



TUGAS AKHIR – TE 145561

**RANCANG BANGUN SISTEM PENGATURAN DAN MONITORING
LAMPU PJU PADA TAMAN KOTA MENGGUNAKAN ANDROID**

Kukuh Ariwibowo
NRP 2213039019
Muhammad Wildan Wicaksono
NRP 2213039045

Dosen Pembimbing
Rachmad Setiawan S.T., M.T
Agus Suhanto, S.Pd

PROGRAM STUDI D3 TEKNIK ELEKTRO
Fakultas Teknologi Industri
Institut Teknologi Sepuluh Nopember
Surabaya 2016



FINAL PROJECT – TE 145561

**CONTROL SYSTEM DESIGN AND MONITORING LAMPS IN THE
PARK CENTER OF CITY USING ANDROID**

Kukuh Ariwibowo
NRP 2213039019
Muhammad Wildan Wicaksono
NRP 2213039045

Advisor
Rachmad Setiawan S.T., M.T
Agus Suhanto, S.Pd

ELECTRICAL ENGINEERING D3 STUDY PROGRAM
Faculty of Industrial Technology
Institut Teknologi Sepuluh Nopember
Surabaya 2016

**RANCANG BANGUN SISTEM PENGATURAN DAN
MONITORING LAMPU PJU PADA TAMAN KOTA
MENGGUNAKAN ANDROID**

TUGAS AKHIR

**Diajukan Guna Memenuhi Sebagian Persyaratan
Untuk Memperoleh Gelar Ahli Madya Teknik
Pada**

**Bidang Studi Elektro Industri
Program Studi D3 Teknik Elektro
Institut Teknologi Sepuluh Nopember**

Menyetujui:

Dosen Pembimbing 1

Dosen Pembimbing 2



Rachmad Setiawan, S.T., M.T
NIP. 19690529 199512 1 001

Agus Suhanto, S.Pd
NIP. 19650821 198603 1 010

**SURABAYA
JUNI, 2016**

RANCANG BANGUN SISTEM PENGATURAN DAN MONITORING LAMPU PJU PADA TAMAN KOTA MENGUNAKAN ANDROID

Nama Mahasiswa :

- 1. Kukuh Ariwibowo (2213039019)**
- 2. Muhammad Wildan Wicaksono (2213039045)**

Dosen Pembimbing :

- 1. Rachmad Setiawan, S.T., M.T. (1969052 9199512 1 001)**
- 2. Agus Suhanto, S.Pd (19650821 198603 1 010)**

ABSTRAK

Lampu jalan atau dikenal juga sebagai Penerangan Jalan Umum (PJU) adalah lampu yang digunakan untuk penerangan jalan dimalam hari sehingga mempermudah pejalan kaki, pesepeda dan pengendara kendaraan dapat melihat dengan lebih jelas jalan/medan yang akan dilalui pada malam hari. Namun, satu kendalanya yaitu sistem Penerangan Jalan Umum (PJU) tersebut masih menggunakan sistem pengendali konvensional yang tidak akan diketahui secara langsung apabila terjadi kerusakan baik lampu maupun jaringan kabelnya.

Maka perlu direalisasikan sebuah sistem Monitoring dan pengendali otomatis yang terpusat, sudah tidak menggunakan saklar lagi karena di sini kami menggunakan software (aplikasi) yang terinstal pada hardware yang berbasis Android. Melalui aplikasi ini kami dapat mengontrol objek (lampu taman kota).

Persentase keberhasilan untuk monitor dapat mencapai 85%. Sedangkan untuk sistem kendali persentase keberhasilan hanya mencapai 90%. Hal ini disebabkan karena koneksi yang kurang stabil dari modul Wi-Fi kami menuju perangkat android yang digunakan, sehingga pengiriman data antara perangkat android menuju sistem kendali dan sebaliknya menjadi terganggu.

Kata Kunci: ESP8266, Mikrokontroler, PJU, *Solid State Relay*, Trafo Arus.

**CONTROL SYSTEM DESIGN AND MONITORING PJU LIGHTS
IN THE PARK CENTER OF CITY USING ANDROID**

Student Name :

- | | |
|-------------------------------------|---------------------|
| 1. Kukuh Ariwibowo | (2213039019) |
| 2. Muhammad Wildan Wicaksono | (2213039045) |

Advisor :

- | | |
|--|--------------------------------|
| 1. Rachmad Setiawan, S.T., M.T. | (1969052 9199512 1 001) |
| 2. Agus Suhanto, S.Pd | (19650821 198603 1 010) |

ABSTRACT

The street lights are lamps used for street lighting at night making it easier for pedestrians, cyclists and motorists can see more clearly the road / terrain to be traversed at night. However, one obstacle that street lighting system is still using a conventional control system that will not be known directly in the event of damage to both lamp and cord tissue.

It is necessary to realize a monitoring system and automatic centralized control, is not using the switch again because we are using the software (app) installed on Android-based hardware. Through this application we can control objects (lights city park).

The percentage of success for the monitor can reach 85%. As for the control system success percentage reached 90%. This is because our Wi-Fi modules have a connection less stable for android devices are used, so that the data transmission between an android device to the control system and otherwise becomes disturbed.

Keywords: *Current transformer, ESP8266, Microcontroller, Solid State Relays, Street Lights.*

DAFTAR ISI

	HALAMAN
HALAMAN JUDUL	i
HALAMAN JUDUL	i
PERNYATAAN KEASLIAN TUGAS AKHIR.....	v
HALAMAN PENGESAHAN	vii
ABSTRAK.....	ix
<i>ABSTRACT</i>	xi
KATA PENGANTAR.....	xiii
DAFTAR ISI.....	xv
DAFTAR GAMBAR	xvii
DAFTAR TABEL.....	xix
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Permasalahan	1
1.3 Batasan Masalah.....	2
1.4 Tujuan.....	2
1.5 Sistematika Laporan	2
1.6 Relevansi.....	3
BAB II TEORI DASAR.....	5
2.1 Lampu PJU.....	5
2.2 <i>Solid State Relay</i>	6
2.3 <i>Light Dependent Resistor</i>	9
2.4 Mikrokontroler ATMega16	11
2.5 <i>Current Transformer</i>	14
2.6 <i>Wireless Router</i>	16
2.7 ESP8266	19
2.8 <i>Real Time Clock</i>	20
BAB III PERANCANGAN DAN PEMBUATAN ALAT	21
3.1 Blok Fungsional Sistem.....	21
3.2 Perancangan Perangkat Keras (<i>Hardware</i>).....	22
3.2.1 Rangkaian LDR	22
3.2.2 Rangkaian Trafo Arus.....	24
3.2.3 Rangkaian Komparator.....	27

3.2.4	Skematik Mikrokontroler AT16.....	28
3.2.5	Rangkaian Pengendali SSR	30
3.2.6	Pengoneksian <i>Wifi Module</i>	31
3.3	Perancangan Perangkat Lunak (<i>Software</i>)	31
3.3.1	Pemrograman ATmega 16.....	31
3.3.2	Pemrograman Koneksi <i>Wifi Module</i>	32
3.3.3	Tampilan pada Komputer dan Android	34
BAB IV HASIL PENGUJIAN DAN ANALISA ALAT		37
4.1	Pengujian Instrumen LDR.....	37
4.2	Pengujian Instrumen Trafo Arus	39
4.3	Pengujian Rangkaian Komparator.....	40
4.4	Pengujian Driver SSR.....	42
4.5	Pengujian Data Mikrokontroler ATmega16	42
4.6	Pengujian <i>Wi-Fi Module</i>	44
4.7	Pengujian Alat Secara Keseluruhan.....	45
BAB V PENUTUP		49
DAFTAR PUSTAKA.....		51
LAMPIRAN A GAMBAR		A-1
LAMPIRAN B PROGRAM		B-1
LAMPIRAN C <i>DATASHEET</i>		C-1
DAFTAR RIWAYAT HIDUP.....		D-1

DAFTAR GAMBAR

HALAMAN

Gambar 2.1	Lampu PJU.....	5
Gambar 2.2	<i>Solid State Relay</i>	6
Gambar 2.3	Blok Diagram SSR	9
Gambar 2.4	<i>Light Dependent Resistor</i>	9
Gambar 2.5	Mikrokontroler ATmega16.....	11
Gambar 2.6	Current Transformer	14
Gambar 2.7	Wireless Router	16
Gambar 2.8	Ilustrasi <i>Wireless Router</i>	18
Gambar 2.9	ESP8266.....	19
Gambar 2.10	<i>Real Time Clock</i>	20
Gambar 3.1	Blok Fungsional	21
Gambar 3.2	Rangkaian Instrumen LDR	23
Gambar 3.3	Rangkaian Peak-Detector Trafo Arus	24
Gambar 3.4	Rangkain Komparator.....	27
Gambar 3.5	Skematik Mikrokontroler ATmega16	28
Gambar 3.6	Rangkaian Pengendalian SSR	30
Gambar 3.7	Koneksi Mikrokontroler dengan <i>Wifi Module</i>	31
Gambar 3.8	Flowchart Pemrograman ATmega16	32
Gambar 3.9	Flowchart Pengambilan Data <i>Wifi Router</i>	33
Gambar 3.10	Flowchart Perintah Pengendali SSR	34
Gambar 3.11	Tampilan Komputer Menggunakan LabView	34
Gambar 3.12	Blok Diagram LabView	35
Gambar 3.13	Tampilan Aplikasi Android	36
Gambar 4.1	Pengujian Komponen LDR.....	37
Gambar 4.2	Pengujian Instrumen LDR	38
Gambar 4.3	Pengujian Instrumen Trafo Arus.....	39
Gambar 4.4	Pengujian Komparator pada Instrumen LDR	40
Gambar 4.5	Pengujian Komparator pada Instrumen Trafo Arus ..	41
Gambar 4.6	Modul SSR 1 <i>Channel</i>	42
Gambar 4.7	Pengujian Alat Menggunakan <i>LabVIEW</i>	42
Gambar 4.8	Pengujian Alar Menggunakan Android.....	42

DAFTAR TABEL

HALAMAN

Tabel 3.1	Pin Mikrokontroler AT16	29
Tabel 3.2	Pin Pengendali SSR.....	30
Tabel 4.1	Hasil Pengujian LDR.....	37
Tabel 4.2	Hasil Pengujian Instrumen LDR	38
Tabel 4.3	Hasil Pengukuran Instrumen Trafo Arus.....	39
Tabel 4.4	Hasil Pengujian Komparator 1	41
Tabel 4.5	Hasil Pengujian Komparator 2	41
Tabel 4.6	Hasil Pengujian Modul SSR	42
Tabel 4.7	Hasil Pengujian ATMega16 1.....	43
Tabel 4.8	Hasil Pengujian ATMega16 2.....	43
Tabel 4.9	Hasil Pengujian ATMega16 3.....	44
Tabel 4.10	Pengujian <i>Wi-Fi Module</i> ESP8266-01 dengan Android	45
Tabel 4.11	Pengujian Sistem Monitor Menggunakan LabView .	45
Tabel 4.12	Pengujian Sistem Kontrol Menggunakan LabView ...	46
Tabel 4.13	Pengujian Sistem Kontrol Secara Keseluruhan	46
Tabel 4.14	Pengujian Sistem Monitor Secara Keseluruhan	47
Tabel 4.15	Pengujian Waktu Penyalaan dan Pemadaman Otomatis.....	47

BAB I

PENDAHULUAN

1.1 Latar Belakang

Lampu jalan atau dikenal juga sebagai Penerangan Jalan Umum (PJU) adalah lampu yang digunakan untuk penerangan jalan di malam hari sehingga mempermudah pejalan kaki, pesepeda dan pengendara kendaraan. Mereka dapat melihat dengan lebih jelas jalan/medan yang akan dilalui pada malam hari. Namun, kendalanya adalah sistem Penerangan Jalan Umum (PJU) tersebut masih menggunakan sistem pengendali konvensional yang tidak akan diketahui secara langsung apabila terjadi kerusakan baik lampu maupun jaringan kabelnya.

Maka perlu direalisasikan sebuah sistem monitoring dan pengendali otomatis yang terpusat, sudah tidak menggunakan saklar lagi karena di sini kami menggunakan software (aplikasi) yang terinstal pada hardware yang berbasis *Android*. Melalui aplikasi ini kita dapat mengontrol status objek (lampu taman kota) ataupun juga bisa menjadi jembatan sebagai kontrol *on/off* nya. Dengan ini maka diharapkan sistem ini dapat meringankan kerja dari petugas yang biasanya harus mondar-mandir untuk menyalakan atau mematikan lampu tersebut menggunakan saklar. Begitu pula jika ada kerusakan pada lampu atau jaringan kabelnya..

1.2 Permasalahan

Adapun permasalahan yang akan kami angkat sebagai bahan Tugas Akhir ini:

- Pembuatan instrumentasi sensor arus AC untuk mendeteksi gangguan jaringan listrik pada jaringan Lampu PJU pada Taman Kota.
- Pembuatan instrumentasi sensor cahaya tampak (LDR) untuk mendeteksi gangguan pada cahaya tampak dari Lampu untuk kondisi mneyala atau mati.
- Pembuatan rancangan sistem kendali lampu dengan mikrokontroler.
- Pembuatan *software* yang dapat mengontrol dan monitor kondisi lampu pada taman dengan Android

1.3 Batasan Masalah

Agar penulisan buku Tugas Akhir ini tidak menyimpang dan mengambang dari tujuan yang semula direncanakan sehingga mempermudah mendapatkan data dan informasi yang diperlukan, maka penulis menetapkan batasan-batasan masalah sebagai berikut :

1. Pembuatan rangkaian LDR sebagai sensor cahaya tampak.
2. Pembuatan rangkaian *Peak-detector* sebagai sensor arus dari trafo arus.
3. Pembuatan rangkaian SSR sebagai pengganti saklar konvensional.
4. Pemakaian ATmega16 sebagai kontroler pada sistem kendali dan monitor.
5. Penggunaan *Wi-Fi* sebagai media komunikasi menggunakan android.

1.4 Tujuan

Pembuatan Rancang Bangun Sistem Pengaturan dan *Monitoring* Lampu PJU pada Taman Kota menggunakan *Android* yang bertujuan untuk :

- Merancang serta membuat suatu sistem pengaturan dan monitoring lampu PJU pada Taman Kota menggunakan *Android*
- Memberi kemudahan kepada petugas untuk memantau dan mengatur kapan waktunya harus menyalakan dan mematikan lampu taman secara otomatis.

1.5 Sistematika Laporan

Sistematika pembahasan Tugas Akhir ini terdiri dari lima bab, yaitu pendahuluan, teori penunjang, perencanaan dan pembuatan alat, pengujian dan analisa alat, serta penutup

BAB I : PENDAHULUAN

Membahas tentang latar belakang, permasalahan, batasan masalah, maksud dan tujuan, sistematika laporan, serta relevansi.

BAB II : TEORI PENUNJANG

Berisi teori penunjang yang mendukung dalam perencanaan dan pembuatan alat.

BAB III : PERANCANGAN ALAT

Membahas tentang perencanaan dan pembuatan perangkat keras yang meliputi rangkaian-rangkaian, desain bangun, dan perangkat lunak yang meliputi program yang akan digunakan untuk mengaktifkan alat tersebut.

