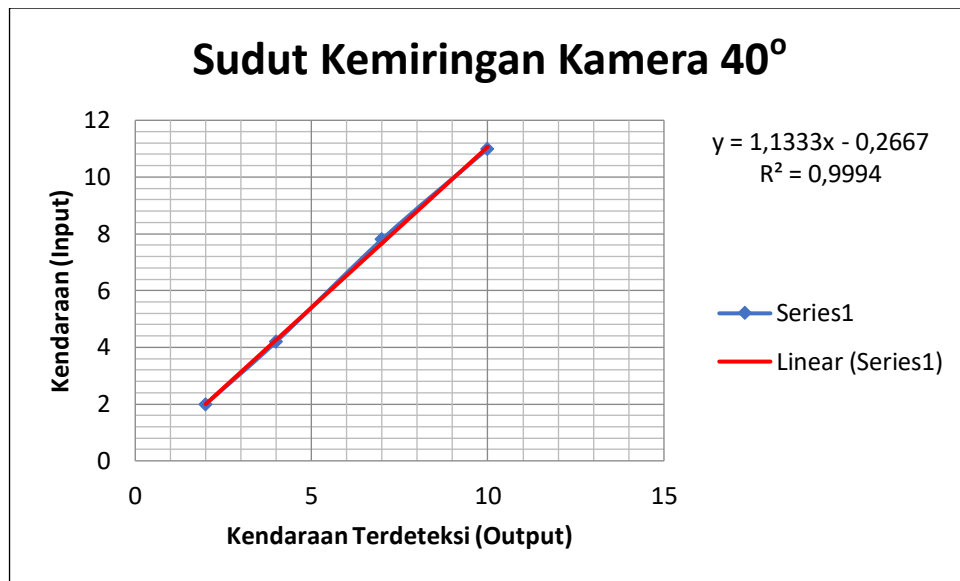
Cara pengujian ini adalah kamera dipasang pada tiang dengan ketinggian 50 cm dari permukaan lantai, kendaraan yang digunakan dalam pengujian sebanyak 2, 4, 7 dan 10 kendaraan yang diletakkan pada lantai dengan jarak 40 cm dari garis sumbu kamera. Percobaan dilakukan sebanyak 5 kali pada setiap jumlah kendaraan dengan sudut kemiringan kamera 40°, 45°, 50°, 55° dan 60°.

4.3.2.3 Hasil Pengujian

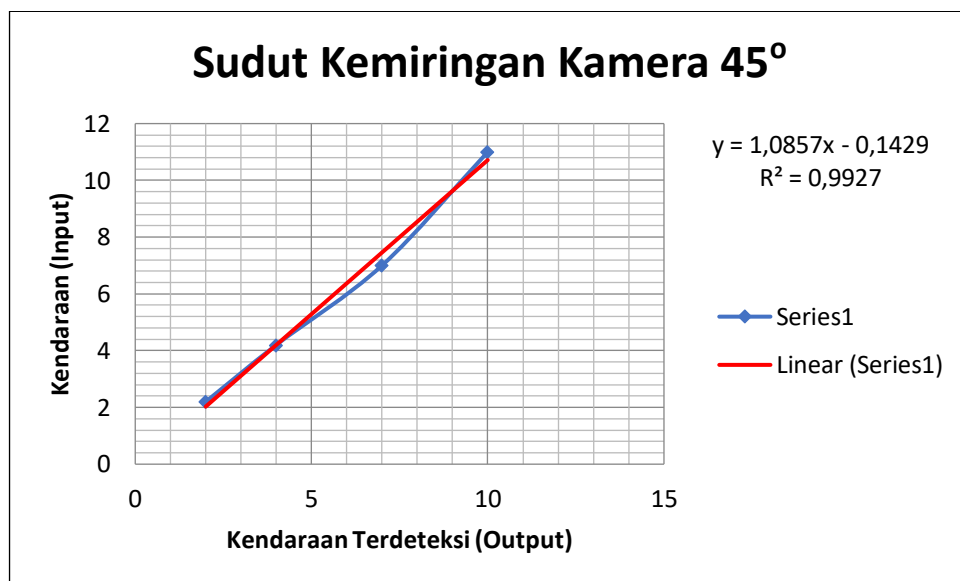
Hasil dari pengujian sudut kemiringan kamera diperlihatkan pada tabel 4.6. Perhitungan regresi liner menggunakan alat bantu dari Microsoft Excel, pada gambar 4.9 memperlihatkan grafik hasil perhitungan regresi liner dari hasil rata-rata pengujian sudut kemiringan 40°, gambar 4.10 untuk sudut kemiringan 45°, gambar 4.11 untuk sudut kemiringan 50°, gambar 4.12 untuk sudut kemiringan 55° dan gambar 4.13 untuk sudut kemiringan 60°.

Tabel 4.6 Pengujian sudut kemiringan kamera

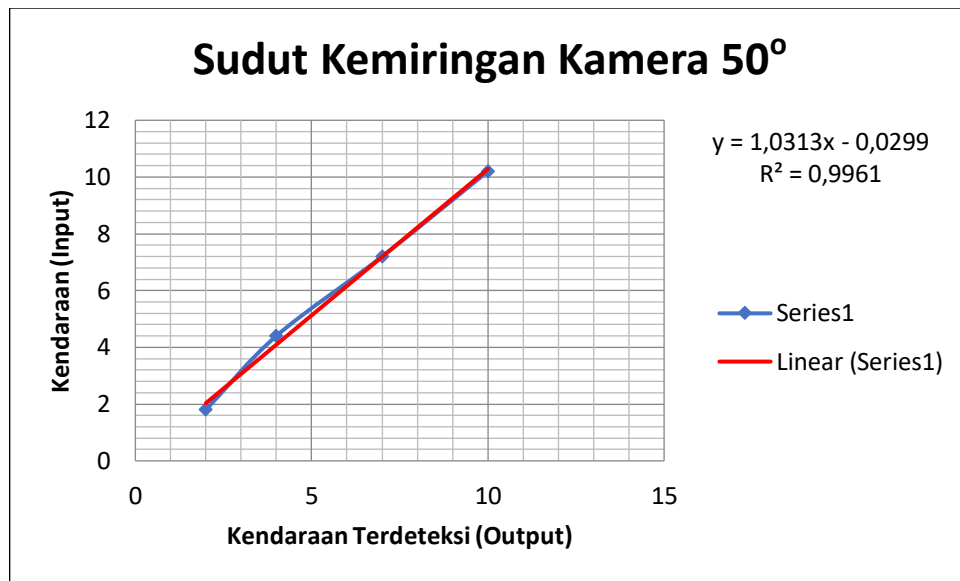
Jumlah Kendaraan	Percobaan ke	Sudut Kemiringan Kamera				
		40°	45°	50°	55°	60°
2	1	2	2	2	2	2
	2	2	2	2	2	2
	3	2	3	2	2	2
	4	2	2	1	3	2
	5	2	2	2	2	2
Rata-rata		2	2,2	1,8	2,2	2
4	1	4	4	4	4	4
	2	5	4	5	4	4
	3	4	4	5	4	5
	4	4	4	4	4	4
	5	4	5	4	4	4
Rata-rata		4,2	4,2	4,4	4	4,2
7	1	8	6	7	7	7
	2	7	7	7	7	7
	3	7	8	7	7	6
	4	7	7	8	7	8
	5	10	7	7	7	7
Rata-rata		7,8	7	7,2	7	7
10	1	11	10	11	10	10
	2	12	11	11	10	10
	3	10	12	9	10	10
	4	11	10	10	10	10
	5	11	12	10	10	10
Rata-rata		11	11	10,2	10	10
R²		0,9994	0,9927	0,9961	0,9996	0,9993
Error (%)		0,06	0,73	0,93	0,04	0,07



