

**SISTEM CERDAS LAMPU PENERANGAN JALAN BERBASIS
*IOT***

PROYEK AKHIR

Laporan akhir ini dibuat dan diajukan untuk memenuhi salah satu syarat kelulusan
Diploma III Politeknik Manufaktur Negeri Bangka Belitung



Disusun Oleh:

Teguh Gesit Aqilla NIM 0031826

Tiara Rahmadani NIM 0031827

**POLITEKNIK MANUFAKTUR NEGERI
BANGKA BELITUNG
TAHUN 2021**

LEMBAR PENGESAHAN

SISTEM CERDAS LAMPU PENERANGAN JALAN BERBASIS IOT

Oleh:

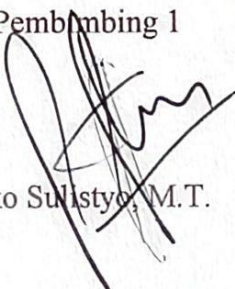
Teguh Gesit Aqilla/0031826

Tiara Rahmadani/0031827

Laporan akhir ini telah disetujui dan disahkan sebagai salah satu syarat kelulusan
Program Dipolma III Politeknik Manufaktur Negeri Bangka Belitung

Menyetujui,

Pembimbing 1




Eko Sulistyono, M.T.

Pembimbing 2




Zanu Saputra, M.Tr.T.

Penguji 1



Indra Dwisaputra, M.T.

Penguji 2



Irwan, M.Sc., Ph.D.

PERNYATAAN BUKAN PLAGIAT

Yang bertanda tangan di bawah ini:

Nama Mahasiswa 1 : Teguh Gesit Aqilla NIM : 0031826

Nama Mahasiswa 2 : Tiara Rahmdani NIM : 0031827

Dengan Judul : Sistem Cerdas Lampu Penerangan Jalan Berbasis *IOT*

Menyatakan bahwa laporan akhir ini adalah hasil kerja kami sendiri dan bukan merupakan plagiat. Pernyataan ini kami buat dengan sebenarnya dan bila ternyata di kemudian hari ternyata melanggar pernyataan ini, kami bersedia menerima sanksi yang berlaku.

Sungailiat, 01 Juni 2021

Nama Mahasiswa

1. Teguh Gesit Aqilla
2. Tiara Rahmadani

Tanda Tangan


.....

.....

ABSTRAK

LPJU merupakan sarana penerangan pada jalan raya. Daya yang digunakan LPJU tidak dapat terkontrol dengan baik sehingga biaya yang dikeluarkan sangat tinggi. Jika terjadi kerusakan, LPJU tidak dapat memberikan sinyal kerusakan tersebut. Untuk itu diperlukan kegiatan pengontrolan dan monitoring secara berkala guna meminimalisir pemborosan daya dan kerusakan pada LPJU. Salah satu cara untuk mempermudah kegiatan tersebut yaitu dengan melakukan pengontrolan kecerahan dan monitoring kerusakan LPJU melalui smartphone. Tujuan yang dicapai pada proyek akhir ini yaitu mampu menampilkan data kerusakan lampu dan pengaturan kecerahan lampu melalui smartphone. Proyek akhir yang dilakukan adalah membuat sistem kontrol dan monitoring LPJU berbasis IoT menggunakan input berupa cahaya dan objek yang terdeteksi oleh sensor GY-302 BH1750, sensor LDR, dan sensor proximity lalu diproses pada Arduino Mega sehingga diperoleh output berupa pembacaan sinyal kerusakan LPJU dan pengaturan kecerahan LPJU yang dapat di-monitoring melalui smartphone. Pengujian proyek akhir dilakukan dengan memberikan input objek yang bergerak melewati LPJU dengan output berupa kontrol kecerahan LPJU serta pengujian dengan input cahaya LPJU dengan output berupa monitoring kerusakan LPJU. Hasil pengujian didapatkan bahwa sistem mampu mendeteksi objek yang bergerak melewati LPJU dengan penyalaan lampu 100 % sejauh 80 cm dengan jarak sudut yang terdeteksi sebesar 45° dan pengontrolan kecerahan lampu dapat diatur melalui aplikasi smartphone serta kerusakan lampu ditandai dengan flashing lamp yang menyala dan dapat terbaca di aplikasi smartphone dan LCD 20x4 dengan persentase error 0%. Dengan sistem ini maka memudahkan petugas dalam memonitoring kerusakan lampu dan pengontrolan kecerahan lampu melalui smartphone.

Kata kunci: LPJU, Smartphone, monitoring, kerusakan, kecerahan

ABSTRACT

LPJU is a means of lighting on the highway. The power used by the LPJU cannot be controlled properly so the costs incurred are very expensive. If there is damage, the LPJU cannot give a signal of the damage. For this reason, periodic control and monitoring activities are needed to minimize power wastage and damage to the LPJU. One way to make this activities easier is by controlling the brightness and monitoring the damage to the LPJU via smartphone. The goal achieved in this final project is to be able to display lamp damage data and lamp brightness settings via smartphone. The final project carried out is to create an IoT-based LPJU control and monitoring system using inputs in the form of light and objects detected by the GY-302 BH1750 sensor, LDR sensor, and proximity sensor and then processed on Arduino Mega so that the output is obtained in the form of reading the LPJU damage signal and setting brightness LPJU which can be monitored via smartphone. The final project testing is carried out by providing input for objects that move through the LPJU with the output in the form of controlling the brightness of the LPJU and testing with LPJU light input with the output in the form of monitoring the damage to the LPJU. The test results show that the system is able to detect objects that move past the LPJU with 100% lamp lighting as far as 80 cm with detected angle distance of 45° and controlling the brightness of the lamps can be adjusted via smartphone application and lamp damage is indicated by flashing lamps. lights up and can be read on smartphone applications and 20x4 LCDs with an error percentage of 0%. This system makes it easier for officers to monitor lamp damage and control lamp brightness via smartphones.

Keywords: LPJU, Smartphone, monitoring, damage, brightness

KATA PENGANTAR

Puji syukur penulis panjatkan kepada Allah SWT atas segala limpahan rahmat dan ridho-Nya sehingga penulis dapat menyelesaikan penyusunan Laporan Proyek Akhir yang berjudul “Sistem Cerdas Lampu Penerangan Jalan Berbasis *IOT*” dengan baik dan tepat waktu. Makalah proyek akhir ini disusun sebagai salah satu syarat untuk menyelesaikan program studi D3 Teknik Elektronika di Politeknik Manufaktur Negeri Bangka Belitung.

Dalam proyek akhir ini penulis merancang dan membuat suatu sistem pada lampu penerangan jalan berupa *monitoring* kerusakan dan pengaturan kecerahan lampu penerangan jalan menggunakan *smartphone*. Dalam pembuatan proyek akhir ini, penulis mendapatkan bantuan dan bimbingan dari berbagai pihak. Maka dari itu, penulis ingin menyampaikan rasa syukur dan terima kasih yang sebesar-besarnya kepada:

1. Kedua orang tua dan keluarga penulis yang telah memberikan dukungan dan doa selama pembuatan proyek akhir serta penyusunan laporan proyek akhir ini;
2. Bapak I Made Andik Setiawan, M.Eng, Ph.D selaku Direktur Politeknik Manufaktur Negeri Bangka Belitung;
3. Bapak Eko Sulistyono, M.T. dan Bapak Zanu Saputra, M.Tr.T. selaku pembimbing proyek akhir yang telah memberikan ilmu dan bimbingan yang sangat bermanfaat kepada penulis dalam pembuatan proyek akhir serta penyusunan laporan proyek akhir ini;
4. Bapak Ocsirensdi selaku dosen wali yang telah memberikan bimbingan dan arahan kepada penulis;
5. Seluruh staf pengajar dari jurusan Teknik Elektronika yang telah memberikan bimbingan kepada penulis;
6. Teman-teman telah banyak memberikan pengetahuan dan dukungan selama penyelesaian proyek akhir ini; dan

7. Semua pihak yang telah membantu dan mendukung penulis tidak dapat disebutkan satu per satu.

Penulis menyadari bahwa penyusunan makalah proyek akhir ini masih jauh dari kata sempurna. Oleh karena itu, penulis sangat mengharapkan kritik, saran, dan masukan yang bersifat membangun dalam perbaikan makalah ini. Akhir kata penulis mengharapkan semoga laporan proyek akhir ini dapat bermanfaat dan sebagai bahan referensi dan informasi bagi pembaca pada umumnya dan adik-adik tingkat pada khususnya serta dapat dikembangkan di kemudian hari, khususnya program studi D3 Elektronika.

Sungailiat, 01 Juni 2021

Penulis

DAFTAR ISI

LEMBAR PENGESAHAN	ii
PERNYATAAN BUKAN PLAGIAT	iii
ABSTRAK	iv
ABSTRACT	v
KATA PENGANTAR	vi
DAFTAR ISI	viii
DAFTAR TABEL	xi
DAFTAR GAMBAR	xii
DAFTAR LAMPIRAN	xv
BAB I	1
PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Batasan Masalah	2
1.4 Tujuan Proyek Akhir	2
BAB II	4
DASAR TEORI	4
2.1 Lampu Penerangan Jalan	4
2.2 <i>Internet of things</i>	6
2.2.1 Prinsip Kerja <i>Internet of Things</i>	6
2.3 <i>Mit App Inventor</i>	7
2.4 Pengontrol	7

2.4.1	NodeMCU ESP8266	7
2.4.2	Arduino Mega	9
2.4.3	Sensor Intensitas Cahaya BH1750 GY-302.....	10
2.4.4	Sensor <i>Proximity Infrared</i>	12
2.4.5	Sensor LDR.....	14
2.4.6	Modul AC <i>Light Dimmer</i>	14
BAB III		17
METODE PELAKSANAAN		17
3.1	Pengumpulan dan Analisis Data.....	20
3.2	Perancangan Konstruksi Kontrol <i>Monitoring</i> LPJU	20
3.3	Perancangan <i>Hardware</i> Elektrik Kontrol <i>Monitoring</i> LPJU	20
3.4	Perancangan <i>Software</i> Kontrol <i>Monitoring</i> LPJU.....	21
3.5	Perakitan Konstruksi dan <i>Hardware</i> Elektrik Kontrol <i>Monitoring</i> LPJU 21	
3.6	Pengujian Kontrol <i>Monitoring</i> LPJU	21
3.7	Pembuatan Makalah Proyek Akhir.....	21
BAB IV		22
PEMBAHASAN		22
4.1	Deskripsi Alat.....	22
4.1.1	Blok Diagram dan <i>Flowchart</i>	22
4.1.2	Prinsip Kerja Blok Diagram dan <i>Flowchart</i>	23
4.2	Perancangan Konstruksi Kontrol <i>Monitoring</i> LPJU	24
4.3	Pembuatan Konstruksi Kontrol <i>Monitoring</i> LPJU.....	24
4.4	Perakitan Konstruksi Kontrol <i>Monitoring</i> LPJU.....	25
4.5	Perancangan <i>Hardware</i> Elektrik Kontrol <i>Monitoring</i> LPJU	25

4.6	Pembuatan <i>Hardware</i> Elektrik Kontrol <i>Monitoring</i> LPJU	26
4.7	Pengujian <i>Hardware</i> Elektrik Kontrol <i>Monitoring</i> LPJU	27
4.7.1	Pengujian Modul Sensor Intensitas Cahaya GY-302 BH1750	27
4.7.2	Pengujian Sensor LDR	31
4.7.3	Pengujian Sensor <i>Proximity Infrared</i>	33
4.7.4	Pengujian Modul <i>Relay</i>	36
4.7.5	Pengujian Modul AC <i>Light Dimmer</i>	38
4.7.6	NodeMCU ESP8266	40
4.8	Perancangan <i>Software</i> Aplikasi	43
4.9	Perakitan <i>Hardware</i> Konstruksi dan Elektrik	49
4.10	Pengujian Prototipe Secara Keseluruhan	50
4.10.1	Pengujian <i>Monitoring</i> Kerusakan Pada Lampu LPJU	51
4.10.2	Pengujian Kecerahan Lampu LPJU	54
4.10.3	Pengujian Kontrol dan <i>Monitoring</i> Kecerahan Lampu LPJU Pada <i>Smartphone</i>	58
BAB V		63
KESIMPULAN DAN SARAN		63
5.1	Kesimpulan	63
5.2	Saran	63
DAFTAR PUSTAKA		64
LAMPIRAN		65

DAFTAR TABEL

Tabel 2. 1 Penataan letak lampu penerangan	4
Tabel 2. 2 Jenis-jenis tiang penopang vertikal berdasarkan bentuk lengannya	5
Tabel 2. 3 Spesifikasi NodeMCU ESP8266	9
Tabel 2. 4 Spesifikasi Arduino Mega.....	10
Tabel 2. 5 Deskripsi pin modul sensor GY-302 BH1750	11
Tabel 2. 6 Deskripsi blok diagram sensor intensitas cahaya BH1750 GY-302....	12
Tabel 2. 7 Spesifikasi sensor proximity infrared	13
Tabel 2. 8 Spesifikasi modul AC light dimmer.....	16
Tabel 4. 1 Hasil pengujian modul sensor GY-302 BH1750	28
Tabel 4. 2 Hasil pengujian sensor LDR	33
Tabel 4. 3 Hasil pengujian modul relay	38
Tabel 4. 4 Hasil pengujian modul AC light dimmer.....	39
Tabel 4. 5 Hasil pengujian monitoring kerusakan lampu LPJU	52
Tabel 4. 6 Hasil pengujian objek yang bergerak sebelum melewati lampu LPJU	56
Tabel 4. 7 Hasil pengujian objek yang bergerak setelah melewati lampu LPJU..	56
Tabel 4. 8 Hasil pengujian jarak objek yang bergerak tepat melewati lampu LPJU	57
Tabel 4. 9 Hasil pengujian kontrol dan monitoring kecerahan lampu LPJU pada smartphone	59

DAFTAR GAMBAR

Gambar 2. 1 Lampu Merkuri	6
Gambar 2. 2 Lampu Sodium	6
Gambar 2. 3 Bentuk fisik NodeMCU ESP8266	7
Gambar 2. 4 Pemetaan pin NodeMCU ESP8266	8
Gambar 2. 5 Pemetaan pin dari chip ESP12-E	8
Gambar 2. 6 Bentuk fisik Arduino Mega.....	9
Gambar 2. 7 Bentuk fisik sensor intensitas cahaya GY-302 BH1750.....	11
Gambar 2. 8 Skematik rangkaian modul sensor GY-302 BH1750.....	11
Gambar 2. 9 Blok diagram sensor intensitas cahaya BH1750 GY-302.....	12
Gambar 2. 10 Skematik rangkaian sensor proximity infrared	13
Gambar 2. 11 Bentuk fisik sensor proximity infrared.....	13
Gambar 2. 12 Skematik rangkaian sensor LDR.....	14
Gambar 2. 13 Bentuk fisik Sensor LDR	14
Gambar 2. 14 Bentuk fisik modul AC light dimmer.....	15
Gambar 2. 15 Rangkaian modul AC light dimmer	15
Gambar 3. 1 Flowchart metode pelaksanaan	19
Gambar 4. 1 Blok diagram sistem lampu penerangan jalan.....	22
Gambar 4. 2 Flowchart sistem lampu penerangan jalan	23
Gambar 4. 3 Desain konstruksi kontrol monitoring LPJU.....	24
Gambar 4. 4 Konstruksi kontrol monitoring LPJU.....	25
Gambar 4. 5 Desain hardware elektrik kontrol monitoring LPJU	26
Gambar 4. 6 Hardware elektrik kontrol monitoring LPJU	26
Gambar 4. 7 Blok diagram pengujian modul sensor GY-302 BH1750.....	27
Gambar 4. 8 Ilustrasi pengujian modul sensor GY-302 BH1750.....	27

Gambar 4. 9 Serial monitor pengujian modul sensor GY-302 BH1750 pada nilai ambang batas gelap 60 Lux.....	29
Gambar 4. 10 Serial monitor pengujian modul sensor GY-302 BH1750 pada nilai ambang batas gelap 400 Lux.....	30
Gambar 4. 11 Serial monitor pengujian modul sensor GY-302 BH1750 pada nilai ambang batas gelap 600 Lux.....	31
Gambar 4. 12 Blok diagram pengujian sensor LDR.....	31
Gambar 4. 13 Ilustrasi pengujian sensor LDR.....	32
Gambar 4. 14 Blok diagram pengujian sensor proximity infrared.....	33
Gambar 4. 15 Ilustrasi pengujian sensor proximity infrared.....	34
Gambar 4. 16 Serial monitor pengujian sensor proximity infrared	35
Gambar 4. 17 Pengujian sensor proximity infrared	35
Gambar 4. 18 Blok diagram pengujian modul relay	36
Gambar 4. 19 Serial monitor pengujian modul relay	37
Gambar 4. 20 Blok diagram pengujian modul AC light dimmer.....	38
Gambar 4. 21 Blok diagram pengujian serial komunikasi untuk pengiriman data dari Arduino Mega ke NodeMCU ESP8266.....	41
Gambar 4. 22 Hasil pengujian serial komunikasi antara Arduino Mega dan NodeMCU ESP8266 pada Arduino Mega	42
Gambar 4. 23 Hasil pengujian serial komunikasi antara Arduino Mega dan NodeMCU ESP8266 pada NodeMCU ESP8266.....	43
Gambar 4. 24 Tampilan screen pertama pada software aplikasi.....	44
Gambar 4. 25 Coding blok untuk screen pertama.....	44
Gambar 4. 26 Tampilan screen kedua pada software aplikasi	45
Gambar 4. 27 Coding blok untuk screen kedua	45
Gambar 4. 28 Tampilan screen ketiga pada software aplikasi.....	46
Gambar 4. 29 Coding blok untuk screen ketiga.....	46
Gambar 4. 30 Tampilan screen keempat pada software aplikasi.....	47
Gambar 4. 31 Coding blok untuk screen keempat	47
Gambar 4. 32 Tampilan screen kelima pada software aplikasi.....	48
Gambar 4. 33 Coding blok untuk screen kelima.....	48

Gambar 4. 34 Tampilan screen keenam pada software aplikasi	49
Gambar 4. 35 Coding blok untuk screen keenam	49
Gambar 4. 36 Hardware kontrol dan elektrik kontrol monitoring LPJU	50
Gambar 4. 37 Blok diagram pengujian monitoring kerusakan pada lampu LPJU	51
Gambar 4. 38 Pengujian monitoring kerusakan lampu LPJU	52
Gambar 4. 39 Pengujian monitoring kerusakan LPJU dari jarak jauh.....	54
Gambar 4. 40 Blok diagram pengujian kecerahan lampu LPJU	54
Gambar 4. 41 Skema pengujian kecerahan lampu LPJU	55
Gambar 4. 42 Pengujian objek yang bergerak sebelum melewati lampu LPJU ...	55
Gambar 4. 43 Pengujian objek yang bergerak setelah melewati lampu LPJU	56
Gambar 4. 44 Pengujian objek yang bergerak tepat melewati lampu LPJU.....	57
Gambar 4. 45 Blok diagram pengujian kontrol dan monitoring kecerahan lampu LPJU pada smartphone	58
Gambar 4. 46 Hubungan sinyal PWM dengan kecerahan lampu LPJU	61
Gambar 4. 47 Pengujian kontrol dan monitoring kecerahan lampu LPJU dari jarak jauh.....	62

DAFTAR LAMPIRAN

Lampiran 1 Daftar Riwayat Hidup.....	65
Lampiran 2 Program Arduino Mega.....	67
Lampiran 3 Program NodeMCU.....	89

BAB I

PENDAHULUAN

1.1 Latar Belakang

Lampu Penerangan Jalan Umum (LPJU) merupakan lampu yang digunakan untuk penerangan jalan pada malam hari sehingga memudahkan para pengguna jalan untuk melihat dengan jelas jalan yang akan dilalui. Selain itu, lampu penerangan jalan ini berfungsi untuk meningkatkan keamanan dan keselamatan para pengguna jalan dari tindak kriminal dan kecelakaan. Saat ini penggunaan lampu penerangan jalan menggunakan kontrol konvensional untuk penghematan energi. Namun karena banyaknya lampu yang terpasang dan menyala pada malam hari maka akan menimbulkan pemborosan dan kegiatan *monitoring* secara berkala harus dilakukan guna meminimalisir kerusakan pada lampu penerangan jalan tersebut. Saat ini petugas yang melakukan kegiatan tersebut masih menggunakan sistem manual yaitu dengan mendatangi lokasi lampu penerangan jalan dipasang. Yang mana kegiatan ini memiliki kekurangan karena petugas memerlukan waktu untuk mengetahui kondisi lampu penerangan jalan tersebut.

Sebelumnya sudah ada beberapa penelitian yang menjelaskan tentang pengontrolan lampu penerangan jalan umum ini. Salah satunya penelitian dari Tresna Umar Syamsuri (2015) yang membuat alat berupa “Kontrol Lampu Jalan Untuk Menghemat Energi”. Hasil penelitian ini membahas tentang kontrol lampu dengan menggunakan sensor cahaya dan sensor kendaraan. Hal ini membuat lampu menyala jika matahari tenggelam pada sore hari dan akan mati jika matahari bersinar kembali pada pagi hari serta lampu akan terang jika ada kendaraan yang akan melewati lampu tersebut dan lampu akan mati jika tidak ada kendaraan yang melewati lampu tersebut.[1] Selanjutnya ada penelitian dari Rijalul Imam, I Gede Putu Wirarama Wedashwara W, Fitri Bimantoro (2020) yang membuat sistem berupa “Rancang Bangun Sistem *Monitoring* dan *Controlling* Penerangan Jalan Umum Berbasis *IOT* dan *Android*”. Dari penelitian tersebut diperoleh sistem *monitoring* kerusakan lampu penerangan jalan umum berbasis mikrokontroler yang memakai *logging time series* dengan notifikasi sms sebagai sarana untuk

memberikan informasi mengenai keadaan lampu jalan. Hal ini memudahkan petugas dalam menyalakan, mematikan, dan mengatur kecerahan lampu jalan melalui android.[2]

Dari pembahasan jurnal yang telah ada maka dalam proyek akhir ini kami akan menambahkan sistem kontrol dan *monitoring* berbasis *IOT* mengenai kecerahan dan kerusakan lampu penerangan jalan yang diatur oleh sensor. Hal ini dapat mengurangi daya yang terpakai sehingga dapat mencegah pemborosan dan petugas dapat mengetahui jika ada kerusakan pada lampu penerangan jalan dari jarak jauh.

1.2 Rumusan Masalah

Rumusan masalah yang dapat diangkat berdasarkan latar belakang masalah proyek akhir yaitu sebagai berikut:

1. Bagaimana cara membuat sistem cerdas pada lampu penerangan jalan berbasis *IoT*?
2. Bagaimana sistem pengontrol lampu penerangan jalan melalui *smartphone*?
3. Bagaimana sistem *monitoring* lampu penerangan jalan sehingga dapat diakses melalui *smartphone*?

1.3 Batasan Masalah

Batasan masalah pembuatan proyek akhir yang berjudul Sistem Cerdas Lampu Penerangan Jalan Berbasis *IoT* ini adalah:

1. *Monitoring* hanya dilakukan terhadap kerusakan lampu penerangan jalan.
2. Pengontrolan kecerahan lampu penerangan jalan hanya berdasarkan objek yang bergerak berdasarkan jangkauan sudut sensor *proximity infrared* dengan jarak maksimal 80 cm.
3. *Monitoring* kerusakan lampu penerangan jalan berdasarkan cahaya yang dihasilkan lampu penerangan jalan itu sendiri.
4. Jangkauan *monitoring* tergantung pada jaringan internet yang tersedia.

1.4 Tujuan Proyek Akhir

Tujuan pembuatan sistem yang ingin dicapai pada proyek akhir ini adalah sebagai berikut:

1. Mampu me-*monitoring* kerusakan lampu penerangan jalan yang ditampilkan melalui *smartphone* melalui *smartphone*.
2. Mampu mengontrol kecerahan lampu penerangan jalan melalui *smartphone*.

BAB II

DASAR TEORI

2.1 Lampu Penerangan Jalan

Lampu Penerangan Jalan Umum (LPJU) merupakan sarana penerangan pada malam hari. Tujuan utama dari pemasangan lampu penerangan ini adalah mempermudah para pengguna jalan untuk melihat dengan jelas jalan yang akan dilalui sehingga dapat meningkatkan keselamatan dan keamanan para pengguna jalan dari kecelakaan dan aksi kriminal.[3]

Pada perencanaan instalasi lampu penerangan jalan harus sesuai dengan Standar Nasional Indonesia (SNI). Pada perencanaan instalasi lampu penerangan jalan harus sesuai dengan Standar Nasional Indonesia (SNI). Penataan letak lampu penerangan jalan perlu dipertimbangkan berdasarkan arah lalu-lintas agar mendapatkan cahaya yang maksimal.[4] Penataan letak lampu penerangan jalan dapat dilihat pada tabel 2.1:

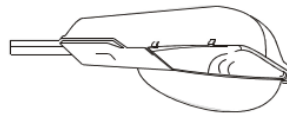
Tempat	Penataan letak lampu penerangan jalan
Jalan satu arah	<ul style="list-style-type: none">• di kanan dan kiri jalan;• di kanan dan kiri jalan berselang-seling;• di kanan dan kiri jalan berhadapan;• di separator jalan;• di median jalan;
Jalan dua arah	<ul style="list-style-type: none">• gabungan antara di kanan dan kiri berhadapan dengan di median jalan;• di bagian tengah jalan dengan cara digantung• dibuat menggunakan lampu menara dengan beberapa
Persimpangan	<p>lampu yang ditempatkan di pulau-pulau, di median jalan, di luar daerah persimpangan.</p>

Tiang penopang vertikal merupakan komponen yang digunakan untuk menopang lampu. Berdasarkan bentuk lengannya, tiang penopang vertikal dapat dibagi menjadi 3 yaitu tanpa lengan, lengan tunggal, lengan ganda.[4] Berikut merupakan tiang lampu tanpa lengan dapat dilihat pada tabel 2.2 berdasarkan SNI (Standar Nasional Indonesia):

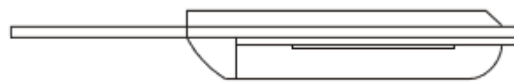
Tabel 2. 2 Jenis-jenis tiang penopang vertikal berdasarkan bentuk lengannya

Bentuk lengan tiang penopang vertikal	Gambar
Tiang lampu tanpa lengan	
Tiang lampu lengan tunggal	
Tiang lampu lengan ganda	

Sumber cahaya merupakan komponen paling penting pada lampu penerangan jalan. Berdasarkan sumber cahaya, lampu penerangan jalan dapat dibedakan menjadi 2 yaitu lampu merkuri, dan lampu sodium.[4] Berikut merupakan lampu merkuri dapat dilihat pada gambar 2.1, dan lampu sodium dapat dilihat gambar 2.2:



Gambar 2. 1 Lampu Merkuri



Gambar 2. 2 Lampu Sodium

2.2 Internet of things

Internet of things (IoT) adalah sebuah konsep yang bertujuan untuk memperluas manfaat dari koneksi internet yang tersambung terus-menerus yang membuat kita dapat menghubungkan benda-benda fisik dengan sensor dan aktuator untuk memperoleh data lalu mengelolanya sendiri sehingga memungkinkan peralatan untuk berkolaborasi dan bertindak berdasarkan perintah yang diperoleh.[5] *Internet of things (IoT)* juga bisa diartikan sebagai sistem yang dapat mengontrol semua perangkat elektronik melalui internet. Yang artinya semua benda fisik dapat dikontrol dan di-*monitoring* dari jarak jauh melalui internet.

2.2.1 Prinsip Kerja Internet of Things

Sederhananya prinsip kerja dari *internet of things* ini memanfaatkan kode-kode pemrograman melalui sebuah modul, dimana modul tersebut berfungsi sebagai penghubung barang fisik ke internet. Kode-kode tersebut akan menjadi sebuah perintah dan akan disimpan dalam modul yang dapat digunakan untuk sistem IoT lalu akan dikirimkan ke aplikasi sehingga dapat digunakan dalam kondisi apapun selama adanya jaringan internet yang terhubung. Internet hanya berfungsi sebagai penghubung interaksi antar kedua barang fisik tersebut dan manusia bertugas sebagai pengawas antar kedua barang fisik tersebut baik secara

langsung maupun tidak langsung. Jika secara tidak langsung, manusia memerlukan sebuah alat sebagai sarana pengontrol jarak jauhnya seperti *smartphone*.