BAB IV : PENGUJIAN DAN ANALISA ALAT

Membahas tentang pengukuran, pengujian, dan penganalisaan terhadap kepresisian sensor dan alat yang telah kami buat.

BAB V : PENUTUP

Menjelaskan tentang kesimpulan dari Tugas Akhir ini dan saran-saran untuk pengembangan alat ini lebih lanjut.

1.6 Relevansi

Diharapkan alat ini dapat terealisasi, alat ini dapat digunakan untuk mengatur sistem penyalan lampu taman. Dengan sistem pengendali penerangan lampu taman ini dapat mempermudah pekerjaan dari petugas penjaga taman dan dapat mengontrol keadaan lampu-lampu yang berada di taman secara real-time.

---Halaman ini sengaja dikosongkan---

BAB II

TEORI PENUNJANG

Pada bab ini membahas tentang teori dasar dan teori penunjang dari peralatan-peralatan yang digunakan dalam pembuatan Rancang Bangun Sistem Pengaturan dan *Monitoring* Lampu PJU pada Taman Kota menggunakan *Android*.

2.1 Lampu PJU

Lampu jalan atau dikenal juga sebagai Penerangan Jalan Umum (PJU) adalah lampu yang digunakan untuk penerangan jalan di malam hari sehingga mempermudah pejalan kaki, pesepeda dan pengendara kendaraan. Mereka dapat melihat dengan lebih jelas jalan/medan yang akan dilalui pada malam hari. Namun, kendalanya adalah sistem Penerangan Jalan Umum (PJU) tersebut masih menggunakan sistem pengendali konvensional yang tidak akan diketahui secara langsung apabila terjadi kerusakan baik lampu maupun jaringan kabelnya.



Gambar 2.1 Lampu PJU

Gambar 2.1 merupakan contoh lampu PJU yang ada di sekitar. Beban lampu PJU biasanya sekitar 50-80W. Biasanya beban ditentukan saat perencanaan pembuatan sarana oleh pihak tertentu misalnya pemerintah untuk taman kota.

2.2 Solid State Relay

Solid state relay adalah relay yang elektronik, yaitu relay yang tidak menggunakan kontaktor mekanik. *Solid state relay* menggunakan kontaktor berupa komponen aktif seperti TRIAC, sehingga solid state relay dapat dikendalikan dengan tegangan rendah dan dapat digunakan untuk mengendalikan tegangan AC dengan voltase besar. Baik relay kontaktor biasa maupun *solid state relay* (SSR) mempunyai keuntungan dan kerugian. Baik keuntungan maupun kerugian tersebut merupakan ‘*trade-off*’ yang harus dipilih bagi disainer sistem *control*.



Gambar 2.2 Solid State Relay Omron G3-202P

Pengertian Solid State Relay

Pada dasarnya *Solid state relay* (SSR) merupakan relay yang dapat didiskripsikan sebagai berikut :

- Mempunyai empat buah terminal, 2 *input* terminal dan 2 buah *output* terminal.
- Tegangan *input* dapat berupa tegangan AC atau DC.
- Antara output dan input diisolasi dengan sistem optikal.
- Output menggunakan keluarga *thyristor*, SCR untuk beban DC dan TRIAC untuk beban AC.
- *Switching ON*, yang sering disebut ‘*firing*’, solid state relay hanya bisa terjadi pada saat tegangan yang masuk ke output pada level yang sangat rendah mendekati nol volt.
- Output berupa tegangan AC (50 Hz atau 60 Hz).

Penggunaan *solid state relay* mempunyai beberapa keuntungan yang menyebabkan *solid-state relay* saat ini menarik untuk digunakan pada aplikasi-aplikasi kontrol untuk beban AC daripada digunakannya relay mekanik (*Electromechanical Relay*, EMR), walaupun biaya sebuah *solid-state relay* lebih mahal daripada biaya sebuah relay mekanik biasa.

Keuntungan dan Kerugian *Solid State Relay*

Keuntungan menggunakan *Solid State Relay* :

1. Pada *solid-state relay* tidak terdapat bagian yang bergerak seperti halnya pada relay. Relay mempunyai sebuah bagian yang bergerak yang disebut kontaktor dan bagian ini tidak ada pada *solid-state relay*. Sehingga tidak mungkin terjadi '*no contact*' karena kontaktor tertutup debu bahkan karat.
2. Tidak terdapat '*bounce*', karena tidak terdapat kontaktor yang bergerak maka pada *solid-state relay* tidak terjadi peristiwa '*bounce*' yaitu peristiwa terjadinya pantulan kontaktor pada saat terjadi perpindahan keadaan. Dengan kata lain dengan tidak adanya *bounce* maka tidak terjadi percikan bunga api pada saat kontaktor berubah keadaan.
3. Proses perpindahan dari kondisi '*off*' ke kondisi '*on*' atau sebaliknya sangat cepat hanya membutuhkan waktu sekitar 10 μ s sehingga *solid-state relay* dapat dengan mudah dioperasikan bersama-sama dengan *zero-crossing* detektor. Dengan kata lain operasi kerja *solid-state relay* dapat disinkronkan dengan kondisi *zero crossing* detektor.
4. *Solid-State relay* kebal terhadap getaran dan guncangan. Tidak seperti relay mekanik biasa yang kontaktornya dapat dengan mudah berubah bila terkena guncangan/getaran yang cukup kuat pada *body* relay tersebut.
5. Tidak menghasilkan suara '*klik*', seperti relay pada saat kontaktor berubah keadaan.
6. Kontaktor output pada *solid-state relay* secara otomatis '*latch*' sehingga energi yang digunakan untuk aktivasi *solid-state relay* lebih sedikit jika dibandingkan dengan energi yang digunakan untuk aktivasi sebuah relay. Kondisi ON sebuah *solid-state relay* akan di-*latch* sampai *solid-state relay* mendapatkan tegangan sangat rendah, yaitu mendekati nol volt.
7. *Solid-State relay* sangat sensitif sehingga dapat dioperasikan langsung dengan menggunakan level tegangan CMOS bahkan level

tegangan TTL. Rangkaian kontrolnya menjadi sangat sederhana karena tidak memerlukan level konverter.

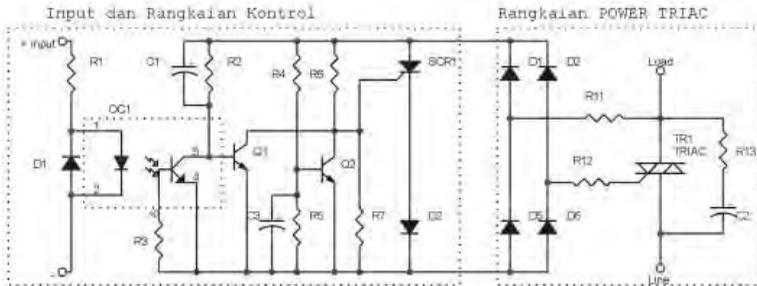
8. Masih terdapat *couple* kapasitansi antara input dan output tetapi sangat kecil sehingga arus bocor antara input output sangat kecil. Kondisi diperlukan pada peralatan *medical* yang memerlukan isolasi yang sangat baik.

Keuntungan *solid-state relay* begitu baik sekali tetapi dibalik keuntungan tersebut terdapat kerugian penggunaan *solid-state relay* yang perlu dipertimbangkan dalam penggunaannya.

Kerugian menggunakan *Solid State Relay* :

1. Resistansi Tegangan transien. Tegangan yang diatur/dikontrol oleh *solid-state relay* benar-benar tidak bersih. Dengan kata lain tidak murni tegangannya berupa sinyal sinus dengan tegangan *peak to peak* 380 vpp tetapi terdapat *spike-spike* yang dihasilkan oleh induksi motor atau peralatan listrik lainnya. *Spike* ini level tegangannya bervariasi jika terlalu besar maka dapat merusakkan *solid-state relay* tersebut. Selain itu sumber-sumber *spike* yang lain adalah sambaran petir, imbas dari *solenoid valve* dan lain sebagainya.
2. Tegangan *drop*. Karena *solid-state relay* dibangun dari bahan silikon maka terdapat tegangan jatuh antara tegangan input dan tegangan output. Tegangan jatuh tersebut kira-kira sebesar 1 volt. Tegangan jatuh ini menyebabkan adanya dissipasi daya yang besarnya tergantung dari besarnya arus yang lewat pada *solid-state relay* ini.
3. Arus bocor-'*leakage current*'. Pada saat *solid-state relay* ini dalam keadaan off atau keadaan open maka dalam kondisi yang idel seharusnya tidak ada arus yang mengalir melewati *solid-state relay* tetapi tidak demikian pada komponen yang sebenarnya. Besarnya arus bocor cukup besar untuk jika dibandingkan arus pada level TTL yaitu sekitar 10mA rms.
4. Sukar dimplementasikan pada aplikasi multi fasa.
5. Lebih mudah rusak jika terkena radiasi nuklir.

Rangkain Solid State Relay



Gambar 2.3 Rangkain SSR

Pada *solid-state relay*, *switching* unit-nya biasanya menggunakan TRIAC sehingga *solid-state relay* ini dapat mengalirkan arus baik arus positif maupun arus negatif. Walaupun demikian untuk mengontrol TRIAC ini digunakan SCR yang mempunyai karakteristik *gate* yang sangat sensitif. Kemudian untuk mengatur *trigger* pada SCR sendiri diatur dengan menggunakan rangkaian transistor. Rangkaian *transistor* ini menjadi penguat level tegangan yang didapat dari *optocoupler*. Penggunaan SCR untuk mengatur *gate* TRIAC karena *gate* SCR mempunyai karakteristik yang lebih sensitif daripada *gate* TRIAC.

Antara bagian *input* dan *output* dipisahkan dengan menggunakan *optocoupler* dan dengan sinyal yang kecil, cukup untuk menyalakan diode saja, maka cukup untuk menggerakkan sebuah beban AC yang besar melalui *solid-state relay*.

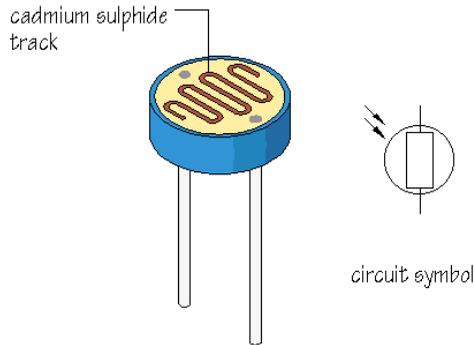
2.3 Light Dependent Resistor

LDR adalah singkatan dari *Light Dependent Resistor* yang merupakan salah satu jenis komponen elektronika resistor. Ya, komponen ini merupakan resistor yang nilai tahanan atau hambatannya sangat peka terhadap intensitas cahaya. Komponen LDR biasanya juga disebut dengan *photo resistor*, atau *photocell*.

Banyak orang yang menggunakan komponen ini sebagai pengganti sensor cahaya. Karena selain efektif, harganya pun lebih murah. Jika anda pernah melihat beberapa rangkaian elektronika sederhana seperti lampu jalan otomatis, pastinya anda pernah melihat komponen yang satu ini sebagai sensornya.

Perlu diketahui bahwa nilai resistansi LDR sangat dipengaruhi oleh intensitas cahaya. Semakin banyak cahaya yang mengenainya, maka

semakin menurun nilai hambatannya. Sebaliknya, jika cahaya yang mengenainya sedikit (gelap), maka nilai hambatannya menjadi semakin besar, sehingga arus listrik yang mengalir akan terhambat.



Gambar 2.4 *Light Dependent Resistor*

Prinsip Kerja LDR (*Light Dependent Resistor*)

Prinsip kerja LDR bisa dibilang sangat sederhana, tak jauh berbeda dari variabel resistor pada umumnya. LDR dipasang pada sebuah rangkaian elektronika dan dapat memutus dan menyambung aliran listrik berdasarkan cahaya. Semakin banyak cahaya yang mengenainya, maka semakin menurun nilai resistansinya. Sebaliknya, jika cahaya yang mengenainya sedikit (gelap), maka nilai hambatannya menjadi semakin besar.

Fungsi LDR (*Light Dependent Resistor*)

Dari penjabaran mengenai arti LDR tadi, fungsi LDR adalah sebagai sakelar otomatis berdasarkan cahaya. Jika cahaya yang diterima oleh LDR banyak, maka nilai resistansi LDR akan menurun, dan listrik dapat mengalir (ON). Sebaliknya, jika cahaya yang diterima LDR sedikit, maka nilai resistansi LDR akan menguat, dan aliran listrik terhambat (OFF).

2.4 Mikrokontroler

Mikrokontroler adalah sebuah sistem komputer fungsional dalam sebuah *chip*. Di dalamnya terkandung sebuah inti prosesor, memori (sejumlah kecil RAM, memori program, atau keduanya), dan perlengkapan *input output*.



Gambar 2.5 Mikrokontroler

Dengan kata lain, mikrokontroler adalah suatu alat elektronika digital yang mempunyai masukan dan keluaran serta kendali dengan program yang bisa ditulis dan dihapus dengan cara khusus, cara kerja mikrokontroler sebenarnya membaca dan menulis data. Sekedar contoh, bayangkan diri Anda saat mulai belajar membaca dan menulis, ketika Anda sudah bisa melakukan hal itu Anda bisa membaca tulisan apapun baik buku, cerpen, artikel dan sebagainya, dan anda pun bisa pula menulis hal-hal sebaliknya. Begitu pula jika Anda sudah mahir membaca dan menulis data maka Anda dapat membuat program untuk membuat suatu sistem pengaturan otomatis menggunakan mikrokontroler sesuai keinginan anda.

Mikrokontroler merupakan komputer di dalam *chip* yang digunakan untuk mengontrol peralatan elektronik, yang menekankan efisiensi dan efektifitas biaya. Secara harfiahnya bisa disebut “pengendali kecil” dimana sebuah sistem elektronik yang sebelumnya banyak memerlukan komponen-komponen pendukung seperti IC TTL dan CMOS dapat direduksi/diperkecil dan akhirnya terpusat serta dikendalikan oleh mikrokontroler ini.

Sistem Kerja Mikrokontroler

Mikrokontroler adalah salah satu dari bagian dasar dari suatu sistem komputer. Meskipun mempunyai bentuk yang jauh lebih kecil dari suatu komputer pribadi dan komputer *mainframe*, mikrokontroler dibangun dari elemen-elemen dasar yang sama. Secara sederhana, komputer akan menghasilkan *output* spesifik berdasarkan inputan yang diterima dan program yang dikerjakan.

Seperti umumnya komputer, mikrokontroler adalah alat yang mengerjakan instruksi-instruksi yang diberikan kepadanya. Artinya, bagian terpenting dan utama dari suatu sistem terkomputerisasi adalah program itu sendiri yang dibuat oleh seorang *programmer*. Program ini menginstruksikan komputer untuk melakukan jalinan yang panjang dari aksi-aksi sederhana untuk melakukan tugas yang lebih kompleks yang diinginkan oleh *programmer*.

Mikrokontroler memiliki berbagai macam sistem, yaitu sebagai berikut :

- **Sistem Input Komputer**

Piranti *input* menyediakan informasi kepada sistem komputer dari dunia luar. Dalam sistem komputer pribadi, piranti *input* yang paling umum adalah *keyboard*. Komputer *mainframe* menggunakan *keyboard* dan pembaca kartu berlubang sebagai piranti inputnya. Sistem dengan mikrokontroler umumnya menggunakan piranti *input* yang jauh lebih kecil seperti sakelar atau *keypad* kecil.