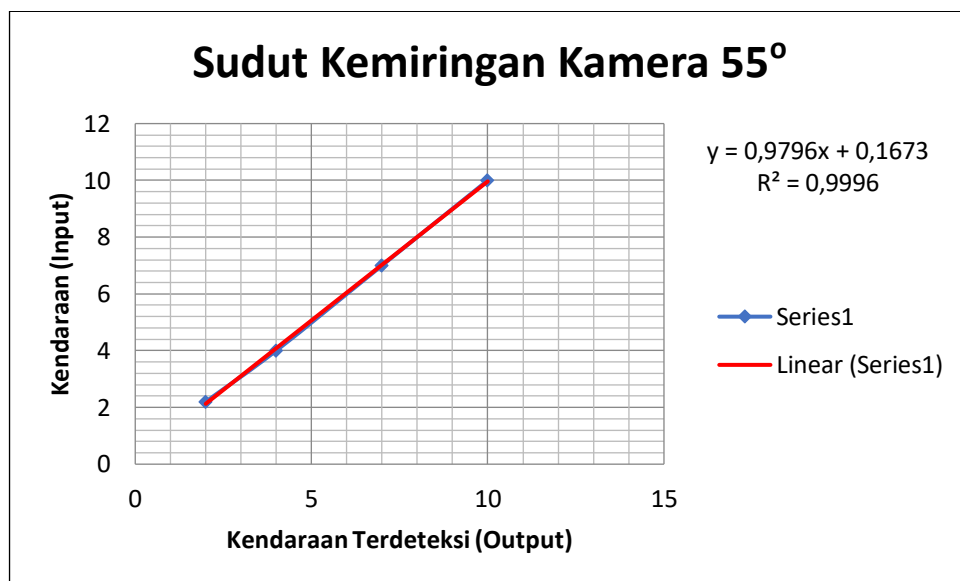
Gambar 4.13 Grafik hasil pengujian sudut kemiringan kamera 40°



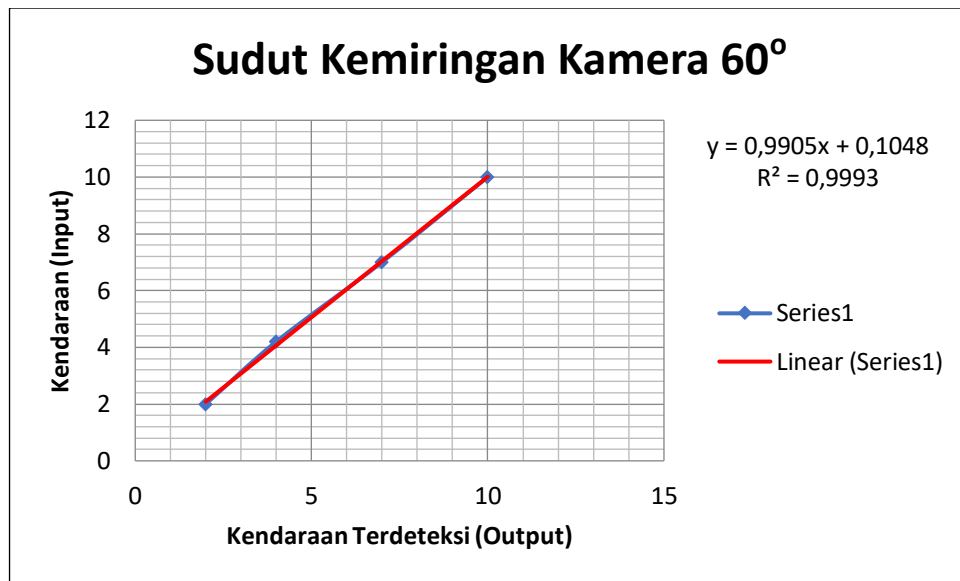
Gambar 4.14 Grafik hasil pengujian sudut kemiringan kamera 45°



Gambar 4.15 Grafik hasil pengujian sudut kemiringan kamera 50°



Gambar 4.16 Grafik hasil pengujian sudut kemiringan kamera 55°



Gambar 4.17 Grafik hasil pengujian sudut kemiringan kamera 60°

4.3.2.4 Analisa dan Pembahasan

Pada pengujian sudut kemiringan ini diuji berdasarkan spesifikasi webcam Logitech C270. Dimana webcam ini dapat dirubah tilting nya pada sudut 0° sampai dengan 90°. Untuk pembacaan objek yang jelas dan akurat, pengujian sudut kemiringan ini hanya dapat dilakukan pada sudut antara 40° sampai 60° saja. Jika sudut lebih atau kurang dari itu, objek akan terlihat lebih jauh dan lebih kecil sehingga mempengaruhi akurasi pembacaan objek dan menjadi tidak akurat.

Pada tabel 4.6 memperlihatkan nilai R^2 dan Error dari hasil pengujian sudut kemiringan seperti diperlihatkan pada gambar 4.9 sampai dengan gambar 4.13. Dari tabel tersebut dapat disimpulkan bahwa nilai Error pada hasil pengujian sudut kemiringan kamera yang terkecil ada pada sudut 55° dengan nilai Error sebesar 0,04%. Maka sudut kemiringan kamera 55° digunakan untuk pengujian secara keseluruhan berdasarkan pengujian yang telah disebutkan diatas.