2.3 *Mit App Inventor*

Mit App Inventor adalah sebuah aplikasi yang dikelola oleh Massachusetts Institute of Technology (MIT) dan berbasis *visual block programming* sehingga memungkinkan pengguna untuk melihat, menggunakan, menyusun dan *men-drag* dan *drops* blok yang merupakan simbol instruksi untuk menciptakan sebuah aplikasi yang dapat berjalan di *smartphone*. Aplikasi ini bertujuan untuk memudahkan pengguna dalam pembuatan aplikasi tanpa harus menguasai bahasa pemrograman. Aplikasi yang dibuat bertujuan untuk dapat mengontrol dan membaca data dari *hardware* yang digunakan. Dalam proyek akhir ini, aplikasi yang dibuat akan digunakan untuk membaca data dari Arduino ataupun NodeMCU.

2.4 Pengontrol

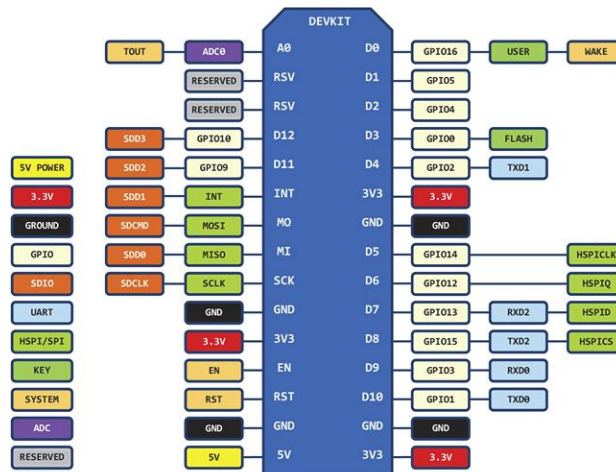
2.4.1 NodeMCU ESP8266

NodeMCU ESP8266 adalah sebuah platform *IoT* yang menggunakan bahasa pemrograman *scripting* Lua dan bersifat *open source* yang terdiri dari perangkat keras buatan *Espressif System* berupa *System On Chip* ESP8266-12. NodeMCU ESP8266 dapat dikatakan sebagai *board* arduino dari ESP8266 karena NodeMCU ESP8266 menggabungkan ESP8266 ke sebuah *board* dengan berbagai fungsi layaknya mikrokontroler ditambah dengan kemampuan akses terhadap WiFi juga *chip* komunikasi USB to *chip* sehingga memprogramnya hanya dibutuhkan ekstensi kabel data mikro USB. Berikut merupakan bentuk fisik NodeMCU ESP8266 dapat dilihat pada gambar 2.3:



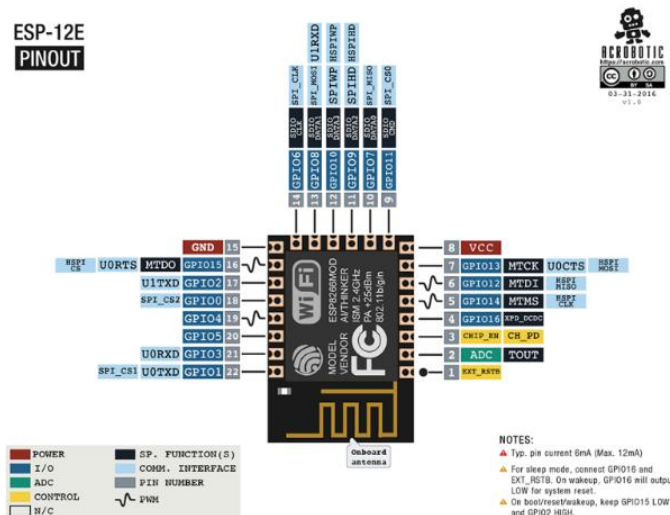
Gambar 2. 3 Bentuk fisik NodeMCU ESP8266

NodeMCU ESP8266 V3 mempunyai ukuran *board* yang lebih besar daripada V3 dan mempunyai 2 pin cadangan untuk daya USB serta pin yang lain untuk GND tambahan. Berikut merupakan pemetaan dari pin NodeMCU ESP8266 V3 dapat dilihat pada gambar 2.4:



Gambar 2. 4 Pemetaan pin NodeMCU ESP8266

NodeMCU ESP8266 dengan *chip* seri ESP12-E memiliki 10 *Port* GPIO dari D0-D10, fungsional PWM, *interface* 12C dan SPI, *interface* 1 Wire serta fitur ADC. Berikut merupakan pemetaan dari pin *chip* ESP12-E dapat dilihat pada gambar 2.5:



Gambar 2. 5 Pemetaan pin dari *chip* ESP12-E

NodeMCU ESP8266 menggunakan tegangan 3.3V untuk bisa berfungsi. Akan tetapi, NodeMCU ESP8266 dapat juga terhubung dengan tegangan

5V melalui *port micro* USB atau pin *V input* yang tersedia pada *board*. Berikut merupakan spesifikasi NodeMCU ESP8266 dapat dilihat pada tabel 2.3:

Tabel 2. 3 Spesifikasi NodeMCU ESP8266

Spesifikasi	Keterangan
Mikrokontroler	ESP8266
Tegangan <i>input</i>	3.3V-5V
GPIO	13 Pin
Kanal PWM	10 Kanal
10 bit ADC Pin	1 Pin
<i>Flash Memory</i>	4 MB
<i>Clock Speed</i>	10/26/24 MHz
WiFi	IEEE 802.11 b/g/in
Frekuensi	2.4 GHz-22,5 GHz
USB <i>Port</i>	<i>Micro</i> USB
USB to <i>Serial Converter</i>	CH340G
Ukuran <i>Board</i>	57mm x 30mm

2.4.2 Arduino Mega

Arduino Mega merupakan mikrokontroler yang berbasis Arduino dengan menggunakan *chip* ATmega256 dan untuk penggunaannya sangat sederhana yaitu dengan menghubungkan *power* dari USB ke PC. Berikut merupakan bentuk fisik Arduino Mega dapat dilihat pada gambar 2.6:



Gambar 2. 6 Bentuk fisik Arduino Mega

Arduino Mega ini memiliki 54 *input* dan *output* digital, dimana 16 buah untuk *analog input* dan 15 buah diantaranya digunakan untuk *output* PWM. Berikut merupakan spesifikasi Arduino Mega dapat dilihat pada tabel 2.4:

Tabel 2. 4 Spesifikasi Arduino Mega

Spesifikasi	Keterangan
Mikrokontroler	ATmega2560
Tegangan Operasional	5V
Tegangan <i>Input</i> Yang Disarankan	7V-12V
Batas Tegangan <i>Input</i>	6V-20V
Pin I/O digital	54 Pin (dimana 15 pin digunakan untuk <i>output</i> PWM)
Pin <i>Input</i> Analog	16 Pin
Arus DC tiap pin I/O	20 mA
Arus DC terhadap pin 3.3 V	50 mA
<i>Flash Memory</i>	256 kB (dimana 8 kB untuk <i>bootloader</i>)
SRAM	8 kB
EEPROM	4 kB
<i>Clock Speed</i>	16 MHz
Ukuran	101mm x 53 mm
Berat	37 gram
LED_BUILTIN	13

2.4.3 Sensor Intensitas Cahaya BH1750 GY-302

Sensor BH1750 GY-302 adalah sensor cahaya digital yang digunakan untuk mengukur intensitas cahaya sekitar dalam satuan lux. Sensor ini dapat mendeteksi cahaya sekitar dengan jangkauan yang luas pada resolusi tinggi (1-65535 lux). Lux merupakan besaran intensitas cahaya. Satu lux sama dengan satu lumen per meter persegi sehingga jika dirumuskan menjadi:

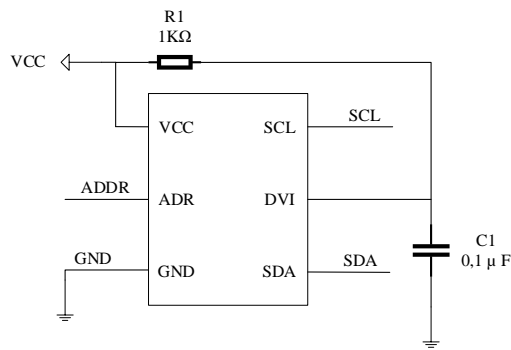
$$1 \text{ lux} = 1 \text{ lm} / \text{m}^2$$

Sensor BH1750 GY-302 merupakan sebuah IC sensor yang menggunakan I2C sebagai media komunikasi dengan mikrokontroler. Berikut merupakan bentuk fisik sensor BH1750 GY-302 dapat dilihat pada gambar 2.7:



Gambar 2. 7 Bentuk fisik sensor intensitas cahaya GY-302 BH1750

Berikut merupakan skematik rangkaian sensor GY-302 BH1750 dapat dilihat pada gambar 2.8 dan deskripsi pin sensor GY-302 BH1750 dapat dilihat pada tabel 2.5:

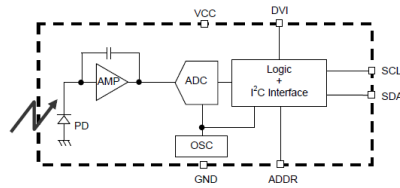


Gambar 2. 8 Skematik rangkaian modul sensor GY-302 BH1750

Tabel 2. 5 Deskripsi pin modul sensor GY-302 BH1750

Nama Pin	Deskripsi
VCC	Tegangan sumber untuk modul sebesar 2,4 V-3,6 V, biasanya digunakan 3,3 V
GND	<i>Ground modul</i> , terhubung ke <i>ground</i> rangkaian
SCL	<i>Serial Clock Line</i> , digunakan untuk menyediakan pulsa clock untuk komunikasi I2C
SDA	<i>Serial Data Adres</i> , digunakan untuk mentransfer data melalui komunikasi I2C
ADDR	Pin alamat perangkat, digunakan untuk memilih alamat ketika lebih dari dua modul terhubung.

Sensor BH1750 GY-302 memiliki blok diagram untuk memudahkan pembacaan cara kerjanya. Berikut merupakan blok diagram sensor BH1750 GY-302 dapat dilihat pada gambar 2.9 dan tabel deskripsi blok diagram sensor BH1750 GY-302 dapat dilihat pada tabel 2.6:



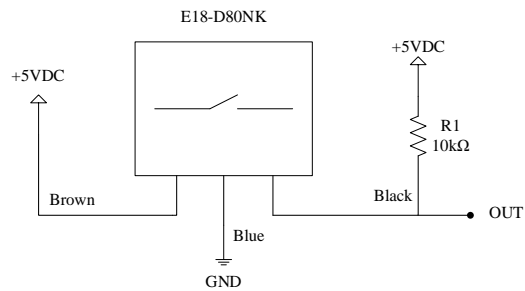
Gambar 2. 9 Blok diagram sensor intensitas cahaya BH1750 GY-302

Tabel 2. 6 Deskripsi blok diagram sensor intensitas cahaya BH1750 GY-302

Nama Bagian	Deskripsi
PD	<i>Photo-diode</i> dengan jangkauan kira-kira respon mata manusia
AMP	<i>Integrator-OP AMP</i> untuk mengubah arus yang dihasilkan PD ke tegangan
ADC	ADC untuk mendapatkan data digital 16 bit.
Logic + I ² C Interface	Perhitungan cahaya sekitar dan antarmuka BUS I ² C. ini termasuk register di bawah. Data register: ini untuk register dari data cahaya sekitar. Nilai awal adalah “0000_0000_0000_0000”. Waktu pengukuran register: ini untuk register dari waktu pengukuran. Nilai awal adalah “0100_0101”.
OSC	<i>Internal Oscillator</i> (typ.320 kHz). Ini adalah CLK untuk internal logic.

2.4.4 Sensor Proximity Infrared

Sensor *proximity infrared* adalah sensor yang dapat mendeteksi objek yang bergerak tanpa adanya kontak fisik. Ketika ada objek yang bergerak di depan sensor dan berada dalam jangkauan sensor maka *output* rangkaian sensor akan berlogika 1. Sebaliknya ketika tidak berada dalam jangkauan sensor maka *output* rangkaian sensor akan berlogika 0. Berikut merupakan skematik sensor *proximity infrared* dapat dilihat pada gambar 2.10:



Gambar 2. 10 Skematik rangkaian sensor *proximity infrared*

Berikut merupakan bentuk fisik sensor *proximity infrared* dapat dilihat pada gambar 2.11:



Gambar 2. 11 Bentuk fisik sensor *proximity infrared*

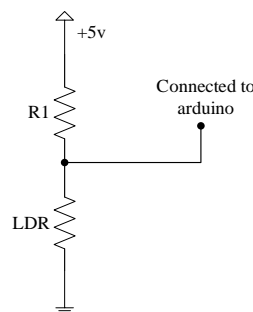
Sensor *proximity infrared* mempunyai sensitivitas yang tinggi terhadap cahaya yang menghalanginya. Sensor ini juga mempunyai penyesuaian untuk mengatur jarak deteksi. Berikut merupakan spesifikasi sensor *proximity infrared* dapat dilihat pada tabel 2.7:

Tabel 2. 7 Spesifikasi sensor *proximity infrared*

Spesifikasi	Keterangan
Tegangan <i>Input</i>	5 VDC
Arus Beban	100 mA
Jarak Deteksi	3 cm-80 cm
Objek Deteksi	<i>Translucency, opaque</i>
Operasi <i>Output</i>	<i>Normally Open</i>
<i>Output DC</i>	<i>Three-wire system NPN</i>
Model No	E 18-D80NK-N
Diameter	18 mm
Panjang	45 mm

2.4.5 Sensor LDR

Sensor *Light Independent Resistor* (LDR) berfungsi untuk mengukur intensitas cahaya. Sensor ini memiliki prinsip kerja yang memanfaatkan terang redup cahaya di sekitarnya dengan cara mengubah besaran intensitas cahaya yang diterima menjadi besaran konduktansi. Berikut merupakan skematik rangkaian sensor LDR dapat dilihat pada gambar 2.12:



Gambar 2. 12 Skematik rangkaian sensor LDR

Semakin banyak intensitas cahaya yang diterima maka semakin rendah resistansinya (konduktansinya semakin tinggi) sehingga relay akan aktif yang membuat lampu akan mati. Semakin sedikit intensitas cahaya yang diterima maka semakin tinggi resistansinya (konduktansinya semakin rendah) sehingga *relay* akan tidak aktif yang membuat lampu akan menyala. Berikut merupakan bentuk fisik sensor LDR dapat dilihat pada gambar 2.13:

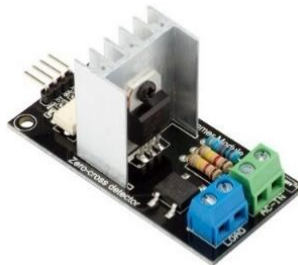


Gambar 2. 13 Bentuk fisik Sensor LDR

2.4.6 Modul AC *Light Dimmer*

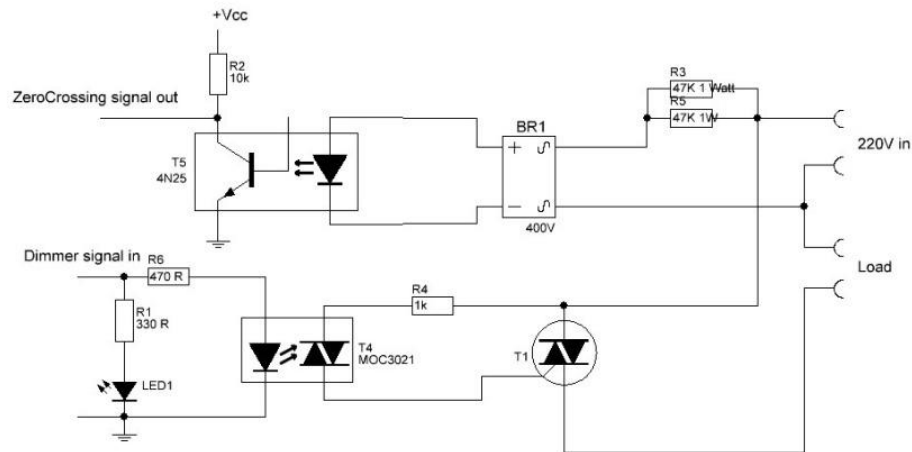
Modul AC *light dimmer* merupakan rangkaian berfungsi untuk mengontrol tegangan AC yang akan diberikan ke perangkat apa pun. Modul ini mempunyai sinyal PWM yang bisa dikontrol langsung menggunakan mikrokontroler. Modul ini juga mempunyai pin *zero crossing detector* yang berfungsi agar mikrokontroler bisa mengetahui waktu yang tepat untuk mengirimkan sinyal PWM. Jika waktunya tidak tepat maka arus AC menggunakan TRIAC jika gate-nya dikontrol oleh sinyal

PWM sinyalnya keluarannya akan kacau dan akan menyebabkan modul tidak berfungsi dalam memperoleh sinyal PSM dengan kurva yang benar.[6] Berikut merupakan bentuk fisik modul AC *light dimmer* dapat dilihat pada gambar 2.14:



Gambar 2. 14 Bentuk fisik modul AC *light dimmer*

Modul AC *light dimmer* terhubung ke Arduino menggunakan dua pin digital, Pertama (*zero crossing detector*) untuk mengontrol lewatnya *phase null* dari AC yang dipakai untuk memulai sinyal interupsi. Kedua (DIM/PSM) untuk mengontrol kecerahan. Berikut merupakan rangkaian modul AC *light dimmer* dapat dilihat pada gambar 2.15:



Gambar 2. 15 Rangkaian modul AC *light dimmer*

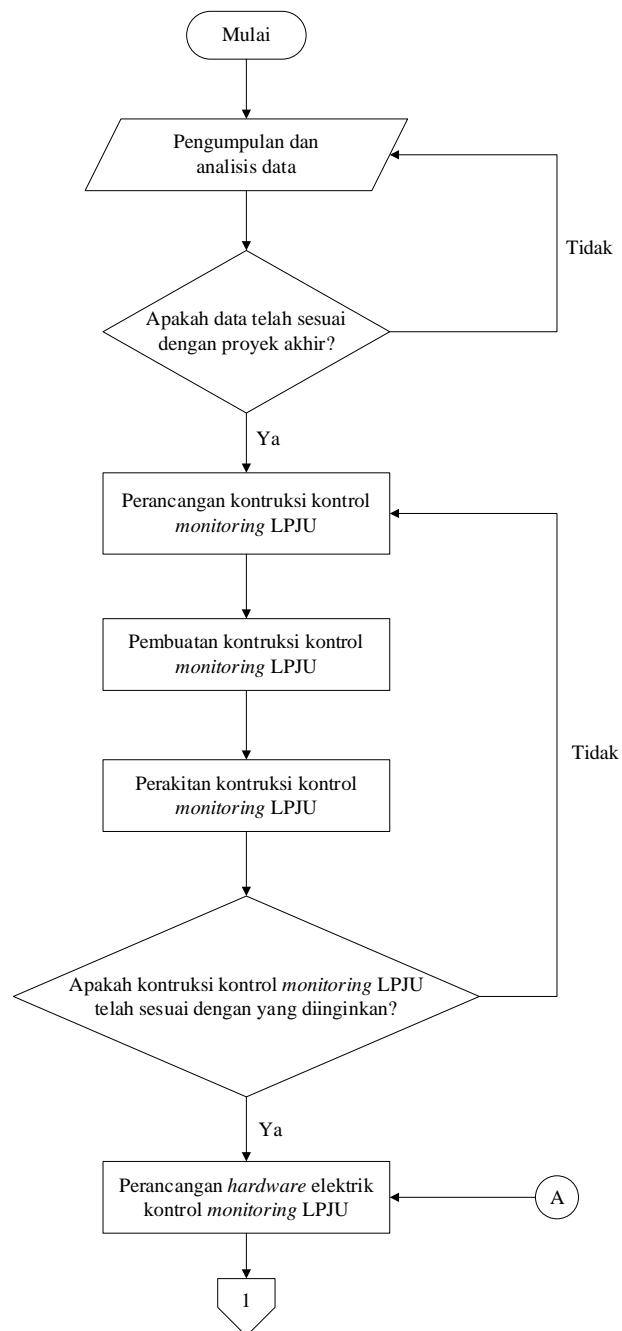
Rangkaian mendeteksi *zero crossing* pada *grid* dan kemudian mikrokontroler menyalakan TRIAC dengan penundaan waktu untuk menentukan kecerahan lampu yang terpasang. Siklus penuh adalah 10mS (pada 50Hz). Bagian daya modul diisolasi dari bagian kontrol untuk meminimalisir gangguan arus tinggi yang masuk ke mikrokontroler. Berikut merupakan spesifikasi modul AC *light dimmer* dapat dilihat pada tabel 2.8:

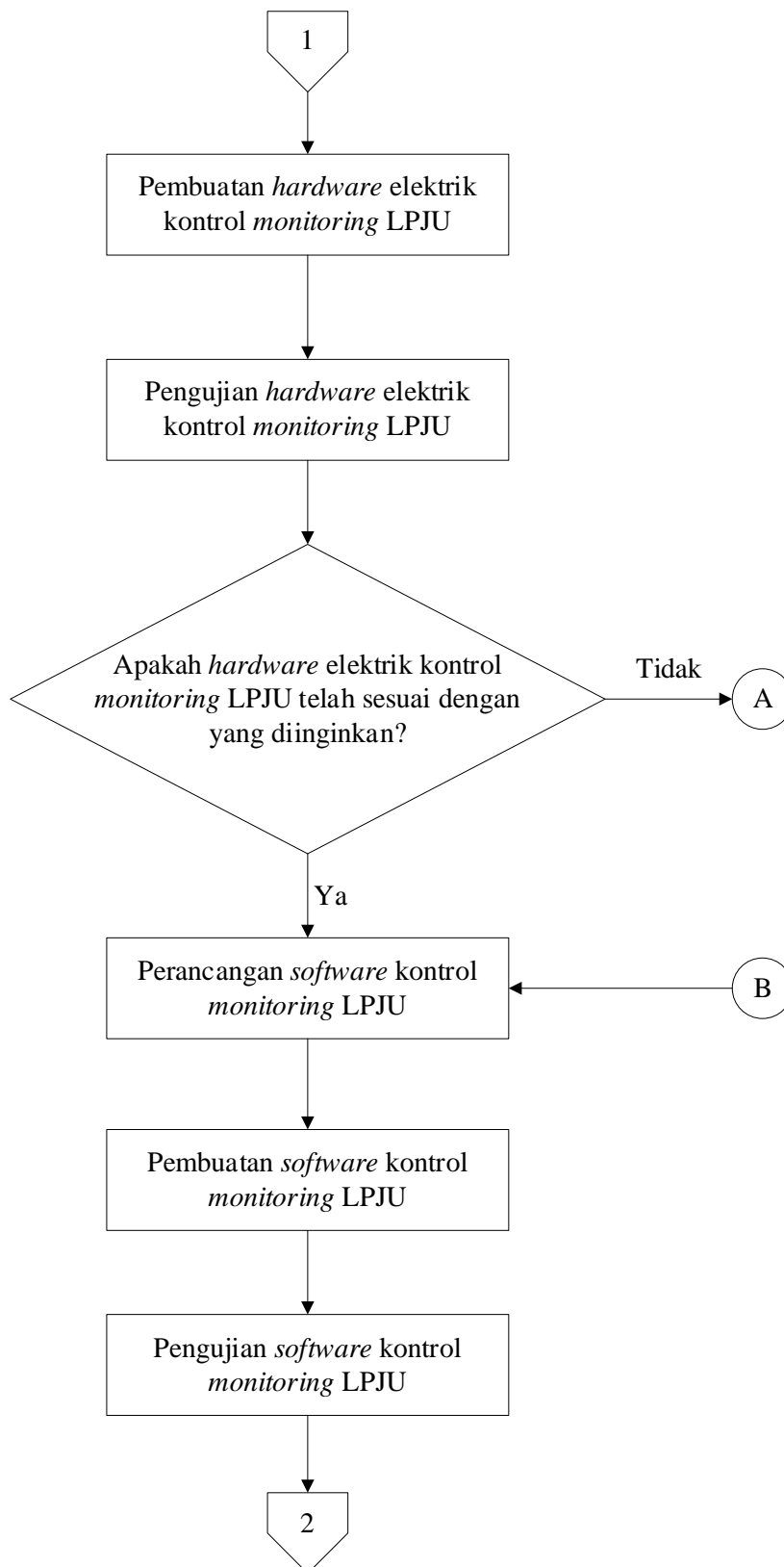
Tabel 2. 8 Spesifikasi modul AC *light dimmer*

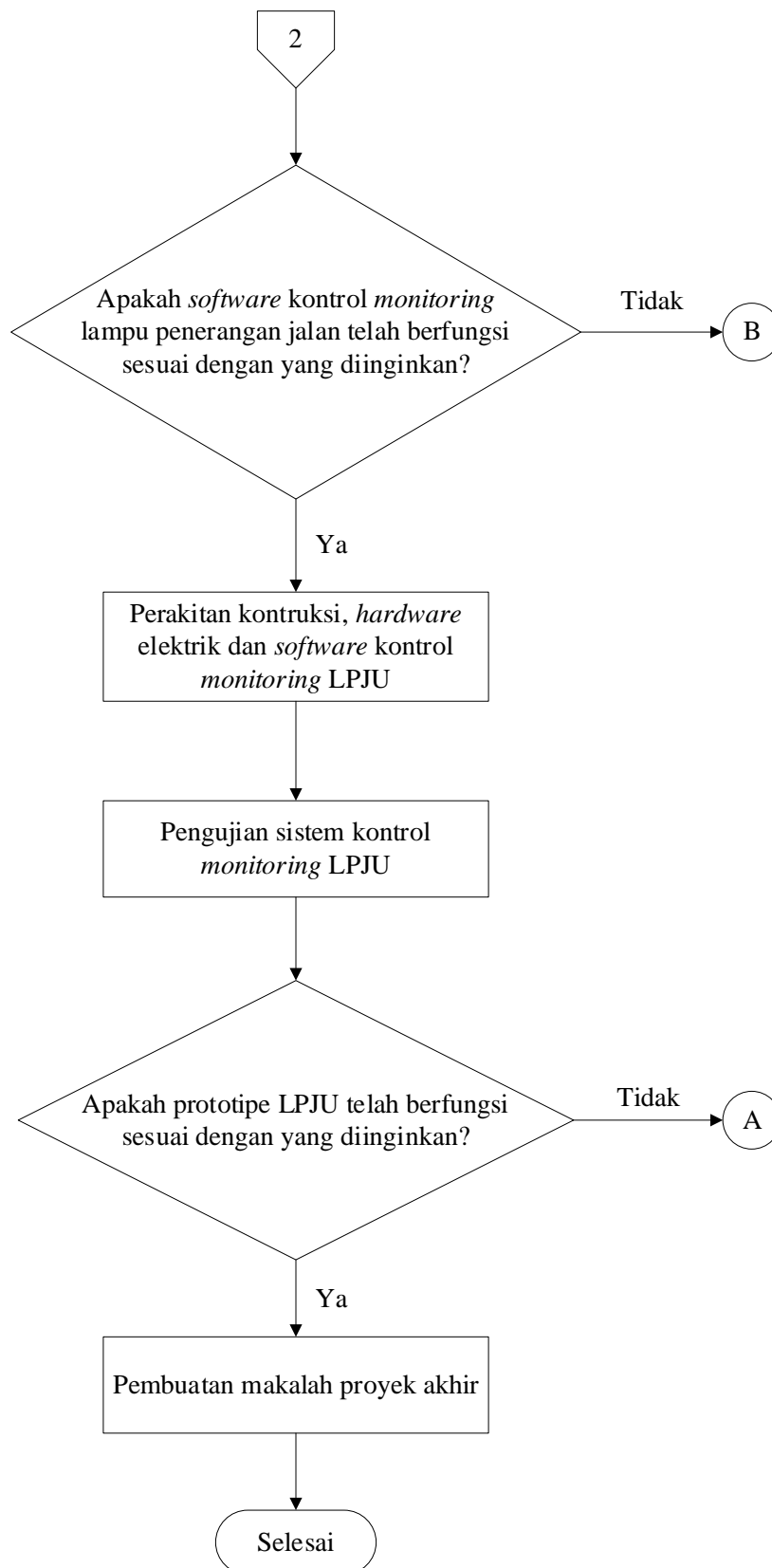
Spesifikasi	Keterangan
<i>Power</i>	600 V-16 A
<i>AC frequency</i>	50/60 Hz
<i>TRIAC</i>	BTA16-600B
<i>Isolation</i>	Optocoupler
<i>Logic level</i>	3,3 V/5 V
<i>Zero point</i>	Logic level
<i>Modulation (PWM with trigger)</i>	Logic level ON/OFF TRIAC
<i>Signal current</i>	<10 mA
<i>Environment</i>	<ul style="list-style-type: none"> • For indoor and outdoor use • Operating temperature 20°C to 80°C
<i>Operating humidity</i>	Dry environment only
<i>ROHS3</i>	Compliant

BAB III METODE PELAKSANAAN

Dalam pembuatan proyek akhir ini dilakukan beberapa tahapan mulai dari tahap pengumpulan data sampai dengan tahap pembuatan makalah. Berikut merupakan *flowchart* metode pelaksanaan proyek akhir dapat dilihat pada gambar 3.1:







Gambar 3. 1 *Flowchart* metode pelaksanaan

3.1 Pengumpulan dan Analisis Data

Pengumpulan data merupakan proses mencari dan mengumpulkan data-data yang diperlukan dalam proses pembuatan proyek akhir. Selanjutnya, data dianalisis untuk menentukan apa saja yang akan diterapkan pada proyek akhir sehingga proyek akhir ini memiliki kelebihan tersendiri dari kontrol *monitoring* lampu penerangan jalan yang pernah dibuat.