Hampir semua *input* mikrokontroler hanya dapat memproses sinyal *input* digital dengan tegangan yang sama dengan tegangan logika dari sumber. Level nol disebut dengan VSS dan tegangan positif sumber (VDD) umumnya adalah 5 volt. Padahal dalam dunia nyata terdapat banyak sinyal analog atau sinyal dengan tegangan level yang bervariasi. Karena itu ada piranti *input* yang mengonversikan sinyal analog menjadi sinyal digital sehingga komputer bisa mengerti dan menggunakannya. Ada beberapa mikrokontroler yang dilengkapi dengan piranti konversi ini, yang disebut dengan ADC, dalam satu rangkaian terpadu.

- **Sistem Output Komputer**

Piranti *output* digunakan untuk berkomunikasi informasi maupun aksi dari sistem komputer dengan dunia luar. Dalam sistem komputer pribadi (PC), piranti *output* yang umum adalah monitor CRT. Sedangkan sistem mikrokontroler mempunyai output yang jauh lebih sederhana seperti lampu indikator atau *beeper*. Frasa kontroler dari kata

mikrokontroler memberikan penegasan bahwa alat ini mengontrol sesuatu.

Mikrokontroler atau komputer mengolah sinyal secara digital, sehingga untuk dapat memberikan output analog diperlukan proses konversi dari sinyal digital menjadi analog. Piranti yang dapat melakukan konversi ini disebut dengan DAC (*Digital to Analog Converter*).

- **Central Processing Unit (CPU)**

CPU adalah otak dari sistem komputer. Pekerjaan utama dari CPU adalah mengerjakan program yang terdiri atas instruksi-instruksi yang diprogram oleh *programmer*. Suatu program komputer akan menginstruksikan CPU untuk membaca informasi dari piranti *input*, membaca informasi dari dan menulis informasi ke memori, dan untuk menulis informasi ke *output*.

Dalam mikrokontroler umumnya hanya ada satu program yang bekerja dalam suatu aplikasi. CPU M68HC05 mengenali hanya 60 instruksi yang berbeda. Karena itu sistem komputer ini sangat cocok dijadikan model untuk mempelajari dasar dari operasi komputer karena dimungkinkan untuk menelaah setiap operasi yang dikerjakan.

- **Clock dan Memori Komputer**

Sistem komputer menggunakan osilator *clock* untuk memicu CPU mengerjakan satu instruksi ke instruksi berikutnya dalam alur yang berurutan. Setiap langkah kecil dari operasi mikrokontroler memakan waktu satu atau beberapa *clock* untuk melakukannya.

Ada beberapa macam tipe dari memori komputer yang digunakan untuk beberapa tujuan yang berbeda dalam sistem komputer. Tipe dasar yang sering ditemui dalam mikrokontroler adalah ROM (*Read Only Memory*) dan RAM (*Random Access Memory*). ROM digunakan sebagai media penyimpan program dan data permanen yang tidak boleh berubah meskipun tidak ada tegangan yang diberikan pada mikrokontroler. RAM digunakan sebagai tempat penyimpanan data sementara dan hasil kalkulasi selama proses operasi. Beberapa mikrokontroler mengikutsertakan tipe lain dari memori seperti EPROM (*Erasable Programmable Read Only Memory*) dan EEPROM (*Electrically Erasable Programmable Read Only Memory*).

- **Program Komputer**

Program digambarkan sebagai awan karena sebenarnya program adalah hasil imajinasi seorang *programmer*. Komponen utama dari program adalah instruksi-instruksi dari instruksi set CPU. Program disimpan dalam memori dalam sistem komputer di mana mereka dapat secara berurutan dikerjakan oleh CPU.

- **Sistem Mikrokontroler**

Setelah dipaparkan bagian-bagian dari suatu sistem komputer, sekarang akan dibahas mengenai mikrokontroler. Digambarkan sistem komputer dengan bagian yang dikelilingi oleh garis putus-putus. Bagian inilah yang menyusun mikrokontroler. Bagian yang dilingkupi kotak bagian bawah adalah gambar lebih detail dari susunan bagian yang dilingkupi garis putus-putus. Kristal tidak termasuk dalam sistem mikrokontroler tetapi diperlukan dalam sirkuit osilator *clock*. Suatu mikrokontroler dapat didefinisikan sebagai sistem komputer yang lengkap termasuk sebuah CPU, memori, osilator *clock*, dan I/O dalam satu rangkaian terpadu. Jika sebagian elemen dihilangkan, yaitu I/O dan memori, maka *chip* ini akan disebut sebagai mikroprosesor.

2.5 Trafo Arus (*Current Transformer*)

Current transformer (CT) atau Trafo Arus adalah peralatan pada sistem tenaga listrik yang berupa trafo yang digunakan untuk pengukuran arus yang besarnya hingga ratusan ampere dan arus yang mengalir pada jaringan tegangan tinggi. Di samping untuk pengukuran arus, trafo arus juga digunakan untuk pengukuran daya dan energi, pengukuran jarak jauh, dan rele proteksi. Kumputan *primer* trafo dihubungkan seri dengan rangkaian atau jaringan yang akan diukur arusnya sedangkan kumputan *sekunder* dihubungkan dengan meter atau dengan rele proteksi.



Gambar 2.6 *Current Transformer*

Prinsip kerja trafo arus sama dengan trafo daya satu fasa. Bila pada kumparan primer mengalir arus I_1 , maka pada kumparan timbul gaya gerak magnet sebesar $N_1 I_1$. Gaya gerak ini memproduksi fluks pada inti, dan fluks ini membangkitkan gaya gerak listrik pada kumparan *sekunder*. Bila terminal kumparan sekunder tertutup, maka pada kumparan sekunder mengalir arus I_2 . Arus ini menimbulkan gaya gerak magnet $N_2 I_2$ pada kumparan *sekunder*. Pada trafo arus biasa dipasang *burden* pada bagian sekunder yang berfungsi sebagai impedansi beban, sehingga trafo tidak benar-benar *short circuit*. Apabila trafo adalah trafo ideal, maka berlaku persamaan :

$$N_1 I_1 = N_2 I_2$$

$$I_1 / I_2 = N_2 / N_1$$

di mana, N_1 : Jumlah belitan kumparan primer

N_2 : Jumlah belitan kumparan sekunder

I_1 : Arus kumparan primer

I_2 : Arus kumparan sekunder

Dalam pemakaian sehari-hari, trafo arus dibagi menjadi jenis-jenis tertentu berdasarkan syarat-syarat tertentu pula, adapun pembagian jenis trafo arus adalah sebagai berikut :

Jenis Trafo Arus Menurut Jumlah Kumparan Primer

a. Jenis Kumparan (*Wound*)

Biasa digunakan untuk pengukuran pada arus rendah, *burden* yang besar, atau pengukuran yang membutuhkan ketelitian tinggi. Belitan primer tergantung pada arus primer yang akan diukur, biasanya tidak lebih dari 5 belitan. Penambahan belitan primer akan mengurangi faktor *thermal* dan dinamis arus hubung singkat.

b. Jenis *Bar (Bar)*

Konstruksinya mampu menahan arus hubung singkat yang cukup tinggi sehingga memiliki faktor *thermis* dan dinamis arus hubung singkat yang tinggi. Keburukannya, ukuran inti yang paling ekonomis diperoleh pada arus pengenal yang cukup tinggi yaitu 1000A.

Jenis Trafo Arus Menurut Jumlah Rasio

a. Jenis Rasio Tunggal

Rasio tunggal adalah trafo arus dengan satu kumparan primer dan satu kumparan *sekunder*.

b. Jenis Rasio Ganda

Rasio ganda diperoleh dengan membagi kumparan *primer* menjadi beberapa kelompok yang dihubungkan seri atau *paralel*.

Jenis Trafo Arus Menurut Jumlah Inti

a. Inti Tunggal

Digunakan apabila sistem membutuhkan salah satu fungsi saja, yaitu untuk pengukuran atau proteksi.

b. Inti Ganda

Digunakan apabila sistem membutuhkan arus untuk pengukuran dan proteksi sekaligus.

Jenis Trafo Arus Menurut Konstruksi Isolasi

a. Isolasi Epoksi-Resin

Biasa dipakai hingga tegangan 110KV. Memiliki kekuatan hubung singkat yang cukup tinggi karena semua belitan tertanam pada bahan isolasi. Terdapat 2 jenis, yaitu jenis *bushing* dan pendukung.

b. Isolasi Minyak-Kertas

Isolasi minyak kertas ditempatkan pada kerangka porselen. Merupakan trafo arus untuk tegangan tinggi yang digunakan pada gardu induk dengan pemasangan luar. Dibedakan menjadi jenis tangki logam, kerangka isolasi, dan jenis gardu. Kelebihannya, penyulang pada sisi primer lebih pendek, digunakan untuk arus pengenal dan arus hubung singkat yang besar.

2.6 *Wireless Router*

Router adalah perangkat network yang digunakan untuk menghubungkan beberapa *network*, baik *network* yang sama maupun berbeda dari segi teknologinya seperti menghubungkan *network* yang menggunakan topologi *Bus*, *Star* dan *Ring*. *Router* minimal memiliki 2 *network interface*. Dalam postingan sebelumnya tentang mengenal teknik *subnetting* telah disinggung bahwa koneksi antar *network* (jaringan dengan subnet IP yang berbeda) hanya bisa terjadi dengan bantuan *Router*.

Router adalah sebuah alat yang mengirimkan paket data melalui sebuah jaringan atau Internet menuju tujuannya, melalui sebuah proses yang dikenal sebagai *routing*. Proses *routing* terjadi pada lapisan 3 (Lapisan jaringan seperti *Internet Protocol*) dari *stack* protokol tujuh-lapis OSI.

Router berfungsi sebagai penghubung antar dua atau lebih jaringan untuk meneruskan data dari satu jaringan ke jaringan lainnya. *Router* berbeda dengan *switch*. *Switch* merupakan penghubung beberapa alat untuk membentuk suatu *Local Area Network* (LAN).

Sebagai ilustrasi perbedaan fungsi dari *router* dan *switch* merupakan suatu jalanan, dan *router* merupakan penghubung antar jalan. Masing-

masing rumah berada pada jalan yang memiliki alamat dalam suatu urutan tertentu. Dengan cara yang sama, *switch* menghubungkan berbagai macam alat, dimana masing-masing alat memiliki alamat IP sendiri pada sebuah LAN.

Router sangat banyak digunakan dalam jaringan berbasis teknologi protokol TCP/IP, dan *router* jenis itu disebut juga dengan IP *Router*. Selain IP *Router*, ada lagi *AppleTalk Router*, dan masih ada beberapa jenis *router* lainnya. Internet merupakan contoh utama dari sebuah jaringan yang memiliki banyak *router* IP.



Gambar 2.7 *Wi-fi Router*

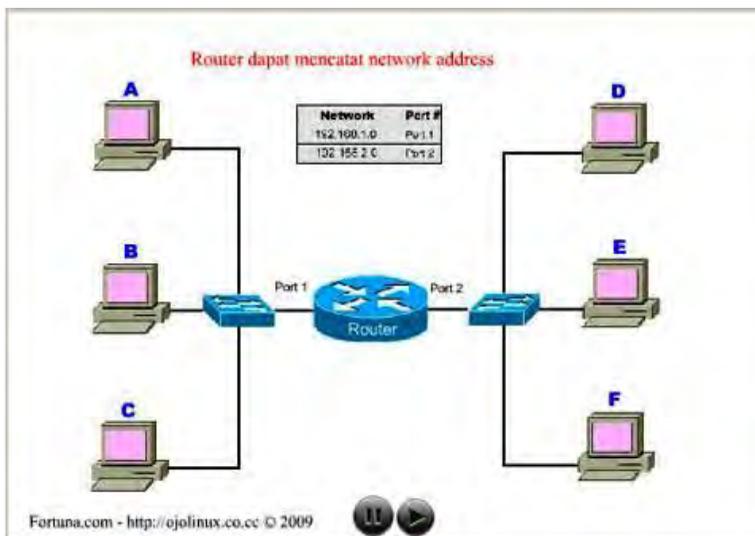
Router dapat digunakan untuk menghubungkan banyak jaringan kecil ke sebuah jaringan yang lebih besar, yang disebut dengan internetwork, atau untuk membagi sebuah jaringan besar ke dalam beberapa *subnetwork* untuk meningkatkan kinerja dan juga mempermudah manajemennya. *Router* juga kadang digunakan untuk mengoneksikan dua buah jaringan yang menggunakan media yang berbeda (seperti halnya *router wireless* yang pada umumnya selain ia dapat menghubungkan komputer dengan menggunakan radio, ia juga mendukung penghubungan komputer dengan kabel UTP), atau berbeda arsitektur jaringan, seperti halnya dari *Ethernet* ke *Token Ring*.

Router juga dapat digunakan untuk menghubungkan LAN ke sebuah layanan telekomunikasi seperti halnya telekomunikasi *leased line* atau *Digital Subscriber Line (DSL)*. *Router* yang digunakan untuk menghubungkan LAN ke sebuah koneksi *leased line* seperti T1, atau T3, sering disebut sebagai *access server*. Sementara itu, *router* yang digunakan untuk menghubungkan jaringan lokal ke sebuah koneksi DSL disebut juga dengan *DSL router*. *Router-router* jenis tersebut umumnya memiliki fungsi *firewall* untuk melakukan penapisan paket berdasarkan alamat sumber dan alamat tujuan paket tersebut, meski beberapa *router*

tidak memilikinya. *Router* yang memiliki fitur penapisan paket disebut juga dengan *packet-filtering router*. *Router* umumnya memblokir lalu lintas data yang dipancarkan secara *broadcast* sehingga dapat mencegah adanya *broadcast storm* yang mampu memperlambat kinerja jaringan.

Cara Kerja *Router*

Fungsi utama *router* adalah merutekan paket (informasi). Sebuah *router* memiliki kemampuan *Routing*, artinya *Router* secara cerdas dapat mengetahui ke mana rute perjalanan informasi (paket) akan dilewatkan, apakah ditujukan untuk *host* lain yang satu *network* ataukah berada di *network* yang berbeda.



Gambar 2.8 Ilustrasi Cara Kerja *Router*

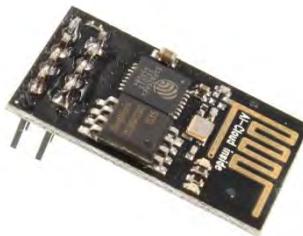
Pada gambar di atas terdapat 2 buah *network* yang terhubung dengan sebuah *router*. *Network* sebelah kiri yang terhubung ke *port 1 router* mempunyai alamat *network* 192.168.1.0 dan *network* sebelah kanan terhubung ke *port 2* dari *router* dengan *network address* 192.155.2.0

- Komputer A mengirim data ke komputer C, maka *router* tidak akan meneruskan data tersebut ke *network* lain.
- Begitu pula ketika komputer F mengirim data ke E, *router* tidak akan meneruskan paket data ke *network* lain.
- Barulah ketika komputer F mengirimkan data ke komputer B, maka *router* akan meneruskan paket data tersebut ke komputer B.

