

# **PROTOTYPE SMARTHOME DENGAN KONSEP INTERNET OF THINGS (IOT) BERBASIS WEBSITE**

## **PROYEK AKHIR**

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



Disusun Oleh :

Muhammad Taufik

NIRM : 0031552

Ria Astuti

NIRM : 0031558

**POLITEKNIK MANUFAKTUR NEGERI  
BANGKA BELITUNG  
TAHUN 2018**

**LEMBAR PENGESAHAN**

**JUDUL PROYEK AKHIR**  
**PROTOTYPE SMARTHOME DENGAN KONSEP INTERNET OF**  
**THINGS (IOT) BERBASIS WEBSITE**

Oleh :

Muhammad Taufik

NIRM : 0031552

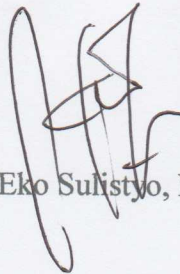
Ria Astuti

NIRM : 0031558

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

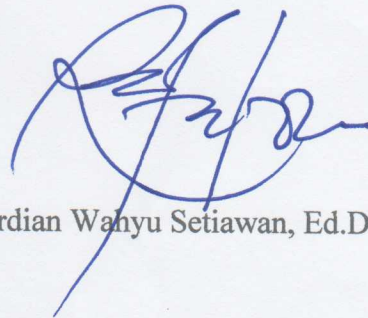
Menyetujui,

Pembimbing 1



Eko Sulistyono, M.T

Pembimbing 2



Ardian Wahyu Setiawan, Ed.D

Penguji 1



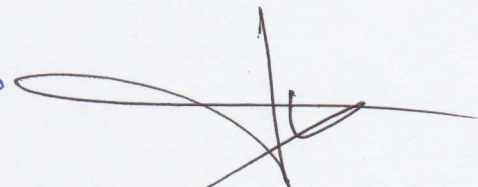
Indra Dwisaputra, M.T

Penguji 2



Aan Febriansyah, M.T

Penguji 3



Surojo, M.T

## PERNYATAAN BUKAN PLAGIAT

Yang bertanda tangan di bawah ini :

Nama Mahasiswa 1 : Muhammad Taufik      NIRM : 0031552

Nama Mahasiswa 2 : Ria Astuti              NIRM : 0031558

Dengan Judul : Prototype Smarthome Dengan Konsep Internet of Things (IoT)  
Berbasis Website

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

Sungailiat, 02 Agustus 2018

Nama Mahasiswa

Tanda Tangan

1. Muhammad Taufik



2. Ria Astuti



## ABSTRAK

*Smarthome adalah sebuah rumah biasa yang terintegrasi dengan konsep internet of things, dimana untuk pengontrolan dan monitoring peralatan listrik dilakukan secara otomatis dari jarak jauh selama 10 detik dan masih terhubung dengan jaringan internet. Sehingga tujuan dari pembuatan sistem ini yaitu efisiensi, efektifitas dan penghematan energi listrik dalam rumah bisa tercapai. Arduino Mega 2560 berfungsi sebagai penghubung antara hardware dan software seperti menggunakan sensor LDR untuk mendeteksi intensitas cahaya di luar ruangan, sensor PIR untuk mendeteksi keberadaan objek bergerak, serta sensor ultrasonik HC-SR04 sebagai water level control, Modul PZEM-004T sensor yang membaca daya yang digunakan. Pembacaan sensor-sensor tersebut dimonitoring dan dikontrol via aplikasi berbasis website yang dibuat menggunakan bahasa pemrograman php dan MySQL. Pengujian dilakukan secara otomatis menggunakan sistem smarthome dan secara manual yang dihitung selama 1 bulan. Hasil yang diperoleh adalah penggunaan secara sebesar 34.860Wh sedangkan secara sistem smarthome sebesar 25.098 Wh atau hemat menghemat sekitar 8,40%*