3.2 Perancangan Konstruksi Kontrol *Monitoring* LPJU

Perancangan konstruksi kontrol *monitoring* LPJU merupakan proses pembuatan desain konstruksi kontrol *monitoring* LPJU untuk meletakkan seluruh komponen yang akan digunakan. *Software* yang digunakan untuk mendesainnya adalah *Inventor*. Saat desain dari LPJU tersebut telah dibuat sesuai dengan kriteria yang diinginkan maka tahap selanjutnya adalah melakukan pembuatan konstruksi LPJU. Semua komponen yang akan digunakan seperti pipa akan terpasang pada tripleks yang dicat berwarna hitam. Setelah tahap pembuatan konstruksi LPJU selesai maka tahap selanjutnya adalah melakukan perakitan konstruksi LPJU agar membentuk suatu konstruksi yang sesuai dengan desain yang telah ditentukan. Perakitan konstruksi kontrol *monitoring* LPJU dilakukan dengan cara menyambungkan tripleks dan pipa menggunakan kabel ties dan menyambungkan kedua buah pipa dengan *elbow* pipa.

3.3 Perancangan *Hardware* Elektrik Kontrol *Monitoring* LPJU

Perancangan *hardware* elektrik kontrol *monitoring* LPJU merupakan proses pembuatan rancangan kelistrikan kontrol *monitoring* LPJU yang bertujuan untuk menentukan komponen yang akan digunakan seperti Arduino Mega, lampu, sensor-sensor, NodeMCU ESP8266, LCD 20x4, dan *relay*. Saat rancangan kelistrikan telah dibuat sesuai dengan kriteria yang diinginkan maka tahap selanjutnya yaitu melakukan perakitan semua komponen menjadi satu sesuai dengan rancangan. Setelah semua terkoneksi atau terhubung dengan baik, lakukan pengujian pada setiap komponen yang telah terkoneksi apakah sudah berfungsi dengan baik atau tidak.

3.4 Perancangan *Software* Kontrol *Monitoring* LPJU

Perancangan *software* kontrol *monitoring* LPJU merupakan proses pembuatan rancangan tampilan aplikasi pada *smartphone* untuk *me-monitoring* kerusakan dan mengatur kecerahan LPJU menggunakan aplikasi Mit App Inventor. Perancangan *software* meliputi rancangan tampilan *log in*, *log out*, pengaturan kecerahan dan *monitoring* kerusakan LPJU. Saat rancangan *software* telah dibuat sesuai dengan kriteria yang diinginkan maka tahap selanjutnya yaitu membuat *software* kontrol *monitoring* LPJU ini meliputi:

- Pembuatan aplikasi pada *smartphone* dengan menggunakan aplikasi *mit app inventor*.
- Pembuatan program keseluruhan pada Arduino Mega.

3.5 Perakitan Konstruksi dan *Hardware* Elektrik Kontrol *Monitoring* LPJU

Perakitan *hardware* konstruksi dan elektrik kontrol *monitoring* LPJU dilakukan dengan cara merakit keseluruhan pada setiap bagian dari *hardware* konstruksi dan elektrik digabung menjadi satu kesatuan.

3.6 Pengujian Kontrol *Monitoring* LPJU

Pada tahap ini dilakukan pengujian kontrol *monitoring* LPJU secara keseluruhan agar mengetahui cara kerja dari sistem cerdas lampu penerangan jalan ini, yaitu dapat menyala dan padam yang diatur oleh sensor intensitas cahaya GY-302 BH170, dapat menyala terang dan redup yang diatur oleh sensor *proximity infrared* dan modul *AC light dimmer* serta sensor LDR untuk mendeteksi kerusakan LPJU. Uji coba ini juga bertujuan untuk memastikan apakah semua komponen telah berkoneksi dan berfungsi sesuai dengan baik atau belum.

3.7 Pembuatan Makalah Proyek Akhir

Pembuatan makalah proyek akhir merupakan tahap terakhir dalam pembuatan proyek akhir yang bertujuan untuk merangkum hasil dari proyek akhir. Dalam makalah proyek akhir ini berisi pendahuluan, landasan teori, pembahasan serta kesimpulan dan saran.

BAB IV PEMBAHASAN

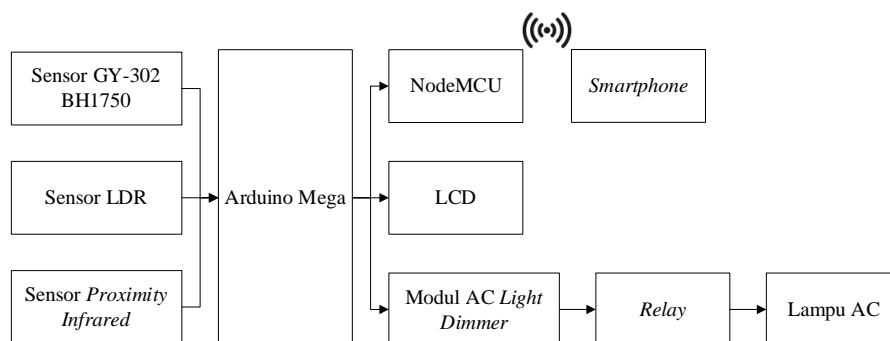
4.1 Deskripsi Alat

Sistem cerdas lampu penerangan jalan berbasis *IoT* ini merupakan sistem kontrol dan *monitoring* pada lampu penerangan jalan. Sistem pengontrolan dan pemantauan menggunakan Arduino Mega untuk mengolah data dan NodeMCU ESP8266 untuk mengirim data yang akan ditampilkan di *smartphone*. Pada sistem kontrol dan *monitoring* ini menggunakan sensor intensitas cahaya, sensor LDR dan sensor *proximity infrared* sebagai kontrol, lampu AC serta menggunakan modul *AC light dimmer*.

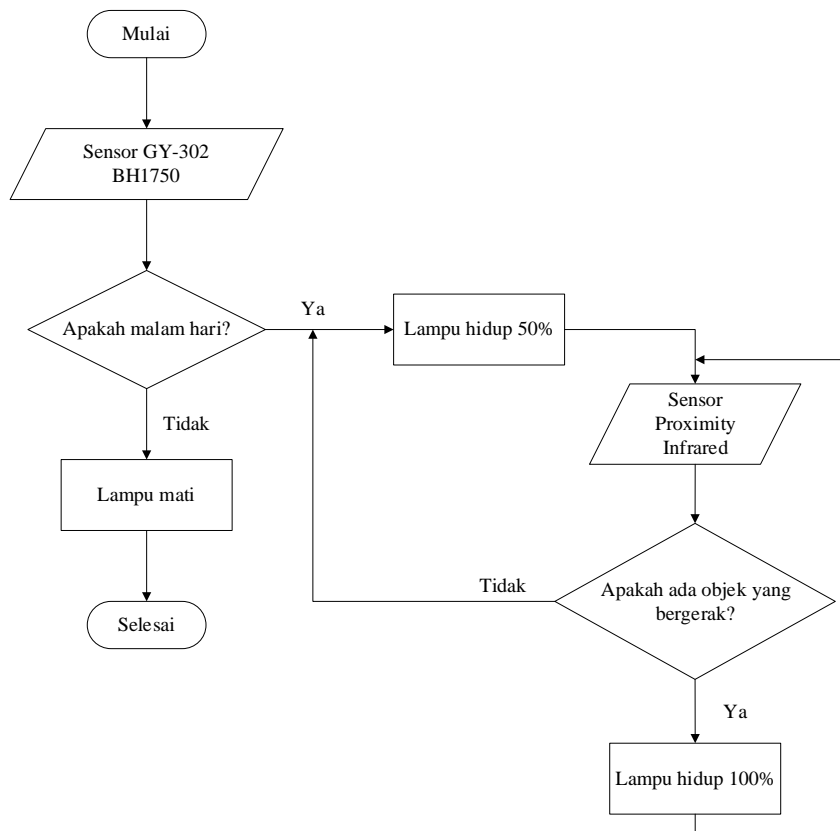
Sistem kontrol dan *monitoring* ini dapat menampilkan kerusakan lampu berdasarkan pengontrolan sensor LDR dan dapat mengontrol kecerahan lampu berdasarkan pengontrolan modul *AC light dimmer*, sensor *proximity infrared* dan sensor intensitas cahaya pada *smartphone*. Beban lampu yang digunakan adalah tiga buah lampu AC 25 watt.

4.1.1 Blok Diagram dan *Flowchart*

Sistem lampu penerangan jalan berbasis *IoT* ini mempunyai blok diagram dan *flowchart* agar mempermudah pembacaan prinsip kerja proyek akhir. Blok diagram dan *flowchart* ini dibuat sesuai dengan konsep dari proyek akhir serta pemakaian komponen-komponen juga disesuaikan dengan fungsi-fungsi komponen pada proyek akhir. Berikut merupakan blok diagram dapat di lihat pada gambar 4.1 dan *flowchart* dapat dilihat pada gambar 4.2:



Gambar 4. 1 Blok diagram sistem lampu penerangan jalan



Gambar 4. 2 *Flowchart* sistem lampu penerangan jalan

4.1.2 Prinsip Kerja Blok Diagram dan *Flowchart*

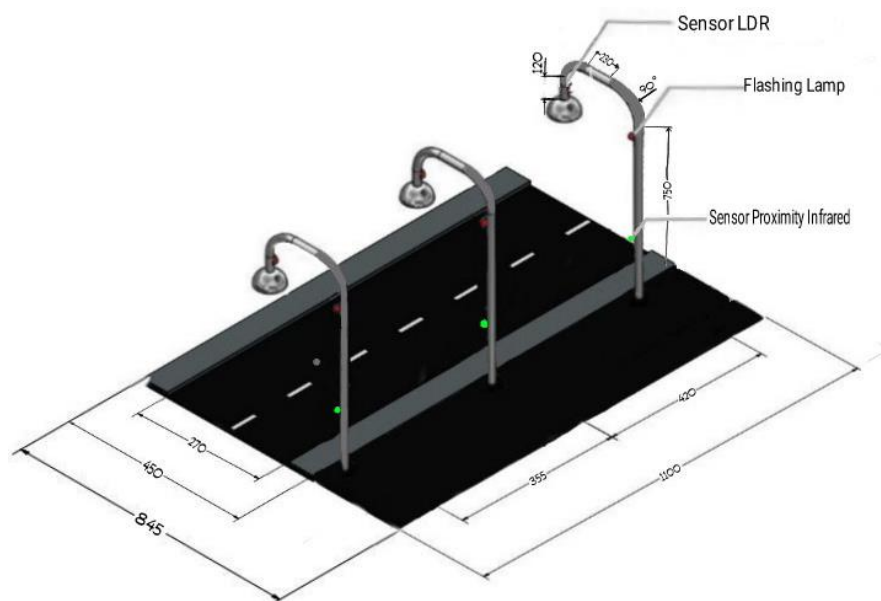
Pada gambar 4.1 merupakan blok diagram sistem lampu penerangan jalan yang di dalamnya terdapat sensor GY-302 BH1750, sensor LDR, sensor *proximity infrared*, Arduino Mega, NodeMCU ESP8266, modul *AC light dimmer*, *relay*, lampu AC, dan *smartphone*. *Input* dari sensor GY-302 BH1750, sensor LDR dan sensor *proximity infrared* akan dikirim ke Arduino. Selanjutnya Arduino Mega akan meneruskan data tersebut ke modul *AC light dimmer* untuk mengatur kecerahan lampu AC dan *relay* untuk menyalakan atau mematikan lampu AC. Data yang telah diolah oleh Arduino Mega akan ditampilkan pada LCD dan dikirimkan ke NodeMCU ESP8266 menggunakan internet agar dapat dilihat melalui *smartphone* petugas.

Pada gambar 4.2 merupakan *flowchart* sistem lampu penerangan jalan yang di dalamnya terdapat sensor GY-302 BH1750, sensor *proximity infrared* dan lampu. Sensor GY-302 BH1750 akan menangkap intensitas cahaya di sekitar sensor

tersebut. Jika intensitas cahaya melewati nilai ambang batas gelap sensor tersebut maka lampu akan padam. Sebaliknya jika intensitas cahaya di bawah nilai ambang batas gelap sensor tersebut maka lampu akan menyala 50%. Lalu jika ada objek yang melewati lampu maka sensor proximity akan mendeteksi objek tersebut sehingga lampu akan menyala 100%. Sebaliknya jika sensor proximity tidak mendeteksi objek apa pun maka lampu akan tetap menyala 50%.

4.2 Perancangan Konstruksi Kontrol *Monitoring* LPJU

Perancangan konstruksi kontrol *monitoring* LPJU dirancang sesuai dengan konsep yang telah ditentukan. Desain ini dibuat menggunakan *software* Inventor. Berikut merupakan desain konstruksi kontrol *monitoring* LPJU dapat dilihat pada gambar 4.3:



Gambar 4. 3 Desain konstruksi kontrol *monitoring* LPJU

4.3 Pembuatan Konstruksi Kontrol *Monitoring* LPJU

Pada bab sebelumnya, telah dijelaskan mengenai desain konstruksi kontrol *monitoring* LPJU. Setelah desain dibuat, langkah selanjutnya yaitu membuat konstruksi yang telah dirancang sebelumnya. Pembuatan konstruksi kontrol *monitoring* LPJU dilakukan di luar kampus Politeknik Manufaktur Negeri Bangka Belitung melainkan di rumah pribadi karena pembuatan prototipe LPJU tidak perlu memakai atau meminjam peralatan yang berada di laboratorium kampus.

Konstruksi kontrol *monitoring* LPJU dibuat sesuai dengan desain yang telah dirancang sebelumnya. Prototipe LPJU terdiri dari jalan dan tiang lampu jalan. Alat dan bahan yang digunakan dalam pembuatan konstruksi prototipe LPJU ini antara lain: tripleks dengan ketebalan 12 mm, pipa dengan diameter 1,5 inch, *elbow* pipa 1,5 inch, *fitting* lampu, sterofom, spidol putih, pilox hitam dan putih.

4.4 Perakitan Konstruksi Kontrol *Monitoring* LPJU

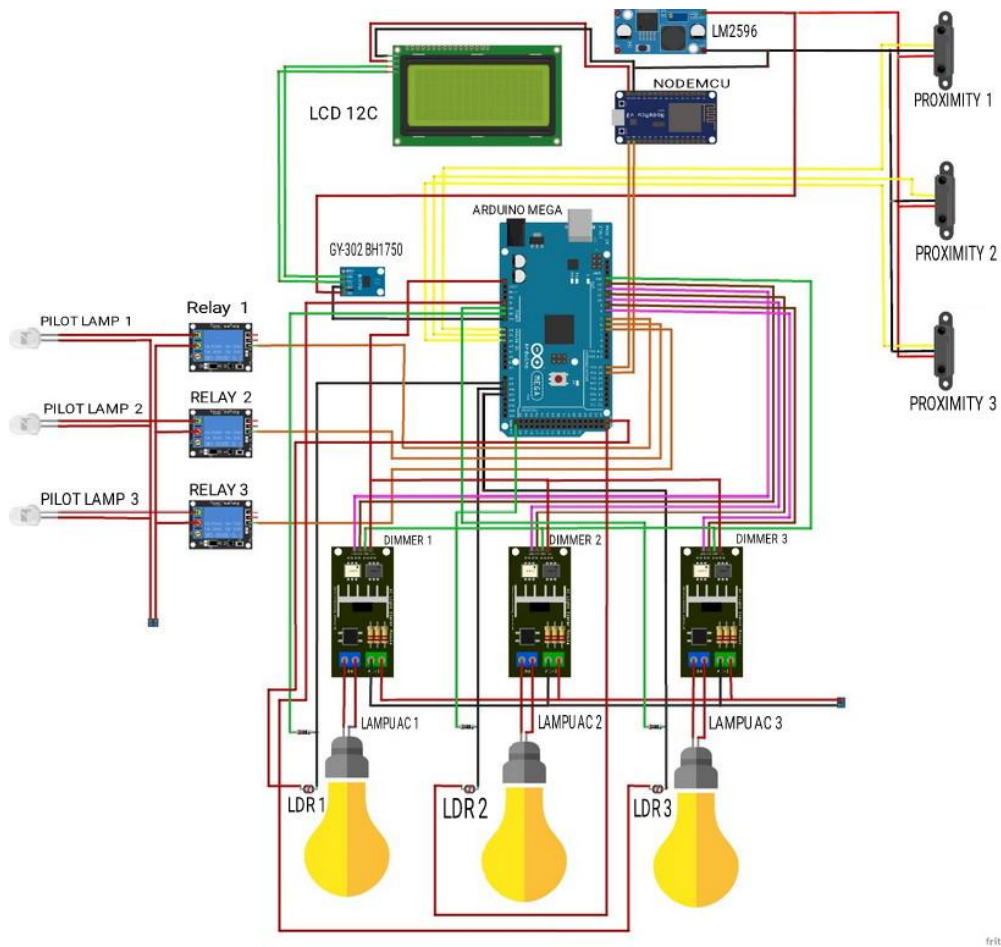
Pada tahap perakitan konstruksi kontrol *monitoring* LPJU ini dirakit secara keseluruhan. Perakitan ini berupa penyambungan dua buah pipa dengan *elbow* pipa, pemasangan *fitting* lampu pada pipa, pemasangan trotoar pada jalan dan pemasangan tiang-tiang lampu dengan jalan yang disesuaikan dengan desain. Selanjutnya tahap pewarnaan marka jalan yang beri warna putih. Berikut merupakan gambar konstruksi kontrol *monitoring* LPJU yang telah selesai pada gambar 4.4:



Gambar 4. 4 Konstruksi kontrol *monitoring* LPJU

4.5 Perancangan *Hardware* Elektrik Kontrol *Monitoring* LPJU

Perancangan *hardware* elektrik kontrol *monitoring* LPJU dilakukan dengan merancang peletakan komponen-komponen elektrik yang terdapat pada sistem prototipe. Komponen yang digunakan meliputi Arduino Mega, sensor GY-302 BH1750, sensor LDR, sensor *proximity infrared*, NodeMCU ESP8266, modul AC *light dimmer*, dan modul *relay*. Desain ini dirancang dengan menggunakan *software fritzing*. Berikut merupakan desain *hardware* elektrik kontrol *monitoring* LPJU dapat dilihat pada gambar 4.5:



Gambar 4. 5 Desain *hardware* elektrik kontrol *monitoring* LPJU

4.6 Pembuatan *Hardware* Elektrik Kontrol *Monitoring* LPJU

Pembuatan *hardware* elektrik kontrol *monitoring* LPJU dilakukan di luar kampus Politeknik Manufaktur Negeri Bangka Belitung melainkan di rumah pribadi karena pembuatan prototipe LPJU tidak perlu memakai atau meminjam peralatan yang ada di laboratorium kampus. Berikut merupakan *hardware* elektrik kontrol *monitoring* LPJU yang telah selesai pada gambar 4.6:



Gambar 4. 6 *Hardware* elektrik kontrol *monitoring* LPJU

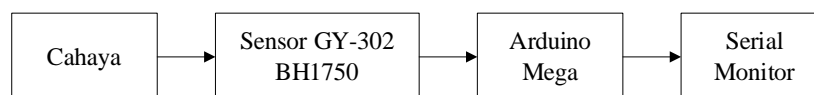
4.7 Pengujian *Hardware* Elektrik Kontrol *Monitoring* LPJU

Pengujian ini dilakukan untuk mengetahui apakah setiap komponen elektrik yang digunakan telah berfungsi sebagaimana mestinya atau belum. Berikut merupakan program utama pengujian setiap komponen yang digunakan:

```
#include "code_main.h"
#include "code_lcd.h"
#include "code_relay.h"
#include "code_proximity.h"
#include "code_light.h"
#include "code_acs.h"
#include "code_NodeMCU ESP8266.h"
void setup() {
  //==> Kode Test Program Komponen .
  testSerialConnection( tidak );
  test_proximity( tidak );
  test_komponen_relay( tidak );
  test_komponen_sensor_cahaya( tidak );
  test_komponen_sensor_LDR( tidak );
  testKomponen_ACLightDimmer( tidak );
  Serial.println("Selsesai");
}
```

4.7.1 Pengujian Modul Sensor Intensitas Cahaya GY-302 BH1750

Pengujian modul sensor GY-302 BH1750 bertujuan untuk mengetahui apakah sensor bekerja dengan baik atau tidak ketika ada cahaya. Pengujian ini dilakukan dengan menghubungkan modul sensor GY-302 BH1750 ke Arduino Mega menggunakan kabel jumper. Berikut merupakan blok diagram pengujian modul sensor GY-302 BH1750 dapat dilihat pada gambar 4.7:



Gambar 4. 7 Blok diagram pengujian modul sensor GY-302 BH1750

Berikut merupakan ilustrasi pengujian modul sensor GY-302 BH1750 dapat dilihat pada gambar 4.8:



Gambar 4. 8 Ilustrasi pengujian modul sensor GY-302 BH1750

Berikut merupakan program kode pengujian modul sensor GY-302 BH1750

ke program utama pengujian:



```




void test_komponen_sensor_cahaya( bool izinkan ){
  //==> cek jika kode di insinkan untuk menyala .
  //==> Parameter untuk cek jika ada perubahan data sinar .
  float value_cahaya = 0.0 ;
  while( izinkan == ya ){
    display_test_sensor_cahaya();
    if( abs(value_cahaya-lightMeter.readLightLevel())>10 ){
      value_cahaya = lightMeter.readLightLevel();
      Serial.print("Perubahan Cahaya : Nilai Lux : " + String(
value_cahaya ) + " , " );
      if( cek_status_sensor_cahaya() == masih_terang ){
        Serial.println("Masih Terang");
      }
      if( cek_status_sensor_cahaya() == sudah_gelap ){
        Serial.println("Sudah Gelap");
      }
    }
  }
}

```

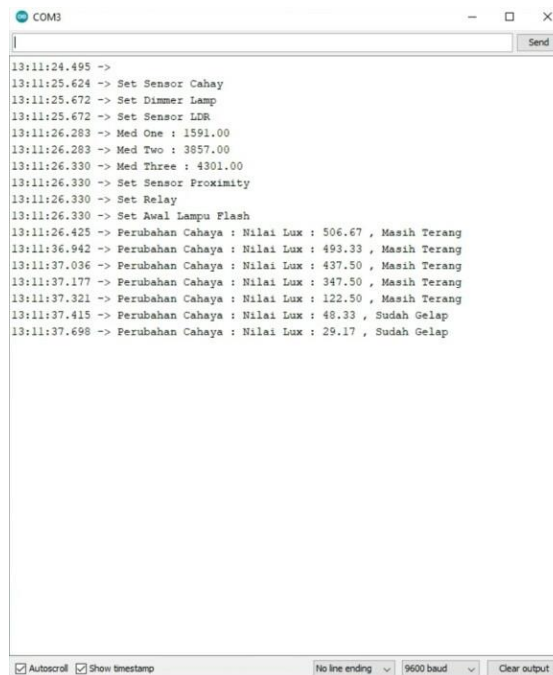
Berikut merupakan hasil pengujian modul sensor GY-302 BH1750 dapat dilihat pada tabel 4.1:

Tabel 4. 1 Hasil pengujian modul sensor GY-302 BH1750

<i>Input</i> (Cahaya)	Nilai Ambang Batas Gelap	<i>Output</i> (Serial Monitor)
	60 Lux	506,67 Lux: Masih Terang
		493,33 Lux: Masih Terang
48,33 Lux: Sudah Gelap		
29,17 Lux: Sudah Gelap		
	400 Lux	506,67 Lux: Masih Terang
		517,50 Lux: Masih Terang

<i>Input (Cahaya)</i>	Nilai Ambang Batas Gelap	<i>Output (Serial Monitor)</i>
		235,00 Lux: Sudah Gelap
		69,17 Lux: Sudah Gelap
	600 Lux	648,33 Lux: Masih Terang
		679,17 Lux: Masih Terang
		506,67 Lux: Sudah Gelap
		467,50 Lux: Sudah Gelap
		

Berikut merupakan serial monitor pengujian modul sensor GY-302 BH1750 pada nilai ambang batas gelap 60 Lux dapat dilihat pada gambar 4.9:



```

13:11:24.495 ->
13:11:25.624 -> Set Sensor Cahay
13:11:25.672 -> Set Dimmer Lamp
13:11:25.672 -> Set Sensor LDR
13:11:26.283 -> Med One : 1591.00
13:11:26.283 -> Med Two : 3857.00
13:11:26.330 -> Med Three : 4301.00
13:11:26.330 -> Set Sensor Proximity
13:11:26.330 -> Set Relay
13:11:26.330 -> Set Awal Lampu Flash
13:11:26.425 -> Perubahan Cahaya : Nilai Lux : 506.67 , Masih Terang
13:11:36.942 -> Perubahan Cahaya : Nilai Lux : 493.33 , Masih Terang
13:11:37.036 -> Perubahan Cahaya : Nilai Lux : 437.50 , Masih Terang
13:11:37.177 -> Perubahan Cahaya : Nilai Lux : 347.50 , Masih Terang
13:11:37.321 -> Perubahan Cahaya : Nilai Lux : 122.50 , Masih Terang
13:11:37.415 -> Perubahan Cahaya : Nilai Lux : 48.33 , Sudah Gelap
13:11:37.698 -> Perubahan Cahaya : Nilai Lux : 29.17 , Sudah Gelap

```

Gambar 4. 9 Serial monitor pengujian modul sensor GY-302 BH1750 pada nilai ambang batas gelap 60 Lux

Berikut merupakan serial monitor pengujian modul sensor GY-302 BH1750 pada nilai ambang batas gelap 400 Lux dapat dilihat pada gambar 4.10:



```
COM3
13:12:33.795 -> Set Relay
13:12:33.795 -> Set Awal Lampu Flash
13:12:33.842 -> Perubahan Cahaya : Nilai Lux : 506.67 , Masih Terang
13:12:37.649 -> Perubahan Cahaya : Nilai Lux : 517.50 , Masih Terang
13:12:37.743 -> Perubahan Cahaya : Nilai Lux : 442.50 , Masih Terang
13:12:37.883 -> Perubahan Cahaya : Nilai Lux : 235.00 , Sudah Gelap
13:12:38.025 -> Perubahan Cahaya : Nilai Lux : 69.17 , Sudah Gelap
13:12:38.494 -> Perubahan Cahaya : Nilai Lux : 103.33 , Sudah Gelap
13:12:38.636 -> Perubahan Cahaya : Nilai Lux : 127.50 , Sudah Gelap
13:12:38.776 -> Perubahan Cahaya : Nilai Lux : 138.33 , Sudah Gelap
13:12:38.917 -> Perubahan Cahaya : Nilai Lux : 176.67 , Sudah Gelap
13:12:39.011 -> Perubahan Cahaya : Nilai Lux : 284.17 , Sudah Gelap
13:12:39.246 -> Perubahan Cahaya : Nilai Lux : 365.00 , Sudah Gelap
13:12:39.246 -> Perubahan Cahaya : Nilai Lux : 375.83 , Sudah Gelap
13:12:39.387 -> Perubahan Cahaya : Nilai Lux : 386.67 , Sudah Gelap
13:12:39.527 -> Perubahan Cahaya : Nilai Lux : 407.50 , Masih Terang
13:12:39.668 -> Perubahan Cahaya : Nilai Lux : 429.17 , Masih Terang
13:12:40.044 -> Perubahan Cahaya : Nilai Lux : 500.83 , Masih Terang
13:12:40.137 -> Perubahan Cahaya : Nilai Lux : 515.83 , Masih Terang
13:12:43.565 -> Perubahan Cahaya : Nilai Lux : 498.33 , Masih Terang
13:12:43.706 -> Perubahan Cahaya : Nilai Lux : 436.67 , Masih Terang
13:12:43.846 -> Perubahan Cahaya : Nilai Lux : 345.83 , Sudah Gelap
13:12:44.035 -> Perubahan Cahaya : Nilai Lux : 150.83 , Sudah Gelap
13:12:44.082 -> Perubahan Cahaya : Nilai Lux : 455.83 , Masih Terang
13:12:44.223 -> Perubahan Cahaya : Nilai Lux : 515.83 , Masih Terang
13:12:45.116 -> Perubahan Cahaya : Nilai Lux : 458.33 , Masih Terang
13:12:45.209 -> Perubahan Cahaya : Nilai Lux : 243.33 , Sudah Gelap
13:12:45.350 -> Perubahan Cahaya : Nilai Lux : 76.67 , Sudah Gelap
13:12:46.239 -> Perubahan Cahaya : Nilai Lux : 105.00 , Sudah Gelap
13:12:46.474 -> Perubahan Cahaya : Nilai Lux : 123.33 , Sudah Gelap
13:12:46.614 -> Perubahan Cahaya : Nilai Lux : 145.83 , Sudah Gelap
13:12:46.755 -> Perubahan Cahaya : Nilai Lux : 171.67 , Sudah Gelap
13:12:46.895 -> Perubahan Cahaya : Nilai Lux : 205.83 , Sudah Gelap
13:12:46.989 -> Perubahan Cahaya : Nilai Lux : 244.17 , Sudah Gelap
13:12:47.131 -> Perubahan Cahaya : Nilai Lux : 200.00 , Sudah Gelap
```