2.7 ESP8266

Internet of things semakin berkembang seiring dengan perkembangan mikrokontroler, modul yang berbasis *ethernet* maupun *wifi* modul yang dikenal dengan ESP8266. ESP8266 adalah *wifi* modul dengan *output* serial TTL yang dilengkapi dengan GPIO. Modul ini dapat dipergunakan secara tunggal maupun ditambahkan mikrokontroler sebagai pengendalinya.

Tegangan kerja ESP 8266 adalah sebesar 3,3 Volt, sehingga untuk penggunaan mikrokontroler tambahannya dapat menggunakan *board Arduino* yang memiliki tegangan 3,3 Volt, tetapi akan lebih baik jika membuat secara terpisah level *shifter* untuk komunikasi dan sumber tegangan untuk *wifi* modul ini.



Gambar 2.9 ESP 8266

2.8 Real Time Clock

RTC (*Real time clock*) adalah jam elektronik berupa *chip* yang dapat menghitung waktu (mulai detik hingga tahun) dengan akurat dan menjaga/menyimpan data waktu tersebut secara *real time*. Karena jam tersebut bekerja *real time*, maka setelah proses hitung waktu dilakukan *output* datanya langsung disimpan atau dikirim ke *device* lain melalui sistem antarmuka.

Chip RTC sering dijumpai pada *motherboard* PC (biasanya terletak dekat *chip* BIOS). Semua komputer menggunakan RTC karena berfungsi menyimpan informasi jam terkini dari komputer yang bersangkutan. RTC dilengkapi dengan baterai sebagai penyuplai daya pada *chip*, sehingga jam akan tetap *up-to-date* walaupun komputer dimatikan. RTC dinilai cukup akurat sebagai pewaktu (*timer*) karena menggunakan osilator kristal.



Gambar 2.10 RTC Module

BAB III PERANCANGAN DAN PEMBUATAN ALAT

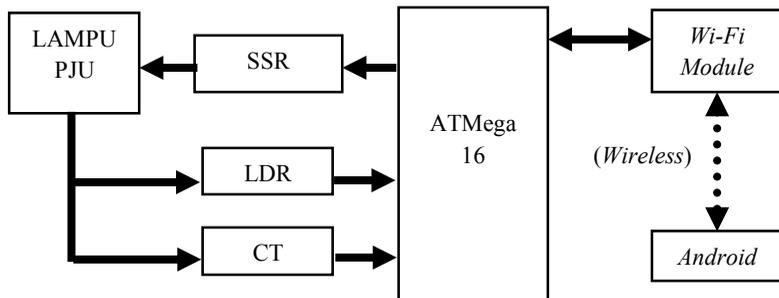
Perancangan alat pada bab ini terdiri dari perancangan perangkat keras (*hardware*) dan perancangan perangkat lunak (*software*).

Perancangan perangkat keras (*hardware*) terdiri dari perancangan elektronik yang meliputi rangkaian-rangkaian yang akan digunakan seperti halnya rangkaian LDR, rangkaian pengondisi sinyal dari trafo arus sebagai sensor arus, rangkaian komparator, skematik I/O mikrokontroler dari ATmega16, rangkaian pengendali SSR (*Solid State Relay*), pengkoneksian *Wi-Fi router* untuk komunikasi serta perancangan maket untuk simulasi lampu taman.

Untuk perancangan perangkat lunak (*software*) meliputi pemrograman mikrokontroler ATmega16, perancangan pemrograman untuk *Wi-Fi router*, dan perancangan tampilan pada komputer maupun perangkat Android.

5.1 Blok Fungsional Sistem

Pada bab ini dibahas mengenai perancangan alat secara keseluruhan. Dijelaskan pula lebih terinci tiap-tiap bagian atau blok-blok penyusun alat ini berupa blok fungsional sistem.



Gambar 3.1 Blok Fungsional

Pada gambar 3.1 merupakan gambar dari blok fungsional untuk perancangan sistem kontrol dan monitor PJU taman kota. Diagram blok tersebut memiliki tahapan sebagai berikut :

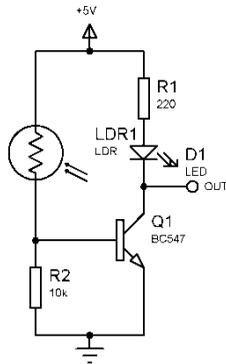
1. Lampu PJU sebagai objek yang akan kami monitor. Pada lampu PJU, saat bekerja terdapat dua hal yang kami monitor yaitu cahaya tampak yang dipancarkan oleh lampu PJU dan arus yang mengalir pada saluran lampu PJU tersebut.
2. LDR adalah komponen yang kami gunakan untuk keperluan sensor cahaya tampak. Pada blok ini terdapat sebuah rangkaian yang bisa membedakan kondisi saat lampu menyala dan mati untuk selanjutnya dikirim data ke kontroler ATmega16.
3. CT adalah *Current Transformer* atau trafo arus yang kami gunakan sebagai sensor arus yang dapat membedakan saluran pada lampu tersebut terbebani atau tidak dan selanjutnya data dikirim ke kontroler ATmega16.
4. Selanjutnya kontroler ATmega16 akan menerima data, mengolah data dan mengirim data tersebut melalui *Wi-Fi module* menuju pengguna yang menggunakan Android.
5. Bila terdapat gangguan pada salah satu keadaan yang dimonitor, maka kontroler akan mengirim sinyal ke pengguna sebagai pemberitahuan bahwa ada yang salah dengan objek yang dimonitor.
6. SSR atau *Solid State Relay* digunakan untuk mengganti sakelar manual yang biasa digunakan pada taman kota sehingga bisa dikontrol oleh pengguna.

5.2 Perancangan Perangkat Keras (*Hardware*)

Perancangan perangkat keras (*hardware*) pada Tugas Akhir ini terdiri dari perancangan elektronik yang meliputi rangkaian-rangkaian yang akan digunakan seperti halnya rangkaian LDR, rangkaian pengondisi sinyal dari trafo arus sebagai sensor arus, rangkaian komparator, skematik I/O mikrokontroler dari ATmega16, rangkaian pengendali SSR (*Solid State Relay*), pengoneksian *Wi-Fi module* untuk komunikasi serta perancangan maket untuk simulasi lampu taman.

3.2.1 Rangkaian LDR

Pada sub bab ini akan kami uraikan perancangan penggunaan LDR serta rangkaian yang mendukung kerja LDR sebagai sensor cahaya tampak untuk memonitor nyala lampu taman.



Gambar 3.2 Rangkaian Rangkaian LDR

Dalam gambar rangkaian di atas, menggunakan kegunaan transistor sebagai sakelar dengan pemicunya adalah LDR. Seperti yang diketahui bahwa sifat LDR saat terkena cahaya terang maka resistansinya menurun, dan sebaliknya saat cahaya semakin gelap maka resistansinya akan naik. Agar *transistor* aktif, diperlukan tegangan V_{be} yang lebih besar dari 0,6 yang diperoleh dari berikut.

$$V_{be} = \frac{R2}{LDR + R2} \times V_{cc}$$

Dalam keadaan terang, nilai resistansi LDR kecil hingga tegangan yang diperoleh R2 akan mendekati sumber. Tegangan tersebut *paralel* dengan kaki basis dan emitor dari *transistor* sehingga kebutuhan V_{be} transistor terpenuhi dan *transistor* bekerja. Saat transistor bekerja maka kolektor dan emitor akan terhubung sehingga arus akan mengalir dan dapat menyalakan LED. Keadaan OUT pada saat ini adalah bernilai 0 atau mendekati 0 volt. Dikarenakan saat OUT yang terhubung langsung dengan kolektor terhubung dengan emitor, dan pada emitor tidak terdapat resistansi apapun sehingga OUT secara tidak langsung akan terhubung dengan sisi *ground*.

$$V_{be} = \frac{10k\Omega}{5\Omega + 10k\Omega} \times 5V$$

$$V_{be} = 4,9V$$

Dalam keadaan gelap, nilai resistansi LDR bernilai besar sehingga terjadi pembagian tegangan antara LDR dengan R2. Saat nilai resistansi LDR lebih besar dari R2 maka tegangan yang diperoleh oleh LDR akan lebih besar dari pada R2. Tegangan pada R2 akan mempengaruhi V_{be} pada *transistor*. Saat tegangan R2 mengecil dan tidak memenuhi tegangan kerja V_{be} , maka hubungan kolektor dan emitor pada *transistor* akan terputus dan memutus arus yang mengalir pada LED sehingga LED akan padam. Keadaan OUT pada saat ini adalah bernilai 1 lebih besar dari 1 volt bahkan bisa mencapai 3,5 volt. Dikarenakan saat hubungan kolektor dan emitor terputus oleh *transistor*, maka hubungan OUT dengan *ground* juga putus, sehingga pada saat ini bila kita ukur V_{ce} (tegangan kolektor emitor) akan mendekati sumber. V_{ce} inilah yang akan menjadi tegangan OUT.

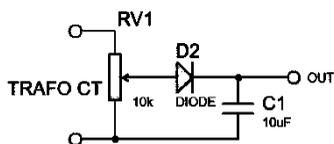
$$V_{be} = \frac{10k\Omega}{538k\Omega + 10k\Omega} \times 5V$$

$$V_{be} = 0.09V$$

3.2.2 Rangkaian Trafo Arus

Pada sub bab ini akan kami uraikan perancangan penggunaan Trafo Arus serta rangkaian yang mendukung kerja Trafo Arus sebagai sensor arus dan menunjukkan status terbebani atau tidak terbebani.

Dalam rangkaian ini, dipergunakan sebuah trafo arus sebagai sensor arus. Keluaran dari trafo arus ini adalah berupa arus bolak-balik (AC). Dalam teori kelistrikan, saat sebuah hambatan teraliri oleh arus, maka hambatan tersebut akan memiliki tegangan sesuai arus yang mengalir. Bila arusnya bernilai tetap maka semakin besar hambatannya semakin besar pula tegangan yang muncul, begitu pula sebaliknya bila nilai hambatannya tetap maka semakin besar arus yang mengalir maka semakin besar pula tegangan yang muncul. Karena itulah kami beri sebuah variabel resistor agar nilai hambatannya bisa sesuai dengan arus yang keluar nantinya.



Gambar 3.3 Rangkaian *Peak-Detector* Trafo Arus

Karena keluaran dari Trafo arus berupa arus bolak-balik maka tegangan dari V_r (variabel resistor) yang muncul adalah tegangan bolak-balik. Oleh karena itu kami beri rangkaian *Peak-Detector* setelah V_r tersebut. Gunanya adalah untuk mengetahui nilai puncak dari tegangan yang muncul pada variabel resistor agar dapat diolah dan berupa tegangan searah.

Dasar dari *Peak-detector* ini adalah sebuah penyearah setengah gelombang yang diwakili oleh sebuah dioda yang diberi filter sebuah kapasitor yang cukup untuk membantu mengurangi ripple yang terjadi pada saat proses penyearahan. Keadaan OUT pada rangkaian ini ada 2 keadaan yang pertama adalah keadaan terbebani dan keadaan tidak terbebani. Beban yang dimaksud adalah lampu PJU. Berikut adalah penjelasannya.

Perlu diketahui trafo arus yang kami dapat mempunyai perbandingan 1:2000. Jadi bila pada arus primer terdapat 5A, maka trafo tersebut akan menghasilkan 2.5mA pada sisi sekundernya.

$$I_{o\ CT} = \frac{I_i\ CT}{2000}$$

$$V_o = \frac{I_{o\ CT} \times VR}{2}$$

Dalam kondisi pertama yaitu keadaan terbebani, lampu PJU dalam kondisi menyala dan mendapat arus (beban lampu kami anggap 23 Watt). Trafo arus akan mendeteksi arus yang mengalir pada lampu dan mengirim sinyal keluaran berupa arus yang menandakan bahwa trafo arus tersebut telah mendeteksi adanya arus mengalir pada saluran itu. Arus keluaran trafo ini akan mengalir variabel *resistor* dan menimbulkan tegangan pada *resistor* ini. Tegangan keluaran dari *resistor* yang berupa tegangan bolak-balik akan disearahkan guna mendapatkan nilai puncak dari tegangan tersebut kemudian melewati proses filter oleh kapasitor agar hasilnya lebih baik. Keadaan OUT pada saat ini bernilai 1 atau lebih dari 1 volt.

$$I_{o\ CT} = \frac{\left(\frac{P_{lampu}}{V_{pln}}\right)}{2000}$$

$$I_{o\ CT} = \frac{\left(\frac{23}{220}\right)}{2000}$$

$$I_{o\ CT} = \frac{0,1045}{2000} A$$

$$I_{o\ CT} = 0.052\ mA$$

$$V_o = \frac{0,052\ mA \times 100k}{2} = 2,61V$$

Kondisi selanjutnya yaitu keadaan tanpa beban. Dalam hal ini lampu PJU dalam kondisi padam sehingga tidak ada arus yang mengalir. Tidak terdeteksinya arus oleh Trafo arus mengakibatkan trafo arus tidak mengeluarkan sinyal yang sebelumnya berupa arus sehingga variabel *resistor* tidak teraliri apapun. Keadaan OUT pada saat ini bernilai 0 atau 0 volt.

$$I_{o\ CT} = \frac{\left(\frac{P_{lampu}}{V_{pln}}\right)}{2000}$$

$$I_{o\ CT} = \frac{\left(\frac{0}{220}\right)}{2000}$$

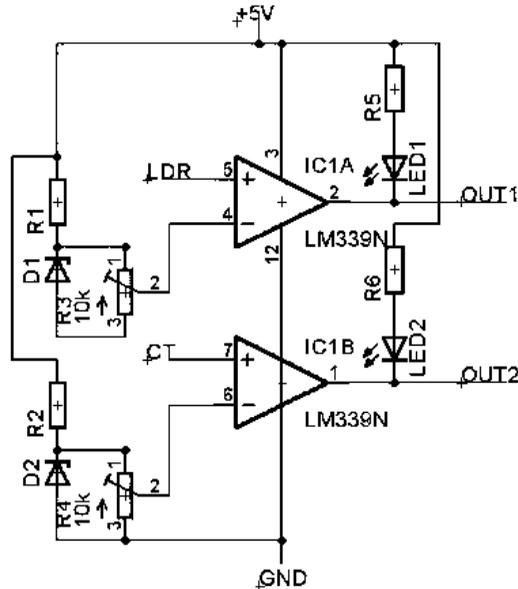
$$I_{o\ CT} = \frac{0}{2000} A$$

$$I_{o\ CT} = 0\ A$$

$$V_o = \frac{0 \times 100k}{2} = 0V$$

3.2.3 Rangkaian Komparator

Pada sub bab ini akan kami uraikan perancangan penggunaan rangkaian komparator sebagai rangkaian yang digunakan untuk menerima sinyal keluaran dari kedua sensor di atas sebelum masuk ke mikrokontroler ATmega16.



Gambar 3.4 Rangkaian Komparator dengan LM339

Dalam rangkaian ini, yang menjadi komponen utama ialah IC LM339 yang berisi empat gerbang komparator. IC ini bekerja dengan membandingkan tegangan *input* dari kaki *non-inverter* dan kaki *inverter*. Saat nilai tegangan dari kaki *non-inverting* lebih besar dari kaki *inverting* maka keluaran komparator adalah bernilai 1, dan sebaliknya bila nilai tegangan dari kaki *non-inverting* lebih kecil dari kaki *inverting* maka keluaran komparator adalah bernilai 0.