**Kata Kunci :** *Modul PZEM-004T, PHP, Sensor PIR, Sensor LDR, Sensor Ultrasonik,*

## **ABSTRACT**

*Smarthome is an ordinary home was integrated with Internet of Things concept to automatically control and monitor electrical equipment to distance long as 10 second and connecting of internet. With this system it can to efficiency, effectiveness, and saving the electrical energy at home. Arduino Mega 2560 serves to link hardware and software as using LDR sensor to detect of light intensity outdoors, PIR sensor to detect the presence of moving objects, and ultrasonic sensor HC-SR04 as water level control. Sensors are monitored and controlled via a website-based application created by php and MySQL program language. Testing in 1 month with automatically using smarthome system and manual system with counted. The results in this test was obtained on manually system in 34.860Wh, and automatic control in 25.098Wh or save about 8,40%*

**Keyword:** *PZEM-004T Modul, PHP, PIR Sensor, LDR Sensor, Ultrasonik Sensor*

## KATA PENGANTAR

Assalamu'alaykum warahmatullahi wabarakatuh. Alhamdulillah, segala puji bagi Allah Ta'ala, karena atas ridho dan rahmat-Nya jualah penulis dapat menyelesaikan proyek akhiryang berjudul"Prototype Smarthome Dengan Konsep Internet Of Thing (Iot) Berbasis Website. Proyek akhir ini disusun sebagai salah satu persyaratan dan kewajiban mahasiswa untuk menyelesaikan kurikulum program pendidikan Diploma III pada Progam Studi Teknik Elekonika Jurusan Teknik Elektronika dan Informatika di Politeknik Manufaktur Negeri Bangka Belitung. Sebagai bahan pembuatan proyek akhir, penulis mengambil bahan dari berbagai informasi, seperti jurnal, buku-buka, dan informasi lainnya.

Selama mengikuti program pendidikan DIII Teknik Elektronika sampai dengan proses penyelesaian proyek akhir ini, penulis telah mendapatkan berbagai macam bimbingan, dorongan, semangat, dan dukungan dari berbagai pihak. Oleh karena itu, perkenankanlah penulis menyampaikan ucapan terima kasih kepada : Tuhan Yang Maha Esa atas segala rezeki, nikmat, rahmat, ridho, dan karunia yang telah diberikan kepada penulis.

1. Orangtua dan keluarga penulis yang selalu memberikan kasih sayang, doa,dukungan moril maupun materil dan semangat.
2. Bapak Sugeng Ariyono, M.Eng., Ph.D.,selaku Direktur Politeknik Manufaktur Negeri Bangka Belitung.
3. Bapak Aan Febriansyah, M.T., selaku Ka. Jurusan Teknik Elektro dan Informatika Politeknik Manufaktur Negeri Bangka Belitung dan penguji 2.
4. Bapak Eko Sulistyono, M.T., selaku Ka. DIII Prodi Teknik Elektronika Politeknik Manufaktur Negeri Bangka Belitung dan pembimbing 1.
5. Bapak Ardian Wahyu Setiawan, Ed.D., selaku pembimbing 2.
6. Bapak Indra Dwisaputra, M.T selaku penguji 1.
7. Bapak Surojo, M.T selaku penguji 3.
8. Seluruh staf pengajar dan civitas di Politeknik Manufaktur Negeri Bangka Belitung.

9. Rekan-rekan mahasiswa Politeknik Manufaktur Negeri Bangka Belitung yang telah banyak membantu selama menyelesaikan proyek akhir.
10. Pihak-pihak lain yang telah memberikan bantuan secara langsung maupun tidak langsung dalam pembuatan proyek akhir ini yang tidak dapat disebutkan satu per satu.

Dalam penyusunan proyek akhir ini, penulis menyadari masih terdapat kekurangan baik sengaja maupun tidak sengaja, dikarenakan keterbatasan ilmu pengetahuan dan wawasan serta pengalaman yang penulis miliki. Oleh karena itu, penulis memohon maaf atas kekurangan dalam penulisan proyek akhir ini dan penulis mengharapkan segala petunjuk, kritik, dan saran yang membangun dari pembaca agar dapat menunjang pengembangan dan perbaikan penulisan selanjutnya. Akhir kata semoga proyek akhir ini dapat bermanfaat untuk menambah ilmu pengetahuan dan wawasan bagi pembaca. Aamiin.