Gambar 4. 10 Serial monitor pengujian modul sensor GY-302 BH1750 pada nilai ambang batas gelap 400 Lux

Berikut merupakan serial monitor pengujian modul sensor GY-302 BH1750 pada nilai ambang batas gelap 600 Lux dapat dilihat pada gambar 4.11:

```

COM3
13:13:16.597 -> Set Sensor LDR
13:13:17.169 -> Med One : 1582.00
13:13:17.169 -> Med Two : 3857.00
13:13:17.215 -> Med Three : 4301.00
13:13:17.215 -> Set Sensor Proximity
13:13:17.262 -> Set Relay
13:13:17.262 -> Set Awal Lampu Flash
13:13:17.356 -> Perubahan Cahaya : Nilai Lux : 506.67 , Sudah Gelap
13:13:31.786 -> Perubahan Cahaya : Nilai Lux : 467.50 , Sudah Gelap
13:13:31.879 -> Perubahan Cahaya : Nilai Lux : 454.17 , Sudah Gelap
13:13:32.020 -> Perubahan Cahaya : Nilai Lux : 514.17 , Sudah Gelap
13:13:32.864 -> Perubahan Cahaya : Nilai Lux : 526.67 , Sudah Gelap
13:13:32.864 -> Perubahan Cahaya : Nilai Lux : 547.50 , Sudah Gelap
13:13:33.051 -> Perubahan Cahaya : Nilai Lux : 587.50 , Sudah Gelap
13:13:33.145 -> Perubahan Cahaya : Nilai Lux : 648.33 , Masih Terang
13:13:33.286 -> Perubahan Cahaya : Nilai Lux : 679.17 , Masih Terang
13:13:33.521 -> Perubahan Cahaya : Nilai Lux : 770.83 , Masih Terang
13:13:33.661 -> Perubahan Cahaya : Nilai Lux : 972.50 , Masih Terang
13:13:33.803 -> Perubahan Cahaya : Nilai Lux : 1139.17 , Masih Terang
13:13:33.898 -> Perubahan Cahaya : Nilai Lux : 1280.00 , Masih Terang
13:13:34.179 -> Perubahan Cahaya : Nilai Lux : 1253.33 , Masih Terang
13:13:34.414 -> Perubahan Cahaya : Nilai Lux : 1230.83 , Masih Terang
13:13:34.555 -> Perubahan Cahaya : Nilai Lux : 1157.50 , Masih Terang
13:13:34.696 -> Perubahan Cahaya : Nilai Lux : 1124.17 , Masih Terang
13:13:34.791 -> Perubahan Cahaya : Nilai Lux : 1059.17 , Masih Terang
13:13:34.931 -> Perubahan Cahaya : Nilai Lux : 1040.00 , Masih Terang
13:13:35.307 -> Perubahan Cahaya : Nilai Lux : 1060.83 , Masih Terang
13:13:35.448 -> Perubahan Cahaya : Nilai Lux : 1095.00 , Masih Terang
13:13:35.542 -> Perubahan Cahaya : Nilai Lux : 1334.17 , Masih Terang
13:13:35.682 -> Perubahan Cahaya : Nilai Lux : 1764.17 , Masih Terang
13:13:35.824 -> Perubahan Cahaya : Nilai Lux : 2310.00 , Masih Terang
13:13:35.918 -> Perubahan Cahaya : Nilai Lux : 2853.33 , Masih Terang
13:13:36.199 -> Perubahan Cahaya : Nilai Lux : 2646.67 , Masih Terang
13:13:36.341 -> Perubahan Cahaya : Nilai Lux : 2414.17 , Masih Terang
13:13:36.434 -> Perubahan Cahaya : Nilai Lux : 2460.83 , Masih Terang
Autoscroll Show timestamp No line ending 9600 baud Clear output

```

Gambar 4. 11 Serial monitor pengujian modul sensor GY-302 BH1750 pada nilai ambang batas gelap 600 Lux

Dari hasil pengujian di atas bisa disimpulkan bahwa sensor intensitas cahaya GY-302 BH1750 bekerja sesuai dengan fungsinya yaitu sensor mampu mendeteksi cahaya di sekitarnya dengan satuan lux.

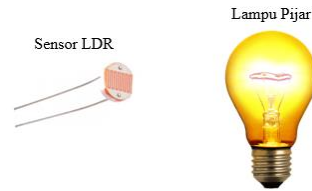
4.7.2 Pengujian Sensor LDR

Pengujian sensor LDR bertujuan untuk mengetahui apakah sensor bekerja dengan baik atau tidak ketika ada cahaya lampu AC. Pengujian ini dilakukan dengan menghubungkan sensor LDR ke Arduino Mega menggunakan kabel jumper. Berikut merupakan blok diagram pengujian sensor LDR dapat dilihat pada gambar 4.12:



Gambar 4. 12 Blok diagram pengujian sensor LDR

Berikut merupakan ilustrasi pengujian sensor LDR dapat dilihat pada gambar 4.13:



Gambar 4. 13 Ilustrasi pengujian sensor LDR

Berikut merupakan program kode pengujian modul sensor LDR ke program utama pengujian:

```
void test_komponen_sensor_LDR( bool izinkan ){
    if( izinkan == ya ){
        dimmer1.setPower( minimal_persen_lampu_dinyalakan ); //==> Set
        Dimmer 10 atau padam .
        dimmer2.setPower( minimal_persen_lampu_dinyalakan ); //==> Set
        Dimmer 10 atau padam .
        dimmer3.setPower( minimal_persen_lampu_dinyalakan ); //==> Set
        Dimmer 10 atau padam .
        Serial.println("Menyalakan Lampu selesai");
    }
    while( izinkan == ya ){
        display_test_sensor_cahaya();
        cahaya_lampu_1 = readLDR712( sensor_LDR_lampu_1 ) -
median_point_lampu_1 ;
        cahaya_lampu_1 = convert2Curent( cahaya_lampu_1 );
        cahaya_lampu_2 = readLDR712( sensor_LDR_lampu_2 ) -
median_point_lampu_2 ;
        cahaya_lampu_2 = convert2Curent( cahaya_lampu_2 );
        cahaya_lampu_3 = readLDR712( sensor_LDR_lampu_3 ) -
median_point_lampu_3 ;
        cahaya_lampu_3 = convert2Curent( cahaya_lampu_3 );
        test_komponen_sensor_LDR();
        Serial.print("Value : " + String(cahaya_lampu_1) );
        if( cahaya_lampu_1 > nilai_ambang_tidak_ada_cahaya ){
            Serial.print(" Lampu 1 Menyala ");
        }else{
            Serial.print(" Lampu 1 Rusak ");
        }
        Serial.print(" , Value : " + String(cahaya_lampu_2) );
        if( cahaya_lampu_2 > nilai_ambang_tidak_ada_cahaya ){
            Serial.print(" Lampu 2 Menyala ");
        }else{
            Serial.print(" Lampu 2 Rusak ");
        }
        Serial.print(" , Value : " + String(cahaya_lampu_3) );
        if( cahaya_lampu_3 > nilai_ambang_tidak_ada_cahaya ){
            Serial.print(" Lampu 3 Menyala ");
        }else{
            Serial.print(" Lampu 3 Rusak ");
        }
    }
}
```




```

    }
    Serial.println();
    delay(1000);
  }
}

```

Berikut merupakan hasil pengujian sensor LDR dapat dilihat pada tabel 4.2:

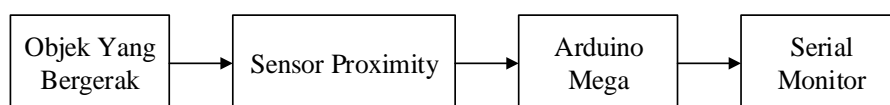
Tabel 4. 2 Hasil pengujian sensor LDR

<i>Input</i> (Cahaya Lampu AC)	Nilai Ambang Batas Gelap	<i>Output</i> (Serial Monitor)
	0,3 V	Value : 0.77 Lampu 1 Menyala Value : 1.51 Lampu 2 Menyala Value : 1.95 Lampu 3 Menyala
		Value : 0.29 Lampu 1 Rusak Value : 1.47 Lampu 2 Menyala Value : 1.94 Lampu 3 Menyala
		Value : 0.77 Lampu 1 Menyala Value : 0.11 Lampu 2 Rusak Value : 1.89 Lampu 3 Menyala

Dari hasil pengujian sensor LDR bisa disimpulkan sensor LDR bekerja sesuai dengan fungsinya yaitu sensor mampu mendeteksi cahaya pada lampu AC 25 watt.

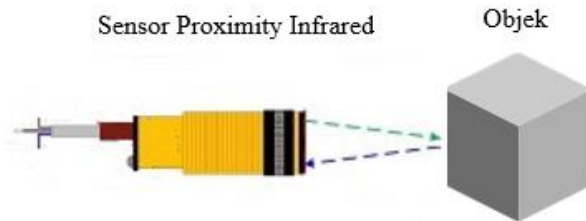
4.7.3 Pengujian Sensor *Proximity Infrared*

Pengujian sensor *proximity infrared* bertujuan untuk mengetahui apakah sensor bekerja dengan baik atau tidak ketika ada objek yang bergerak di depannya. Pengujian ini dilakukan dengan menghubungkan modul sensor *proximity infrared* ke Arduino Mega menggunakan kabel jumper. Berikut merupakan blok diagram pengujian sensor *proximity infrared* dapat dilihat pada gambar 4.14:



Gambar 4. 14 Blok diagram pengujian sensor *proximity infrared*

Berikut merupakan ilustrasi pengujian sensor *proximity infrared* dapat dilihat pada gambar 4.15:



Gambar 4. 15 Ilustrasi pengujian sensor *proximity infrared*

Berikut merupakan program kode pengujian sensor *proximity infrared* ke program utama pengujian:

```
void test_proximity( bool izinkan ){
  while( izinkan == ya ){
    //==> Reset lcd terlebih dahulu .
    lcd.clear();
    //==> Test proximity lampu 1 .
    lcd.setCursor( 0 , 0 );
    if( baca_status_proximity( proximity_lampu_1 ) == ya ){
      lcd.print("Prox 1 Deteksi");
      Serial.print("Prox 1 Deteksi");
    }
    else{
      lcd.print("Prox 1 non");
      Serial.print(" , Prox 1 non");
    }
    //==> Test proximity lampu 2 .
    lcd.setCursor( 0 , 1 );
    if( baca_status_proximity( proximity_lampu_2 ) == ya ){
      lcd.print("Prox 2 Deteksi");
      Serial.print(" , Prox 2 Deteksi");
    }
    else{
      lcd.print("Prox 2 non");
      Serial.print(" , Prox 2 non");
    }
    //==> Test proximity lampu 3 .
    lcd.setCursor( 0 , 2 );
    if( baca_status_proximity( proximity_lampu_3 ) == ya ){
      lcd.print("Prox 3 Deteksi");
      Serial.print(" , Prox 3 Deteksi");
    }
    else{
      lcd.print("Prox 3 non");
      Serial.print(" , Prox 3 non");
    }
    Serial.println();
    delay(1500);
  }
}
```

Berikut merupakan serial monitor pengujian sensor *proximity infrared* dapat dilihat pada gambar 4.16:



Gambar 4. 16 Serial monitor pengujian sensor *proximity infrared*

Berikut merupakan salah satu pengujian sensor *proximity infrared* dapat dilihat pada gambar 4.17:

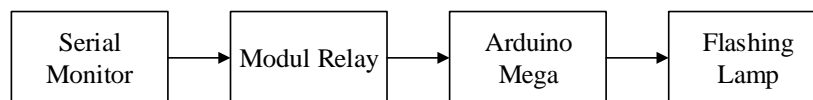


Gambar 4. 17 Pengujian sensor *proximity infrared*

Dari hasil pengujian sensor *proximity infrared* bisa disimpulkan sensor *proximity infrared* bekerja sesuai dengan fungsinya yaitu sensor mampu mendeteksi objek yang bergerak bergerak didepannya.

4.7.4 Pengujian Modul Relay

Pengujian modul *relay* bertujuan untuk mengetahui apakah *relay* bekerja dengan baik atau tidak menggunakan *flashing lamp*. Pengujian ini dilakukan dengan menghubungkan modul *relay* ke Arduino Mega menggunakan kabel jumper. Berikut merupakan blok diagram pengujian modul *relay* dapat dilihat pada gambar 4.18:



Gambar 4. 18 Blok diagram pengujian modul *relay*

Berikut merupakan program kode pengujian modul *relay* ke program utama pengujian:

```
void test_komponen_relay( bool izinkan ){
  String msg = "" ;
  //==> status .
  bool relay_1 = true ;
  bool relay_2 = true ;
  bool relay_3 = true ;
  if( izinkan == ya ){
    display_test_proximity( relay_1 , relay_2 , relay_3 );
    settingStatusRelay( lamp_flasing_1 , relay_1 );
    relay_1 = !relay_1 ;
    settingStatusRelay( lamp_flasing_2 , relay_2 );
    relay_2 = !relay_2 ;
    settingStatusRelay( lamp_flasing_3 , relay_3 );
    relay_3 = !relay_3 ;
  }
  //==> cek jika test komponen relay di izinkan menyala .
  while( izinkan == ya ){
    Serial.println("=====>");
    Serial.println("Ketik A [ Enter ] untuk menyalakan dan
mematikan relay 1");
    Serial.println("Ketik B [ Enter ] untuk menyalakan dan
mematikan relay 2");
    Serial.println("Ketik C [ Enter ] untuk menyalakan dan
mematikan relay 3");
    while( true ){
      msg = readMessageCom();
      if( msg.length() > 0 ){
        break ;
      }
    }
    if( msg.indexOf("A") > -1 ){
      settingStatusRelay( lamp_flasing_1 , relay_1 );
      if( relay_1 )
        Serial.println("Relay 1 Di Nyalakan ");
      else
```

```

        Serial.println("Relay 1 Di Padamkan ");
        relay_1 = !relay_1 ;
    }
    if( msg.indexOf("B") > -1 ){
        settingStatusRelay( lamp_flasing_2 , relay_2 );
        if( relay_2 )
            Serial.println("Relay 2 Di Nyalakan ");
        else
            Serial.println("Relay 2 Di Padamkan ");
        relay_2 = !relay_2 ;
    }
    if( msg.indexOf("C") > -1 ){
        settingStatusRelay( lamp_flasing_3 , relay_3 );
        if( relay_3 )
            Serial.println("Relay 3 Di Nyalakan ");
        else
            Serial.println("Relay 3 Di Padamkan ");
        relay_3 = !relay_3 ;
    }
    display_test_proximity( relay_1 , relay_2 , relay_3 );
}
}

```

Berikut merupakan serial monitor pengujian modul *relay* dapat dilihat pada gambar 4.19:

```


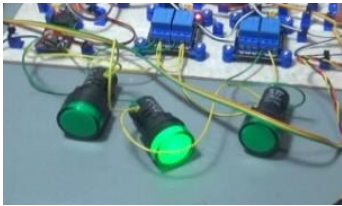

COM3
d
13:07:55.200 -> Set Awal Lampu Flash
13:07:55.295 -> =====>
13:07:55.295 -> Ketik A [ Enter ] untuk menyalakan dan mematikan relay 1
13:07:55.343 -> Ketik B [ Enter ] untuk menyalakan dan mematikan relay 2
13:07:55.438 -> Ketik C [ Enter ] untuk menyalakan dan mematikan relay 3
13:08:09.569 -> Relay 1 Di Padamkan
13:08:09.663 -> =====>
13:08:09.663 -> Ketik A [ Enter ] untuk menyalakan dan mematikan relay 1
13:08:09.757 -> Ketik B [ Enter ] untuk menyalakan dan mematikan relay 2
13:08:09.804 -> Ketik C [ Enter ] untuk menyalakan dan mematikan relay 3
13:08:24.310 -> Relay 1 Di Nyalakan
13:08:24.403 -> =====>
13:08:24.403 -> Ketik A [ Enter ] untuk menyalakan dan mematikan relay 1
13:08:24.497 -> Ketik B [ Enter ] untuk menyalakan dan mematikan relay 2
13:08:24.544 -> Ketik C [ Enter ] untuk menyalakan dan mematikan relay 3
13:08:26.563 -> Relay 1 Di Padamkan
13:08:26.705 -> =====>
13:08:26.705 -> Ketik A [ Enter ] untuk menyalakan dan mematikan relay 1
13:08:26.752 -> Ketik B [ Enter ] untuk menyalakan dan mematikan relay 2
13:08:26.846 -> Ketik C [ Enter ] untuk menyalakan dan mematikan relay 3
13:08:32.952 -> Relay 2 Di Padamkan
13:08:33.093 -> =====>
13:08:33.093 -> Ketik A [ Enter ] untuk menyalakan dan mematikan relay 1
13:08:33.140 -> Ketik B [ Enter ] untuk menyalakan dan mematikan relay 2
13:08:33.234 -> Ketik C [ Enter ] untuk menyalakan dan mematikan relay 3
13:08:36.668 -> Relay 2 Di Nyalakan
13:08:36.809 -> =====>
13:08:36.809 -> Ketik A [ Enter ] untuk menyalakan dan mematikan relay 1
13:08:36.904 -> Ketik B [ Enter ] untuk menyalakan dan mematikan relay 2
13:08:36.951 -> Ketik C [ Enter ] untuk menyalakan dan mematikan relay 3
13:08:42.819 -> Relay 3 Di Padamkan
13:08:42.960 -> =====>
13:08:42.960 -> Ketik A [ Enter ] untuk menyalakan dan mematikan relay 1
13:08:43.006 -> Ketik B [ Enter ] untuk menyalakan dan mematikan relay 2
13:08:43.053 -> Ketik C [ Enter ] untuk menyalakan dan mematikan relay 3
 Autoscroll  Show timestamp No line ending 9600 baud Clear output

```

Gambar 4. 19 Serial monitor pengujian modul *relay*

Berikut merupakan hasil pengujian modul *relay* dapat dilihat pada tabel 4.3:

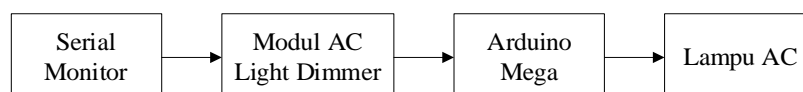
Tabel 4. 3 Hasil pengujian modul *relay*

<i>Input</i> (Serial Monitor) dan Keterangan	<i>Output</i> (<i>Flashing lamp</i>)
A <i>Relay 1</i> Dipadamkan	
B <i>Relay 2</i> Di Nyalakan	
C <i>Relay 3</i> Di Nyalakan	

Dari hasil pengujian modul *relay* bisa disimpulkan modul *relay* bekerja sesuai dengan fungsinya yaitu modul *relay* dapat memadamkan dan menyalakan *flashing lamp*.

4.7.5 Pengujian Modul AC *Light Dimmer*

Pengujian ini bertujuan untuk mengetahui apakah modul AC *light dimmer* bekerja dengan baik atau tidak ketika mengatur kecerahan lampu AC. Pengujian ini dilakukan dengan menghubungkan modul AC *light dimmer* ke Arduino Mega menggunakan kabel jumper. Berikut merupakan blok diagram pengujian modul AC *light dimmer* dapat dilihat pada gambar 4.20:



Gambar 4. 20 Blok diagram pengujian modul AC *light dimmer*

Berikut merupakan program kode pengujian modul AC *light dimmer* ke program utama pengujian:




```





void testKomponen_ACLightDimmer( bool izinkan ){
    String msg = "";
    while( izinkan == ya ){
        Serial.println("=====>");
        Serial.println("Ketik Nilai 0 - 100% untuk setting nilai
kecerahan : " );
        while( true ){
            msg = readMessageCom();
            if( msg.length() > 0 ){
                break ;
            }
        }
        Serial.println("Set Nilai Kecerahan : " + String(
msg.toInt() ));
        int value = msg.toInt();
        dimmer1.setPower( value );
        dimmer2.setPower( value );
        dimmer3.setPower( value );
    }
}

```

Berikut merupakan hasil pengujian modul AC *light dimmer* dapat dilihat pada tabel 4.4:

Tabel 4. 4 Hasil pengujian modul AC *light dimmer*

No	Input (Serial Monitor)	Output (Lampu AC)
1	0 %	
2	10 %	
3	25 %	

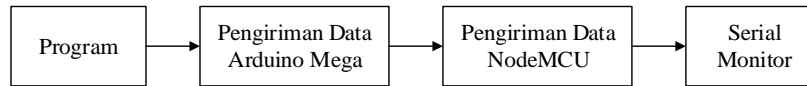
No	Input (Serial Monitor)	Output (Lampu AC)
4	50 %	
5	75 %	
6	90 %	
7	95 %	

Dari hasil pengujian modul AC *light dimmer* bisa disimpulkan modul AC *light dimmer* bekerja sesuai dengan fungsinya yaitu modul AC *light dimmer* dapat mengatur kecerahan lampu AC 25 watt.

4.7.6 NodeMCU ESP8266 ESP8266

Pengujian ini bertujuan untuk mengetahui apakah pengiriman data dari Arduino Mega ke NodeMCU ESP8266 melalui jaringan WiFi bekerja dengan baik

atau tidak. Berikut merupakan blok diagram pengujian serial komunikasi untuk pengiriman data dari Arduino Mega ke NodeMCU ESP8266 ESP8266 dapat dilihat pada gambar 4.21:



Gambar 4. 21 Blok diagram pengujian serial komunikasi untuk pengiriman data dari Arduino Mega ke NodeMCU ESP8266 ESP8266

Berikut merupakan program pengujian serial komunikasi untuk pengiriman data dari Arduino Mega ke NodeMCU ESP8266 ESP8266:

```

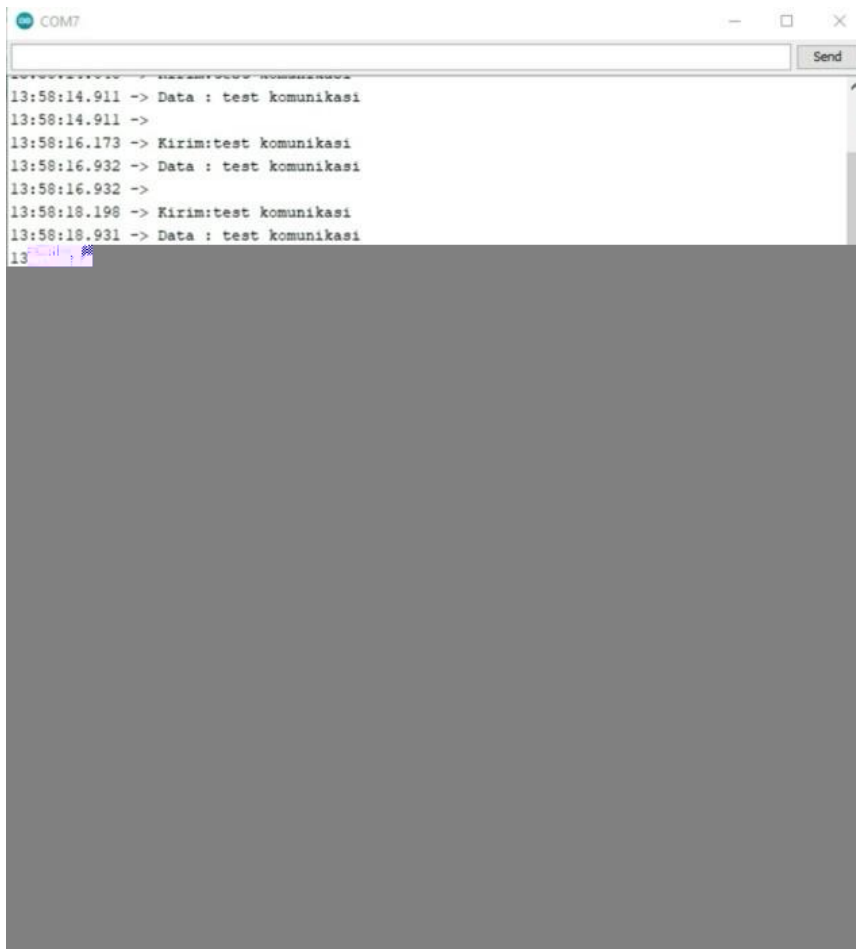
#include "FirebaseESP8266.h"
#include <SoftwareSerial.h>
#include <ESP8266HTTPClient.h>
#include <ESP8266WiFi.h>
#include "code_ard.h"
#include "code_server.h"
#include "code_firebase.h"
void setup() {
  Serial.begin(9600);
  esp.begin(9600);
  testSerialConnection( false );
  koneksiWifi();
  inisialKomunikasiFirebase();
}
unsigned long tmSendLogData = millis();
float maxTmMenit = 30 ;
void loop() {
  String msg = readMessageEsp();
  if( msg.length() > 0 ){
    Serial.println( msg );
    if( msg.indexOf("kirim") > -1 ){
      //==> Perintah untuk mengirim data ke server .
      updateDataServer_setSetting();
    }
    else{
      //===> data diterima .
      Serial.println("Data Diterima : " + msg );
      //===> parse data .
      parseData( msg );
      //===> Tampilkan data .
      showDataReceiver();
    }
  }
  //=====> Update informasi dari database .
  updateInformasiPerintahSettingDariDatabase();
  updateInformasiArduino();
}
  
```

Berikut merupakan hasil pengujian serial komunikasi antara Arduino Mega dan NodeMCU ESP8266 pada Arduino Mega dapat dilihat pada gambar 4.22:



Gambar 4. 22 Hasil pengujian serial komunikasi antara Arduino Mega dan NodeMCU ESP8266 pada Arduino Mega

Berikut merupakan hasil pengujian serial komunikasi antara Arduino Mega dan NodeMCU ESP8266 pada NodeMCU ESP8266 dapat dilihat pada gambar 4.23:

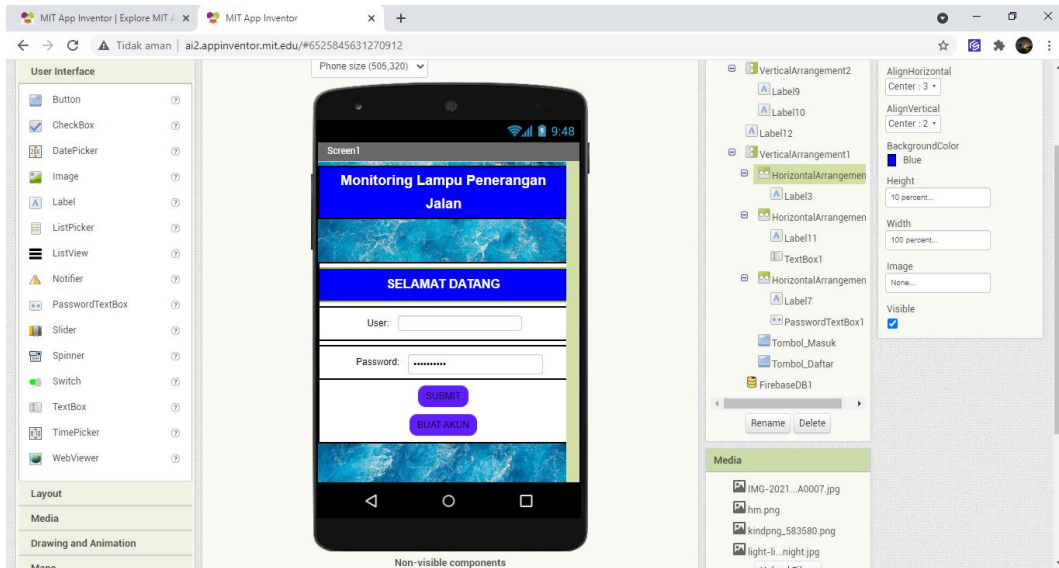


Gambar 4. 23 Hasil pengujian serial komunikasi antara Arduino Mega dan NodeMCU ESP8266 pada NodeMCU ESP8266

Dari hasil pengujian NodeMCU ESP8266 bisa disimpulkan NodeMCU ESP8266 bekerja sesuai dengan fungsinya yaitu NodeMCU ESP8266 dapat menerima data yang dikirimkan dari Arduino Mega.