4.3.3 Perhitungan Sudut Efisien Objek Kamera

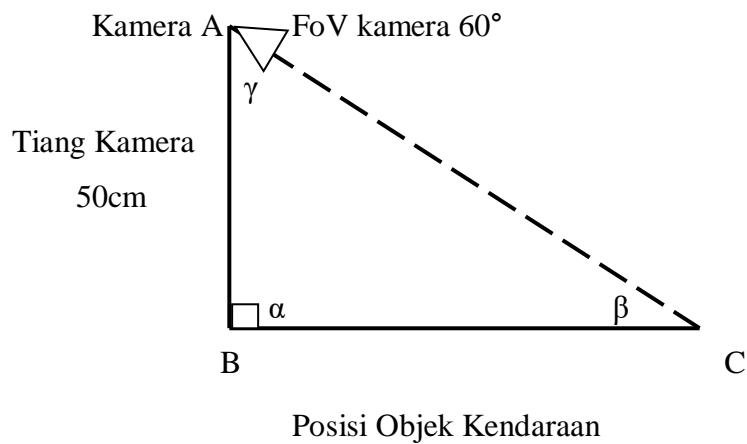
4.3.3.1 Tujuan

Tujuan dari percobaan ini adalah untuk mengetahui berapa *FoV* (*Fiel of View*) atau ruang pandang yang efisien serta optimal dari persepsi kamera terhadap posisi objek kendaraan yang berada di jalanan. Pada perhitungan ini terdapat nilai fix yang tidak dapat dirubah, yaitu ketinggian tiang kamera setinggi 50cm, ukuran

objek kendaraan, tilting kamera 0° sampai 90° , serta Fov kamera 60° yang berdasarkan spesifikasi.

4.3.3.2 Cara Pengukuran

Dapat digambarkan posisi kamera sebagai berikut :



- Keterangan :
- AB = Tinggi tiang kamera.
 - BC = Ruas posisi kendaraan yang ditangkap kamera.
 - AC = Jarak dan ruang pandang kamera terhadap objek kendaraan.
 - γ = Sudut tilting kamera.

Diketahui : $\gamma = 60^\circ$; $\alpha = 90^\circ$; $\beta = 30^\circ$; AB = 50cm

Dicari : AC dan BC

$$\begin{aligned} \text{Perhitungan : } AC &= (\sin \beta = 50/AC) & BC &= \sqrt{AC^2 - AB^2} \\ \sin 30^\circ &= 50/AC & &= \sqrt{100^2 + 50^2} \\ 1/2 &= 50/AC & BC &= \underline{86,6 \text{ cm}} \end{aligned}$$

$$AC = 50/(1/2)$$

$$AC = \underline{100 \text{ cm}}$$

4.3.3.3 Hasil Pengukuran

Dari perhitungan diatas, didapat jarak ruang pandang kamera terjauh yang termasuk kategori efisien terhadap kendaraan yaitu sejauh 100cm, dengan tinggi tiang kamera sepanjang 50cm. Untuk area kendaraan yang dapat ditangkap oleh kamera secara efisien yaitu berada pada jarak maksimal sepanjang 86,6cm dari sudut siku tiang kamera. Untuk angka perhitungan ini didapat dari spesifikasi alat yang dibuat untuk system *traffic light* ini saja.

4.3.3.4 Analisa Dan Pembahasan

Setelah didapat sudut dan ruang pandang kamera yang pas serta area kendaraan yang tertangkap kamera, maka pengujian system *traffic light* dapat dilakukan untuk meminimalisir persentase error yang terjadi. Jadi area kendaraan yang dapat ditangkap kamera hanya dapat maksimal 86,6cm dari sudut siku tiang kamera, jika lebih maka pembacaan objek kendaraan dari kamera kurang akurat. Bisa terbaca lebih sedikit atau bahkan objek tersebut tidak terbaca.

Dengan sudut tilting kamera yang mempunyai persentase error paling kecil yaitu 55°, maka ruang pandang kamera untuk menangkap dan membaca serta mengolah jumlah objek menjadi lebih akurat jika dilakukan percobaan pada sudut tersebut. Namun jika kurang atau lebih dari sudut tersebut, percobaan akan memiliki persentase error yang lebih besar. Namun hal ini juga dipengaruhi dari spesifik kamera serta pencahayaan yang baik.

4.4 Pengujian Keseluruhan Sistem

4.4.1 Pengujian Durasi Kondisi *Traffic Light*

4.4.1.1 Tujuan

Tujuannya adalah untuk mengetahui apakah sistem keseluruhan dapat bekerja dengan baik, dapat mendeteksi kendaraan pada jumlah kendaraan yang bervariasi dengan posisi kendaraan yang dikombinasikan antara seri dan paralel, sistem dapat menghasilkan durasi lampu lalu lintas sesuai dengan kategori lalu-lintas yang sedang terjadi berdasarkan jumlah kendaraan yang terdeteksi.

4.4.1.2 Cara Pengujian

1. Menentukan seluruh parameter pengujian (jumlah kendaraan dan durasi lampu) untuk masing-masing kategori kondisi lalu-lintas, parameter yang

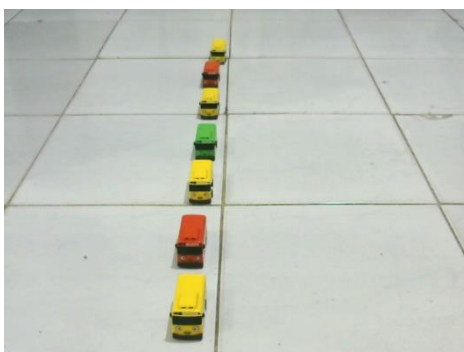
digunakan dalam pengujian keseluruhan sistem dapat dilihat pada tabel 4.8.