Sungailiat, 02 Agustus 2018

Penulis

## DAFTAR ISI

	Halaman
<b>HALAMAN JUDUL</b> .....	i
<b>LEMBAR PENGESAHAN</b> .....	ii
<b>PERNYATAAN BUKAN PLAGIAT</b> .....	iii
<b>ABSTRAK</b> .....	iv
<b>ABSTRACT</b> .....	v
<b>KATA PENGANTAR</b> .....	vi
<b>DAFTAR ISI</b> .....	viii
<b>DAFTAR GAMBAR</b> .....	xi
<b>DAFTAR TABEL</b> .....	xiv
<b>DAFTAR LAMPIRAN</b> .....	xv
<b>BAB I PENDAHULUAN</b>	
1.1 Latar Belakang Masalah .....	1
1.2 Rumusan Masalah .....	3
1.3 Batasan Masalah .....	3
1.4 Tujuan Proyek Akhir .....	3
<b>BAB II DASAR TEORI</b>	
2.1 <i>Internet Of Things (IoT)</i> .....	4
2.2 <i>Smarthome</i> .....	5
2.3 Daya .....	5
2.4 Energi Listrik .....	6
2.5 Aplikasi Berbasis <i>Website</i> .....	6
2.6 <i>MySQL</i> .....	7
2.7 <i>PHP</i> .....	8
2.8 <i>HTML</i> .....	9
2.9 Perangkat Keras ( <i>Hardware</i> ) .....	9



2.9.1	<i>Arduino Mega2560</i> .....	9
2.9.2	<i>Modul Relay</i> .....	11
2.9.3	<i>Modul PZEM-004T</i> .....	12
2.9.4	<i>Buck Converter</i> .....	13
2.9.5	<i>ESP8266</i> .....	13
2.9.6	<i>Power Supply</i> .....	13
2.9.7	<i>Sensor Passive Infrared Receiver (PIR)</i> .....	14
2.9.8	<i>Sensor LDR(Light Dependent Resistor)</i> .....	15
2.9.9	<i>Sensor Ultrasonik</i> .....	16
2.9.10	<i>Buzzer</i> .....	17
2.11	Perangkat Lunak ( <i>Software</i> ) .....	18
2.11.1	<i>Software Arduino IDE Version 1.8.1</i> .....	18
2.11.2	<i>Notepad ++</i> .....	18
2.11.3	<i>XAMPP</i> .....	19
<b>BAB III METODE PELAKSANAAN</b>		
3.1	Flow Chart Pembuatan Alat.....	20
3.2	Survei, Pengumpulan Data dan Pengolahan Data.....	21
3.2.1	Metode Pengumpulan Data <i>Primer</i> .....	21
3.2.2	Metode Pengumpulan Data <i>Sekunder</i> .....	21
3.3	Rancangan dan Pembuatan <i>Hardware Smarthome</i> .....	22
3.4	Rancangan dan Pembuatan <i>Hardware Smarthome</i> .....	22
3.5	Uji Coba <i>Smarthome</i> .....	22
3.6	Analisa Data.....	23
3.7	Pembuatan Laporan.....	23
<b>BAB IV PEMBAHASAN</b>		
4.1	Perancangan dan Pembuatan Konstruksi .....	24
4.1.1	Perancangan <i>Smarthome</i> .....	25
4.1.2	Hasil <i>Smarthome</i> .....	26
4.2	<i>Box Panel</i> .....	26
4.3	Bagian <i>Hardware Elektrik</i> .....	28
4.3.1	<i>Arduino Mega 2560</i> .....	29

4.3.2 <i>Power supply</i> .....	32
4.3.3 Modul <i>Relay</i> .....	34
4.3.4 <i>Buck Converter</i> .....	36
4.3.5 Sensor <i>LDR</i> .....	39
4.3.6 Sensor Ultrasonik.....	40
4.3.7 Sensor <i>PIR</i> .....	43
4.3.8 Modul PZEM-004T.....	46
4.3.9 <i>Buzzer</i> .....	49
4.3.10 Modul WiFi ESP8266.....	50
4.4 <i>Assembly</i> Komponen Keseluruhan.....	53
4.5 Pembuatan Program .....	53
4.5.1 Sistem Kontrol Pada <i>Smarthome</i> .....	53
4.5.2 Pembuatan Aplikasi <i>Website</i> .....	55
4.6 Uji Coba Sistem Keseluruhan .....	56
4.6.1 Pengujian Dengan <i>Website</i> .....	63
4.6.2 Penggunaan Energi Listrik Harian .....	64
4.7 Analisa.....	69
<b>BAB V PENUTUP</b>	
5.1 Kesimpulan .....	70
5.2 Saran.....	71
<b>DAFTAR PUSTAKA</b> .....	72
<b>LAMPIRAN</b>	