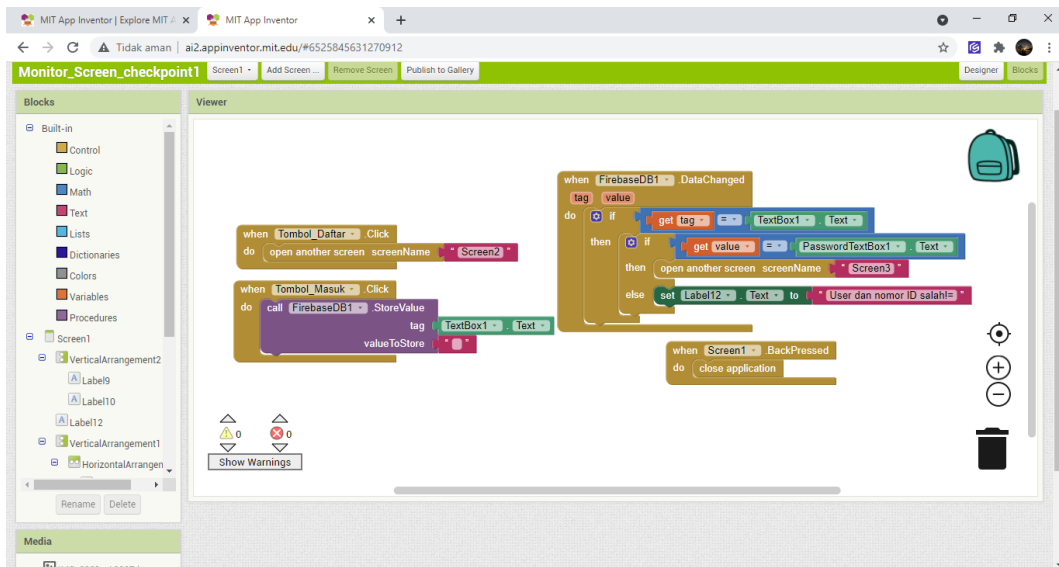
4.8 Perancangan *Software* Aplikasi

Pada perancangan *software* aplikasi pada *smartphone* ini menggunakan *Mit App Inventor* secara *online*. Pertama, buka *Mit App Inventor* di *google*, lalu klik *Create Apps!* dan *login* menggunakan *email*. Kemudian klik *Start new project* lalu tulis nama *project*-nya dan klik OK. Setelah itu, atur tampilan sesuai dengan rancangan yang telah ada dan buat *coding* blok agar aplikasi yang dibuat bisa berfungsi sesuai dengan yang diharapkan. Berikut merupakan tampilan *screen* pertama pada *software* aplikasi dapat dilihat pada gambar 4.24:



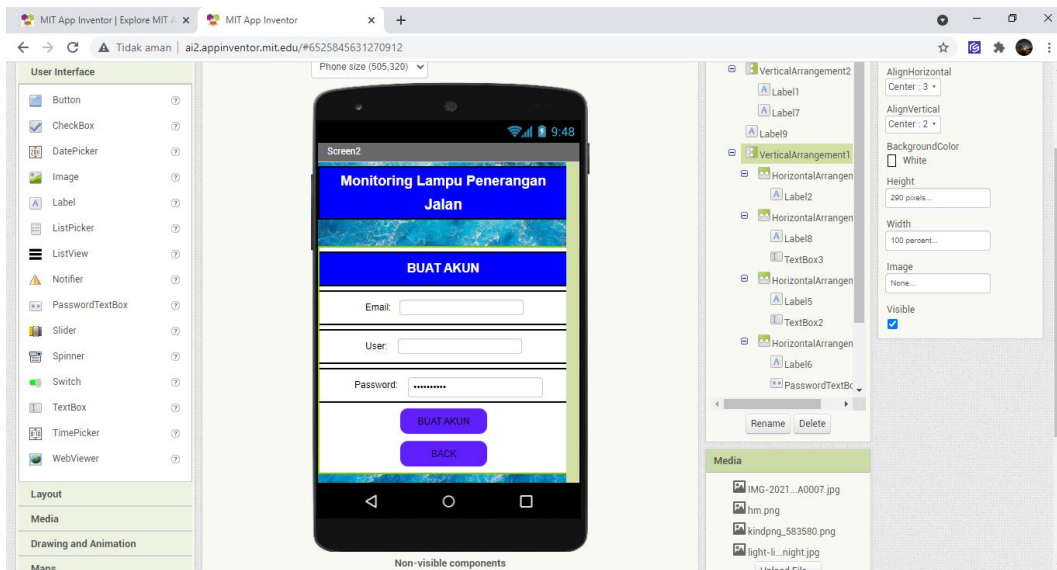
Gambar 4. 24 Tampilan *screen* pertama pada *software* aplikasi

Berikut merupakan *coding* blok untuk *screen* pertama dapat dilihat pada gambar 4.25:



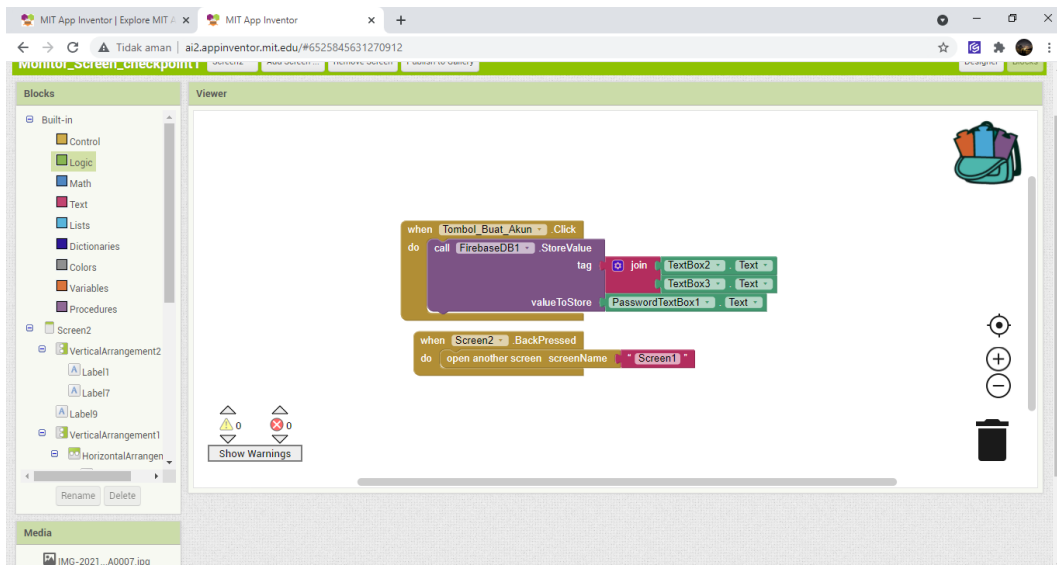
Gambar 4. 25 *Coding* blok untuk *screen* pertama

Berikut merupakan tampilan *screen* kedua pada *software* aplikasi dapat dilihat pada gambar 4.26:



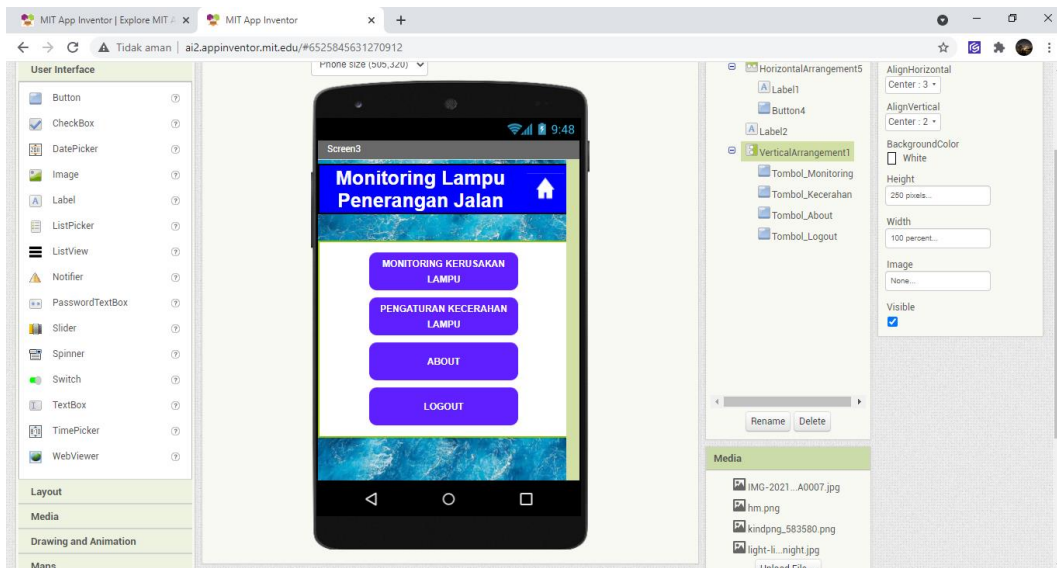
Gambar 4. 26 Tampilan *screen* kedua pada *software* aplikasi

Berikut merupakan *coding* blok untuk *screen* kedua dapat dilihat pada gambar 4.27:



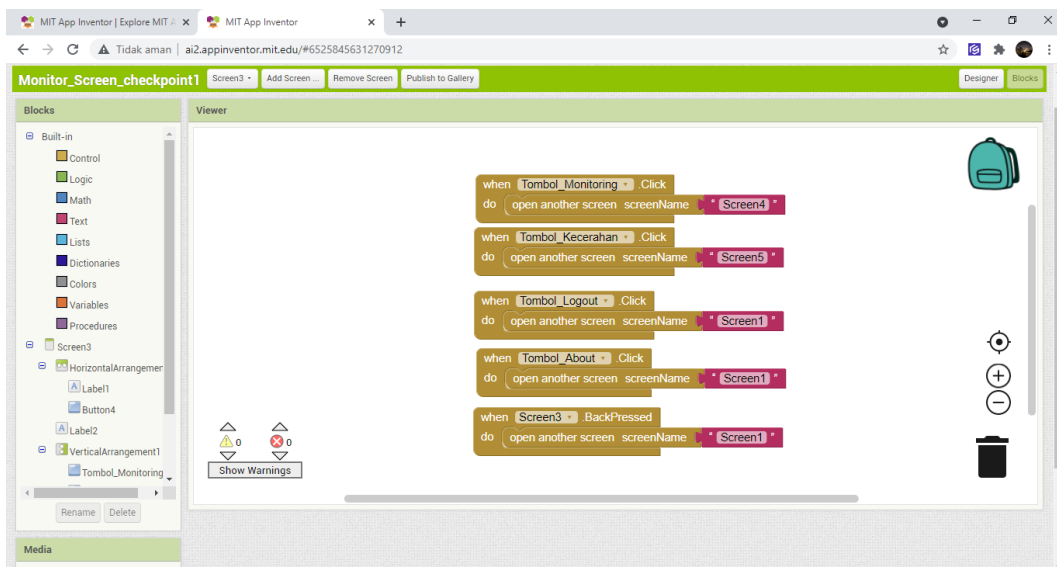
Gambar 4. 27 *Coding* blok untuk *screen* kedua

Berikut merupakan tampilan *screen* ketiga pada *software* aplikasi dapat dilihat pada gambar 4.28:



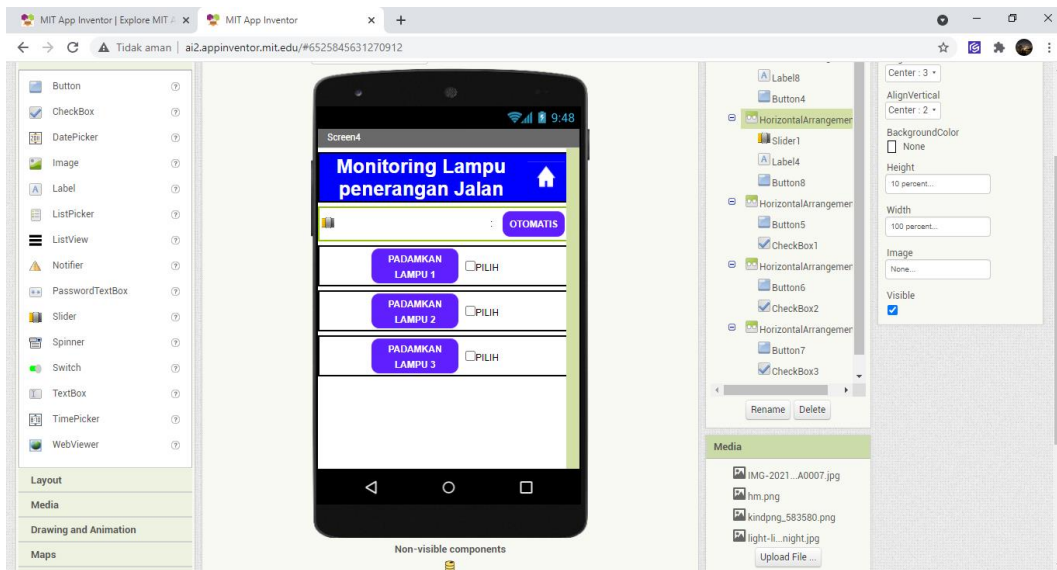
Gambar 4. 28 Tampilan *screen* ketiga pada *software* aplikasi

Berikut merupakan *coding* blok untuk *screen* ketiga dapat dilihat pada gambar 4.29:



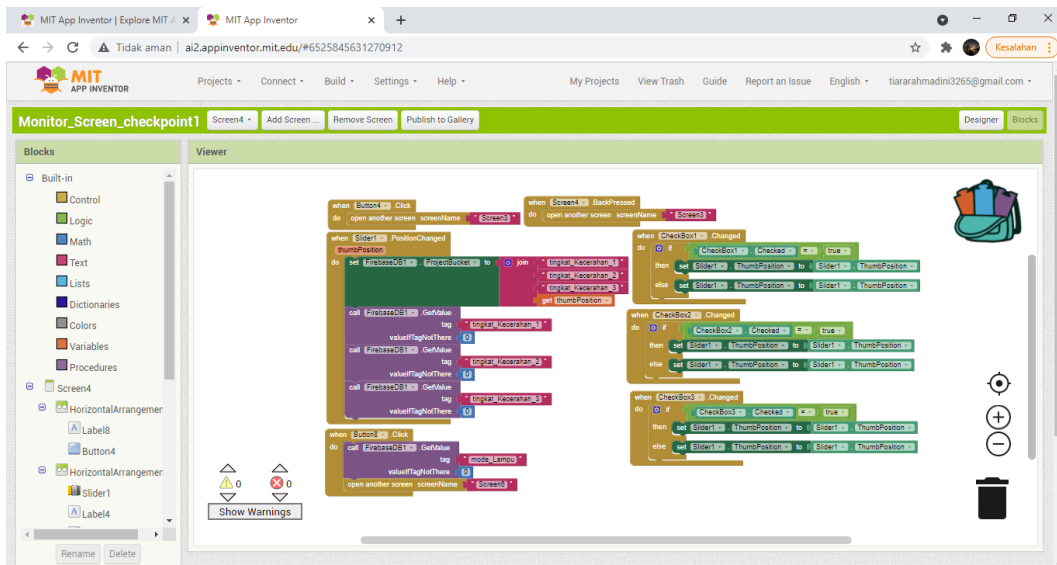
Gambar 4. 29 *Coding* blok untuk *screen* ketiga

Berikut merupakan tampilan *screen* keempat pada *software* aplikasi dapat dilihat pada gambar 4.30 dan *coding* blok untuk *screen* keempat dapat dilihat pada gambar 4.31:



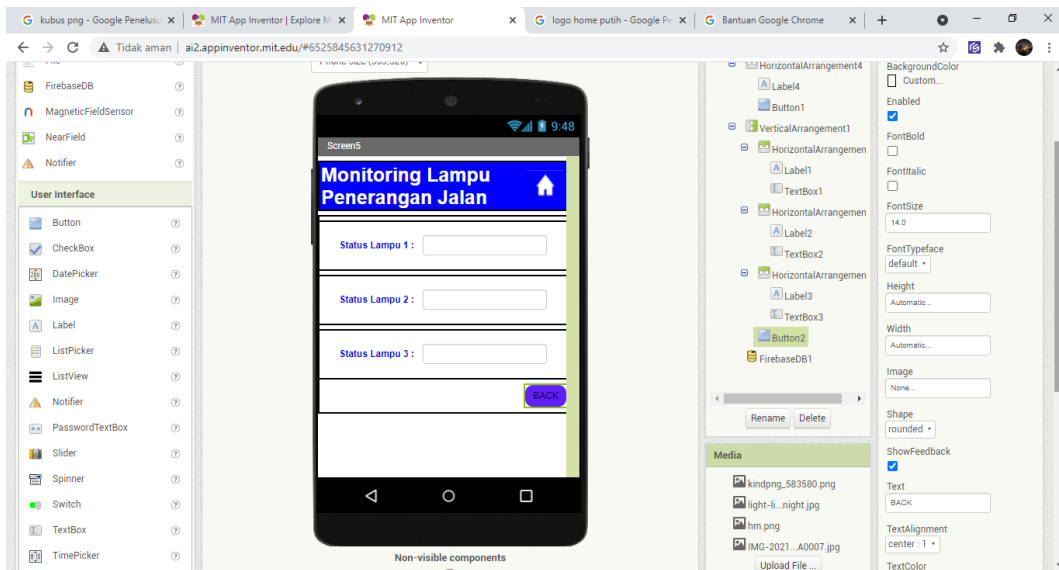
Gambar 4. 30 Tampilan *screen* keempat pada *software* aplikasi

Berikut merupakan *coding* blok untuk *screen* keempat dapat dilihat pada gambar 4.31:



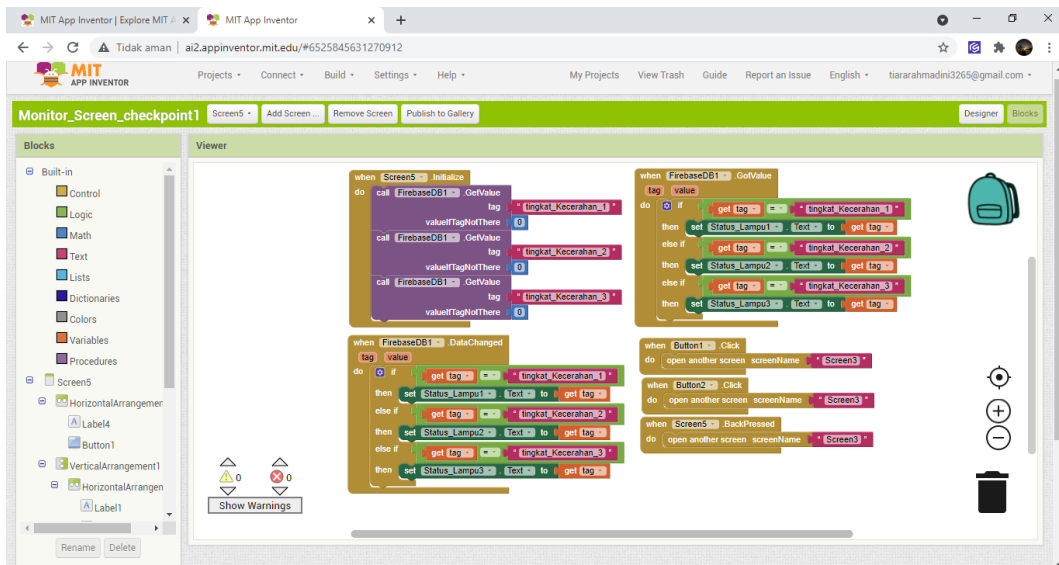
Gambar 4. 31 *Coding* blok untuk *screen* keempat

Berikut merupakan tampilan *screen* kelima pada *software* aplikasi dapat dilihat pada gambar 4.32:



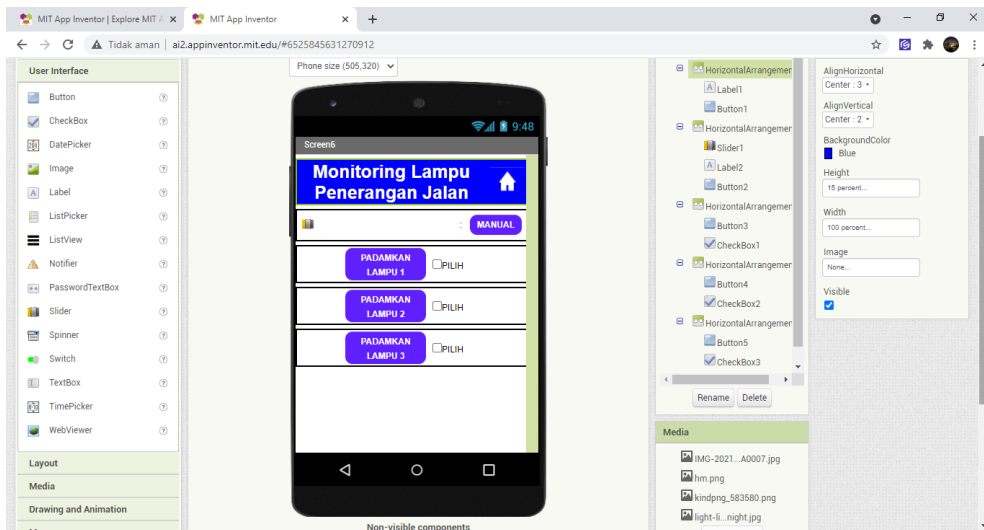
Gambar 4. 32 Tampilan *screen* kelima pada *software* aplikasi

Berikut merupakan *coding* blok untuk *screen* kelima dapat dilihat pada gambar 4.33:



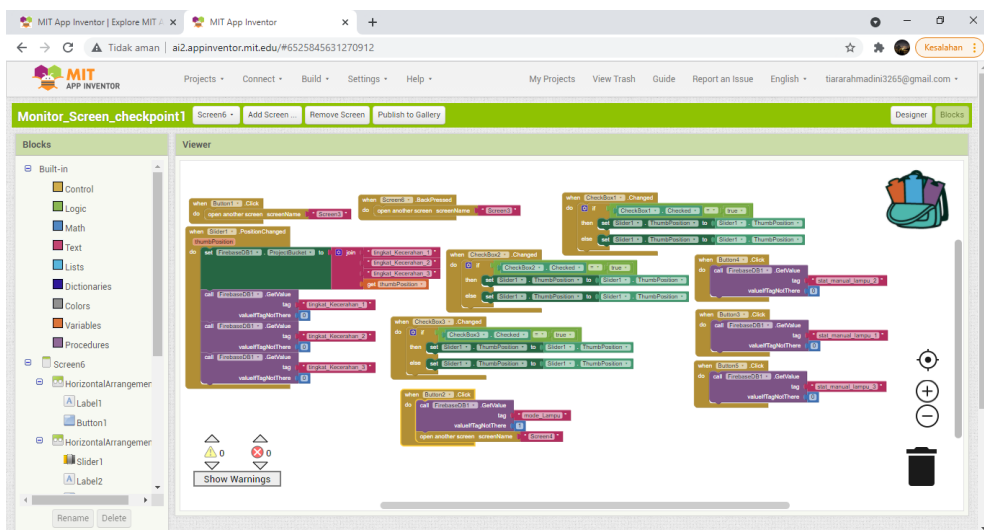
Gambar 4. 33 *Coding* blok untuk *screen* kelima

Berikut merupakan tampilan *screen* keenam pada *software* aplikasi dapat dilihat pada gambar 4.34:



Gambar 4. 34 Tampilan *screen* keenam pada *software* aplikasi

Berikut merupakan *coding* blok untuk *screen* keenam dapat dilihat pada gambar 4.35:



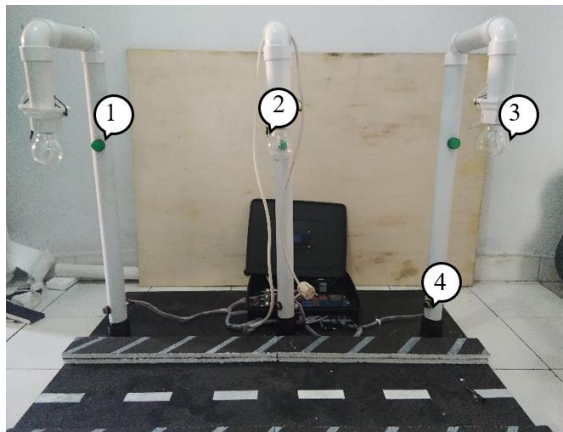
Gambar 4. 35 *Coding* blok untuk *screen* keenam

Setelah pengaturan tampilan selesai, data dikirimkan oleh NodeMCU ESP8266 lalu ditampilkan pada *software* aplikasi yang telah kita buat dan *software* aplikasi telah siap digunakan.

4.9 Perakitan *Hardware* Konstruksi dan Elektrik

Pada tahap ini dilakukan perakitan keseluruhan pada setiap bagian dari *hardware* konstruksi dan elektrik digabung menjadi satu kesatuan. Perakitan

hardware konstruksi dan elektrik kontrol *monitoring* LPJU ini dilakukan di luar kampus Politeknik Manufaktur Negeri Bangka Belitung melainkan di rumah pribadi karena pembuatan prototipe LPJU tidak perlu memakai atau meminjam peralatan yang berada di laboratorium ataupun bengkel kampus. Berikut merupakan gambar *hardware* kontrol dan elektrik kontrol *monitoring* LPJU yang telah selesai pada gambar 4.29:



Gambar 4. 36 *Hardware* kontrol dan elektrik kontrol *monitoring* LPJU

Keterangan:

1. *Flasing Lamp*
2. Sensor LDR
3. Lampu AC 25 Watt
4. Sensor *Proximity Infrared*

4.10 Pengujian Prototipe Secara Keseluruhan

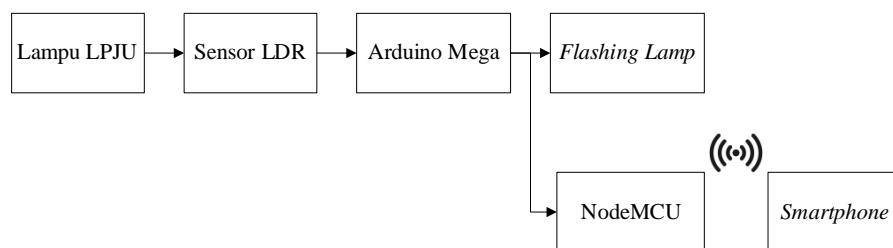
Pengujian prototipe secara keseluruhan dilakukan untuk mengetahui apakah sistem kontrol dan *monitoring* dari proyek akhir ini dapat bekerja sesuai dengan fungsinya. Dalam pengujian prototipe secara keseluruhan dilakukan tiga jenis pengujian yaitu sebagai berikut:

1. Pengujian kontrol dan *monitoring* kerusakan pada lampu LPJU.
2. Pengujian kecerahan lampu LPJU.
3. Pengujian *monitoring* kecerahan lampu LPJU pada *smartphone*.