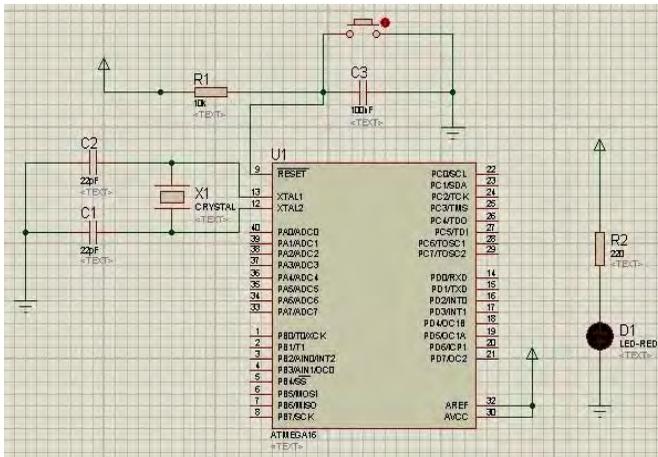
Pada kaki *non-inverting* (+) mendapat *input* tegangan dari masing-masing Rangkaian LDR dan Rangkaian trafo arus. Sebagai pembandingnya pada kaki *inverting* diberi tegangan dari rangkaian pembagi tegangan berupa resistor dan zener. Karena keluaran dari kedua

Rangkaian tidak sama, maka masing-masing juga memerlukan pembagi tegangan yang berbeda pula.

Pada pembagi tegangan keluarannya harus lebih rendah tetapi tidak 0 volt dari tegangan keluaran masing-masing Rangkaian. Misalnya bila keluaran dari Rangkaian LDR adalah 3 volt maka pembagi tegangan harus mengeluarkan 2 atau 2,5 volt. Dengan begitu komparator akan membandingkan 3 volt dan 2 volt tersebut dan memutuskan untuk mengeluarkan sinyal bernilai 1. Dan saat Rangkaian LDR mengeluarkan 1 volt dan pembagi tetap 2 volt, maka komparator akan membandingkan 1 volt dan 2 volt tersebut dan mengeluarkan keluaran bernilai 0.

3.2.4 Skematik Mikrokontroler ATmega16

Pada sub bab ini akan kami uraikan perancangan skematik untuk pin I/O (*input* dan *output*) yang kami gunakan dalam mikrokontroler ATmega16 dan bagaimana rangkaian yang kami buat.



Gambar 3.5 Skematik Mikrokontroler ATmega16

Pada gambar 3.5 skematik mikrokontroler ATmega16 dapat diketahui rancangan koneksi dari *port-port* yang akan dipakai seperti halnya untuk Port A 0 – 5 digunakan untuk masuknya sinyal yang

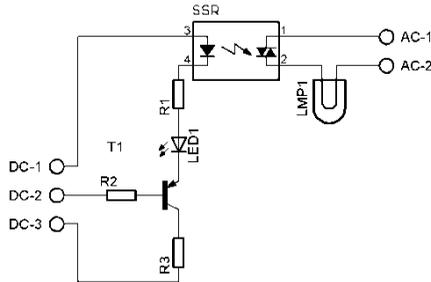
dikeluarkan oleh rangkaian trafo arus, kemudian *Port B* 0-5 digunakan untuk masuknya sinyal yang dikeluarkan oleh rangkaian LDR dan *Port C* 0 – 6 digunakan untuk mengontrol sebuah SSR guna mengendalikan penyalan lampu PJU yang kami gunakan. Untuk *Port D* 0 dan 1 (atau RX dan TX) digunakan untuk komunikasi dengan *Wi-Fi router*. Berikut adalah rincian *port-port* yang digunakan dalam bentuk tabel.

Tabel 3.1 Pin Mikrokontroler ATmega16 Yang Digunakan

	PIN	FUNGSI
1	PB0	Rangkaian LDR Lampu 1
2	PB1	Rangkaian LDR Lampu 2
3	PB2	Rangkaian LDR Lampu 3
4	PB3	Rangkaian LDR Lampu 4
5	PB4	Rangkaian LDR Lampu 5
6	PB5	Rangkaian LDR Lampu 6
9	<i>RESET</i>	Tombol <i>Reset</i>
12	XTAL2	<i>Ground</i>
13	XTAL1	<i>Ground</i>
14	PD0	<i>TX WiFi Router</i>
15	PD1	<i>RX WiFi Router</i>
22	PC0	Pengendali SSR Lampu 1
23	PC1	Pengendali SSR Lampu 2
24	PC2	Pengendali SSR Lampu 3
25	PC3	Pengendali SSR Lampu 4
26	PC4	Pengendali SSR Lampu 5
27	PC5	Pengendali SSR Lampu 6
35	PA5	Rangkaian Trafo Arus Lampu 6
36	PA4	Rangkaian Trafo Arus Lampu 5
37	PA3	Rangkaian Trafo Arus Lampu 4
38	PA2	Rangkaian Trafo Arus Lampu 3
39	PA1	Rangkaian Trafo Arus Lampu 2
40	PA0	Rangkaian Trafo Arus Lampu 1

3.2.5 Rangkaian Pengendali SSR

Pada sub bab ini akan kami uraikan perancangan rangkaian pengendali SSR agar kerja dari SSR sesuai yang kita harapkan.



Gambar 3.6 Rangkaian Pengendali SSR

Berikut adalah rincian dari pin pada gambar di atas.

Tabel 3.2 Pin Pengendali SSR

No.	Pin	Fungsi
1	DC-1	+5V DC
2	DC-2	Trigger dari Mikrokontroler
3	DC-3	GND
4	AC-1	Fasa 220 V AC
5	AC-2	Netral 220 V AC

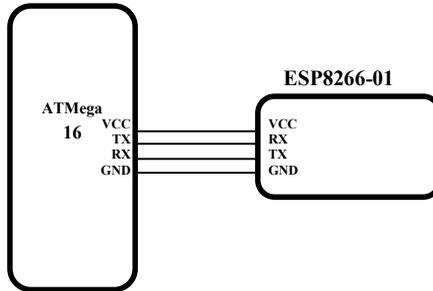
Dalam gambar 3.6 di atas, dengan memanfaatkan fungsi saklar elektronik menggunakan *transistor* PNP maka SSR akan aktif saat DC-2 diberi tegangan 0 volt. Berikut rincian kerja dari rangkaian di atas.

Saat pin DC-2 tidak dihubungkan atau diberi tegangan 5V, maka arus dari kaki emitor tidak bisa mengalir menuju basis karena tegangan di kaki emitor juga 5V dan bila tidak ada beda potensial (tegangan) maka arus tidak mengalir, sehingga SSR tidak aktif dan lampu padam.

Saat pin DC-2 diberi tegangan 0V atau bisa dikatakan diberi *ground*, maka arus dari kaki emitor akan mengalir menuju basis dan pada transistor akan terpenuhi tegangan V_{be} dengan demikian *transistor* dalam keadaan aktif dan emitor-pun terhubung dengan kaki kolektor, sehingga SSR menjadi aktif dan lampu akan menyala.

3.2.6 Pengoneksian *Wi-Fi Module*

Pada sub bab ini akan kami uraikan perancangan pengoneksian dari *Wi-Fi module* dengan Mikrokontroler ATmega16 sehingga perangkat *Wi-Fi* bisa digunakan sebagai alat komunikasi.



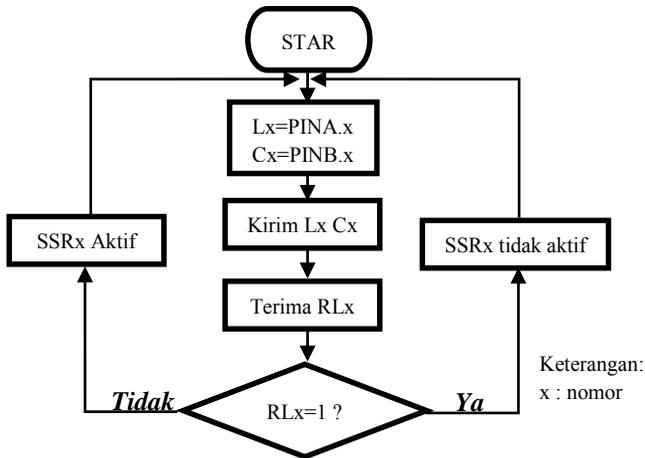
Gambar 3.7 Koneksi Mikrokontroler Dengan *Wi-Fi Module*

Dalam gambar 3.7 di atas, sama halnya dengan *modul, Wi-Fi Module* ESP8266 hanya perlu terhubung secara serial yakni dengan menyambungkan pin TX dan RX saja untuk bisa berkomunikasi dengan ATmega16 atau mikrokontroler lainnya. Efek dari mudahnya merangkai modul, maka untuk *software* dalam ESP8266 ini cukup rumit sehingga butuh banyak referensi dalam pemrogramannya.

5.3 Perancangan Perangkat Lunak (*Software*)

3.3.1 Pemrograman ATmega16

Pada sub bab ini akan kami uraikan perancangan pemrograman yang kami lakukan pada perangkat mikrokontroler ATmega16 sehingga bisa bekerja sebagai kontroler pada sistem kontrol dan monitor lampu taman kota yang kami lakukan.

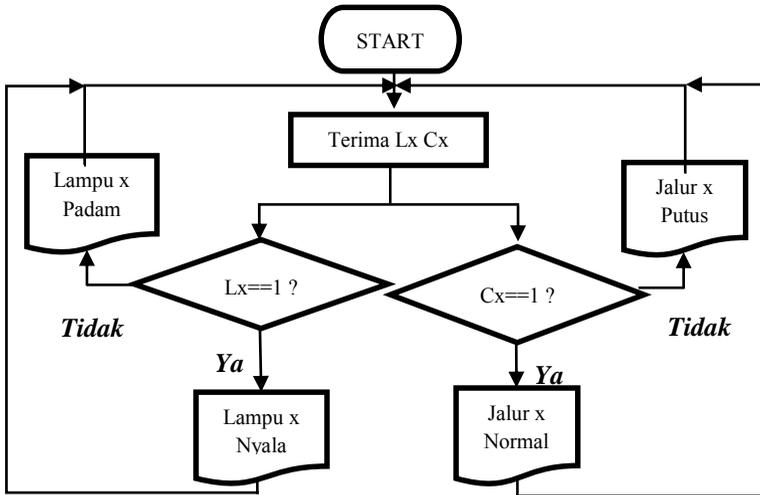


Gambar 3.8 Flowchart Diagram Pemrograman ATmega16

Sebenarnya pemrograman dari mikrokontroler yang kami buat cukup sederhana, hanya perlu menerima sinyal masukan dari Rangkaian sensor dan mengirimkannya melalui pin TX dan menerima perintah dari pin RX yang pin-pin tersebut terhubung menuju *Wi-Fi module*.

3.3.2 Pemrograman Koneksi *Wi-Fi Module*

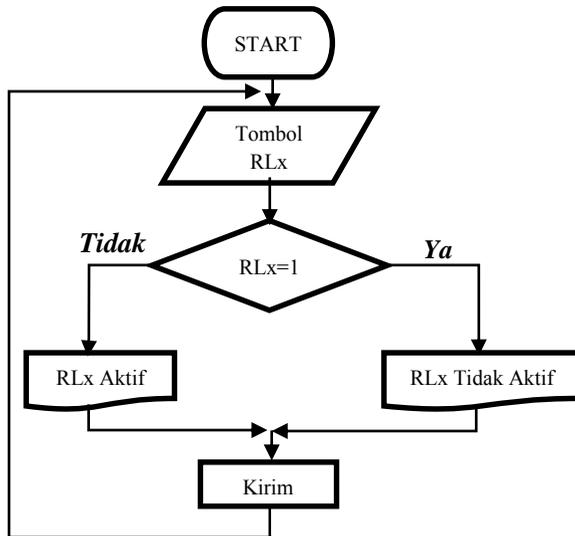
Pada sub bab ini akan kami uraikan perancangan pemrograman pada koneksi *Wi-Fi module* yang kami pakai sehingga bisa digunakan sebagai alat komunikasi antara kontroler dan pengguna.



Gambar 3.9 Flowchart Pengambilan Data pada *Wi-Fi Module*

Pada *Wi-Fi module* ini pemrogramannya sedikit rumit, karena yang memerlukan kemampuan untuk mengetahui beberapa bahasa program seperti halnya bahasa *lua*, *html*, *php*, dan *javascript*. Berikut alur kerja dari program *Wi-Fi module*.

Yang kali pertama dilakukan adalah pengecekan hubungan antara *module* dengan mikrokontroler. Bila sudah terdeteksi maka program inti bisa dilakukan. Data dari mikrokontroler diterima dan ditampilkan pada halaman *web* untuk menunjukkan status dari masing-masing sensor. Pada halaman *web* juga tersedia perintah untuk menyalakan dan mematikan lampu PJU, dengan mengirim sinyal data tertentu akan memicu mikrokontroler untuk menjalankan program untuk *driver SSR*. Dan berikut adalah *flowchart* untuk perintah *driver SSR*.



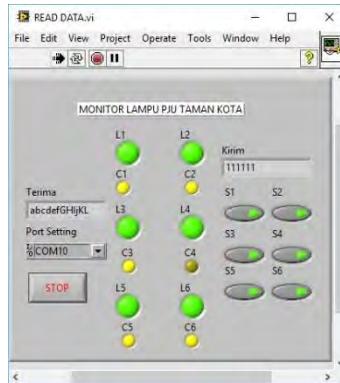
Gambar 3.10 *Flowchart* Perintah Pengendali SSR

3.3.3 Tampilan pada Komputer dan Android

Pada sub bab ini akan kami uraikan perancangan tampilan yang kami gunakan untuk pengujian pada komputer serta pada android.

Rancangan tampilan ini terbagi menjadi dua proses, pertama dengan komputer dengan komunikasi saja tanpa *Wi-Fi module* untuk proses pengujian alat. Selanjutnya menggunakan halaman *web* yang kemudian dikonversikan menjadi aplikasi android.

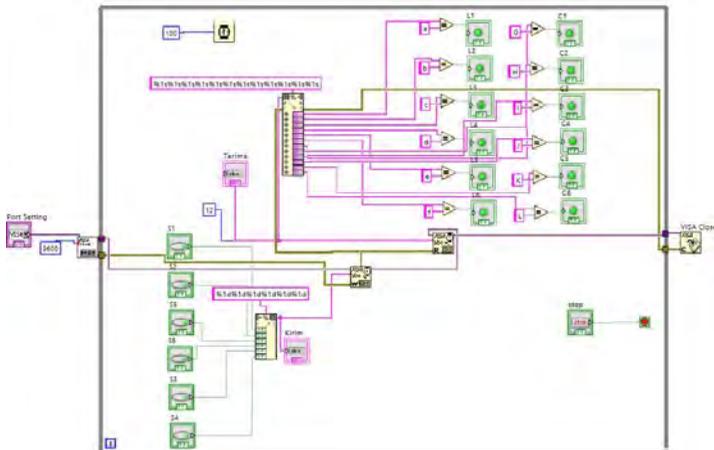
Untuk pengujian pada komputer kami memakai program aplikasi *LabVIEW* sebagai media antarmuka alat kami. Dengan tampilan sederhana dan fungsi yang cukup *LabVIEW* kami jadikan sebagai aplikasi untuk menguji sistem dari rangkaian kami bekerja dengan baik atau tidak. Berikut adalah halaman antarmuka pada *LabVIEW*.