2. Mempersiapkan seluruh modul sampai semua modul lampu lalu lintas dan modul display informasi lalu lintas terhubung ke jaringan WiFi HotSpot modul kendali utama dengan baik.
3. Mempersiapkan laptop/smartphone sampai terhubung dengan jaringan WiFi HotSpot modul kendali utama yang kemudian melakukan koneksi remote desktop protokol dengan menggunakan bantuan program Putty.
4. Menjalankan program 'TrafficManager.py'.
5. Ketinggian kamera dari permukaan lantai sebesar 50 cm dengan sudut kemiringan kamera yang di *setting* adalah 55° .
6. Meletakkan kendaraan dengan jarak dari sumbu kamera sebesar 40 cm, dengan jumlah 0 sampai dengan 10 kendaraan.
7. Posisi susunan kendaraan dikombinasikan antara seri dan paralel pada setiap pengambilan sampling.
8. Melakukan pengambilan sampling sebanyak 5 kali untuk setiap jumlah kendaraan.

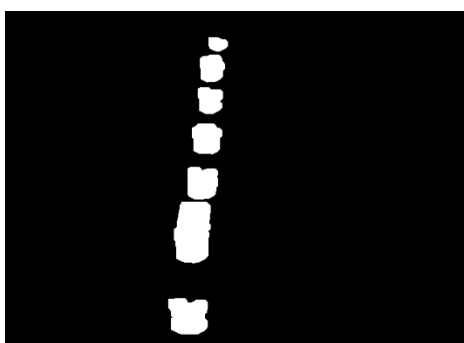
4.4.1.3 Hasil Pengujian

Pada gambar 4.14 dan gambar 4.17 memperlihatkan input dari pengujian keseluruhan sistem, gambar 4.15 dan gambar 4.18 memperlihatkan proses 'Background Subtractor' dari sistem pendeteksian kendaraan, pada gambar 4.16 dan gambar 4.19 memperlihatkan output dari sistem pendeteksian. Pendeteksian jumlah kendaraan tidak selalu akurat berdasarkan jumlah kendaraan. Hal ini disebabkan beberapa factor seperti kondisi cahaya, rapat renggangnya kendaraan, warna objek, serta kapasitas tangkapan kamera terhadap objek.

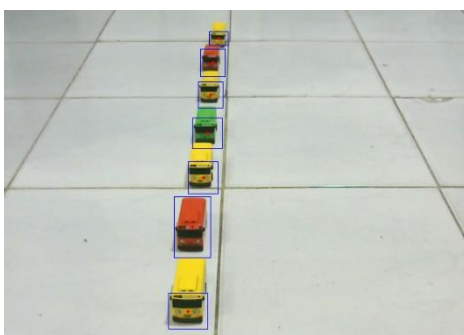
Pada tabel 4.7 memperlihatkan data dari hasil dari pengujian keseluruhan sistem dan pada gambar 4.20 memperlihatkan grafik regresi dari sistem pendeteksian.



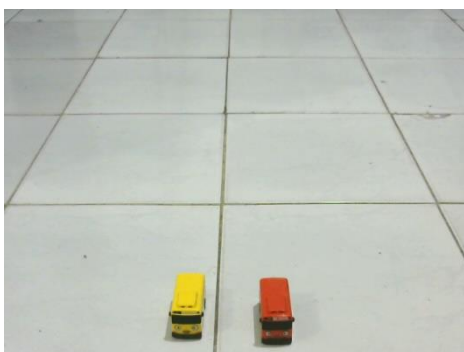
Gambar 4.18 Input 7 kendaraan tersusun secara seri



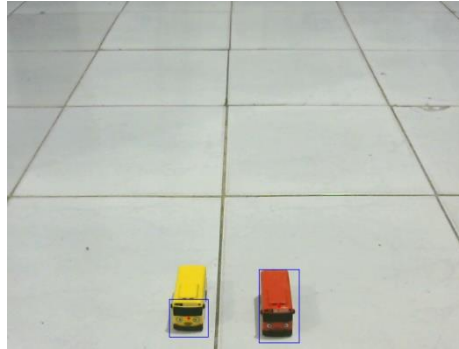
Gambar 4.19 Proses 'Background subtractor' pada pendeteksian 7 kendaraan



Gambar 4.20 Output dari pendeteksian 7 kendaraan



Gambar 4.21 Input 2 kendaraan tersusun secara paralel



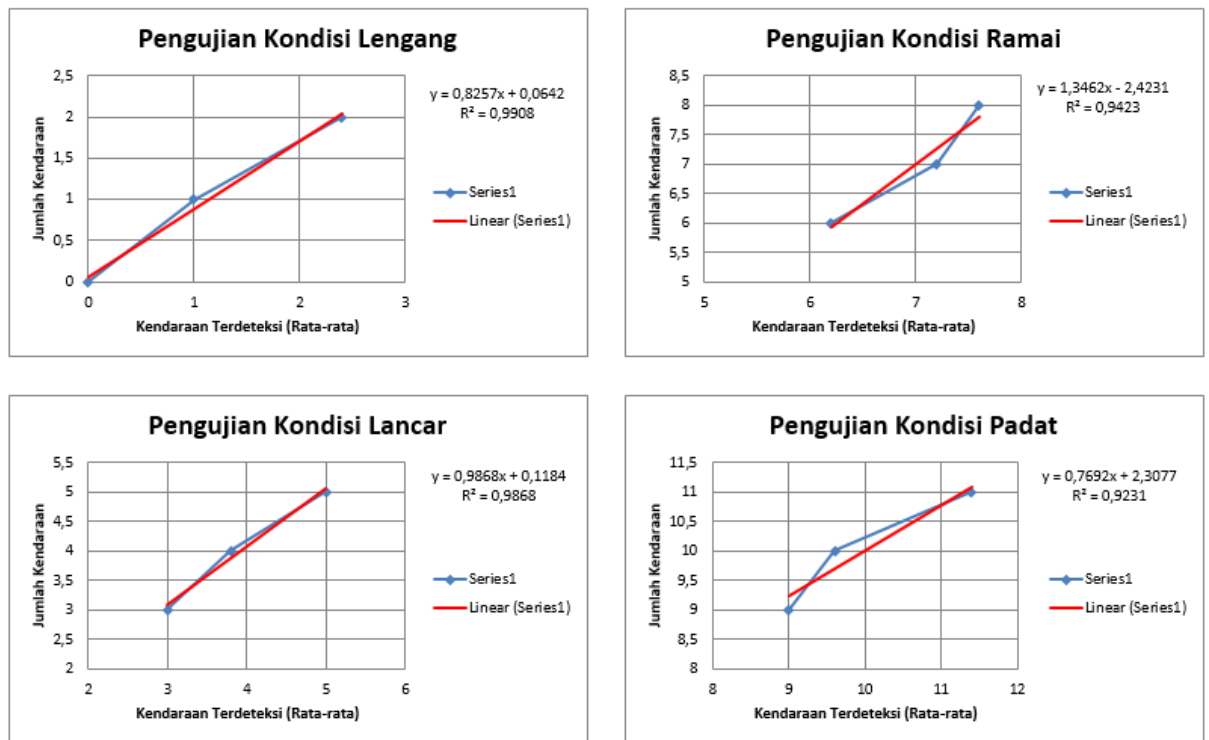
Gambar 4.22 Proses 'Background subtractor' pada pendeteksian 2 kendaraan