Untuk detail pembahasan ketiga jenis pengujian tersebut dapat dilihat pada pembahasan di bawah ini:

4.10.1 Pengujian *Monitoring* Kerusakan Pada Lampu LPJU

Pengujian ini dilakukan untuk mendeteksi kerusakan pada lampu LPJU yang ditandai dengan menyalnya *flashing lamp* serta untuk melihat hasil *monitoring* kerusakan lampu LPJU melalui *smartphone*. Prosedur pengujian ini dilakukan dengan memasang dan melepas lampu LPJU, dimana jika lampu LPJU dilepas maka lampu LPJU diindikasikan dalam keadaan rusak yang akan ditandai dengan menyalnya *flashing lamp*, begitu juga sebaliknya jika lampu LPJU dipasang maka lampu LPJU diindikasikan dalam keadaan baik yang akan ditandai dengan padamnya *flashing lamp* lalu data tersebut akan dikirimkan ke aplikasi *smartphone*. Berikut merupakan blok diagram pengujian *monitoring* kerusakan lampu LPJU dapat dilihat pada gambar 4.37:

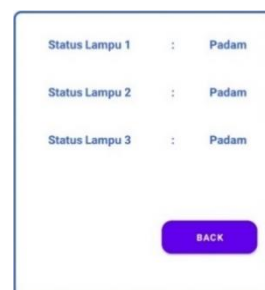


Gambar 4. 37 Blok diagram pengujian *monitoring* kerusakan pada lampu LPJU

Hasil pengujian akan dikonversikan dalam tabel untuk melihat apakah *monitoring* kerusakan pada lampu LPJU berfungsi dengan baik atau tidak dan untuk menyinkronkan data yang ditampilkan pada aplikasi *smartphone* dengan *flashing lamp*. Berikut merupakan beberapa gambar pengujian *monitoring* kerusakan lampu LPJU dapat dilihat pada gambar 4.38:



a. Kondisi *flashing lamp* jika semua lampu dilepas



b. Tampilan *smartphone* jika semua lampu dilepas



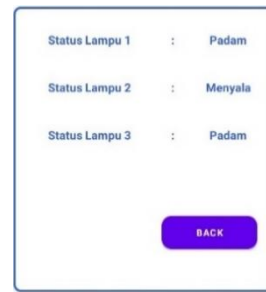
c. Kondisi *flashing lamp* jika lampu 1 dilepas, lampu 1 dan lampu 2 dipasang



d. Tampilan *smartphone* jika lampu 1 dilepas, lampu 1 dan lampu 2 dipasang



e. Kondisi *flashing lamp* jika lampu 1 dan lampu 3 dilepas, lampu 2 dipasang



f. Tampilan *smartphone* jika lampu 1 dan lampu 3 dilepas, lampu 2 dipasang

Gambar 4. 38 Pengujian *monitoring* kerusakan lampu LPJU

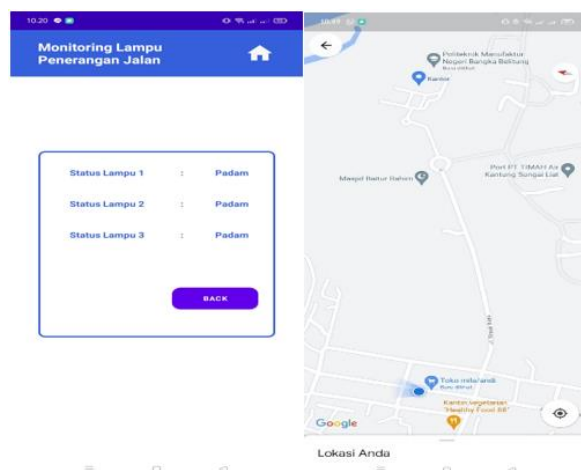
Dari gambar pengujian di atas, hasil pengujiannya dapat dikonversikan dalam bentuk tabel. Berikut merupakan hasil pengujian *monitoring* kerusakan lampu kerusakan lampu LPJU dapat dilihat pada tabel 4.6:

Tabel 4. 5 Hasil pengujian *monitoring* kerusakan lampu LPJU

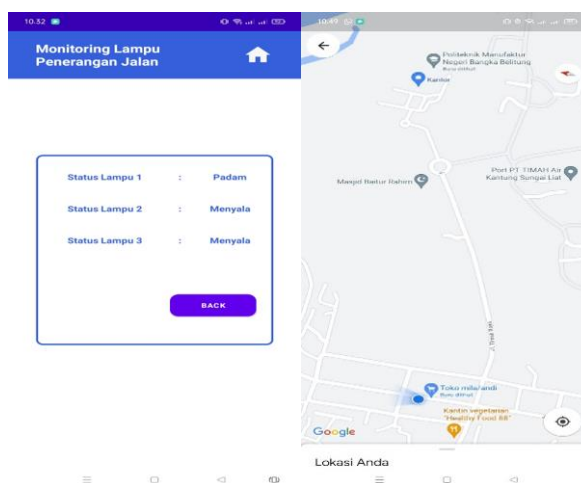
Input (Lampu LPJU)			Output					
Dipasang/Dilepas			<i>Flashing Lamp</i>			<i>Tampilan Smartphone</i>		
Lampu 1	Lampu 2	Lampu 3	<i>Flashing Lamp 1</i>	<i>Flashing Lamp 2</i>	<i>Flashing Lamp 3</i>	Lampu 1	Lampu 2	Lampu 3
Dilepas	Dilepas	Dilepas	Menyala	Menyala	Menyala	Menyala	Menyala	Menyala
Dilepas	Dipasang	Dipasang	Menyala	Padam	Padam	Menyala	Padam	Padam
Dipasang	Dilepas	Dipasang	Padam	Menyala	Padam	Padam	Menyala	Padam
Dipasang	Dipasang	Dilepas	Padam	Padam	Menyala	Padam	Padam	Menyala
Dipasang	Dilepas	Dilepas	Padam	Menyala	Menyala	Padam	Menyala	Menyala
Dilepas	Dipasang	Dilepas	Menyala	Padam	Menyala	Menyala	Padam	Menyala
Dilepas	Dilepas	Dipasang	Menyala	Menyala	Padam	Menyala	Menyala	Padam
Dipasang	Dipasang	Dipasang	Padam	Padam	Padam	Padam	Padam	Padam

Dari hasil pengujian di atas dapat diketahui bahwa ketika lampu dilepas maka lampu dinyatakan padam/rusak yang ditandai dengan *flashing lamp* menyala dan ketika lampu dipasang maka lampu dinyatakan menyala/baik yang ditandai dengan *flashing lamp* padam. Dari hasil pengujian di atas, data kerusakan lampu LPJU yang ditampilkan *smartphone* sesuai dengan *input* yang diberikan, jika lampu dilepas maka tampilan pada *smartphone* adalah lampu padam dan jika lampu dipasang maka tampilan pada *smartphone* adalah lampu menyala.

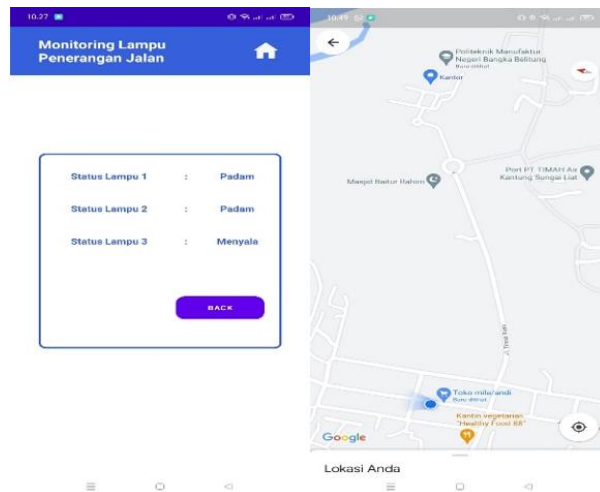
Selain itu, tidak hanya pengujian dari jarak dekat saja aplikasi ini juga diuji oleh pengguna yang berada di beda tempat dengan perangkat. Berikut data-data reponden yang diuji oleh pengguna yang berada di beda kota ditunjukkan pada gambar 4.39:



(a) Data responden ketika semua lampu padam



(b) Data responden ketika lampu 1 padam, lampu 2 dan lampu 3 menyala



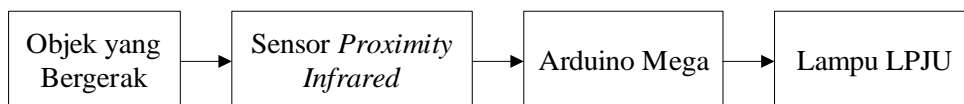
(c) Data responden ketika lampu 1 dan lampu 2 padam, lampu 3 menyala

Gambar 4. 39 Pengujian monitoring kerusakan LPJU dari jarak jauh

Dari hasil pengujian bisa disimpulkan bahwa *monitoring* kerusakan LPJU ditandai dengan *flashing lamp* yang menyala dan dapat terbaca dengan baik pada aplikasi *smartphone* dengan persentase 0%.

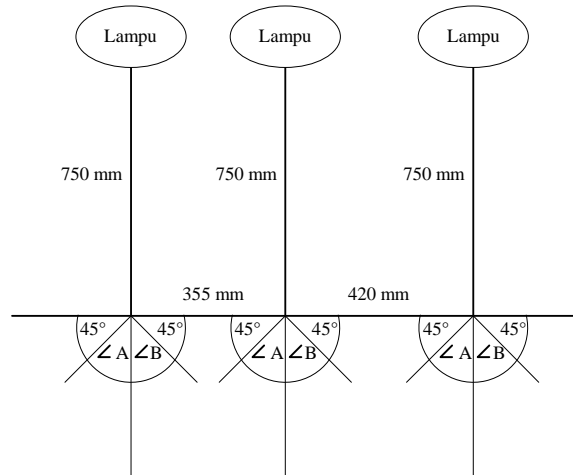
4.10.2 Pengujian Kecerahan Lampu LPJU

Pengujian ini dilakukan untuk melihat kondisi lampu LPJU ketika ada objek yang melewati lampu LPJU sesuai dengan jangkauan sudut dan jarak sensor *proximity infrared*. Prosedur pengujian ini dilakukan dengan meletakkan objek yang bergerak melewati lampu LPJU. Jika objek yang bergerak melewati lampu LPJU berada dalam jangkauan sudut dan sensor *proximity infrared* maka lampu akan menyala dengan kecerahan 100% sedangkan jika objek yang bergerak melewati lampu LPJU berada di luar jangkauan sudut dan sensor *proximity infrared* maka lampu akan menyala dengan kecerahan 50%. Pada pengujian ini, objek yang bergerak berupa mobil mainan dan miniatur robot. Berikut merupakan blok diagram pengujian kecerahan lampu LPJU dapat dilihat pada gambar 4.40:



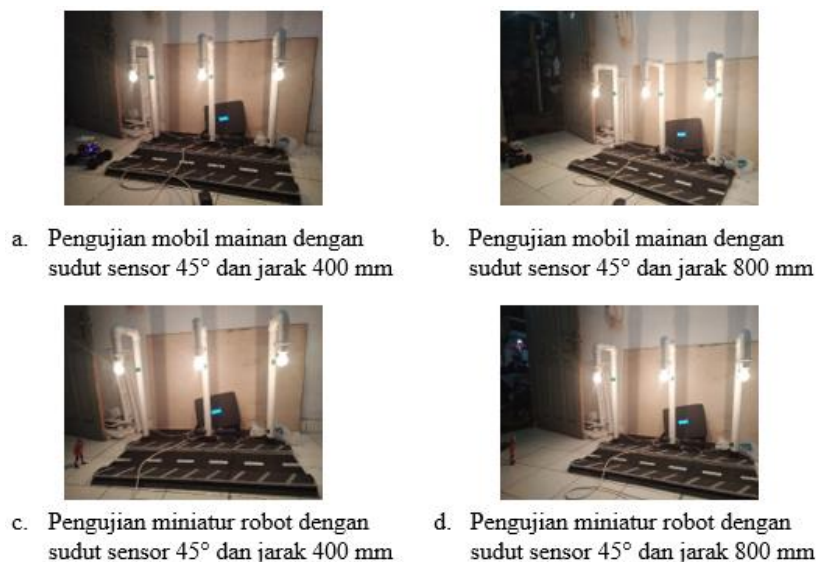
Gambar 4. 40 Blok diagram pengujian kecerahan lampu LPJU

Berikut merupakan skema pengujian kecerahan lampu LPJU dapat dilihat pada gambar 4.41:



Gambar 4. 41 Skema pengujian kecerahan lampu LPJU

Hasil pengujian akan dikonversikan dalam tabel untuk melihat apakah kecerahan lampu LPJU berfungsi dengan baik atau tidak ketika ada objek yang melewati lampu LPJU sesuai dengan jangkauan sudut dan jarak sensor *proximity infrared*. Berikut merupakan salah satu gambar pengujian objek yang bergerak sebelum melewati lampu LPJU dapat dilihat pada gambar 4.42:



Gambar 4. 42 Pengujian objek yang bergerak sebelum melewati lampu LPJU

Berikut merupakan hasil pengujian objek yang bergerak sebelum melewati lampu LPJU dapat dilihat pada tabel 4.6:

Tabel 4. 6 Hasil pengujian objek yang bergerak sebelum melewati lampu LPJU

<i>Input</i> (Objek yang Bergerak)	Sudut Sensor Proximity Terhadap Objek ($\angle A$)	Jarak Tiang dengan Objek (mm)	<i>Output</i> (Lampu LPJU)
Mobil mainan	45°	400 mm	Menyala 100%
		800 mm	Menyala 100%
		900 mm	Menyala 50%
	65°	400 mm	Menyala 100%
		800 mm	Menyala 50%
		400 mm	Menyala 100%
Miniatur robot	45°	800 mm	Menyala 100%
		900 mm	Menyala 50%
	75°	800 mm	Menyala 50%
		400 mm	Menyala 100%

Berikut merupakan gambar pengujian objek yang bergerak setelah melewati lampu LPJU dapat dilihat pada gambar 4.43:



a. Pengujian mobil mainan dengan sudut sensor 45° dan jarak 400 mm



b. Pengujian miniatur robot dengan sudut sensor 65° dan jarak 250 mm

Gambar 4. 43 Pengujian objek yang bergerak setelah melewati lampu LPJU

Berikut merupakan hasil pengujian objek yang bergerak setelah melewati lampu LPJU dapat dilihat pada tabel 4.7:

Tabel 4. 7 Hasil pengujian objek yang bergerak setelah melewati lampu LPJU

<i>Input</i> (Objek yang Bergerak)	Sudut Sensor Proximity Terhadap Objek ($\angle B$)	Jarak Tiang dengan Objek (mm)	<i>Output</i> (Lampu LPJU)
Mobil mainan	45°	400 mm	Menyala 100%

<i>Input</i> (Objek yang Bergerak)	Sudut Sensor Proximity Terhadap Objek ($\angle B$)	Jarak Tiang dengan Objek (mm)	<i>Output</i> (Lampu LPJU)
Miniatur robot	65°	800 mm	Menyala 50%
		400 mm	Menyala 100%
		800 mm	Menyala 50%
	75°	400 mm	Menyala 100%
		800 mm	Menyala 50%
		250 mm	Menyala 100%
	65°	350 mm	Menyala 50%
		250 mm	Menyala 100%
		350 mm	Menyala 50%

Berikut merupakan gambar pengujian objek yang bergerak tepat melewati lampu LPJU dapat dilihat pada gambar 4.44:



a. Pengujian mobil mainan dengan jarak lurus 400 mm



b. Pengujian miniatur robot dengan jarak lurus 400 mm

Gambar 4. 44 Pengujian objek yang bergerak tepat melewati lampu LPJU

Berikut merupakan hasil pengujian jarak objek yang bergerak tepat melewati lampu LPJU dapat dilihat pada tabel 4.8:

Tabel 4. 8 Hasil pengujian jarak objek yang bergerak tepat melewati lampu LPJU

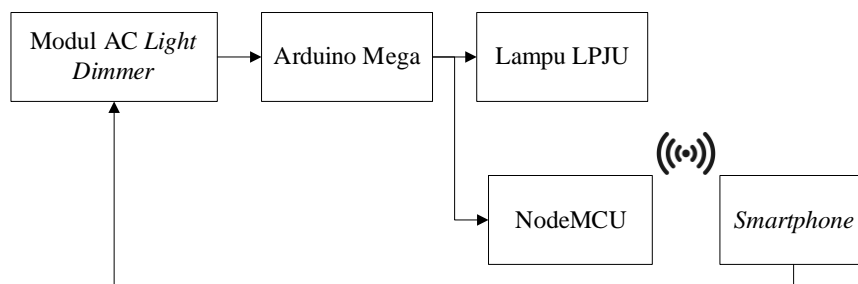
<i>Input</i> (Objek yang Bergerak)	Jarak Lurus Tiang Dengan Objek (mm)	<i>Output</i> (Lampu LPJU)
Mobil Mainan	400 mm	Menyala 100%
	550 mm	Menyala 100%
	800 mm	Menyala 50%
Miniatur Robot	400 mm	Menyala 50%
	800 mm	Menyala 50%

Dari hasil pengujian di atas dapat diketahui bahwa jika ada objek yang melewati lampu LPJU dengan sudut 45° sejauh 80 cm maka lampu akan menyala 100% dan jika ada objek yang melewati lampu LPJU dengan sudut 45° sejauh 90 cm maka lampu akan menyala 50%.

Dari hasil pengujian kecerahan lampu LPJU bisa disimpulkan bahwa sistem mampu mendeteksi objek yang bergerak melewati LPJU dengan penyalaaan lampu 100% sejauh 80 cm dengan jarak sudut yang terdeteksi sebesar 45° .

4.10.3 Pengujian Kontrol dan *Monitoring* Kecerahan Lampu LPJU Pada *Smartphone*

Pengujian ini dilakukan untuk melihat kondisi lampu LPJU ketika kecerahan lampu diatur oleh *smartphone*. Prosedur pengujian ini adalah mengeser *slider* kecerahan lampu LPJU pada *smartphone* kemudian hasilnya akan dilihat pada lampu LPJU, apakah kecerahan lampu LPJU dapat diatur melalui *smartphone* atau tidak.. Berikut merupakan blok diagram pengujian kontrol dan *monitoring* kecerahan lampu LPJU pada *smartphone* dapat dilihat pada gambar 4.45:



Gambar 4. 45 Blok diagram pengujian kontrol dan *monitoring* kecerahan lampu LPJU pada *smartphone*

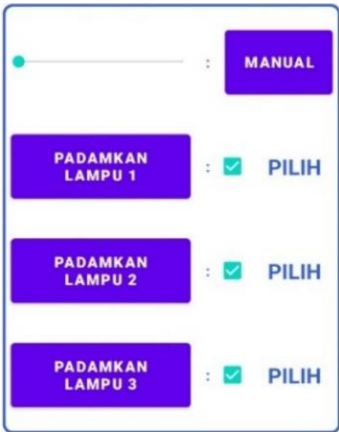



Pengujian kontrol dan *monitoring* kecerahan lampu LPJU pada *smartphone* ini dilakukan pada siang hari di ruangan tertutup yang berukuran 2,4 m x 3,3 m. Pengujian ini dilakukan menggunakan alat intensitas cahaya dengan satuan lux. Besaran intensitas cahaya yang digunakan pada proyek akhir ini ada dua yaitu lux dan lumen. Lux merupakan satuan ketika mengukur intensitas cahaya yang diterima akibat adanya sumber cahaya sedangkan lumen merupakan satuan ketika mengukur

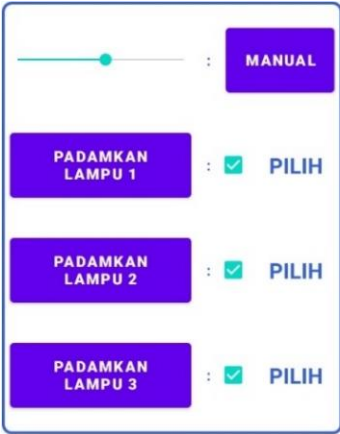

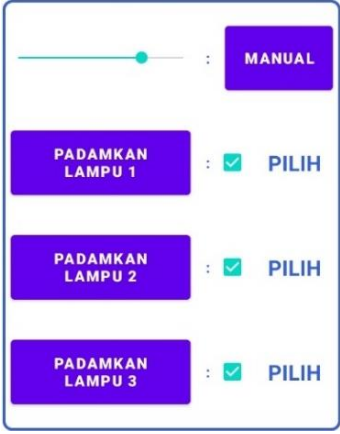

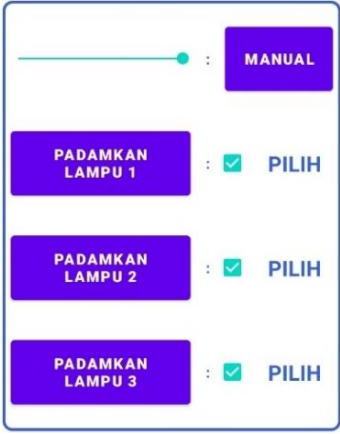

intensitas cahaya yang dihasilkan oleh sumber cahaya. Satu lux sama dengan satu lumen per meter persegi sehingga jika dirumuskan menjadi:

$$1 \text{ lux} = 1 \text{ lumen} / \text{m}^2$$

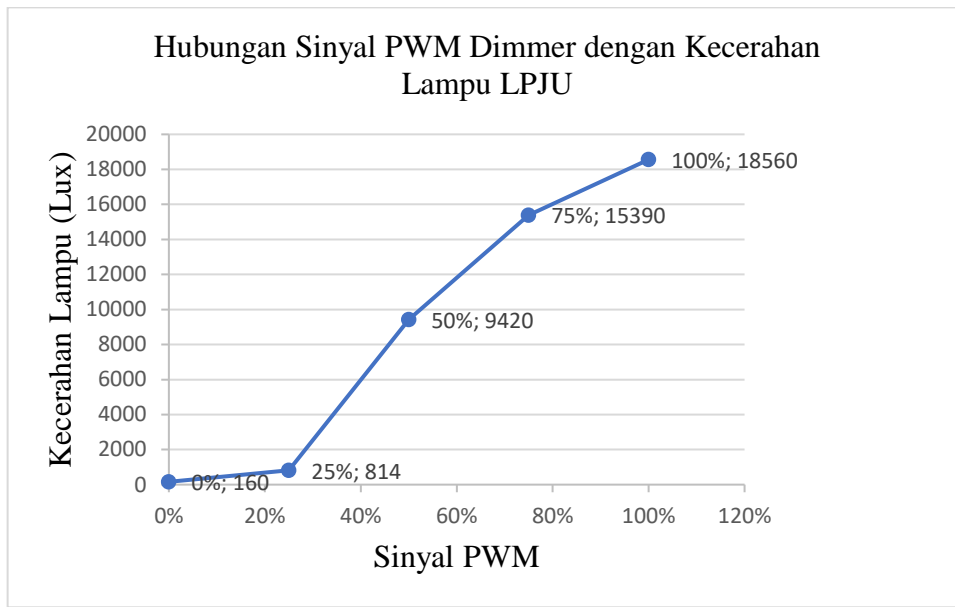
Hasil pengujian pada akan dikonversikan dalam tabel untuk melihat kondisi lampu LPJU berfungsi dengan baik atau tidak ketika kecerahan lampu diatur oleh *smartphone*. Berikut merupakan hasil pengujian kontrol dan *monitoring* kecerahan lampu LPJU pada *smartphone* dapat dilihat pada tabel 4.9:

Tabel 4. 9 Hasil pengujian kontrol dan *monitoring* kecerahan lampu LPJU pada *smartphone*

<i>Input (Smartphone)</i>	<i>Output (Lampu LPJU)</i>		
	Keadaan lampu LPJU	Alat ukur	Perhitungan
<p>Slider kecerahan 0%</p> 		160 lux	1.280 lumen
<p>Slider kecerahan 25%</p> 		814 lux	6.512 lumen

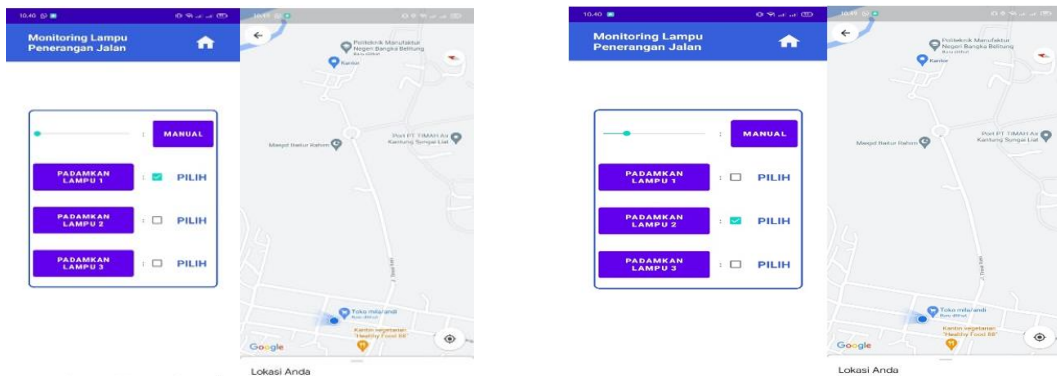
		<i>Output (Lampu LPJU)</i>	
<i>Input (Smartphone)</i>	Keadaan lampu LPJU	Alat ukur	Perhitungan
Slider kecerahan 50%			
		9.420 lux	75.360 lumen
Slider kecerahan 75%			
		15.390 lux	123.120 lumen
Slider kecerahan 100%			
		18.560 lux	148.480 lumen

Dari hasil pengujian di atas dapat dalam bentuk grafik untuk melihat hubungan sinyal PWM dengan kecerahan lampu LPJU. Berikut merupakan grafik hubungan sinyal PWM dengan kecerahan lampu LPJU dapat dilihat pada gambar 4.46:



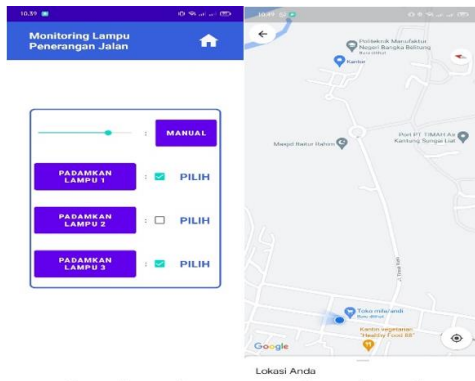
Gambar 4. 46 Hubungan sinyal PWM dengan kecerahan lampu LPJU

Dari hasil pengujian di atas dapat diketahui jika *slider* kecerahan lampu LPJU digeser maka lampu LPJU akan menyala sesuai dengan kecerahan yang telah ditentukan oleh *slider* tersebut. Selain itu, tidak hanya pengujian dari jarak dekat saja aplikasi ini juga diuji oleh pengguna yang berada di beda tempat dengan perangkat. Berikut data-data reponden yang diuji oleh pengguna yang berada di beda kota ditunjukkan pada gambar 4.47:

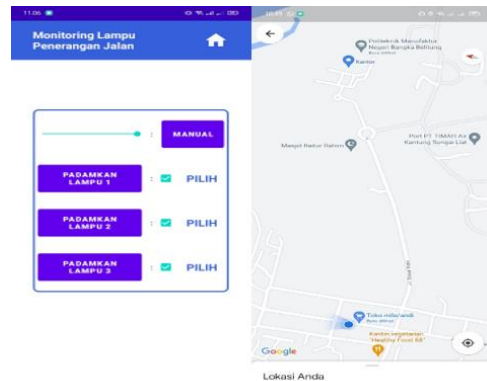


(a) Data responden kecerahan lampu 0%

(b) Data responden kecerahan lampu 25%



(c) Data responden kecerahan lampu 75%



(d) Data responden kecerahan lampu 100%

Gambar 4. 47 Pengujian kontrol dan *monitoring* kecerahan lampu LPJU dari jarak jauh

Dari hasil pengujian di atas dapat disimpulkan bahwa pengontrolan kecerahan lampu dapat diatur melalui aplikasi *smartphone*.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil dari pengujian dan analisa serta fungsi prototipe yang berjudul “Sistem Cerdas Lampu Penerangan Jalan Berbasis *IoT*” dapat disimpulkan bahwa:

- Kerusakan lampu ditandai dengan *flashing lamp* yang menyala dan dapat terbaca pada aplikasi *smartphone* dan LCD 20x4 dengan persentase error 0%.
- Sistem mampu mendeteksi objek yang bergerak melewati LPJU dengan penyalan lampu 100 % sejauh 80 cm dengan jarak sudut yang terdeteksi sebesar 45°.
- Pengontrolan kecerahan lampu dapat diatur melalui aplikasi *smartphone*

5.2 Saran

Dari proyek akhir yang telah dikerjakan ini ada beberapa saran yang disampaikan untuk mengembangkan proyek akhir ini ke depannya, yaitu:

- *Monitoring* dapat dilakukan pada pemakaian daya lampu penerangan jalan.
- Untuk menghemat energi listrik, dianjurkan untuk menggunakan *solar cell*.

DAFTAR PUSTAKA

- [1] T. U. Syamsuri, P. N. Malang, A. Pendahuluan, and B. Teori, “B-28 B-29,” vol. 7, pp. 28–33, 2015.
- [2] R. Imam, F. Bimantoro, P. Studi, T. Informatika, F. Teknik, and U. Mataram, “RANCANG BANGUN SISTEM MONITORING DAN CONTROLLING PENERANGAN JALAN UMUM BERBASIS IOT DAN ANDROID (Design of Monitoring and Controlling for Public Street Lighting Based in IoT and Android),” vol. 2, no. 1, pp. 101–112, 2020.
- [3] I. M. A. Suyadnya, “Rancang Bangun Alat Monitoring Kerusakan Lampu Penerangan Jalan Umum Berbasis Mikrokontroler dengan Notifikasi SMS,” vol. 2, no. 2, pp. 90–99, 2018.
- [4] S. Nasional, I. Ics, and B. S. Nasional, “Spesifikasi penerangan jalan di kawasan perkotaan,” 2008.
- [5] P. Issn, “INTERNET OF THINGS (IOT) SISTEM PENGENDALIAN LAMPU,” vol. 4, no. 1, pp. 19–26, 2018.
- [6] B. Wibowo, H. S. Utama, and N. Kusumaningrum, “Perancangan dan Realisasi Sistem Kendali Lampu, Air Conditioner Berbasis Android,” vol. 21, no. 1, pp. 36–43, 2019.