Gambar 3.11 Tampilan pada Komputer Menggunakan *LabVIEW*

Pada gambar 3.11 terdapat 6 indikator hijau sebagai indikator dari sensor cahaya dan 6 indikator kuning sebagai indikator dari sensor arus. Terdapat 6 buah tombol *on/off* sebagai pengendali SSR dan tombol *STOP* untuk menghentikan program. *Port Setting* digunakan untuk memilih *Port COM* pada PC sesuai dengan *COM* serial yang digunakan.

Untuk blok diagram dari *LabVIEW* bisa dilihat pada gambar di bawah ini.



Gambar 3.12 Blok Diagram *LabVIEW*

Berikutnya adalah tampilan pada aplikasi yang diakses menggunakan android.



Gambar 3.13 Tampilan Aplikasi Android

Gambar 3.13 adalah sebuah tampilan aplikasi android yang dibuat menggunakan *App Inventor* yang merupakan sebuah layanan pembuatan aplikasi android gratis yang mudah dalam penggunaannya.

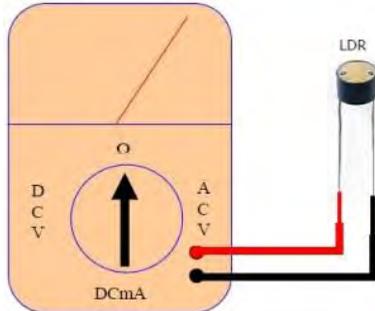
BAB IV HASIL PENGUJIAN DAN ANALISA ALAT

Pada bab ini akan dibahas mengenai pengujian dan analisa alat dari sistem kontrol dan monitor lampu PJU yang telah kami buat. Pengujian dan analisa ini ditujukan untuk memastikan peralatan yang dibuat dapat berfungsi dengan baik sesuai yang telah direncanakan.

Pengujian sistem kontrol dan monitor lampu PJU ini meliputi pengujian instrumen LDR, pengujian instrumen trafo arus, pengujian pada komparator, pengujian data pada mikrokontroler, dan pengujian *Wi-Fi module* yang tersambung dengan mikrokontroler. Setelah melakukan beberapa pengujian alat, data yang diperoleh akan dianalisa untuk mengetahui proses kerja dari seluruh sistem alat yang dibuat.

4.1 Pengujian Instrumen LDR

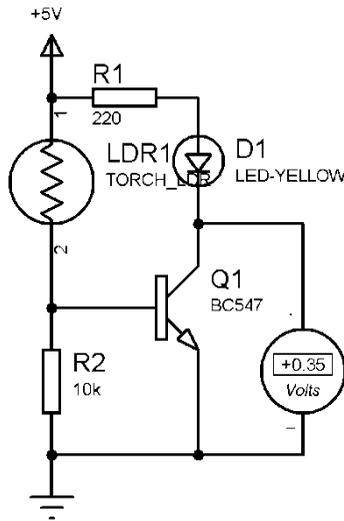
Pengujian instrumen LDR dilakukan dua tahap. Pertama pengukuran nilai LDR saat terang, redup dan gelap. Selanjutnya dengan mengukur tegangan keluaran dari beberapa kondisi yang akan dilakukan sesuai keadaan yang mungkin terjadi di lapangan sesungguhnya.



Gambar 4.1 Pengujian Komponen LDR

Tabel 4.1 Hasil Pengujian LDR

No.	Keadaan	Tahanan Keluaran
1	Terang lampu	3 ohm
2	Dalam ruangan	110 ohm
3	Gelap	20500 ohm



Gambar 4.2 Pengujian Instrumen LDR

Dan berikut adalah hasil dari pengujian dan pengukuran yang terjadi.

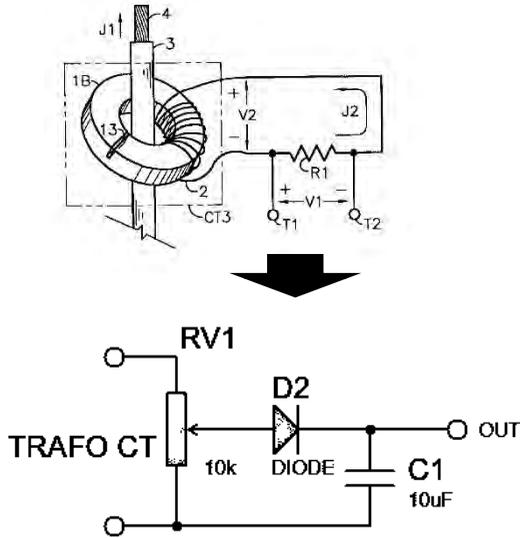
Tabel 4.2 Hasil pengujian Instrumen LDR

No.	Kedaaan	Tegangan Keluaran
1	Terang lampu	0,34 volt
2	Dalam ruangan	1,22 volt
3	Gelap	3,58 volt

Kedaaan dalam ruangan menghasilkan tegangan keluaran yang kecil karena pada kedaaan dalam ruangan terdapat cahaya yang masuk melalui jendela ruangan maupun lampu ruangan yang yang masih menyala sehingga mempengaruhi nilai LDR.

4.2 Pengujian Instrumen Trafo Arus

Pengujian instrumen trafo arus ini dilakukan dengan melilitkan 1 fasa pada gulungan trafo dan mengukur arus yang mengalir pada fasa tersebut serta mengukur keluaran dari instrumen trafo arus ini.



Gambar 4.3 Pengukuran Instrumen Trafo Arus

Pada gambar di atas, $R1$ pada gambar bagian atas sama dengan $RV1$ pada gambar bagian bawah. Dan berikut adalah hasil pengukuran OUT .

Tabel 4.3 Hasil pengukuran instrumen trafo arus

No.	Beban	Arus	Out
1	- Watt	0 A	0 Volt
2	23 Watt	0,104 A	1,04 Volt
3	50 Watt	0,227 A	2,27 Volt

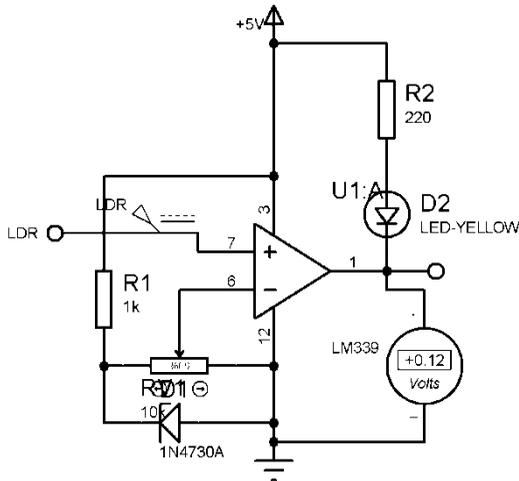
Diberi tiga macam beban karena dimisalkan dalam sebuah proyek kota telah diperhitungkan daya lampu sebesar 23 watt. Apabila ada

kesalahan pembelian barang akan terdeteksi apabila daya lampu lebih kecil ataupun lebih besar.

Hasil keluaran tersebut dihasilkan dari keluaran trafo arus yang per rasio 2000:1, jadi keluaran arus pada 23 Watt sekitar 0,054 mA dan pada 50 Watt sekitar 0,117 mA. Dengan memasang resistor sebesar 10K Ohm maka pada resistor tersebut akan muncul tegangan sebesar nilai tahanan dikalikan arus yang mengalir.

4.3 Pengujian Komparator

Pengujian komparator tidak terlepas dari pengujian kedua sensor sebelumnya dan pengujian pada pembagi tegangan. Maka dari itu pengujian akan terbagi dua dan masing-masing adalah untuk instrumen LDR dan untuk trafo arus.

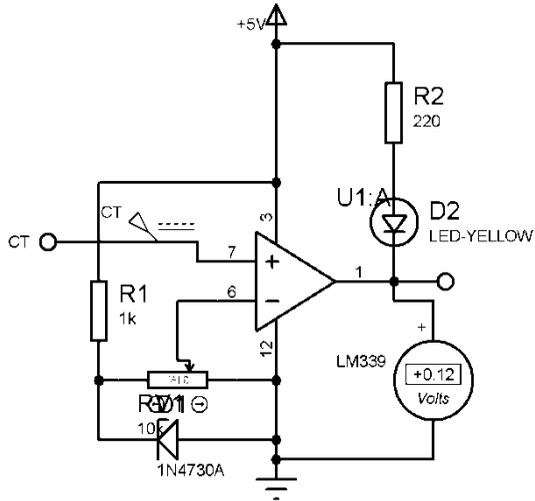


Gambar 4.4 Pengujian Komparator Pada Instrumen LDR

Berdasarkan hasil pengujian instrumen LDR, maka diputuskan untuk mengatur pembagi tegangan dengan nilai 1,1 volt. Kami beranggapan bahwa saat lampu PJU padam maka penerangan sekitar akan seperti keadaan dalam ruangan, dan berikut adalah hasil pengujian komparator di atas.

Tabel 4.4 Hasil Pengujian Komparator 1

No.	Keadaan	LED	Tegangan OUT
1	Lampu pijar Nyala	Nyala	0 volt
2	Lampu pijar Padam	Padam	3,58 Volt



Gambar 4.5 Pengujian Komparator pada Instrumen Trafo Arus

Berikutnya untuk pengujian kedua, berdasarkan pengujian instrumen trafo arus sebelumnya didapat saat menggunakan beban 23 Watt keluarannya bernilai 1,04 volt. Maka diputuskan untuk mengatur pembagi tegangan dengan nilai 0,9 volt. Dan berikut adalah hasil dari pengujian kedua.

Tabel 4.5 Hasil pengujian komparator 2

No.	Keadaan	LED	Tegangan OUT
1	Lampu pijar Nyala	Padam	3,58volt
2	Lampu pijar Padam	Nyala	0 Volt

4.4 Pengujian Pengendali SSR



Gambar 4.6 Modul SSR 1 *channel*

Pada modul ini terdapat *driver* seperti yang ada pada perencanaan bab III, dan berikut adalah hasil dari pengujian modul tersebut.

Tabel 4.6 Hasil Pengujian Modul SSR

No.	Tegangan Ch	LED DC	Lampu AC
1	5 V	Padam	Padam
2	Tak terhubung	Padam	Padam
3	GND	Nyala	Nyala
4	Tak terhubung	Padam	Nyala
5	5V	Padam	Padam

Dari hasil data yang diperoleh, SSR bekerja dengan baik apabila pin CH diberi tegangan yang sesuai dan sedikit terganggu bila tidak terhubung dengan tegangan.

4.5 Pengujian Data Mikrokontroler ATmega16

Pengujian data pada mikrokontroler ATmega16 ini akan dibagi menjadi 2, yaitu uji pengambilan data dan uji pengiriman data terhadap alat yang telah kami buat, bukan pada *Wi-Fi module*.

Tabel 4.7 Pengambilan Data Dari Rangkaian LDR

Lampu PJU						Rang. LDR						Data pada PORTB
1	2	3	4	5	6	1	2	3	4	5	6	
1	1	1	1	1	1	0	0	0	0	0	0	0
0	1	1	1	1	1	1	0	0	0	0	0	1
1	0	1	1	1	1	0	1	0	0	0	0	2
1	1	0	1	1	1	0	0	1	0	0	0	4
1	1	1	0	1	1	0	0	0	1	0	0	8
1	1	1	1	0	1	0	0	0	0	1	0	16
1	1	1	1	1	0	0	0	0	0	0	1	32
1	0	1	1	0	1	0	1	0	0	1	0	18
0	1	0	1	0	0	1	0	1	0	1	1	53
1	0	1	0	0	0	0	1	0	1	1	1	58
0	1	0	0	0	0	1	0	1	1	1	1	61
0	0	0	0	0	0	1	1	1	1	1	1	63

Saat lampu PJU menyala (1) rangkaian LDR akan mengirim data (0) sebagai tanda bahwa lampu normal. Dan saat lampu PJU padam (0) rangkaian LDR akan mengirim data (1) sebagai tanda bahwa lampu sedang bermasalah.

Tabel 4.8 Pengambilan Data Dari Rangkaian Trafo Arus

Lampu PJU						Rang. LDR						Data pada PORTA
1	2	3	4	5	6	1	2	3	4	5	6	
1	1	1	1	1	1	1	1	1	1	1	1	63
0	1	1	1	1	1	0	1	1	1	1	1	62
1	0	1	1	1	1	1	0	1	1	1	1	61
1	1	0	1	1	1	1	1	0	1	1	1	59
1	1	1	0	1	1	1	1	1	0	1	1	55
1	1	1	1	0	1	1	1	1	1	0	1	47
1	1	1	1	1	0	1	1	1	1	1	0	31
1	0	1	1	0	1	1	0	1	1	0	1	45
0	1	0	1	0	0	0	1	0	1	0	0	10
1	0	1	0	0	0	1	0	1	0	0	0	5
0	1	0	0	0	0	0	1	0	0	0	0	2
0	0	0	0	0	0	0	0	0	0	0	0	0

Saat lampu PJU menyala (1) rangkaian trafo arus akan mengirim data (1) sebagai tanda pada saluran tersebut terdapat arus yang mengalir. Dan saat lampu PJU padam (0) rangkaian trafo arus akan mengirim data (0) sebagai tanda pada saluran tersebut tidak ada arus yang mengalir (bermasalah).

Tabel 4.9 Pengiriman Data Menuju Pengendali SSR

Data pada PORTB	SSR					
	1	2	3	4	5	6
0	1	1	1	1	1	1
1	0	1	1	1	1	1
2	1	0	1	1	1	1
4	1	1	0	1	1	1
8	1	1	1	0	1	1
16	1	1	1	1	0	1
32	1	1	1	1	1	0
18	1	0	1	1	0	1
53	0	1	0	1	0	0
58	1	0	1	0	0	0
61	0	1	0	0	0	0
63	0	0	0	0	0	0

Karena pengendali SSR mempunyai sifat aktif saat diberi 0, maka saat mikrokontroler mengeluarkan data 0 (dalam biner 000000) maka seluruh SSR akan aktif. Begitu juga sebaliknya bila mikrokontroler mengeluarkan data 63 (dalam biner 111111) maka seluruh SSR akan padam.

4.6 Pengujian *Wi-Fi Module*

Pengujian *Wi-Fi Module* ini akan menguji kelangsungan lamanya dan lancarnya sebuah perangkat android terhubung dengan modul yang kami miliki. Dan berikut adalah hasil dari beberapa tipe android yang kami coba untuk terhubung dengan modul *Wi-Fi* kami. Modul *Wi-Fi* yang kami gunakan adalah ESP8266-01.

Tabel 4.10 Pengujian *Wi-Fi Module* ESP8266-01 dengan Android

No.	Merek Android	Hasil
1	Andromax Ec	Tidak bisa terhubung
2	Xiaomi RedMi2	Terhubung 3 detik lalu putus 10 detik
3	ASUS ZenFon	Terhubung 30 detik lalu putus 10 detik
4	ASUS Lenovo A7000	Terhubung lancar

Dengan hasil pengujian seperti di atas, maka modul *Wi-Fi* milik kami memiliki kelemahan yaitu sinyal yang tidak stabil. Terbukti kekuatan sinyal yang terbaca pada beberapa merek android berbeda-beda sehingga perangkat android tidak semuanya bisa terhubung dengan lancar sebagaimana fungsi *Wi-Fi* semestinya.