Gambar 4.23 Output dari pendeteksian 2 kendaraan

Tabel 4.7 Data hasil pengujian keseluruhan sistem kendali traffic light

Jumlah Kendaraan		Kondisi Lalu-lintas	Durasi (Output)					
Terpasang (Input)	Terdeteksi (Rata-rata)		Hijau	Kuning	Merah	Transisi	Blink On	Blink Off
0	0	Lengang	-	-	-	-	2	1
1	1	Lengang	-	-	-	-	2	1
2	2,4	Lengang	-	-	-	-	2	1
3	3	Lancar	10	3	53	2	-	-
4	3,8	Lancar	10	3	53	2	-	-
5	5	Lancar	10	3	53	2	-	-
6	6,2	Ramai	20	3	83	2	-	-
7	7,2	Ramai	20	3	83	2	-	-
8	7,6	Ramai	20	3	83	2	-	-
9	9	Padat	30	3	113	2	-	-
10	9,6	Padat	30	3	113	2	-	-
R²	0,9954							
Error (%)	0,46							



Gambar 4.24 Grafik regresi dari sistem pendeteksian kendaraan pada pengujian empat kondisi sistem kendali traffic light

4.4.2 Pengujian Implementasi Sistem *Traffic Light*

4.4.2.1 Tujuan

Pengujian Ini bertujuan untuk menampilkan gambaran dimana antar *traffic light* tidak terjadi tumpang tindih kondisi nyala lampunya. Dimana satu *traffic light* tidak menjalankan fungsi yang sama atau berbenturan dengan tiga *traffic light* lainnya yang sudah sesuai perhitungan durasi nyala lampunya. Dan juga membuktikan *traffic light* yang sedang beroperasi sesuai rumus perhitungan yang telah ditentukan.

4.4.2.2 Cara Pengujian

Untuk meringkas pengambilan visual nyala *traffic light* yang berkesinambungan, pengujian ini mengambil contoh kondisi lancar dimana durasinya yang paling singkat. Yaitu dengan menghitung kondisi nyala lampu dengan memasukan symbol warna sesuai kondisi kedalam table penggambaran warna *traffic light*. Hasil implementasi dari *system Traffic Light* tersebut dapat dilihat pada table 4.8.

4.4.2.3 Hasil Pengujian

Tabel 4.8 Implementasi sistem traffic light keseluruhan

Durasi (Detik)	TL 1	TL 2	TL 3	TL 4
1	Red	Red	Red	Red
2	Red	Red	Red	Red
3	Yellow	Red	Red	Red
4	Yellow	Red	Red	Red
5	Green	Red	Red	Red
6	Green	Red	Red	Red
7	Green	Red	Red	Red
8	Green	Red	Red	Red
9	Green	Red	Red	Red
10	Green	Red	Red	Red
11	Green	Red	Red	Red
12	Green	Red	Red	Red
13	Green	Red	Red	Red
14	Green	Red	Red	Red
15	Yellow	Red	Red	Red
16	Yellow	Red	Red	Red
17	Yellow	Red	Red	Red
18	Red	Red	Red	Red
19	Red	Red	Red	Red
20	Red	Yellow	Red	Red
21	Red	Yellow	Red	Red
22	Red	Green	Red	Red
23	Red	Green	Red	Red
24	Red	Green	Red	Red
25	Red	Green	Red	Red
26	Red	Green	Red	Red
27	Red	Green	Red	Red
28	Red	Green	Red	Red
29	Red	Green	Red	Red
30	Red	Green	Red	Red
31	Red	Green	Red	Red
32	Red	Yellow	Red	Red
33	Red	Yellow	Red	Red
34	Red	Yellow	Red	Red
35	Red	Red	Red	Red
36	Red	Red	Red	Red
37	Red	Red	Yellow	Red
38	Red	Red	Yellow	Red
39	Red	Red	Green	Red
40	Red	Red	Green	Red
41	Red	Red	Green	Red
42	Red	Red	Green	Red
43	Red	Red	Green	Red
44	Red	Red	Green	Red
45	Red	Red	Green	Red
46	Red	Red	Green	Red
47	Red	Red	Green	Red
48	Red	Red	Green	Red
49	Red	Red	Yellow	Red
50	Red	Red	Yellow	Red
51	Red	Red	Yellow	Red
52	Red	Red	Red	Red
53	Red	Red	Red	Red
54	Red	Red	Red	Yellow
55	Red	Red	Red	Yellow
56	Red	Red	Red	Green
57	Red	Red	Red	Green
58	Red	Red	Red	Green
59	Red	Red	Red	Green
60	Red	Red	Red	Green
61	Red	Red	Red	Green
62	Red	Red	Red	Green
63	Red	Red	Red	Green
64	Red	Red	Red	Green
65	Red	Red	Red	Green
66	Red	Red	Red	Yellow
67	Red	Red	Red	Yellow
68	Red	Red	Red	Yellow

4.4.3 Analisa dan Pembahasan

Dari hasil pengujian keseluruhan sistem seperti yang diperlihatkan pada tabel 4.7, didapatkan hasil pengujian dengan $R^2 = 0.9954$ dengan error pada pendeteksian sebesar 0,46%. Output dari Sistem Kendali Traffic Light menghasilkan pengenalan kategori kondisi lalu lintas dan durasi lampu yang sudah sesuai dengan parameter yang telah ditetapkan pada saat pengujian seperti yang diperlihatkan pada tabel 4.7. Tetapi pembacaan jumlah objek masih terdapat yang tidak sesuai dikarenakan beberapa factor seperti objek kendaraan yang tidak boleh berdempet, kondisi cahaya, sudut pengambilan gambar dari kamera, kualitas kamera, warna objek, serta objek yang terlalu kecil yang ditangkap oleh kamera.