LAMPIRAN

Lampiran 1 Daftar Riwayat Hidup

DAFTAR RIWAYAT HIDUP

1. Data Pribadi

Nama Lengkap : Teguh Gesit Aqilla
Tempat, Tanggal Lahir : Sungailiat, 29 November 1999
Alamat Rumah : Jl. Komplek Plaben, Baturusa
No. HP : 085778907687
Email : teguhga7@gmail.com
Jenis Kelamin : Laki-Laki
Agama : Islam



2. Riwayat Pendidikan

SD Negeri 2 Merawang	Lulus 2012
SMP Negeri 1 Merawang	Lulus 2015
SMK Negeri 2 Pangkal Pinang	Lulus 2018
DIII Politeknik Manufaktur Negeri Bangka Belitung	Lulus 2021

3. Pendidikan Non Formal

Sungailiat, 25 Agustus 2021

Teguh Gesit Aqilla

DAFTAR RIWAYAT HIDUP

1. Data Pribadi

Nama Lengkap : Tiara Rahmadani
Tempat, Tanggal Lahir : Sungailiat, 15 Desember 2000
Alamat Rumah : Jalan Jendral Sudirman, Belinyu
No. HP : 082281965920
Email : tiararahmadani3265@gmail.com
Jenis Kelamin : Perempuan
Agama : Islam



2. Riwayat Pendidikan

SD Negeri 5 Belinyu	Lulus 2012
SMP Negeri 1 Belinyu	Lulus 2015
SMA Negeri 1 Belinyu	Lulus 2018
DIII Politeknik Manufaktur Negeri Bangka Belitung	Lulus 2021

3. Pendidikan Non Formal

Sungailiat, 25 Agustus 2021



Tiara Rahmadani

Lampiran 2 Program Arduino Mega

Kode program keseluruhan Arduino Mega:

```
#include <RBDdimmer.h>
#include <LiquidCrystal_I2C.h>
#include <BH1750.h>
#include "code_main.h"
#include "code_lcd.h"
#include "code_relay.h"
#include "code_proximity.h"
#include "code_light.h"
#include "code_acs.h"
#include "code_nodemcu.h"
void setup() {
    inisialDimmer_lampu();
    Serial.println("PadamKan Lampu");
    dimmer1.setPower( nilai_persen_lampu_padam ); //==> Set Dimmer
10 atau padam .
    dimmer2.setPower( nilai_persen_lampu_padam ); //==> Set Dimmer
10 atau padam .
    dimmer3.setPower( nilai_persen_lampu_padam ); //==> Set Dimmer
10 atau padam .
    //=====> Inisial Serial USart .
    Serial.begin(9600);
    Serial3.begin(9600);
    //=====> Inisial LCD .
    Wire.begin();
    lcd.begin();
    lcd.print("System Ready");
    //=====> Inisial awal sensor cahaya .
    Serial.println("Set Sensor Cahay");
    inisialSensor();
    //=====> Inisial Dimmer Lamp .
    Serial.println("Set Dimmer Lamp");
    //=====> Inisial Sensor LDR .
    //=====> Fungsi ini untuk kalibrasi sensor LDR .
    Serial.println("Set Sensor LDR");
    median_point_lampu_1 = readLDR712( sensor_LDR_lampu_1 );
    median_point_lampu_2 = readLDR712( sensor_LDR_lampu_2 );
    median_point_lampu_3 = readLDR712( sensor_LDR_lampu_3 );
    Serial.println("Med One : " + String(median_point_lampu_1));
    Serial.println("Med Two : " + String(median_point_lampu_2));
    Serial.println("Med Three : " + String(median_point_lampu_3));
    //=====> Inisial Sensor Proximity ,
    Serial.println("Set Sensor Proximity");
    inisialProximity();
    //=====> Panggil Fungsi untuk inisial relay .
    Serial.println("Set Relay");
    inisialRelay();
    //=====> setting awal lampu saat program baru dinyalakan .
    Serial.println("Set Awal Lampu Flash");
    settingStatusRelay( lamp_flasing_1 , padam );
    settingStatusRelay( lamp_flasing_2 , padam );
    settingStatusRelay( lamp_flasing_3 , padam );
    //====> Kode Test Program Komponen .
```

```

testSerialConnection( tidak );
test_proximity( tidak );
test_komponen_relay( tidak );
test_komponen_sensor_cahaya( tidak );
test_komponen_sensor_LDR( tidak );
testKomponen_ACLightDimmer( tidak );
Serial.println("Selsesai");
}
void kode_proses_lampu_pilot_jika_terjadi_kerusakan(){
//====> Baca Informasi Sensor LDR untuk analisa jika lampu
gagal menyala karena -
//====> mengalami masalah .
//=> cek ambang batas lampu 1 .
//=> lampu masih menyala dengan di tandai ada cahaya keluar .
//=> atau ada listrik mengalir melewati lampu dan sensor
cahaya .
if( cahaya_lampu_1 > nilai_ambang_tidak_ada_cahaya ){
    settingStatusRelay( lamp_flasing_1 , padam );
}
//=> lampu masih padam dengan di tandai tidak ada cahaya
keluar .
else{
    settingStatusRelay( lamp_flasing_1 , menyala );
}
//=> cek ambang batas lampu 2 .
//=> lampu masih menyala dengan di tandai ada cahaya keluar .
//=> atau ada listrik mengalir melewati lampu dan sensor
cahaya .
if( cahaya_lampu_2 > nilai_ambang_tidak_ada_cahaya ){
    settingStatusRelay( lamp_flasing_2 , padam );
}
//=> lampu masih padam dengan di tandai tidak ada cahaya
keluar .
else{
    settingStatusRelay( lamp_flasing_2 , menyala );
}
//=> cek ambang batas lampu 3 .
//=> lampu masih menyala dengan di tandai ada cahaya keluar .
//=> atau ada listrik mengalir melewati lampu dan sensor
cahaya .
if( cahaya_lampu_3 > nilai_ambang_tidak_ada_cahaya ){
    settingStatusRelay( lamp_flasing_3 , padam );
}
//=> lampu masih padam dengan di tandai tidak ada cahaya
keluar .
else{
    settingStatusRelay( lamp_flasing_3 , menyala );
}
}
void loop(){
//====> Panggil kode program untuk membaca pesan dari nodemcu .
//====> Proses untuk membaca informasi yang dikirimkan dari
nodemcu .
String temps = readMessageEsp();
//====> Jika ada pesan masuk , langsung saja parsing data atau
pemecahan -

```

```

//==> data menjadi satuan fungsi .
if( temps.length() > 0 ){
    //====> Tampilkan informasi pada serial monitor .
    Serial.println("Data Diterima : " + temps );
    //====> Panggil fungsi untuk parsing atau pemecahan data .
    parseData( temps );
}
//==> Baca nilai cahaya listrik dari sensor LDR lampu 1 .
cahaya_lampu_1 = readLDR712( sensor_LDR_lampu_1 ) -
median_point_lampu_1 ;
cahaya_lampu_1 = convert2Curent( cahaya_lampu_1 );
//==> Baca nilai cahaya listrik dari sensor LDR lampu 2 .
cahaya_lampu_2 = readLDR712( sensor_LDR_lampu_2 ) -
median_point_lampu_2 ;
cahaya_lampu_2 = convert2Curent( cahaya_lampu_2 );
//==> Baca nilai cahaya listrik dari sensor LDR lampu 3 .
cahaya_lampu_3 = readLDR712( sensor_LDR_lampu_3 ) -
median_point_lampu_3 ;
cahaya_lampu_3 = convert2Curent( cahaya_lampu_3 );
//====> Panggil fungsi untuk membaca nilai sensor cahaya dan cek
jika sudah gelap .
//====> cek jika sinar matahari sudah tidak ada atau gelap .
//====> Sebelum masuk ke proses , cek jika . Jika mode lampu
dalam posisi otomatis.
//====> karena proses if di bawah di peruntukan untuk mode
otomatis . Tergantung dari-
//====> cahaya lampu atau tingkat kegelapan pada lingkungan
sekitar .
if(
    cek_status_sensor_cahaya() == sudah_gelap && //==> Cek Jika
sudah gelap .
    mode_Lampu.toInt() == lampu_dalam_mode_otomatis //==> Cek Jika
lampu dalam mode otomatis .
){
    //====> Panggil fungsi untuk memproses lampu pilot jika
terjadi kerusakan .
    //====> Jadi kode proses ini melakukan cek , jika tidak ada
cahaya listrik -
    //====> saat lampu dinyalakan malam hari .
    kode_proses_lampu_pilot_jika_terjadi_kerusakan();
    //====> Reset parameter terlebih dahulu .
    detect_Objeck = tidak ;
    str_prox = "" ;
    //==> Cek sensor Proximity .
    //==> Jika Terdeteksi Objeck Pada Proximity 1 .
    if(
        baca_status_proximity( proximity_lampu_1 ) == ada_object
    ){
        dimmer1.setPower( set_kecerahan_lampu_1.toInt() ); //====>
Nyalakan full .
        detect_Objeck = ya ;
        //====> Set Proximity 1 mendeteksi .
        str_prox += "1" ;
    }
    //==> Cek sensor Proximity .
    //==> Jika tidak Terdeteksi Objeck .

```

```

        else if(
            baca_status_proximity( proximity_lampu_1 ) ==
tidak_ada_objeck
        ){
            dimmer1.setPower( minimal_persen_lampu_dinyalakan );
//====> Nyalakan setengah ,
        }
        //====> Jika Terdeteksi Object Pada Proximity 2 .
        if(
            baca_status_proximity( proximity_lampu_1 ) == ada_object
||
            baca_status_proximity( proximity_lampu_2 ) == ada_object
||
            baca_status_proximity( proximity_lampu_3 ) == ada_object
        ){
            dimmer1.setPower( set_kecerahan_lampu_1.toInt() ); //====>
Nyalakan full .
            dimmer2.setPower( set_kecerahan_lampu_2.toInt() ); //====>
Nyalakan full .
            dimmer3.setPower( set_kecerahan_lampu_3.toInt() ); //====>
Nyalakan full .
            detect_Object = ya ;
            //====> Set Proximity 2 mendeteksi .
            str_prox += "2" ;
        }
        //====> Cek sensor Proximity .
        //====> Jika tidak Terdeteksi Object .
        else if(
            baca_status_proximity( proximity_lampu_1 ) ==
tidak_ada_objeck ||
            baca_status_proximity( proximity_lampu_2 ) ==
tidak_ada_objeck ||
            baca_status_proximity( proximity_lampu_3 ) ==
tidak_ada_objeck
        ){
            dimmer1.setPower( minimal_persen_lampu_dinyalakan );
//====> Nyalakan setengah ,
            dimmer2.setPower( minimal_persen_lampu_dinyalakan );
//====> Nyalakan setengah ,
            dimmer3.setPower( minimal_persen_lampu_dinyalakan );
//====> Nyalakan setengah ,
        }
    }
    //====> Panggil fungsi untuk membaca nilai sensor cahaya dan cek
jika masih terang .
    //====> cek jika sinar matahari masih bersinar.
    //====> Sebelum masuk ke proses , cek jika . Jika mode lampu
dalam posisi otomatis.
    //====> karena proses else if di bawah di peruntukan untuk mode
otomatis . Tergantung dari-
    //====> cahaya lampu atau tingkat kegelapan pada lingkungan
sekitar .
    else if(
        cek_status_sensor_cahaya() == masih_terang && //==> Cek Jika
sudah gelap .

```

```

        mode_Lampu.toInt() == lampu_dalam_mode_otomatis //==> Cek
Jika lampu dalam mode otomatis .
    ){
        dimmer1.setPower( nilai_persen_lampu_padam ); //==> Set Dimmer
10 atau padam .
        dimmer2.setPower( nilai_persen_lampu_padam ); //==> Set Dimmer
10 atau padam .
        dimmer3.setPower( nilai_persen_lampu_padam ); //==> Set Dimmer
10 atau padam .
        settingStatusRelay( lamp_flasing_1 , padam );
        settingStatusRelay( lamp_flasing_2 , padam );
        settingStatusRelay( lamp_flasing_3 , padam );
    }
//====> Kode program di bawah ini adalah proses dimana perangkat
akan bekerja secara -
//====> manual atau di kendalikan lewat hp android .
    else if(
        mode_Lampu.toInt() == lampu_dalam_mode_manual
    ){
        //====> cek jika lampu 1 di setting menyala .
        if( stat_manual_lampu_1.toInt() ==
lampu_di_set_nyala_mode_manual ){
            //====> Pada saat di setting menyala , maka setting tingkat
kecerahan lampu sesuai .
            dimmer1.setPower( set_kecerahan_lampu_1.toInt() );
        }
        //====> cek jika lampu 1 di setting padam .
        else if( stat_manual_lampu_1.toInt() ==
lampu_di_set_padam_mode_manual ){
            //====> Pada saat di setting padam , maka setting tingkat
kecerahan lampu sesuai .
            dimmer1.setPower( nilai_persen_lampu_padam );
        }
        //====> cek jika lampu 2 di setting menyala .
        if( stat_manual_lampu_2.toInt() ==
lampu_di_set_nyala_mode_manual ){
            //====> Pada saat di setting menyala , maka setting tingkat
kecerahan lampu sesuai .
            dimmer2.setPower( set_kecerahan_lampu_2.toInt() );
        }
        //====> cek jika lampu 2di setting padam .
        else if( stat_manual_lampu_2.toInt() ==
lampu_di_set_padam_mode_manual ){
            //====> Pada saat di setting padam , maka setting tingkat
kecerahan lampu sesuai .
            dimmer2.setPower( nilai_persen_lampu_padam );
        }
        //====> cek jika lampu 3 di setting menyala .
        if( stat_manual_lampu_3.toInt() ==
lampu_di_set_nyala_mode_manual ){
            //====> Pada saat di setting menyala , maka setting tingkat
kecerahan lampu sesuai .
            dimmer3.setPower( set_kecerahan_lampu_3.toInt() );
        }
        //====> cek jika lampu 3 di setting padam .

```

```
    else if( stat_manual_lampu_3.toInt() ==
lampu_di_set_padam_mode_manual ){
        //===> Pada saat di setting padam , maka setting tingkat
kecerahan lampu sesuai .
        dimmer3.setPower( nilai_persen_lampu_padam );
    }
}
//====> Panggil program untuk melakukan update lcd .
//====> untuk tampilan utama .
display_TampilanUtama();
//====> Update Informasi ke nodemcu .
updateInformasiServer();
}
```

Kode akses untuk program utama:

```
//====> library lcd .
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27, 20 , 4);
#include <BH1750.h>
#include <Wire.h>
BH1750 lightMeter;
bool cek_status_sensor_cahaya();
//====> Parameter boolean untuk nyala dan padam .
bool menyala = true ;
bool padam = false ;
//====> parameter boolean ya dan tidak .
bool ya = true ;
bool tidak = false ;
//====> Status sensor cahaya .
bool masih_terang = true ;
bool sudah_gelap = false ;
//====> Status lampu dinyalakan saat mode manual .
bool lampu_di_set_nyala_mode_manual = 1 ;
bool lampu_di_set_padam_mode_manual = 0 ;
//====> Status sensor proximity .
bool ada_object = true ;
bool tidak_ada_objeck = false ;
//====> Status mode lampu ,
int lampu_dalam_mode_manual = 0 ;
int lampu_dalam_mode_otomatis = 1 ;
//====> Kode Parameter Setting Port Komponen <====//
//====> Pin Sensor LDR .
//====> Pin untuk Sensor LDR hcahaya pada pin analog Karena
informasi output -
//====> adsalah analog , Tidak bisa diletakan pada pin digital .
const byte sensor_LDR_lampu_1 = A8 ;
const byte sensor_LDR_lampu_2 = A9 ;
const byte sensor_LDR_lampu_3 = A10 ;
//====> Pin Sensor Proximity .
//====> Pin Sensor Proximity dapat menggunakan pin analog ataupun
digital -
//====> akan tetapi lebih mudah menggunakan pin analog karena
analisa sinyal out -
//====> dapat menggunakan grafik pada serial plotter.
const byte proximity_lampu_1 = A0 ;
const byte proximity_lampu_2 = A1 ;
const byte proximity_lampu_3 = A2 ;
//====> Pin Dimmer Lamp .
int pin_pwm_1 = 8 ;
int pin_pwm_2 = 9 ;
int pin_pwm_3 = 10 ;
int pin_zero = 5 ;
//====> Pin Relay .
//====> Pin relay hcahaya diletakan pada pin digital . Pin Input
relay menggunakan-
//====> sinyal HIGH dan LOW untuk memadamkan dan menyalakan relay
. Setiap module-
```

```

//====> relay yang di jual di pasaran memiliki jenis input yang
berbeda - beda .
//====> Adaya yang aktif HIGH dan yang aktif LOW . jadi perlu di
ingat pada saat-
//====> pembelian .
//====> Setting jika aktif HIGH maka beri keterangan true .
//====> Jika aktif LOW maka beri keterangan false ;
bool isAktifHigh = true ;
//====> Setting sesuai dengan yang di gunaka .
const byte lampu_jalan = 4 ;
//====> setting sesuai dengan pin yang di gunakan .
const byte lamp_flasing_1 = 7 ;
const byte lamp_flasing_2 = 6 ;
const byte lamp_flasing_3 = 5 ;
//====> Setting batas ambang nilai lux cahaya saat gelap atau
malam hari .
float nilai_Ambang_Batas_gelap = 60 ;
//====> Setting ambang batas nilai sensor cahaya , saat
mendeteksi lampu rusak .
float nilai_ambang_tidak_ada_cahaya = 1.3 ;
//=====> Status parameter , menyimpan informasi dari nodemcu .
//=====> informasi mengenai status wifi dalam kondisi terkoneksi
-
//=====> atau tidak .
bool wifiConnected = false ;
//=====> status mode arduino otomatis atau manual .
String mode_Lampu = "1" ;
//=====> Setting tingkat kecerahan lampu .
String set_kecerahan_lampu_1 = "95" ;
String set_kecerahan_lampu_2 = "95" ;
String set_kecerahan_lampu_3 = "95" ;
//=====> Setting On/Off lampu Manual.
String stat_manual_lampu_1 = "0" ;
String stat_manual_lampu_2 = "0" ;
String stat_manual_lampu_3 = "0" ;
String readMessageCom(){
    String msg = "" ;
    unsigned long tm = millis() ;
    while( (millis()-tm) < 150 ){
        if( Serial.available() ){
            char cht = Serial.read();
            msg += cht ;
            tm = millis();
        }
    }
    return msg ;
}
//====> parameter yang menyimpan informasi ketika ada object
terdeteksi oleh proximity .
//====> variabel ini di gunaksn untuk pendukung informasi pada
display lcd .
bool detect_Object = false ;
//====> Variabel yang menampung prproximity mana yang mendeteksi
object .
String str_prox = "" ;
//====> Parameter minimum % lampu dinyalakan .

```

```
int minimal_persen_lampu_dinyalakan = 50 ;
int nilai_persen_lampu_padam = 10 ;
///<====> parameter nilai yang menampung nilai hasil pembacaan
sensor cahaya .
///<====> variabel yang menampung hasil pembacaan sensor cahaya .
float cahaya_lampu_1 = 0.0 ;
float cahaya_lampu_2 = 0.0 ;
float cahaya_lampu_3 = 0.0 ;
///<====> median nilai yang menampung hasil pembacaan nilai media
///<====> atau nilai awal sensor LDR tidak mendapatkan cahaya yang
mengalir .
float median_point_lampu_1 = 0 ;
float median_point_lampu_2 = 0 ;
float median_point_lampu_3 = 0 ;
```

Kode akses untuk sensor LDR:

```
// Arduino UNO has 5.0 volt with a max ADC value of 1023 steps
// LDR712 5A uses 185 mV per A
// LDR712 20A uses 100 mV per A
// LDR712 30A uses 66 mV per A
float readLDR712( byte pin_sensor_LDR ){
  //=====> lakukan pembacaan dan gunakan RMS untuk mendapatkan
  tegangan stabil .
  //=====> lakukan pembacaan selama 1 second .
  unsigned long tmWait = millis();
  float temporaryBiner = 0 ;
  float counterData = 0 ;
  while( (millis()-tmWait) < 200 ){
    temporaryBiner += float(analogRead( pin_sensor_LDR ));
    counterData = counterData + 1 ;
    delay(2);
  }
  float rms = temporaryBiner / counterData ;
  return float(map( rms , 0 , 1024 , 0 , 5000 ));
}
float convert2Curent( float voltageSensor ){
  return abs(voltageSensor / 1000.0) ;
}
void test_komponen_sensor_LDR( bool izinkan ){
  if( izinkan == ya ){
    dimmer1.setPower( minimal_persen_lampu_dinyalakan ); //==> Set
    Dimmer 10 atau padam .
    dimmer2.setPower( minimal_persen_lampu_dinyalakan ); //==> Set
    Dimmer 10 atau padam .
    dimmer3.setPower( minimal_persen_lampu_dinyalakan ); //==> Set
    Dimmer 10 atau padam .
    Serial.println("Menyalakan Lampu selesai");
  }
  //====> cek jika kode test di izinkan .
  while( izinkan == ya ){
    //====> Tampilkan display sensor cahaya .
    display_test_sensor_cahaya();
    //==> Baca nilai cahaya listrik dari sensor LDR lampu 1 .
    cahaya_lampu_1 = readLDR712( sensor_LDR_lampu_1 ) -
    median_point_lampu_1 ;
    cahaya_lampu_1 = convert2Curent( cahaya_lampu_1 );
    //==> Baca nilai cahaya listrik dari sensor LDR lampu 2 .
    cahaya_lampu_2 = readLDR712( sensor_LDR_lampu_2 ) -
    median_point_lampu_2 ;
    cahaya_lampu_2 = convert2Curent( cahaya_lampu_2 );
    //==> Baca nilai cahaya listrik dari sensor LDR lampu 3 .
    cahaya_lampu_3 = readLDR712( sensor_LDR_lampu_3 ) -
    median_point_lampu_3 ;
    cahaya_lampu_3 = convert2Curent( cahaya_lampu_3 );
    test_komponen_sensor_LDR();
    Serial.print("Value : " + String(cahaya_lampu_1) );
    if( cahaya_lampu_1 > nilai_ambang_tidak_ada_cahaya ){
      Serial.print(" Lampu 1 Menyala ");
    }else{
      Serial.print(" Lampu 1 Rusak ");
    }
  }
}
```

```
}
Serial.print(" , Value : " + String(cahaya_lampu_2) );
if( cahaya_lampu_2 > nilai_ambang_tidak_ada_cahaya ){
  Serial.print(" Lampu 2 Menyala ");
}else{
  Serial.print(" Lampu 2 Rusak ");
}
Serial.print(" , Value : " + String(cahaya_lampu_3) );
if( cahaya_lampu_3 > nilai_ambang_tidak_ada_cahaya ){
  Serial.print(" Lampu 3 Menyala ");
}else{
  Serial.print(" Lampu 3 Rusak ");
}
Serial.println();
delay(1000);
}
}
```

Kode akses untuk lcd:

```
unsigned long tmUpdateDisplay = millis();
int MaxTMDisplay = 500 ;
//====> Tampilan informasi form user <====//
void display_TampilanUtama(){
    if( (millis()-tmUpdateDisplay) > MaxTMDisplay ){
        lcd.clear();
        //==> Tampilkan informsi analisa cahaya .
        lcd.setCursor( 0 , 0 );
        lcd.print("Lux: ");
        lcd.print( String(lightMeter.readLightLevel(),0) );
        lcd.print(" ,");
        if( cek_status_sensor_cahaya() == masih_terang ){
            lcd.print("Siang");
        }
        if( cek_status_sensor_cahaya() == sudah_gelap ){
            lcd.print("Malam");
        }
        lcd.setCursor( 0 , 1 );
        if( mode_Lampu.toInt() == lampu_dalam_mode_manual ){
            lcd.print("Manual");
        }
        if( mode_Lampu.toInt() == lampu_dalam_mode_otomatis ){
            lcd.print("Otomatis");
        }
        if( detect_Objeck == ya ) lcd.print(", Object");
        else lcd.print(", Non ");
        lcd.print(" " + str_prox );
        lcd.setCursor( 0 , 2 );
        //==> Tampilkan informsi Sensor cahaya 1
        if( cahaya_lampu_1 > nilai_ambang_tidak_ada_cahaya ){
            lcd.print("1 On");
        }else{
            lcd.print("1 Off");
        }
        if( cahaya_lampu_2 > nilai_ambang_tidak_ada_cahaya ){
            lcd.print(",2 On");
        }else{
            lcd.print(",2 Off");
        }
        if( cahaya_lampu_3 > nilai_ambang_tidak_ada_cahaya ){
            lcd.print(", 3 On");
        }else{
            lcd.print(", 3 Off");
        }
        //==> Reset Timer .
        tmUpdateDisplay = millis();
    }
}
//====> Tampilan Test Proximity <====//
void display_test_proximity( bool stat1 , bool stat2 , bool stat3
){
    if( (millis()-tmUpdateDisplay) > MaxTMDisplay ){
        lcd.clear();
        //==> Tampilkan form .
    }
}
```

```

        lcd.setCursor( 0 , 0 );
        lcd.print("Status Relay");
        //==> Tampilkan form .
        lcd.setCursor( 0 , 1 );
        lcd.print("Proxim 1 :");
        if( stat1 ) lcd.print("Aktif");
        else lcd.print("Non Aktif");
        //==> Tampilkan form .
        lcd.setCursor( 0 , 2 );
        lcd.print("Proxim 2 :");
        if( stat2 ) lcd.print("Aktif");
        else lcd.print("Non Aktif");
        //==> Tampilkan form .
        lcd.setCursor( 0 , 3 );
        lcd.print("Proxim 3 :");
        if( stat3 ) lcd.print("Aktif");
        else lcd.print("Non Aktif");
        //==> Reset Timer .
        tmUpdateDisplay = millis();
    }
}
void display_test_sensor_cahaya( ){
    if( (millis()-tmUpdateDisplay) > MaxTMDisplay ){
        lcd.clear();
        //==> Tampilkan informsi analisa cahaya .
        lcd.setCursor( 0 , 0 );
        lcd.print("Sensor Cahaya");
        //==> Tampilkan informasi nilai lux .
        lcd.setCursor( 0 , 1 );
        lcd.print( lightMeter.readLightLevel() );
        //==> Tampilkan Status sensor cahaya .
        lcd.setCursor( 0 , 2 );
        lcd.print("Status Sensor Cahaya");
        lcd.setCursor( 0 , 3 );
        if( cek_status_sensor_cahaya() == masih_terang ){
            lcd.print("Masih Terang");
        }
        if( cek_status_sensor_cahaya() == sudah_gelap ){
            lcd.print("Sudah Gelap");
        }
        //==> Reset Timer .
        tmUpdateDisplay = millis();
    }
}
void test_komponen_sensor_LDR( ){
    if( (millis()-tmUpdateDisplay) > MaxTMDisplay ){
        lcd.clear();
        //==> Tampilkan informsi Sensor cahaya 1
        lcd.setCursor( 0 , 1 );
        lcd.print("S cahaya 1: " + String(cahaya_lampu_1));
        //==> Tampilkan informsi Sensor cahaya 2
        lcd.setCursor( 0 , 2 );
        lcd.print("S cahaya 2: " + String(cahaya_lampu_2));
        //==> Tampilkan informsi Sensor cahaya 1
        lcd.setCursor( 0 , 3 );
        lcd.print("S cahaya 3: " + String(cahaya_lampu_3));
    }
}

```

```
    //==> Reset Timer .  
    tmUpdateDisplay = millis();  
  }  
}
```

Kode akses untuk sensor intensitas cahaya dan modul ac dimmer light:

```
//====> parameter yang menyimpan informasi informasi cahaya -
//====> berdasarkan hasil pembacaan sensor cahaya .
float nilai_cahaya = 0 ;
//====> parameter nilai yang menyimpan informasi bahwa cahaya -
//====> sekarang dalam kondisi gelap .
bool sinar_Sekitar_Sudah_Gelap = 0 ;
void inisialSensor(){
    lightMeter.begin();
}
bool cek_status_sensor_cahaya(){
    nilai_cahaya = lightMeter.readLightLevel();
    sinar_Sekitar_Sudah_Gelap=nilai_cahaya>nilai_Ambang_Batas_gelap;
    return sinar_Sekitar_Sudah_Gelap ;
}
void test_komponen_sensor_cahaya( bool izinkan ){
    //====> cek jika kode di izinkan untuk menyala .
    //====> Parameter untuk cek jika ada perubahan data sinar .
    float value_cahaya = 0.0 ;
    while( izinkan == ya ){
        display_test_sensor_cahaya();
        if( abs( value_cahaya - lightMeter.readLightLevel() ) > 10 ){
            value_cahaya = lightMeter.readLightLevel();
            Serial.print("Perubahan Cahaya : Nilai Lux : " + String(
value_cahaya ) + " , " );
            if( cek_status_sensor_cahaya() == masih_terang ){
                Serial.println("Masih Terang");
            }
            if( cek_status_sensor_cahaya() == sudah_gelap ){
                Serial.println("Sudah Gelap");
            }
        }
    }
}
#include <RBDdimmer.h>
dimmerLamp dimmer1(pin_pwm_1);
dimmerLamp dimmer2(pin_pwm_2);
dimmerLamp dimmer3(pin_pwm_3);
void inisialDimmer_lampu() {
    dimmer1.begin(NORMAL_MODE, ON); //dimmer initialisation:
name.begin(MODE, STATE)
    dimmer2.begin(NORMAL_MODE, ON); //dimmer initialisation:
name.begin(MODE, STATE)
    dimmer3.begin(NORMAL_MODE, ON); //dimmer initialisation:
name.begin(MODE, STATE)
    dimmer1.setPower(100); // setPower(0-100%);
    dimmer2.setPower(100); // setPower(0-100%);
    dimmer3.setPower(100); // setPower(0-100%);
}
void testKomponen_ACLightDimmer( bool izinkan ){
    //====> cek jika test komponen relay di izinkan menyala .
    String msg = "" ;
    while( izinkan == ya ){
        Serial.println("=====>");
    }
}
```

```
    Serial.println("Ketik Nilai 0 - 100% untuk setting nilai
kecerahan : " );
    while( true ){
        msg = readMessageCom();
        if( msg.length() > 0 ){
            break ;
        }
    }
    Serial.println("Set Nilai Kecerahan:"+String( msg.toInt() ));
    int value = msg.toInt();
    dimmer1.setPower( value );
    dimmer2.setPower( value );
    dimmer3.setPower( value );
}
}
```