## DAFTAR GAMBAR

Gambar	Halaman
2.1 <i>Arduino Mega2560</i> .....	10
2.2 Hubungan Modul Relay dengan <i>Arduino Mega2560</i> .....	12
2.3 Modul <i>PZEM-004T</i> .....	12
2.4 <i>Buck Converter</i> .....	13
2.5 <i>ESP8266</i> .....	13
2.6 <i>Power Supply</i> .....	14
2.7 Sensor <i>PIR</i> .....	14
2.8 Sensor <i>LDR</i> .....	15
2.9 Bentuk Fisik Sensor <i>Ultrasonik</i> .....	16
2.10 <i>Buzzer</i> .....	17
2.11 Tampilan dari <i>Software Arduino IDE</i> .....	18
2.12 Tampilan Awal <i>Notepad ++</i> .....	19
2.13 Tampilan Awal <i>XAMPP</i> .....	19
3.1 <i>Flow Chart</i> Metode Pelaksanaan .....	20
4.1 Rancangan denah 2D dari rumah pintar.....	24
4.2 Tampak depan dari <i>smarthome</i> .....	25
4.3 Hasil <i>smarthome</i> .....	26
4.4 Rancangan <i>box panel</i> .....	27
4.5 Hasil dari <i>box panel</i> .....	28
4.6 Pemasangan komponen-komponen pada arduino .....	28
4.7 Bentuk fisik arduino mega 2560 .....	30
4.8 Blok diagram pengujian arduino.....	30
4.9 Rangkaian skematik <i>power supply</i> .....	32
4.10 Bentuk fisik dari <i>power supply</i> .....	33
4.11 Blok diagram pengujian <i>power supply</i> .....	33

4.12 Hasil pengukuran <i>power supply</i> .....	33
4.13 Rangkaian skematik <i>relay</i> .....	34
4.14 Modul <i>relay</i> yang sudah dirangkai .....	35
4.15 Blok diagram pengujian modul <i>relay</i> .....	35
4.16 Hasil pengujian modul <i>relay</i> .....	35
4.17 Rangkaian skematik <i>buck converter</i> .....	36
4.18 <i>Buck converter</i> .....	36
4.19 Blok diagram pengujian <i>buck converter</i> .....	36
4.20 Pengukuran tegangan <i>output buck converter</i> .....	37
4.21 Rangkaian skematik sensor <i>LDR</i> .....	38
4.22 Modul sensor <i>LDR</i> .....	38
4.23 Rangkaian pengujian sensor <i>LDR</i> .....	38
4.24 Rangkaian skematik sensor <i>ultrasonik</i> .....	40
4.25 Sensor <i>ultrasonik</i> .....	40
4.26 Skema pengujian sensor <i>ultrasonik</i> .....	41
4.27 Pengujian sensor ultrasonik pada tandon air .....	42
4.28 Rangkaian skematik sensor <i>PIR</i> .....	43
4.29 Sensor <i>PIR</i> yang sudah dirangkai .....	43
4.30 Rangkaian pengujian sensor <i>PIR</i> .....	44
4.31 Pengujian ruangan dengan sensor <i>PIR</i> .....	44
4.32 Rangkaian skematik modul <i>PZEM-004T</i> .....	45
4.33 Modul <i>PZEM-004T</i> yang sudah dirangkai .....	46
4.34 Rangkaian pengujian sensor <i>PZEM-004T</i> .....	47
4.35 Pembacaan daya dengan <i>wattmeter</i> .....	48
4.36 Rangkaian skematik <i>buzzer</i> .....	49
4.37 <i>Buzzer</i> .....	50
4.38 Rangkaian skematik modul <i>wifi ESP8266</i> .....	50
4.39 Modul <i>wifi ESP8266</i> .....	51
4.40 Koneksi modul <i>wifi ESP8266</i> dengan <i>hotspot</i> .....	52
4.41 Tampak atas hasil perakitan <i>hardware</i> .....	53
4.42 <i>Flow chart</i> sistem kerja <i>smarthome</i> .....	54