Namun hal ini masih dalam angka persentase error yang normal yaitu 0,46%. Dari analisa dan data yang didapat diatas, dapat disimpulkan bahwa Sistem Kendali Traffic Light dapat bekerja dengan normal sesuai dengan parameter yang telah ditentukan. Pada saat pengujian keseluruhan sistem, nilai error tertinggi pada 7,69% yang didapat pada saat pengujian dengan kondisi padat, dan nilai error terendah pada 0,92% yang didapat pada saat pengujian kondisi lengan.

Sistem kendali *traffic light* ini memiliki kekurangan dimana jika koneksi antar modul lampu ke Orang Pi Zero terdapat ping dengan latency yang tinggi, maka akan menyebabkan delay atau keterlambatan antar modul lampu untuk menjalankan urutan. Namun tidak mempengaruhi durasi lampu hijau, kuning dan merah pada tiap modul. Kekurangan ini tidak dapat dihindari dikarenakan masalah kualitas fisik komponen. Dalam hal ini diakibatkan spesifikasi dan kualitas dari komponen yang masih terbilang standar.

4.5 Kesimpulan

Berdasarkan pengujian tegangan, pada tegangan output IC Regulator 7805 terjadi penurunan tegangan pada saat kondisi 'TRANSISI', dimana pada saat kondisi ini, kedua relay (relay lampu merah dan relay lampu kuning) dalam keadaan aktif secara bersamaan, hal ini menyebabkan beban daya meningkat pada IC Regulator. Penurunan tegangan yang terjadi dan besarnya deviasi perubahan tegangan yaitu 0,76 dianggap masih dalam batas wajar sesuai dengan spesifikasi komponen datasheet UA7800.

Pada saat pengujian keseluruhan sistem, nilai error yang di dapat dari kondisi lengang adalah 0,92%, kondisi lancer adalah 1,32%, kondisi ramai adalah 5,77% dan kondisi padat adalah 7,69%. Dapat dikatakan semakin banyak jumlah kendaraan, maka semakin besar persentase errornya. Ini dikarenakan semakin kecil ukuran objek karena terlihat lebih jauh maka akan kurang akurat pembacaan objek tersebut. Dan itu menyebabkan persentase error semakin tinggi jika kendaraan yang ditangkap kamera semakin banyak.

Terdapat beberapa factor yang mempengaruhi akurasi pembacaan objek tersebut yaitu ukuran objek, kondisi cahaya, warna objek serta kapasitas tangkapan dari kamera. Namun persentase error tersebut masih dalam kategori normal karena hanya sedikit mempengaruhi status *traffic light*. Dan secara keseluruhan, system ini dapat dikatakan bekerja dengan baik.

BAB V

PENUTUP

5.1 Kesimpulan

Selama perancangan, pembuatan dan pengujian keseluruhan Sistem Kendali Traffic Light ini, diperoleh beberapa kesimpulan sebagai berikut:

1. Sistem pendeteksian kendaraan dengan metode blob cukup akurat dengan persentase kesalahan deteksi objek sebesar 0,46%. Proses *Background Substactor* membuat sistem pendeteksian kendaraan lebih mudah dan akurat.
2. Sistem pendeteksian dengan metode blob membuat sistem tidak kaku terhadap ketinggian dan sudut tangkap kamera lalu lintas, hal ini disebabkan karena metode tersebut menggunakan pengenalan kontur yang lebih dinamis.
3. Pada NodeMCU, menerima data berupa string yang dikirim secara wireless menggunakan fungsi '*client.read()*' lebih cepat yang hanya membutuhkan sekitar 15 millidetik untuk menerima data yang dikirim melalui wireless, dibandingkan menggunakan fungsi '*client.readString()*' yang membutuhkan 5000 millidetik.
4. Penggunaan modul relai membuat modul lampu lalu-lintas dapat menggerakkan/menyalakan lampu dengan jangkauan tegangan yang luas (dari 3 V sampai dengan 230 V baik tegangan DC maupun AC).
5. Fungsi untuk mengaktifkan dan me-nonaktifkan modul relai membuat konsumsi daya pada modul lampu lalu-lintas relative kecil.
6. Perbedaan level tegangan kerja pada keseluruhan sistem hanya terdapat pada modul relai, dimana nodemcu bekerja pada level tegangan 3.3 V dan relai bekerja pada level tegangan 5 V. Modul relai tersebut menggunakan optocoupler sebagai isolasi kepada basis transistor penggerak relai, sehingga optocoupler itu dapat pula berfungsi sebagai level *shifter* satu arah.
7. NodeMCU memiliki beberapa kekurangan, terutama dalam menangani frekuensi transfer data yang tinggi melalui wireless. Hal ini dapat menyebabkan 'tabrakan' terhadap urutan nyala lampu.

5.2 Saran

Untuk pengembangan Sistem Kendali *Traffic Light* berbasis IoT yang jauh lebih baik, penulis mendapatkan beberapa point selama pembuatan Sistem Kendali *Traffic Light* ini, yaitu:

1. Penggunaan mikrokontroller yang jauh lebih baik semisal mikrokontroller ARM Cortex v.7 ataupun mikokontroller SoC (*System On a Chip*) lainnya yang memiliki kinerja yang lebih baik terhadap penanganan frekuensi komunikasi data secara wireless yang cukup intensif.
2. Perangkat komunikasi wireless yang memiliki daya pancar yang lebih tinggi, sehingga Sistem Kendali *Traffic Light* ini dapat digunakan secara luas.
3. Pengembangan sistem deteksi yang lebih baik seperti dengan menambahkan modul kamera pada setiap ruas jalan, sehingga didapatkan informasi ruas jalan yang mengalami kepadatan, fungsi mengenali arah kendaraan, dsb.