Kode akses untuk nodemcu:

```
//=====> Fungsi untuk membaca pesan masuk dari
nodemcu/ESP8266 .
String readMessageEsp(){
    String msg = "" ;
    while( Serial3.available() ){
        char cht = Serial3.read();
        msg += cht ;
        delay(5);
    }
    return msg ;
}
void testSerialConnection( bool stat ){
    unsigned long tmsend = millis();
    while( stat ){
        String temps = readMessageEsp();
        if( temps.length() > 0 ){
            Serial.println("Data : " + temps );
        }
        if( (millis()-tmsend) > 2000 ){
            Serial3.println("test komunikasi");
            Serial.println("Kirim:test komunikasi");
            tmsend = millis();
        }
        delay(3);
    }
}
//=====> Generated Data Yang Akan Di Kirim Menuju Node //
//=====> Nodemcu .
String createDataSend(){
    String pesan = "" ;
    pesan += String( nilai_cahaya ) + "," ;
    pesan += String( sinar_Sekitar_Sudah_Gelap ) + "," ;
    pesan += String( cahaya_lampu_1 ) + "," ;
    pesan += String( cahaya_lampu_2 ) + "," ;
    pesan += String( cahaya_lampu_3 ) + "," ;
    pesan += String( cahaya_lampu_1 > nilai_ambang_tidak_ada_cahaya
) + "," ;
    pesan += String( cahaya_lampu_2 > nilai_ambang_tidak_ada_cahaya
) + "," ;
    pesan += String( cahaya_lampu_3 > nilai_ambang_tidak_ada_cahaya
) + "," ;
    pesan += "0,0" ;
    return pesan ;
}
//==> Kode timer untuk pewaktuan dalam pengiriman informasi ke
nodemcu .
unsigned long timerUpdateServer = millis();
float maxTimerUpdateServer = 1500 ;
//==> Kode timer untuk pewaktuan dalam pengiriman informasi ke
nodemcu .
unsigned long timerUpdateNodeMCU = millis();
float maxTimerUpdateNodeMCU = 500 ;
//=====> Fungsi untuk mengirim informasi
//=====> 1. Informasi Sensor .
```

```

//=====> 2. Informasi kondisi cahaya matahari .
//=====> 3. Informasi kondisi lampu .
void updateInformasiServer(){
    if( ( millis() - timerUpdateServer ) > maxTimerUpdateServer ){
        Serial.println("Pesan Kirim : Ke Node MCU");
        Serial3.print( "kirim#" );
        timerUpdateServer = millis(); ;
    }
    else if( ( millis() - timerUpdateNodeMCU ) >
maxTimerUpdateNodeMCU ){
        String pesan = createDataSend() ;
        Serial.println("Pesan Kirim :" + pesan);
        Serial3.print( pesan + "#" );
        timerUpdateNodeMCU = millis(); ;
    }
}
}
//====> Fungsi untuk menguraikan informasi yang dikirim dari
nodemcu .
//====> karena informasi data yang di kirim dari nodemcu masih
dalam bentuk -
//====> gabungan karakter , contoh 023,232,332,34e3,232,0 .
//====> maka hcahaya di pecah menjadi per informasi .
//====> nomor 1 => 023 .
//====> nomor 2 => 232 .
//====> nomor 3 => 332 .
//====> nomor 4 => 34e3 .
//====> nomor 4 => 232
void parseData( String msg ){
    String temp = "" ;
    int index_ = 0 ;
    for( int idx = 0 ; idx < msg.length() ; idx++ ){
        char temps = msg.charAt( idx );
        if( temps == ',' ){
            if( index_ == 0 ){ wifiConnected = temp.toInt() == 1 ;}
            else if( index_ == 1 ){ mode_Lampu = temp ;}
            else if( index_ == 2 ){ set_kecerahan_lampu_1 = temp ;}
            else if( index_ == 3 ){ set_kecerahan_lampu_2 = temp ;}
            else if( index_ == 4 ){ set_kecerahan_lampu_3 = temp ;}
            else if( index_ == 5 ){ stat_manual_lampu_1 = temp ;}
            else if( index_ == 6 ){ stat_manual_lampu_2 = temp ;}
            else if( index_ == 7 ){ stat_manual_lampu_3 = temp ;}
            temp = "" ;
            index_++ ;
        }else{
            temp += temps ;
        }
    }
    Serial.println("=====>");
    Serial.println("Status Wifi : " + String(wifiConnected));
    Serial.println("Mode : " + String(mode_Lampu));
    Serial.println("Set Lampu 1 : " +
String(set_kecerahan_lampu_1));
    Serial.println("Set Lampu 2 : " +
String(set_kecerahan_lampu_2));
    Serial.println("Set Lampu 3 : " +
String(set_kecerahan_lampu_3));
}

```

```
    Serial.println("Stat manual lampu 1 : " +  
String(stat_manual_lampu_1));  
    Serial.println("Stat manual lampu 2 : " +  
String(stat_manual_lampu_2));  
    Serial.println("Stat manual lampu 3 : " +  
String(stat_manual_lampu_3));  
}
```

Kode akses untuk sensor *proximity infrared*:

```
//==> Fungsi untuk inisial relay .
void inisialProximity(){
    pinMode( proximity_lampu_1 , INPUT );
    pinMode( proximity_lampu_2 , INPUT );
    pinMode( proximity_lampu_3 , INPUT );
}
bool baca_status_proximity( int pin_proximity ){
    return digitalRead( pin_proximity ) == tidak ;
}
//==> Kode fungsi untuk test komponen proximity .
void test_proximity( bool izinkan ){
    while( izinkan == ya ){
        //==> Reset lcd terlebih dahulu .
        lcd.clear();
        //==> Test proximity lampu 1 .
        lcd.setCursor( 0 , 0 );
        if( baca_status_proximity( proximity_lampu_1 ) == ya ){
            lcd.print("Prox 1 Deteksi");
            Serial.print("Prox 1 Deteksi");
        }
        else{
            lcd.print("Prox 1 non");
            Serial.print(" , Prox 1 non");
        }
        //==> Test proximity lampu 2 .
        lcd.setCursor( 0 , 1 );
        if( baca_status_proximity( proximity_lampu_2 ) == ya ){
            lcd.print("Prox 2 Deteksi");
            Serial.print(" , Prox 2 Deteksi");
        }
        else{
            lcd.print("Prox 2 non");
            Serial.print(" , Prox 2 non");
        }
        //==> Test proximity lampu 3 .
        lcd.setCursor( 0 , 2 );
        if( baca_status_proximity( proximity_lampu_3 ) == ya ){
            lcd.print("Prox 3 Deteksi");
            Serial.print(" , Prox 3 Deteksi");
        }
        else{
            lcd.print("Prox 3 non");
            Serial.print(" , Prox 3 non");
        }
        Serial.println();
        delay(1500);
    }
}
```

Kode akses untuk modul *relay*:

```
//====> Fungsi untuk inisial relay .
void inisialRelay(){
    pinMode( lampu_jalan , OUTPUT );
    pinMode( lamp_flasing_1 , OUTPUT );
    pinMode( lamp_flasing_2 , OUTPUT );
    pinMode( lamp_flasing_3 , OUTPUT );
}
//=====> Nyalakan Relay , untuk jenis relay aktif LOW .
void nyalakanRelay_activeLOW( byte pinRelay ){
    digitalWrite( pinRelay , LOW );
}
//=====> Padamkan Relay , untuk jenis relay aktif LOW .
void padamkanRelay_activeLOW( byte pinRelay ){
    digitalWrite( pinRelay , HIGH );
}
//=====> Nyalakan Relay , untuk jenis relay aktif HIGH .
void nyalakanRelay_activeHigh( byte pinRelay ){
    digitalWrite( pinRelay , HIGH );
}
//=====> Padamkan Relay , untuk jenis relay aktif HIGH .
void padamkanRelay_activeHigh( byte pinRelay ){
    digitalWrite( pinRelay , LOW );
}
void settingStatusRelay( byte alamat_pinRelay , bool relayState ){
    if( isAktifHigh == false ){
        if( relayState == true ){
            nyalakanRelay_activeHigh( alamat_pinRelay );
        }else{
            padamkanRelay_activeHigh( alamat_pinRelay );
        }
    }
    else{
        if( relayState == true ){
            nyalakanRelay_activeLOW( alamat_pinRelay );
        }else{
            padamkanRelay_activeLOW( alamat_pinRelay );
        }
    }
}
//====> Kode fungsi untuk test relau .
void test_komponen_relay( bool izinkan ){
    //====>
    String msg = "" ;
    //====> status .
    bool relay_1 = true ;
    bool relay_2 = true ;
    bool relay_3 = true ;
    //====>
    if( izinkan == ya ){
        display_test_proximity( relay_1 , relay_2 , relay_3 );
        settingStatusRelay( lamp_flasing_1 , relay_1 );
        relay_1 = !relay_1 ;
        settingStatusRelay( lamp_flasing_2 , relay_2 );
        relay_2 = !relay_2 ;
    }
}
```

```

        settingStatusRelay( lamp_flasing_3 , relay_3 );
        relay_3 = !relay_3 ;
    }
    //==> cek jika test komponen relay di izinkan menyala .
    while( izinkan == ya ){
        Serial.println("=====>");
        Serial.println("Ketik A [ Enter ] untuk menyalakan dan
mematikan relay 1");
        Serial.println("Ketik B [ Enter ] untuk menyalakan dan
mematikan relay 2");
        Serial.println("Ketik C [ Enter ] untuk menyalakan dan
mematikan relay 3");
        while( true ){
            msg = readMessageCom();
            if( msg.length() > 0 ){
                break ;
            }
        }
        if( msg.indexOf("A") > -1 ){
            settingStatusRelay( lamp_flasing_1 , relay_1 );
            if( relay_1 )
                Serial.println("Relay 1 Di Nyalakan ");
            else
                Serial.println("Relay 1 Di Padamkan ");
            relay_1 = !relay_1 ;
        }
        if( msg.indexOf("B") > -1 ){
            settingStatusRelay( lamp_flasing_2 , relay_2 );
            if( relay_2 )
                Serial.println("Relay 2 Di Nyalakan ");
            else
                Serial.println("Relay 2 Di Padamkan ");
            relay_2 = !relay_2 ;
        }
        if( msg.indexOf("C") > -1 ){
            settingStatusRelay( lamp_flasing_3 , relay_3 );
            if( relay_3 )
                Serial.println("Relay 3 Di Nyalakan ");
            else
                Serial.println("Relay 3 Di Padamkan ");
            relay_3 = !relay_3 ;
        }
        display_test_proximity( relay_1 , relay_2 , relay_3 );
    }
}

```

Lampiran 3 Program NodeMCU

Kode program keseluruhan NodeMCU:

```
#include "FirebaseESP8266.h"
#include <SoftwareSerial.h>
#include <ESP8266HTTPClient.h>
#include <ESP8266WiFi.h>
#include "code_ard.h"
#include "code_server.h"
#include "code_firebase.h"

void setup() {
  Serial.begin(9600);
  esp.begin(9600);
  testSerialConnection( false );
  koneksiWifi();
  inisialKomunikasiFirebase();
}

unsigned long tmSendLogData = millis();
float maxTmMenit = 30 ;

void loop() {
  String msg = readMessageEsp();
  if( msg.length() > 0 ){
    Serial.println( msg );
    if( msg.indexOf(" kirim") > -1 ){
      //==> Perintah untuk mengirim data ke server .
      updateDataServer_setSetting();
    }
    else{
      //==> data diterima .
      Serial.println("Data Diterima : " + msg );
      //==> parse data .
      parseData( msg );
      //==> Tampilkan data .
      showDataReceiver();
    }
  }
  //=====> Update informasi dari database .
  updateInformasiPerintahSettingDariDatabase();
  updateInformasiArduino();
}
```

Kode akses untuk firebase:

```
#define FIREBASE_HOST "https://projectstreetlamp-ala64-default-
rtbd.firebaseio.com/"
#define FIREBASE_AUTH "yvOP2urWAVTHQumoANjQrzJYyBlmviCzmWdkdU2B"
FirebaseData firebaseData;
FirebaseJson json;
String path = "/";
String path_cahaya = "nilaicahaya" ;
String path_kondisiCahaya = "kondisicahaya" ;
//====> Arus Lampu .
String path_arusLampu1 = "arus_lampu_1" ;
String path_arusLampu2 = "arus_lampu_2" ;
String path_arusLampu3 = "arus_lampu_3" ;
//====> Status Lampu .
String path_stat_arusLampu1 = "stat_lampu_1" ;
String path_stat_arusLampu2 = "stat_lampu_2" ;
String path_stat_arusLampu3 = "stat_lampu_3" ;
//====> Mode Lampu .
String path_modeLampu = "mode_Lampu" ;
//====> Status Tingkat Kecerahan .
String path_tingkat_Kecerahan1 = "tingkat_Kecerahan_1" ;
String path_tingkat_Kecerahan2 = "tingkat_Kecerahan_2" ;
String path_tingkat_Kecerahan3 = "tingkat_Kecerahan_3" ;
//====> Status On/Off Lampu
String path_stat_manual_lampu_1 = "stat_manual_lampu_1" ;
String path_stat_manual_lampu_2 = "stat_manual_lampu_2" ;
String path_stat_manual_lampu_3 = "stat_manual_lampu_3" ;
void printResult(FirebaseData &data);
void inisialKomunikasiFirebase(){
    //=====> inisial firebase .
    Firebase.begin(FIREBASE_HOST, FIREBASE_AUTH);
    Firebase.reconnectWiFi(true);
    //Set the size of WiFi rx/tx buffers in the case where we want
to work with large data.
    firebaseData.setBSSLBufferSize(1024, 1024);
    //Set the size of HTTP response buffers in the case where we
want to work with large data.
    firebaseData.setResponseSize(1024);
    //Set database read timeout to 1 minute (max 15 minutes)
    Firebase.setReadTimeout(firebaseData, 1000 * 60);
    //tiny, small, medium, large and unlimited.
    //Size and its write timeout e.g. tiny (1s), small (10s), medium
(30s) and large (60s).
    Firebase.setwriteSizeLimit(firebaseData, "tiny");
}
void getInformationFromServer(){
    //====> ambil informsai tingkat kecerahan lampu 1.
    if(Firebase.getString( firebaseData , path + path_modeLampu ) )
    {
        String msg = firebaseData.stringData();
        mode_Lampu = msg ;
    }
    else
    {
```

```

        Serial.println("Failed mode lampu: " +
firebaseData.errorReason());
    }
    //==> ambil informsai tingkat kecerahan lampu 1.
    if(Firebase.getString( firebaseData , path +
path_tingkat_Kecerahan1 ) )
    {
        String msg = firebaseData.stringData();
        set_kecerahan_lampu_1 = msg ;
    }
    else
    {
        Serial.println("Failed Kecerahan 1: " +
firebaseData.errorReason());
    }
    //==> ambil informsai tingkat kecerahan lampu 2.
    if(Firebase.getString( firebaseData , path +
path_tingkat_Kecerahan2 ) )
    {
        String msg = firebaseData.stringData();
        set_kecerahan_lampu_2 = msg ;
    }
    else
    {
        Serial.println("Failed kecerahan 2: " +
firebaseData.errorReason());
    }
    //==> ambil informsai tingkat kecerahan lampu 3.
    if(Firebase.getString( firebaseData , path +
path_tingkat_Kecerahan3 ) )
    {
        String msg = firebaseData.stringData();
        set_kecerahan_lampu_3 = msg ;
    }
    else
    {
        Serial.println("Failed kecerahan 3: " +
firebaseData.errorReason());
    }
    //==> ambil informsai status lampu manual untuk lampu 1
    if(Firebase.getString( firebaseData , path +
path_stat_manual_lampu_1 ) )
    {
        String msg = firebaseData.stringData();
        stat_manual_lampu_1 = msg ;
    }
    else
    {
        Serial.println("Failed stat manual 1: " +
firebaseData.errorReason());
    }
    //==> ambil informsai status lampu manual untuk lampu 2
    if(Firebase.getString( firebaseData , path +
path_stat_manual_lampu_2 ) )
    {
        String msg = firebaseData.stringData();

```

```

        stat_manual_lampu_2 = msg ;
    }
    else
    {
        Serial.println("Failed stat manual 2: " +
firebaseData.errorReason());
    }
    //==> ambil informsai status lampu manual untuk lampu 3
    if(Firebase.getString( firebaseData , path +
path_stat_manual_lampu_3 ) )
    {
        String msg = firebaseData.stringData();
        stat_manual_lampu_3 = msg ;
    }
    else
    {
        Serial.println("Failed stat manual 3: " +
firebaseData.errorReason());
    }
}
unsigned long tmGetDataFromDatabase = millis();
float maxTimeGetDataFromDatabase = 2000 ;
void updateInformasiPerintahSettingDariDatabase() {
    if( (millis()-tmGetDataFromDatabase) >
maxTimeGetDataFromDatabase ) {
        getInformationFromServer();
        tmGetDataFromDatabase = millis();
    }
}
//==> update data dari server .
void updateDataServer_setSetting(){
    //==> Set String Untuk Sensor Cahaya .
    if( Firebase.setString( firebaseData , path + path_cahaya ,
String( nilai_cahaya ) ) )
        printResult(firebaseData);
    else
        Serial.println("Failed: " + firebaseData.errorReason());
    //==> Set String Untuk Kondisi Cahaya .
    if( Firebase.setString( firebaseData , path +
path_kondisiCahaya , String( sinar_Sekitar_Sudah_Gelap ) ) )
        printResult(firebaseData);
    else
        Serial.println("Failed: " + firebaseData.errorReason());
    //==> Set String Untuk Arus Lampu 1 .
    if( Firebase.setString( firebaseData , path + path_arusLampul
, String( arus_lampu_1 ) ) )
        printResult(firebaseData);
    else
        Serial.println("Failed: " + firebaseData.errorReason());
    //==> Set String Untuk Arus Lampu 2 .
    if( Firebase.setString( firebaseData , path + path_arusLampu2
, String( arus_lampu_2 ) ) )
        printResult(firebaseData);
    else
        Serial.println("Failed: " + firebaseData.errorReason());
    //==> Set String Untuk Arus Lampu 3 .

```

```

    if( Firebase.setString( firebaseData , path + path_arusLampu3
, String( arus_lampu_3 ) ) )
        printResult(firebaseData);
    else
        Serial.println("Failed: " + firebaseData.errorReason());
        //==> Set String Untuk Status Lampu 1 .
        if( Firebase.setString( firebaseData , path +
path_stat_arusLampu1 , String( stat_lampu_1 ) ) )
            printResult(firebaseData);
        else
            Serial.println("Failed: " + firebaseData.errorReason());
            //==> Set String Untuk Status Lampu 2 .
            if( Firebase.setString( firebaseData , path +
path_stat_arusLampu2 , String( stat_lampu_2 ) ) )
                printResult(firebaseData);
            else
                Serial.println("Failed: " + firebaseData.errorReason());
                //==> Set String Untuk Status Lampu 3 .
                if( Firebase.setString( firebaseData , path +
path_stat_arusLampu3 , String( stat_lampu_3 ) ) )
                    printResult(firebaseData);
                else
                    Serial.println("Failed: " + firebaseData.errorReason());
    }
void printResult(FirebaseData &data)
{
    if (data.dataType() == "int")
        Serial.println(data.intData());
    else if (data.dataType() == "float")
        Serial.println(data.floatData(), 5);
    else if (data.dataType() == "double")
        printf("%.9lf\n", data.doubleData());
    else if (data.dataType() == "boolean")
        Serial.println(data.boolData() == 1 ? "true" : "false");
    else if (data.dataType() == "string")
        Serial.println(data.stringData());
    else if (data.dataType() == "json")
    {
        Serial.println();
        FirebaseJson &jjson = data.jsonObject();
        //Print all object data
        Serial.println("Pretty printed JSON data:");
        String jsonStr;
        jjson.toString(jsonStr,true);
        Serial.println(jsonStr);
        Serial.println();
        Serial.println("Iterate JSON data:");
        Serial.println();
        size_t len = jjson.iteratorBegin();
        String key, value = "";
        int type = 0;
        for (size_t i = 0; i < len; i++)
        {
            jjson.iteratorGet(i, type, key, value);
            Serial.print(i);
            Serial.print(", ");

```

```

        Serial.print("Type: ");
        Serial.print(type == FirebaseJson::JSON_OBJECT ?
"object" : "array");
        if (type == FirebaseJson::JSON_OBJECT)
        {
            Serial.print(", Key: ");
            Serial.print(key);
        }
        Serial.print(", Value: ");
        Serial.println(value);
    }
    json.iteratorEnd();
}
else if (data.dataType() == "array")
{
    Serial.println();
    //get array data from FirebaseData using FirebaseJsonArray
object
    FirebaseJsonArray &arr = data.jsonArray();
    //Print all array values
    Serial.println("Pretty printed Array:");
    String arrStr;
    arr.toString(arrStr,true);
    Serial.println(arrStr);
    Serial.println();
    Serial.println("Iterate array values:");
    Serial.println();
    for (size_t i = 0; i < arr.size(); i++)
    {
        Serial.print(i);
        Serial.print(", Value: ");
        FirebaseJsonData &jsonData = data.jsonData();
        //Get the result data from FirebaseJsonArray object
        arr.get(jsonData, i);
        if (jsonData.typeNum == FirebaseJson::JSON_BOOL)
            Serial.println(jsonData.boolValue ? "true" :
"false");
        else if (jsonData.typeNum == FirebaseJson::JSON_INT)
            Serial.println(jsonData.intValue);
        else if (jsonData.typeNum == FirebaseJson::JSON_FLOAT)
            Serial.println(jsonData.floatValue);
        else if (jsonData.typeNum ==
FirebaseJson::JSON_DOUBLE)
            printf("%.9lf\n", jsonData.doubleValue);
        else if (jsonData.typeNum == FirebaseJson::JSON_STRING
||
            jsonData.typeNum == FirebaseJson::JSON_NULL
||
            jsonData.typeNum == FirebaseJson::JSON_OBJECT
||
            jsonData.typeNum == FirebaseJson::JSON_ARRAY)
            Serial.println(jsonData.stringValue);
    }
}
else
{

```

```
        Serial.println(data.payload());  
    }  
}
```

Kode akses untuk server:

```
//=====> Parameter setting untuk akses ke wireless .
#define WIFI_SSID "sakha"
#define WIFI_PASSWORD "prime468328"
//=====> Fungsi koneksi ke wireless
void koneksiWifi(){
  WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
  Serial.print("Connecting to Wi-Fi");
  int countForSpace = 0 ;
  while (WiFi.status() != WL_CONNECTED)
  {
    Serial.print(".");
    delay(300);
    if( countForSpace > 30 ){
      countForSpace = 0 ;
      Serial.println();
    }
    countForSpace++ ;
  }
  Serial.println();
  Serial.print("Connected with IP: ");
  Serial.println(WiFi.localIP());
  Serial.println();
}
```

Kode akses untuk ard

```
SoftwareSerial esp(14, 12);
//=====> parameter data .
String nilai_cahaya = "" ;
String sinar_Sekitar_Sudah_Gelap = "" ;
//=====> Informasi Status Lampu Rusak/Tidak Dari sensor LDR ,
String stat_lampu_1 = "0" ;
String stat_lampu_2 = "0" ;
String stat_lampu_3 = "0" ;
//=====> status mode arduino otomatis atau manual .
String mode_Lampu = "1" ;
//=====> Setting tingkat kecerahan lampu .
String set_kecerahan_lampu_1 = "0" ;
String set_kecerahan_lampu_2 = "0" ;
String set_kecerahan_lampu_3 = "0" ;
//=====> Setting On/Off lampu Manual.
String stat_manual_lampu_1 = "0" ;
String stat_manual_lampu_2 = "0" ;
String stat_manual_lampu_3 = "0" ;
//=====> Fungsi untuk membaca pesan masuk dari
nodemcu/ESP8266 .
String readMessageEsp(){
    String msg = "" ;
    unsigned long timeOut = millis();
    while( (millis()-timeOut) < 100 ){
        if( esp.available() ){
            char cht = esp.read();
            if( cht == '#' ) break ;
            msg += cht ;
            timeOut = millis();
        }
        delay(1);
    }
    return msg ;
}
String readMessageArd(){
    String temp = "" ;
    unsigned long tmWait = millis();
    while( (millis()-tmWait ) < 500 ){
        if( esp.available() ){
            char temps = esp.read();
            if( (int)temps > 34 && (int)temps < 65 ){
                temp += char( temps );
            }
            tmWait = millis();
        }
        if( Serial.available() ){
            char temps = Serial.read();
            if( (int)temps > 34 && (int)temps < 65 ){
                temp += char( temps );
            }
            tmWait = millis();
        }
        delay(1);
    }
}
```

```

    return temp ;
}
void showDataReceiver(){
    Serial.println();
    Serial.println("=====>");
    Serial.println("Cahaya : " + nilai_cahaya );
    Serial.println("Status cahaya : " + sinar_Sekitar_Sudah_Gelap );
    Serial.println("arus_lampu_1 : " + arus_lampu_1 );
    Serial.println("arus_lampu_2 : " + arus_lampu_2 );
    Serial.println("arus_lampu_3 : " + stat_lampu_3 );
    Serial.println("stat_lampu_1 : " + stat_lampu_1 );
    Serial.println("stat_lampu_2 : " + stat_lampu_2 );
    Serial.println("stat_lampu_3 : " + stat_lampu_3 );
}
void parseData( String msg ){
    String temp = "" ;
    int index_ = 0 ;
    for( int idx = 0 ; idx < msg.length() ; idx++ ){
        char temps = msg.charAt( idx );
        if( temps == ',' ){
            if( index_ == 0 ){
                nilai_cahaya = temp ;
            }
            else if( index_ == 1 ){
                sinar_Sekitar_Sudah_Gelap = temp ;
            }
            else if( index_ == 2 ){
                arus_lampu_1 = temp ;
            }
            else if( index_ == 3 ){
                arus_lampu_2 = temp ;
            }
            else if( index_ == 4 ){
                arus_lampu_3 = temp ;
            }
            else if( index_ == 5 ){
                stat_lampu_1 = temp ;
            }
            else if( index_ == 6 ){
                stat_lampu_2 = temp ;
            }
            else if( index_ == 7 ){
                stat_lampu_3 = temp ;
            }
            temp = "" ;
            index_++ ;
        }else{
            temp += temps ;
        }
    }
}
void testSerialConnection( bool stat ){
    unsigned long tmsend = millis();
    while( stat ){
        String temps = readMessageEsp();
        if( temps.length() > 0 ){

```

```

        Serial.println("Data : " + temps );
    }
    if( (millis()-tmsend) > 2000 ){
        esp.println("test komunikasi");
        Serial.println("Kirim:test komunikasi");
        tmsend = millis();
    }
    delay(3);
}
}
unsigned long timerUpdateServer = millis();
float maxTimerUpdateServer = 1000 ;
String createDataSend(){
    String pesan = "" ;
    pesan += String( WiFi.status() != WL_CONNECTED ) + "," ;
    pesan += mode_Lampu + "," ;
    pesan += set_kecerahan_lampu_1 + "," ;
    pesan += set_kecerahan_lampu_2 + "," ;
    pesan += set_kecerahan_lampu_3 + "," ;
    pesan += stat_manual_lampu_1 + "," ;
    pesan += stat_manual_lampu_2 + "," ;
    pesan += stat_manual_lampu_3 + "," ;
    pesan += "0" ;
    return pesan ;
}
void updateInformasiArduino(){
    if( ( millis() - timerUpdateServer ) > maxTimerUpdateServer ){
        String temp = createDataSend();
        Serial.println("Data Kirim : " + esp);
        esp.print( temp );
        timerUpdateServer = millis(); ;
    }
}
}

```