4.43	Tampilan awal XAMPP .....	55
4.44	Halaman pembuatan <i>database</i> .....	55
4.45	Tampilan notepad ++ .....	56
4.46	Flowchart program LDR .....	57
4.47	Blok Diagram pengujian sensor <i>LDR</i> .....	58
4.48	Pengujian sensor LDR .....	55
4.49	Flowchart sensor <i>PIR</i> .....	59
4.50	Blok diagram pengujian sensor <i>LDR</i> .....	60
4.51	Pengujian ruangan dengan sensor PIR .....	60
4.52	<i>Flowchat</i> sensor ultrasonik .....	61
4.53	Blok Diagram Pengujian .....	62
4.54	Pengujian sensor.....	62
4.55	Pengujian kontrol pada website .....	63
4.56	Pengujian lampu pada website .....	63
4.57	Hasil monitoring peralatan listrik .....	64
4.58	Grafik <i>Monitoring</i> Penggunaan Energi Listrik .....	66
4.59	Hasil dari <i>Data Base</i> pada <i>Website</i> .....	67
4.60	Pembacaan Daya Dengan <i>Wattmeter</i> .....	68

## DAFTAR TABEL

Tabel	Halaman
2.1 Data teknik <i>Board Arduino Mega 2560</i> .....	10
4.1 Pin pada <i>arduino mega 2560</i> .....	29
4.2 Hasil pengukuran <i>active high pin arduino mega 2560</i> .....	30
4.2 Hasil pengukuran <i>active high pin arduino mega 2560</i> .....	30
4.3 Hasil pengukuran <i>active low pin arduino mega 2560</i> .....	31
4.4 Pengujian <i>power supply</i> .....	34
4.5 Hasil pengukuran <i>buck converter</i> .....	37
4.6 Hasil pengujian sensor LDR .....	39
4.7 Hasil perbandingan sensor ultrasonik dengan penggaris .....	42
4.8 Hasil pengujian sensor PIR dengan berbagai objek.....	45
4.9 Hasil pengujian sensor <i>PIR</i> .....	45
4.10 Pengujian Modul PZEM-004T .....	48
4.11 Pengujian Koneksi ESP8266 dengan <i>hotspot</i> .....	52
4.8 Hasil pengujian sensor PIR dengan berbagai objek.....	45
4.9 Hasil pengujian sensor <i>PIR</i> .....	45
4.10 Pengujian Modul PZEM-004T .....	48
4.11 Pengujian Koneksi ESP8266 dengan <i>hotspot</i> .....	52
4.12 Hasil pengujian sensor LDR .....	58
4.13 Pengujian lampu.....	60
4.14 Hasil pengujian sensor Ultrasonik .....	62
4.15 Perhitungan Pemakaian Manual.....	64
4.16 Perhitungan Otomatis.....	65
4.17 Hasil <i>Monitoring</i> Penggunaan Energi Listrik .....	66
4.18 Perbandingan Perhitungan Secara Amanual dan Otomatis.....	69

## DAFTAR LAMPIRAN

Lampiran 1 : Daftar Riwayat Hidup

Lampiran 2 : Program *Smarthome*

Lampiran 3 : Lembar *Datasheet*

## BAB I

### PENDAHULUAN

#### 1.1 Latar Belakang Masalah

Pesatnya perkembangan ilmu pengetahuan dan teknologi terutama dalam bidang elektronika telah menghasilkan banyak dampak positif dalam kehidupan manusia. Salah satu dampak positifnya adalah terciptanya suatu sistem yang bernama *smarthome*. Sistem rumah otomatis atau *smarthome system* adalah sebuah rumah biasa yang berintegrasi dengan suatu system dalam pengontrolan peralatan listrik seperti lampu, kipas angin, tandon air dan peralatan listrik lainnya. Pada tugas akhir sebelumnya telah dibuat alat yang berjudul “MANAJEMEN PEMAKAIAN ENERGI LISTRIK UNTUK RUMAH PINTAR”[15].