DAFTAR PUSTAKA

- [1] MANUAL KAPASITAS JALAN INDONESIA (MKJI), 1997, DIREKTORAT JENDERAL BINA MARGA DIREKTORAT, BINA JALAN KOTA (BINKOT).
- [2] Apache HTTP Server (tanpa tanggal). Diakses pada 5 Januari 2018, dari https://id.wikipedia.org/wiki/Apache_HTTP_Server
- [3] PHP (tanpa tanggal). Diakses pada 5 Januari 2018, dari <https://en.wikipedia.org/wiki/PHP>
- [4] Betha Sidik, Ir, 2004, Pemograman WEB dengan PHP, Penerbit Informatika Bandung, cetakan pertama.
- [5] C++ (tanpa tanggal). Diakses pada 7 Januari 2018, dari [https://en.wikipedia.org/wiki/Python_\(programming_language\)](https://en.wikipedia.org/wiki/Python_(programming_language))
- [6] Python (programming language) (tanpa tanggal). Diakses pada 5 Januari 2018, dari [https://en.wikipedia.org/wiki/Python_\(programming_language\)](https://en.wikipedia.org/wiki/Python_(programming_language))
- [7] OpenCV (tanpa tanggal). Diakses pada 5 Januari 2018, dari <https://en.wikipedia.org/wiki/OpenCV>
- [8] Haar-like feature (tanpa tanggal). Diakses pada 6 Januari 2018, dari https://en.wikipedia.org/wiki/Haar-like_feature
- [9] Orange Pi Zero (tanpa tanggal). Diakses pada 10 Mei 2018, dari <http://www.orangepi.org/orangepizero/>
- [10] NodeMCU (tanpa tanggal) Diakses pada 6 Januari 2018, dari <https://en.wikipedia.org/wiki/NodeMCU>
- [11] Dot-matrix display (tanpa tanggal). Diakses pada 6 Januari 2018, dari https://en.wikipedia.org/wiki/Dot-matrix_display
- [12] OV7670. Diakses pada 6 Januari 2018, dari <http://www.arducam.com/camera-modules/0-3mp-ov7670/>
- [13] Jogyanto Hartono, MBA, Ph.D, 2000, Konsep Dasar Pemograman Bahasa C, Penerbit Andi Yogyakarta, edisi kedua cetakan pertama.

- [14] Andi Kristanto, S.Kom, 2003, *Algoritma dan Pemrograman dengan C++*, Penerbit Graha Ilmu Yogyakarta, edisi pertama cetakan pertama.
- [15] Suryatini Fitria, “Perancangan dan Implementasi Pengendali Lampu Lalu Lintas Berdasarkan Kepadatan Kendaraan Menggunakan Logika Fuzzy”, Tesis, ITB, Bandung, 2015.
- [16] D. Beymer, P. McLauchlan, B. Coifman, and J. Malik, “*A real-time computer vision system for measuring traffic parameters,*” *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, Puerto Rico, June 1997, pp. 496–501.
- [17] K. P. Karmann and A. von Brandt, “*Moving object recognition using an adaptive background memory*”, *Proc. Time-Varying Image Processing and Moving Object Recognition*, vol. 2, V. Capellini, Ed., 1990
- [18] Rensso V. H. Mora Colque, and Guillermo, “*Robust Model for Vehicle Type Identification in Video Traffic Surveillance*” *SIBGRAPI-Conference Graphics, Patterns and Images*, Peru, 2013
- [19] Gonzales, Rafael, C., *Digital Image Processing*, Addison-wesley publishing, 2, 760 – 783, 1992.
- [20] Willey, John. Sons, *Digital Image Processing*, A Wiley-Interscience Publication, 3, 401 – 566, 2001.
- [21] Willow Garage, Open Computer Vision, [Online]. Available: [http://
http://docs.opencv.org/ OpenCV 2.4.11.0 documentation »](http://http://docs.opencv.org/ OpenCV 2.4.11.0 documentation »)
- [22] Solomon, Chris. Toby, Breckon., *Fundamentals of Digital Image Processing*, Wileyblackwell Press, 1, 197-200, 2011.
- [23] Intel, *Open Source Computer Vision Library*. U.S.A: Intel Corporation, 1999-2001.
- [24] Ritter G.X, Wilson J. N, *Hanbook of Computer Vision Algorithms in Image Algebra*, CRC Press, Washington D.C, 2001
- [25] Kaspers, Anne., “*Blob Detection Biomedical Image Sciences*”, Image Sciences Institute, UMC Utrecht, 2011.
- [26] Atkociounas, et al., “*Image Processing in Road Traffic Analysis*”, *Nonlinear Analysis: Modelling and Control*, Vol. 10, No. 4, 315” 332,200.

- [27] Serially Interfaced, 8-Digit LED Display Driver (tanpa tanggal). Diakses pada 6 Januari 2018, dari <https://www.sparkfun.com/datasheets/Components/Generals/COM-09622-MAX7219-MAX7221.pdf>.
- [28] UA7800 Series Positive-Voltage Regulators, Rev. J (tanpa tanggal). Diakses pada 6 Januari 2018, dari <https://www.sparkfun.com/datasheets/LM7805.pdf>.
- [29] ESP8266EX Datasheet (tanpa tanggal). Diakses pada 6 Januari 2018, dari https://www.espressif.com/sites/default/files/documentation/0a-esp8266ex_datasheet_en.pdf.
- [30] COM-09590-YSL-R531R-3D-D2 (tanpa tanggal). Diakses pada 6 Januari 2018, dari <https://www.sparkfun.com/datasheets/Components/LED/COM-09590-YSL-R531R-3D-D2.pdf>.
- [31] 4 Chanel 5V Optical Isolated Relay Module (tanpa tanggal). Diakses pada 6 Januari 2018, dari <https://www.handsontec.com/dataspecs/4Ch-relay.pdf>.