4.7 Pengujian Sistem Secara Keseluruhan

Pengujian sistem secara keseluruhan dilakukan dengan cara merangkai seluruh rangkaian menjadi satu dan menghubungkannya dengan sebuah komputer. Pada komputer terdapat sebuah antarmuka yang telah disiapkan sebelumnya pada aplikasi *LabVIEW* sebagai antarmuka sistem pada alat kami.

Berikut adalah hasil dari pengujian sistem menggunakan *LabVIEW*.

Tabel 4.11 Pengujian Sistem Monitor Menggunakan *LabVIEW*

No.	Objek	Kondisi	Indikator <i>LabVIEW</i>		Persentase	
			Cahaya	Arus	Cahaya	Arus
1	L1	Nyala	<i>on</i>	<i>on</i>	100%	100%
		Mati	<i>off</i>	<i>off</i>		
2	L2	Nyala	<i>on</i>	<i>on</i>	100%	100%
		Mati	<i>off</i>	<i>off</i>		
3	L3	Nyala	<i>on</i>	<i>on</i>	100%	100%
		Mati	<i>off</i>	<i>off</i>		
4	L4	Nyala	<i>on</i>	<i>off</i>	100%	0%
		Mati	<i>off</i>	<i>off</i>		
5	L5	Nyala	<i>on</i>	<i>on</i>	100%	100%
		Mati	<i>off</i>	<i>off</i>		
6	L6	Nyala	<i>on</i>	<i>on</i>	100%	100%
		Mati	<i>off</i>	<i>off</i>		

Tabel 4.12 Pengujian Sistem Kontrol Menggunakan *LabVIEW*

No.	Tombol <i>LabVIEW</i>	Kondisi	Keadaan Objek	Persentase
1	S1	<i>on</i>	Nyala	30%
		<i>off</i>	Berkedip	
2	S2	<i>on</i>	Nyala	100%
		<i>off</i>	Mati	
3	S3	<i>on</i>	Nyala	100%
		<i>off</i>	Mati	
4	S4	<i>on</i>	Nyala	0%
		<i>off</i>	Nyala	
5	S5	<i>on</i>	Nyala	100%
		<i>off</i>	Mati	
6	S6	<i>on</i>	Nyala	100%
		<i>off</i>	Mati	

Selanjutnya adalah pengujian menggunakan aplikasi android yang telah dibuat sebelumnya dan ini merupakan pengujian sistem secara keseluruhan. Merek android yang kami gunakan adalah ASUS Zenfon.

Tabel 4.13 Pengujian Sistem Kontrol Keseluruhan

No.	Tombol	Kondisi	Keadaan Objek	Persentase
1	Lampu 1	<i>on</i>	Nyala	100%
		<i>off</i>	Mati	
2	Lampu 2	<i>on</i>	Nyala	95%
		<i>off</i>	Mati	
3	Lampu 3	<i>on</i>	Nyala	98%
		<i>off</i>	Mati	
4	Lampu 4	<i>on</i>	Nyala	100%
		<i>off</i>	Mati	
5	Lampu 5	<i>on</i>	Nyala	100%
		<i>off</i>	Mati	
6	Lampu 6	<i>on</i>	Nyala	95%
		<i>off</i>	Mati	

Tabel 4.14 Pengujian Sistem Monitor Keseluruhan

No.	Objek	Kondisi	Indikator <i>LabVIEW</i>		Persentase	
			Cahaya	Arus	Cahaya	Arus
1	L1	Nyala	<i>on</i>	<i>on</i>	80%	80%
		Mati	<i>off</i>	<i>off</i>		
2	L2	Nyala	<i>on</i>	<i>on</i>	85%	80%
		Mati	<i>off</i>	<i>off</i>		
3	L3	Nyala	<i>on</i>	<i>on</i>	85%	75%
		Mati	<i>off</i>	<i>off</i>		
4	L4	Nyala	<i>on</i>	<i>off</i>	90%	75%
		Mati	<i>off</i>	<i>off</i>		
5	L5	Nyala	<i>on</i>	<i>on</i>	80%	80%
		Mati	<i>off</i>	<i>off</i>		
6	L6	Nyala	<i>on</i>	<i>on</i>	95%	90%
		Mati	<i>off</i>	<i>off</i>		

Dengan hasil pengujian ini, maka secara keseluruhan sistem dapat bekerja dengan baik. Dan selanjutnya pengujian fungsi *timer* untuk pengaturan waktu penyalan dan pemadaman secara otomatis.

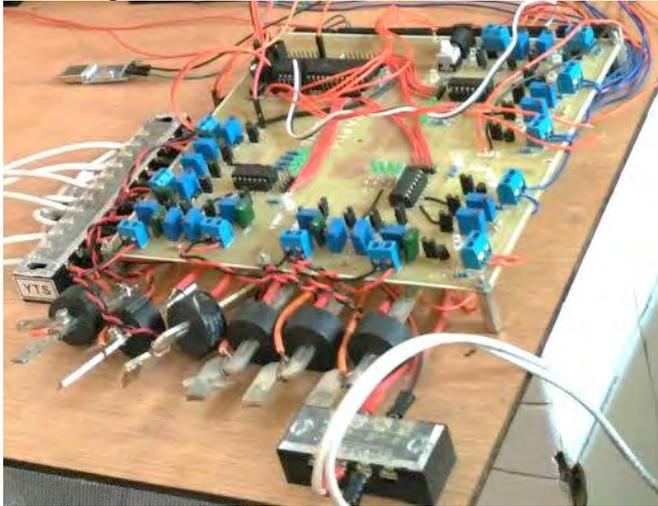
Tabel 4.15 Pengujian Waktu Penyalan dan Pemadaman Otomatis

No.	Set Waktu	Hasil
1	Nyala (10.00)	Nyala (10.00)
	Padam (10.01)	Padam (10.02)
2	Nyala (10.15)	Nyala (10.15)
	Padam (10.17)	Padam (10.17)
3	Nyala (10.20)	Nyala (10.20)
	Padam (10.21)	Padam (10.21)
4	Nyala (10.25)	Nyala (10.26)
	Padam (10.27)	Padam (10.27)

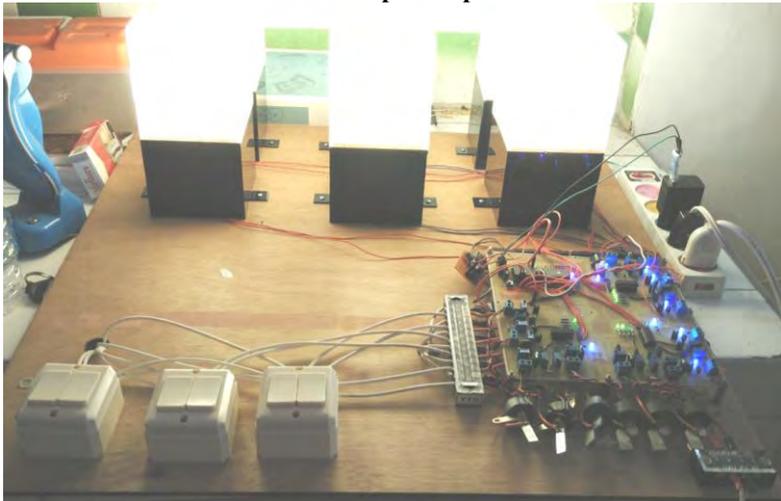
Dengan hasil pengujian di atas, perbedaan waktu antara waktu set dan hasil dikarenakan kelancaran data terkirim dari android menuju perangkat terganggu oleh sinyal. Bisa dilihat pada hasil pengujian *Wi-Fi Module*.

LAMPIRAN A FOTO

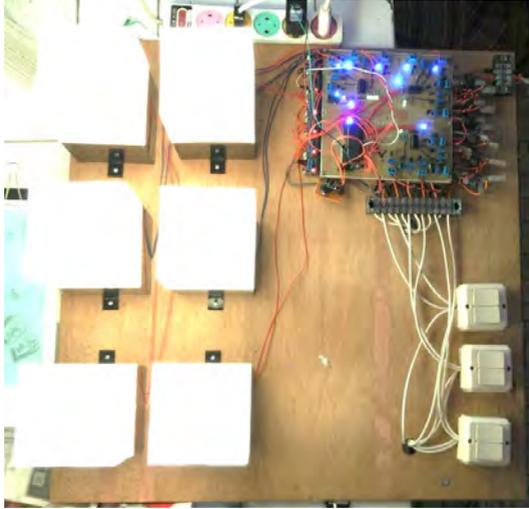
1. Foto Rangkaian Kendali dan Monitor



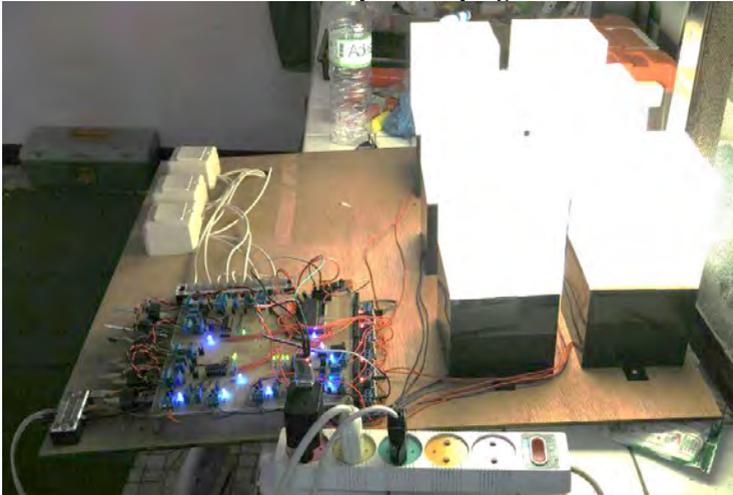
2. Foto Keseluruhan Alat Tampak Depan



3. Foto Keseluruhan Alat Tampak Atas



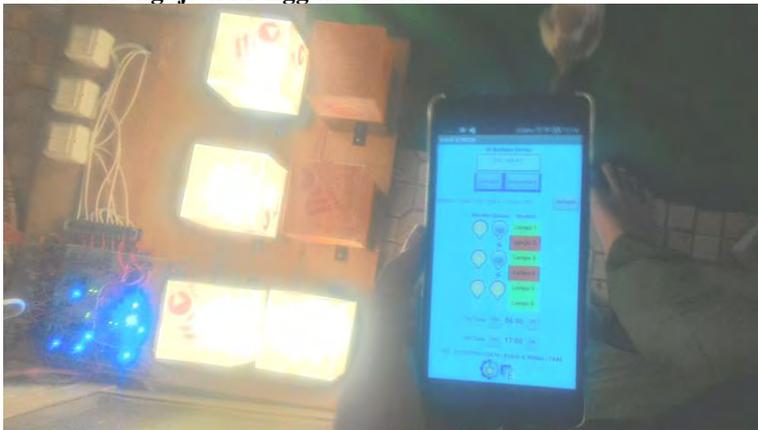
4. Foto Keseluruhan Alat Tampak Samping



5. Foto Pengujian Menggunakan *LabVIEW*



6. Foto Pengujian Menggunakan *Android*



LAMPIRAN B PROGRAM

A. Program ATmega16

```
#include <stdio.h>
#include <stdlib.h>
#include <avr/io.h>
#include <util/delay.h>
#include <avr/interrupt.h>
#include <avr/pgmspace.h>
#include "twi.h"
#include "rtc.h"

#define F_CPU
    12000000
#define BAUD
    9600
#define BAUDRATE
    ((F_CPU)/(BAUD*16UL)-1)

#define ON 1
#define OFF 0
#define delay    _delay_ms
#define print    uart_puts

const uint8_t daysinmonth [] PROGMEM =
{ 31,28,31,30,31,30,31,31,30,31,30,31 };
unsigned char buffer[20];
char RxBuff[200], buff[100];
int RxComplete, RxEn, index, meniton, menitoff,
jamon, jamoff;
int detik, menit, jam, hari, tanggal, bulan,
tahun;

uint8_t readRTC(uint8_t *year, uint8_t *month,
uint8_t *date, uint8_t *day, uint8_t *hour,
uint8_t *min, uint8_t *sec);
void setRTC(uint8_t year, uint8_t month, uint8_t
date, uint8_t hour, uint8_t min, uint8_t sec);
```

```

void uart_putc(const uint8_t c)
{
    while (!(UCSRA & (1<<UDRE))); // wait
until ready to send
    UDR = c; // send
    karakter
}

void uart_puts (const char *s)
{
    do
    {
        uart_putc (*s); //send one
        char of string
    }
    while (*s++); //next char
of string until end of string sign
}

void println(char str[])
{
    print(str);
    print("\n\r");
}

void uart_puti(const uint16_t var)
{
    char cache[7];
    uart_puts(utoa(var, cache, 10));
}

void uart_init()
{
    UBRRH = (BAUDRATE>>8);
    UBRRL = BAUDRATE;
    UCSRB|= (1<<TXEN) | (1<<RXEN) | (1<<RXCIE);
    UCSRC|= (1<<URSEL) | (1<<UCSZ0) | (1<<UCSZ1);
}

```

```

void portC(int pin, int state)
{
    if(state == 1) PORTC |= (1<<pin);
    else PORTC &= ~(1<<pin);
}

int pinB(int pin)
{
    if(PINB & (1<<pin)) return 1;
    else return 0;
}
int pinA(int pin)
{
    if(PINA & (1<<pin)) return 1;
    else return 0;
}

ISR(USART_RXC_vect)
{
    if(UDR != '\n' && UDR != '\r')
    {
        uart_putc(UDR);
        RxBuff[index] = UDR;
        index++;
    }
    else
    {
        //uart_putc(RxBuff[2]);

        if(RxBuff[2] == '#')
        {
            jamon = ((RxBuff[3]-'0') * 10)
+ (RxBuff[4]-'0');
            meniton = ((RxBuff[5]-'0') *
10) + (RxBuff[6]-'0');

            sprintf(buff, "Waktu ON
= %d:%d\n\r", jamon, meniton);
            print(buff);
        }
    }
}

```

```

        if(RxBuff[2] == '*')
        {
            jamoff = ((RxBuff[3]-'0') *
10) + (RxBuff[4]-'0');
            menitoff = ((RxBuff[5]-'0') *
10) + (RxBuff[6]-'0');

            sprintf(buff, "Waktu OFF
= %d:%d\n\r", jamoff, menitoff);
            print(buff);
        }

        if(RxBuff[3] == 'N')
        {
            switch(RxBuff[4])
            {
                case '1' :
                    portC(2, ON);
                    break;
                case '2' :
                    portC(3, ON);
                    break;
                case '3':
                    portC(4, ON);
                    break;
                case '4' :
                    portC(5, ON);
                    break;
                case '5' :
                    portC(6, ON);
                    break;
                case '6':
                    portC(7, ON);
                    break;
            }
        }

        else
        {