Alat ini terdiri dari miniatur rumah dengan peralatan listrik yang akan dikendalikan menggunakan pesan *SMS* untuk mengontrol lampu yang menyala secara otomatis dengan mendeteksi objek yang bergerak dan berdasarkan intensitas cahaya lingkungan. Namun pada sistem ini terdapat kekurangan dikarenakan kontrol yang dilakukan tidak dapat *memonitoring* langsung kondisi peralatan listrik dan jumlah pulsa yang ada pada modul *GSM*. Oleh karena itu perlunya pengembangan sistem dengan berbasis pada *Internet of Things( IoT)*, dimana sistem ini dapat memudahkan pemilik rumah untuk melakukan pengontrolan dan *monitoring* peralatan listrik dari jarak jauh selama masih terhubung dengan jaringan internet.

Sistem ini sangatlah cocok diterapkan bagi pemilik rumah apabila sedang berpergian tanpa mengkhawatirkan lampu rumah yang padam pada malam hari atau tandon air yang dalam kondisi kosong serta peralatan listrik lainnya yang lupa dalam pemadamannya. Selain memberikan kemudahan, sistem ini juga memberikan kenyamanan bagi penderita difabel dan juga dapat mengontrol penggunaan energi listrik sedemikian rupa sehingga meningkatkan efisiensi



penggunaan listrik. Saat ini kebutuhan masyarakat akan energi listrik sangat tinggi terlebih penggunaan listrik, pada rumah tangga. Sebagai contoh adalah kebiasaan lupa mematikan tandon air, atau lupa mematikan lampu kamar, kipas angin dan kebiasaan lainnya.

Padahal kebiasaan tersebut dapat menambah biaya listrik bulanan serta pemakaian energi yang digunakan. Mengapa demikian, karena energi listrik merupakan daya listrik yang terpakai selama waktu tertentu. Artinya semakin besar daya dari peralatan maka semakin besar energi yang terpakai, serta semakin lama waktu pemakaiannya maka semakin besar energi yang dihabiskan [1]. Telah diketahui jumlah sumber energi dialam terbatas, sementara kebutuhan atas listrik terus meningkat. Jika kita menggunakan energi secara berlebihan hal tersebut akan mengancam keberlangsungan alam ini.

Pada percobaan lainnya juga dalam jurnal berjudul “Aplikasi Rumah Pintar (Smarthome) Pengendali Peralatan Elektronik Rumah Tangga Berbasis Web”, dimana alat tersebut peralatan listrik yang akan dikendalikan menggunakan simulasi LED yang akan dikontrol pada *interface web* dengan raspberry pi sebagai *server* penghubung antara *hardware* dan *software* yang digunakan oleh *user* [2]. Penelitian lainnya dengan judul jurnal “Efisiensi Energi Smarthome (Rumah Pintar) berbasis Remote Relay dan LDR ( Light Dependent Resistansi)”, dimana alat tersebut bisa mengontrol lampu menggunakan remote dengan jarak maksimalnya adalah 90cm serta lampu bisa menyala secara otomatis dengan mendeteksi intensitas cahaya [3].

Oleh karena itu, perlu adanya alat yang bisa selain mengontrol peralatan listrik secara langsung, namun juga bisa memonitoring penggunaan daya yang digunakan peralatan listrik tersebut dengan *platform* yang mudah seperti *website*. Penulis mempunyai ide membuat alat untuk dengan judul “*Prototype smarthome* dengan konsep *Internet of Things* (IoT) berbasis *website*” dengan alat *Arduino Mega2560* sebagai pengendalinya. Hal ini merupakan upaya untuk penghematan listrik serta memudahkan pengguna untuk mengendalikan peralatan elektronik rumah tangga secara otomatis dari jarak jauh sehingga penggunaan daya listrik lebih efektif.

## **1.2 Rumusan Masalah**

Rumusan masalah yang dibuat pada proyek akhir ini adalah :

1. Bagaimana merancang dan membuat konstruksi alat *prototype smarthome* dengan konsep *internet of things* berbasis *website*?
2. Bagaimana merancang dan membuat sistem kontrol pada alat *prototype smarthome* dengan konsep *internet of things* berbasis *website* ?
3. Bagaimana mengetahui perbandingan antara penggunaan daya secara manual dan secara sistem *smarthome*?