LAMPIRAN

Program Traffic Manager

```
from threading import Timer, Thread, Event
import threading
import subprocess
import datetime
import time
import json
import os
import os.path
import urllib2
import socket
import ssl
import os
from os import walk
import sys
from random import randint

import cv2
import numpy as np

local_path = os.path.abspath("")

IMAGE_FILENAME = local_path + "/images/images.png"
IMAGE_BACKGROUND = local_path + '/images/background.png'
IMAGE_PROCESSED = local_path + '/images/processed.png'
IMAGE_MASK = local_path + '/images/mask.png'
KONFIGURASI_LAST_MODIFIED = "
FILE_KONFIGURASI = local_path + "/config.json"
FILE_STATUS = local_path + "/status.json"
```

```
FILE_CONTOUR = local_path + "/contour.json"
```

```
IMAGE_SOURCE = -1
```

```
Status_Kamera = False
```

```
Tampilkan_Gambar = False
```

```
# Kontur disesuaikan dengan jarak kamera
```

```
# (ukuran kendaraan setelah tertangkap oleh kamera)
```

```
DEF_CONTOUR_WIDTH = 15
```

```
DEF_CONTOUR_HEIGHT = 15
```

```
contour_config = {'min_contour_width': '15', 'min_contour_height': '15'}
```

```
# Warna untuk menggambar frame yang sudah diproses
```

```
DIVIDER_COLOUR = (255, 255, 0)
```

```
BOUNDING_BOX_COLOUR = (255, 0, 0)
```

```
CENTROID_COLOUR = (0, 0, 255)
```

```
Display_Teks = "
```

```
Jumlah_Kendaraan = 0
```

```
Perubahan_Kondisi = False
```

```
# Status_Lampu: 0 - Lampu Kuning Berkedip, sedangkan Lampu Hijau dan Lampu Merah dalam keadaan off.
```

```
#           1 - Lampu Aktif (Hijau, Kuning dan Merah).
```

```
Status_Lampu = 0
```

```
# Status_Lampu_n: 0 - Lampu Kuning Berkedip, sedangkan Lampu Hijau dan Lampu Merah dalam keadaan off.
```

```
#           1 - Lampu Hijau.
```

```
#          2 - Lampu Kuning.  
#          3 - Lampu Merah.  
#          4 - Transisi setelah Lampu Merah keadaan on,  
#          pada kondisi ini, lampu kuning keadaan on bersamaan dengan lampu  
merah,  
#          kemudian akan kembali ke 1.
```

```
Status_Lampu_1 = 0
```

```
Status_Lampu_2 = 0
```

```
Status_Lampu_3 = 0
```

```
Status_Lampu_4 = 0
```

```
# Status_Blink: 0 - Lampu Kuning keadaan off, sedangkan Lampu Hijau dan Lampu  
Merah dalam keadaan off.
```

```
#          1 - Lampu Kuning keadaan on, sedangkan Lampu Hijau dan Lampu Merah  
dalam keadaan off.
```

```
Status_Blink_1 = 0
```

```
Status_Blink_2 = 0
```

```
Status_Blink_3 = 0
```

```
Status_Blink_4 = 0
```

```
# Durasi Dalam Satuan Detik
```

```
Durasi_Sample = 15
```

```
Durasi_Merah = 45
```

```
Durasi_Kuning = 3
```

```
Durasi_Hijau = 10
```

```
Durasi_Transisi = 2
```

```
Durasi_Blink_On = 2
```

```
Durasi_Blink_Off = 1
```

```
Timer_Lalin_1 = 0
```

```
Timer_Lalin_2 = 0
```



```

Timer_Lalin_3 = 0
Timer_Lalin_4 = 0
Timer_Sampling = 0

# Durasi Pemeriksaan Status Koneksi Modul (dalam satuan detik)
Timer_Koneksi = 0
Durasi_Koneksi = 5

# Durasi Pemeriksaan Konfigurasi (dalam satuan detik)
Timer_Konfigurasi = 0
Durasi_Konfigurasi = 10

status_kondisi = {}
# Kondisi Lalu Lintas
#          0 - Lengah
#          1 - Lancar
#          2 - Ramai
#          3 - Padat
status_kondisi = {'kondisi_lalin': '0', 'jumlah_kendaraan': '0'}

konfigurasi = {}

# Array Untuk Parameter Pada Modul
konfigurasi['modul'] = []

# Array Untuk Parameter Kondisi Lalu Lintas
konfigurasi['setting'] = []

# Parameter Pada Modul Lampu Lalu Lintas dan Display Informasi Lalu Lintas

```

Nilai yang tertera adalah nilai default untuk kondisi lembang

yang akan dipergunakan untuk setiap modul

```
konfigurasi['modul'].append({
    'modul_type': 'Lampu-Lalin',
    'status_koneksi': '0',
    'terminal_id': 'Lampu-Lalin-1',
    'terminal_key': '0123456789',
    'ip_address': '192.168.0.21',
    'ssid': 'KendaliAPILL',
    'password': 'KendaliAPILL',
    'status_lampu': '0',
    'durasi_merah': '38',
    'durasi_kuning': '3',
    'durasi_merah': '5',
    'durasi_transisi': '2',
    'durasi_blink_on': '2',
    'durasi_blink_off': '1',
    'status_relay': '1',
    'set_timer': '0'
})
```

```
konfigurasi['modul'].append({
    'modul_type': 'Lampu-Lalin',
    'status_koneksi': '0',
    'terminal_id': 'Lampu-Lalin-2',
    'terminal_key': '0123456789',
    'ip_address': '192.168.0.22',
    'ssid': 'KendaliAPILL',
```

```
'password': 'KendaliAPILL',  
'status_lampu': '0',  
'durasi_merah': '38',  
'durasi_kuning': '3',  
'durasi_merah': '5',  
'durasi_transisi': '2',  
'durasi_blink_on': '2',  
'durasi_blink_off': '1',  
'status_relay': '1',  
'set_timer': '0'  
})
```

```
konfigurasi['modul'].append({  
    'modul_type': 'Lampu-Lalin',  
    'status_koneksi': '0',  
    'terminal_id': 'Lampu-Lalin-3',  
    'terminal_key': '0123456789',  
    'ip_address': '192.168.0.23',  
    'ssid': 'KendaliAPILL',  
    'password': 'KendaliAPILL',  
    'status_lampu': '0',  
    'durasi_merah': '38',  
    'durasi_kuning': '3',  
    'durasi_merah': '5',  
    'durasi_transisi': '2',  
    'durasi_blink_on': '2',  
    'durasi_blink_off': '1',  
    'status_relay': '1',  
    'set_timer': '0'
```

```
)
```

```
konfigurasi['modul'].append({  
    'modul_type': 'Lampu-Lalin',  
    'status_koneksi': '0',  
    'terminal_id': 'Lampu-Lalin-4',  
    'terminal_key': '0123456789',  
    'ip_address': '192.168.0.24',  
    'ssid': 'KendaliAPILL',  
    'password': 'KendaliAPILL',  
    'status_lampu': '0',  
    'durasi_merah': '38',  
    'durasi_kuning': '3',  
    'durasi_merah': '5',  
    'durasi_transisi': '2',  
    'durasi_blink_on': '2',  
    'durasi_blink_off': '1',  
    'status_relay': '1',  
    'set_timer': '0'  
})
```

```
konfigurasi['modul'].append({  
    'modul_type': 'Display-Lalin',  
    'status_koneksi': '0',  
    'terminal_id': 'Display-Lalin-1',  
    'terminal_key': '0123456789',  
    'ip_address': '192.168.0.20',  
    'ssid': 'KendaliAPILL',  
    'password': 'KendaliAPILL',
```

```
'status_display': '1',  
'display_text': '',  
'display_brightness': '0.5',  
'display_speed': '100'  
})
```