```

```

switch(RxBuff[5])
{
    case '1' :
        portC(2, OFF); //
Pneumatik 1 "ON"
        break;
    case '2' :
        portC(3, OFF); //
Pneumatik 2 "ON"
        break;
    case '3':
        portC(4, OFF); //
Pneumatik 3 "ON"
        break;
    case '4' :
        portC(5, OFF); //
Pneumatik 1 "ON"
        break;
    case '5' :
        portC(6, OFF); //
Pneumatik 2 "ON"
        break;
    case '6':
        portC(7, OFF); //
Pneumatik 3 "ON"
        break;
}
}

index = 0;
for(int i=0; i<20; i++) RxBuff[i] =
'\0';
}

// Sensor LDR
if (PINB.0==1)
{putchar('A');}
else
{putchar('a');}

```

```
delay_ms(1);
if (PINB.1==1)
{putchar('B');}
else
{putchar('b');}
delay_ms(1);
if (PINB.2==1)
{putchar('C');}
else
{putchar('c');}
delay_ms(1);
if (PINB.3==1)
{putchar('D');}
else
{putchar('d');}
delay_ms(1);
if (PINB.4==1)
{putchar('E');}
else
{putchar('e');}
delay_ms(1);
if (PINB.5==1)
{putchar('F');}
else
{putchar('f');}
delay_ms(1);

// Sensor Arus
if (PINA.0==1)
{putchar('G');}
else
{putchar('g');}
delay_ms(1);
if (PINA.1==1)
{putchar('H');}
else
{putchar('h');}
delay_ms(1);
if (PINA.2==1)
{putchar('I');}
```

```

else
    {putchar('i');}
delay_ms(1);
if (PINA.3==1)
    {putchar('J');}
else
    {putchar('j');}
delay_ms(1);
if (PINA.4==1)
    {putchar('K');}
else
    {putchar('k');}
delay_ms(1);
if (PINA.5==1)
    {putchar('L');}
else
    {putchar('l');}

void read_rtc(void)
{
    if(rtc_is_ds3231()) {
        int8_t t;
        uint8_t f;
        ds3231_get_temp_int(&t, &f);
    }
    else {
        uint8_t hour, min, sec;
        rtc_get_time_s(&hour, &min, &sec);
    }
}

////////////////////////////////////
/// Main Program
int main()
{
    DDRC = 0xFF;
    PORTC = 0x00;

    PORTA = 0xFF;
    PORTB = 0xFF;
}

```

```

    uart_init();
    sei();

    twi_init_master();
    rtc_init();

    while(1)
    {
        rtc_get_time_s(&jam, &menit,
&detik);

        uart_puts("Waktu saat ini = ");
        uart_puti(jam); uart_putc(':');
        uart_puti(menit); uart_putc(':');
        uart_puti(detik); uart_puts("\n\r");
        delay(1000);

        if(jam == jamoff && menit ==
menitoff && detik <= 1)
        {
            for(int i=2; i<8; i++)
portC(i, OFF);
            print("Lampu OFF !\r\n");
        }
        else if(jam == jamon && menit ==
meniton && detik <= 1)
        {
            for(int i=2; i<8; i++)
portC(i, ON);
            print("Lampu ON !\r\n");
        }
    }
}

```

B. Program ESP8266

```
wifi.setmode(wifi.SOFTAP)

colorOn = "skyblue";
colorOff = "silver";
l1 = "silver";
l2 = "silver";
l3 = "silver";
l4 = "silver";
l5 = "silver";
l6 = "silver";
a1 = "silver";
a2 = "silver";
a3 = "silver";
a4 = "silver";
a5 = "silver";
a6 = "silver";

srv=net.createServer(net.TCP)
srv:listen(80,function(conn)
    conn:on("receive", function(client,request)

        buf = "";
        buf2 = "";
        buf3 = "";
        buf4 = "";
        buf5 = "";

        local _, _, method, path, vars =
string.find(request, "[A-Z]+ (.+)?(.+) HTTP");
        if(method == nil)then
            _, _, method, path =
string.find(request, "[A-Z]+ (.+) HTTP");
        end
        local _GET = {}
        if (vars ~= nil)then
            for k, v in string.gmatch(vars,
"(%w+)=(%w+)&*") do
                _GET[k] = v
            end
        end
    end
end)
```

```

end
end

local _on, _off = "", ""
if(_GET.pin == "ON1")then
    l1 = colorOn;
    a1 = colorOn;
    print("!ON1");
elseif(_GET.pin == "OFF1")then
    l1 = colorOff;
    a1 = colorOff;
    print("!OFF1");
elseif(_GET.pin == "ON2")then
    l2 = colorOn;
    a2 = colorOn;
    print("!ON2");
elseif(_GET.pin == "OFF2")then
    l2 = colorOff;
    a2 = colorOff;
    print("!OFF2");
elseif(_GET.pin == "ON3")then
    l3 = colorOn;
    a3 = colorOn;
    print("!ON3");
elseif(_GET.pin == "OFF3")then
    l3 = colorOff;
    a3 = colorOff;
    print("!OFF3");
elseif(_GET.pin == "ON4")then
    l4 = colorOn;
    a4 = colorOn;
    print("!ON4");
elseif(_GET.pin == "OFF4")then
    l4 = colorOff;
    a4 = colorOff;
    print("!OFF4");
elseif(_GET.pin == "ON5")then
    l5 = colorOn;
    a5 = colorOn;
    print("!ON5");

```

```

elseif(_GET.pin == "OFF5")then
    l5 = colorOff;
    a5 = colorOff;
    print("!OFF5");
elseif(_GET.pin == "ON6")then
    l6 = colorOn;
    a6 = colorOn;
    print("!ON6");
elseif(_GET.pin == "OFF6")then
    a6 = colorOff;
    l6 = colorOff;
    print("!OFF6");
elseif(_GET.waktuon ~= nil)then
    print("#".._GET.waktuon);
elseif(_GET.waktuoff ~= nil)then
    print("!*".._GET.waktuoff);
end

buf = buf.."<h1> MONITORING PJU</h1>";
buf = buf.."<div style=\"border: 2px
solid black;padding: 5px 20px;background:
cornflowerblue; width: 400px;height: 340px;\">";
buf = buf.."<p>Lampu 1 : <a
href=\"?pin=ON1\"><button>ON</button></a>&nbsp;<
a
href=\"?pin=OFF1\"><button>OFF</button></a></p>"
;
buf = buf.."<p>Lampu 2 : <a
href=\"?pin=ON2\"><button>ON</button></a>&nbsp;<
a
href=\"?pin=OFF2\"><button>OFF</button></a></p>"
;
buf = buf.."<p>Lampu 3 : <a
href=\"?pin=ON3\"><button>ON</button></a>&nbsp;<
a
href=\"?pin=OFF3\"><button>OFF</button></a></p>"
;
buf = buf.."<p>Lampu 4 : <a
href=\"?pin=ON4\"><button>ON</button></a>&nbsp;<
a

```

```

href="\ "?pin=OFF4\""><button>OFF</button></a></p>"
;
    buf = buf.."<p>Lampu 5 : <a
href="\ "?pin=ON5\""><button>ON</button></a>&nbsp;<
a
href="\ "?pin=OFF5\""><button>OFF</button></a></p>"
;
    buf = buf.."<p>Lampu 6 : <a
href="\ "?pin=ON6\""><button>ON</button></a>&nbsp;<
a
href="\ "?pin=OFF6\""><button>OFF</button></a></p>"
;
    buf = buf.."<div
style="\ height:300px;width: 180px;background:
bisque;position: absolute;margin: -230px 0px 0px
200px;border: 1px solid black;\ ">"
    buf = buf.."<h2 style="\ margin-left:
50px;\ ">Status</h2>"
;
    buf2 = buf2.."<div style="\ background:";
    buf2 = buf2..l1;
    buf2 = buf2.." ; width: 50px; height:
20px; text-align: center; margin: 20px;
\">L1</div>";
    buf2 = buf2.."<div style="\ background:";
    buf2 = buf2..l2;
    buf2 = buf2.." ; width: 50px; height:
20px; text-align: center; margin: 20px;
\">L2</div>";
    buf2 = buf2.."<div style="\ background:";
    buf2 = buf2..l3;
    buf2 = buf2.." ; width: 50px; height:
20px; text-align: center; margin: 20px;
\">L3</div>";
    buf2 = buf2.."<div style="\ background:";
    buf2 = buf2..l4;
    buf2 = buf2.." ; width: 50px; height:
20px; text-align: center; margin: 20px;
\">L4</div>";
    buf2 = buf2.."<div style="\ background:";

```

```

        buf2 = buf2..l5;
        buf2 = buf2.."; width: 50px; height:
20px; text-align: center; margin: 20px;
\">L5</div>";
        buf2 = buf2.."<div style=\"background:\";
        buf2 = buf2..l6;
        buf2 = buf2.."; width: 50px; height:
20px; text-align: center; margin: 20px;
\">L6</div>";

        buf3 = buf3.."<div style=\"background:\";
        buf3 = buf3..a1;
        buf3 = buf3..";width: 50px;height:
20px;text-align: center;margin: -240px 0px 0px
100px;\">A1</div>";
        buf3 = buf3.."<div style=\"background:\";
        buf3 = buf3..a2;
        buf3 = buf3..";width: 50px;height:
20px;text-align: center;margin: 20px 0px 0px
100px;\">A2</div>";
        buf3 = buf3.."<div style=\"background:\";
        buf3 = buf3..a3;
        buf3 = buf3..";width: 50px;height:
20px;text-align: center;margin: 20px 0px 0px
100px;\">A3</div>";
        buf3 = buf3.."<div style=\"background:\";
        buf3 = buf3..a4;
        buf3 = buf3..";width: 50px;height:
20px;text-align: center;margin: 20px 0px 0px
100px;\">A4</div>";
        buf3 = buf3.."<div style=\"background:\";
        buf3 = buf3..a5;
        buf3 = buf3..";width: 50px;height:
20px;text-align: center;margin: 20px 0px 0px
100px;\">A5</div>";
        buf3 = buf3.."<div style=\"background:\";
        buf3 = buf3..a6;
        buf3 = buf3..";width: 50px;height:
20px;text-align: center;margin: 20px 0px 0px
100px;\">A6</div>";

```

```

buf3 = buf3.."</div><br/>";

buf4 = buf4.."<form id=\"usrform\">";
buf4 = buf4.."Waktu ON: <input
style=\"margin-left:5px; width:50px;\"
type=\"text\" name=\"waktuon\"><input
type=\"submit\">";
buf4 = buf4.."</form>";

buf4 = buf4.."<form id=\"usrform\">";
buf4 = buf4.."Waktu OFF: <input
style=\"width:50px;\" type=\"text\"
name=\"waktuoff\"><input type=\"submit\">";
buf4 = buf4.."</form>";

client:send(buf);
client:send(buf2);
client:send(buf3);
client:send(buf4);
client:send("</div>");
client:close();
collectgarbage();
end)
end)

```

KESIMPULAN

Setelah melakukan pembuatan dan pengujian sistem pengaturan dan monitor lampu PJU, dapat diambil kesimpulan bahwa sistem monitor bekerja dengan baik. Persentase keberhasilan untuk monitor dapat mencapai 85%. Sedangkan untuk sistem kendali persentase keberhasilan hanya mencapai 90%. Hal ini disebabkan karena koneksi yang kurang stabil dari modul Wi-Fi kami menuju perangkat android yang digunakan, sehingga pengiriman data antara perangkat android menuju sistem kendali dan sebaliknya menjadi terganggu. Dengan menggunakan perangkat android lain, hasil bisa akan berubah dan tidak seefektif ini.

Saran untuk pengembangan sistem selanjutnya adalah dengan mencoba metode Wi-Fi lain selain menggunakan ESP8266-01 untuk meningkatkan kelancaran sinyal yang nantinya mempengaruhi kelancaran pengiriman dan penerimaan data dari dan menuju perangkat android.

Daftar Pustaka

- [1] W. Budiharto, “Aneka Proyek Mikrokontroler”, Graha Ilmu, Yogyakarta, 2011.
- [2] N.H. Safaat, “Pemrograman Aplikasi Android *Smartphone* dan Tablet PC Android”, Informatika, Bandung, 2012.
- [3] A.A. Huda, “24 Jam Pintar Pemrograman Android”, Andi Offset, Yogyakarta., 2012.
- [4] Andrianto, Heri, “Pemrograman mikrokontroler AVR Atmega16”, Informatika Bandung, 2008.
- [5] Maravitsas, Nikos, “*Android Socket Example*”, tersedia di: <http://examples.javacodegeeks.com/android/core/socket-core/android-socket-example/>, diakses 12 Januari 2016.
- [6] Ogata, Katsuhiko, “Teknik Kontrol Otomatik”, Jilid 1, Erlangga Jakarta, 1997.
- [7] Sigiros, Stevani Agnesia, “Instrumentasi Virtual Menggunakan *LabVIEW* dan *Soundcard*”, tersedia di: <http://repository.usu.ac.id/handle/123456789/45580>, diakses 30 Mei 2016.
- [8] Prastomo, Ragil Aji, “Perancangan Fungsi Perintah pada Mikrokontroler Atmega8535 Melalui Router Wi-Fi”, Tugas Akhir, Universitas Diponegoro Semarang, 2013.

DAFTAR RIWAYAT HIDUP



Nama : Kukuh Ariwibowo
Jenis Kelamin : Laki-laki
TTL : Surabaya, 25 Agustus 1994
Agama : Islam
Tinggi Badan : 169 cm
Berat Badan : 54 kg
Alamat : Dukuh Bulak Banteng Sek.
4A/6, Surabaya
Telep / HP : 085203442231
E-mail : kkh.takuru@gmail.com

RIWAYAT PENDIDIKAN

1. 2000-2006 : SDN Sidotopo Wetan IV/558 Surabaya
2. 2006-2009 : SMPN 15 Surabaya
3. 2009-2013 : SMKN 5 Surabaya
4. 2013-2016 : D3 Teknik Elektro, Program Studi Teknik Elektro Industri – FTI Institut Teknologi Sepuluh Nopember (ITS)

PENGALAMAN KERJA

1. Magang Kerja di PT. SCOMPTEC Surabaya
2. Kerja Praktek di PT. PAL (Persero) Surabaya
3. Kerja Praktek di KANGEAN ENERGY INDONESIA Ltd.

PENGALAMAN ORGANISASI

DAFTAR RIWAYAT HIDUP



Nama : Muhammad Wildan Wicaksono
Jenis Kelamin : Laki-laki
TTL : Surabaya, 15 April 1995
Agama : Islam
Tinggi Badan : 169 cm
Berat Badan : 50 kg
Alamat : Dsn. Sidomukti RT.04/RW.02
Kel. Kraton Kec. Krian
Kab. Sidoarjo
Telep / HP : 082257173483
E-mail : edelweisluluh95@gmail.com

RIWAYAT PENDIDIKAN

5. 2001-2007 : SDN IV Krian
6. 2007-2010 : SMPN 1 Krian
7. 2010-2013 : SMAN 1 Krian
8. 2013-2016 : D3 Teknik Elektro, Program Studi Teknik Elektro Industri – FTI Institut Teknologi Sepuluh Nopember (ITS)

PENGALAMAN KERJA

1. Kerja Praktek di PT. PLN Area Surabaya Barat divisi AMR

PENGALAMAN ORGANISASI

1. Staff Kementerian Dalam Negeri Periode 2014/2015 HIMAD3TEKTRO, FTI – ITS
2. Wakil Ketua FORUM KOMUNIKASI D3 ELEKTRO INDUSTRI Periode 2015/2016
3. Wakil Ketua Karang Taruna RT.04 Dsn. Sidomukti