## **1.3 Batasan Masalah**

Batasan masalah yang dibuat pada proyek akhir ini adalah :

1. *Smarthome* ini diterapkan dalam bentuk *prototype* rumah/miniatur rumah dengan ukuran panjang 80cm, lebar 80cm dan tinggi 40cm.
2. *Prototype smarthome* pada tipe rumah 1 lantai dengan akses kontrol dan monitoring 6 lampu, 1 kipas angin, dan 1 tandon air.
3. Sistem *smarthome* ini hanya bisa digunakan ditempat yang memiliki jaringan internet stabil.

## **1.4 Tujuan Proyek Akhir**

Tujuan dari proyek akhir ini adalah :

1. Memudahkan pengguna dalam memonitoring penggunaan daya secara langsung di rumah tanpa batasan jarak.
2. Meminimalisasikan daya yang berlebihan secara optimal.

## BAB II

### DASAR TEORI

#### 2.1 *Internet of Things (IoT)*

*Internet of Things (IoT)*, sebuah istilah yang belakangan ini mulai ramai diperbincangkan, namun masih belum ada definisi yang baku dari IoT itu sendiri. Burange dan Misalkar (2015) menyatakan *Internet of Things (IoT)* adalah struktur di mana objek, orang disediakan dengan identitas eksklusif dan kemampuan untuk pindah data melalui jaringan tanpa memerlukan dua arah antara manusia ke manusia yaitu sumber ke tujuan atau interaksi manusia ke computer [4].

Ide awal *Internet of Things (IoT)* pertama kali dicetuskan oleh Kevin Ashton pada tahun 1999 dalam sebuah presentasinya. Dalam artikelnya untuk RFID Journal, “*That ‘Internet of things’ Thing*”, Kevin Asthon (2009) menyebutkan: “Hampir semua dari sekitar 50 petabyte (1,024 terabyte) dari data yang tersedia di internet dibuat oleh manusia yang mengetik, merekam, memotret, atau membaca *barcode*. Tetapi dikarenakan keterbatasan waktu, ketelitian, dan data/informasi dari didunia nyata tidak dapat diperoleh secara langsung. Hal ini dikarenakan teknologi informasi saat ini sangat bergantung dari data yang berasal dari manusia. Sehingga kita membutuhkan komputer yang tahu segala sesuatu, yang bisa mengolah datanya sendiri tanpa bantuan manusia, agar bisa melacak dan menghitung segala sesuatu dan kita akan tahu kapan suatu hal perlu diganti, diperbaiki, atau diingatkan, dan apakah mereka masih baru atau sudah usang.

*Internet of things* memiliki potensi untuk mengubah dunia, seperti yang telah dilakukan oleh internet. Bahkan lebih dari pada itu” [5]. *Internet of Things (IoT)* menggunakan metode *nirkabel* atau pengendalian secara otomatis tanpa mengenal jarak. Pengimplemtasian *Internet of Things* sendiri biasanya disesuaikan dengan keinginan *developer* dalam pengembangan aplikasi yang diciptakannya. Oleh karena itu IoT dapat memberikan inovasi kemudahan dalam pengontrolan dan monitoring dari jarak jauh dengan konektifitas internet yag baik.

## 2.2 *Smarthome*

*Smarthome* atau rumah cerdas merupakan kesatuan sistem yang terdiri dari kontrol, monitoring, dan otomatisasi yang pengendaliannya melalui *smartphone* atau komputer secara terpusat. Pada proyek akhir ini, sistem *smarthome* diimplementasikan dalam ruang lingkup yang lebih kecil atau sederhana yaitu pengontrolan terhadap lampu ruangan rumah, tandon air serta kipas angin.

Rumah merupakan bangunan yang berfungsi sebagai tempat tinggal dan berlindung manusia. Pada dasarnya sebuah rumah terdiri atas ruangan-ruangan terpisah dengan fungsi tersendiri dan berisi alat-alat elektronika dan perabot rumah tangga. Sistem *smarthome* adalah sistem aplikasi yang merupakan gabungan antara teknologi dan pelayanan yang dikhususkan pada lingkungan rumah dengan fungsi tertentu yang bertujuan meningkatkan efisiensi, kenyamanan dan keamanan penghuninya.

## 2.3 Daya

Daya Listrik atau dalam bahasa Inggris disebut dengan *electrical power* adalah jumlah energi yang diserap atau dihasilkan dalam sebuah sirkuit/rangkaian listrik. Satuan SI daya listrik adalah *watt* yang menyatakan banyaknya tenaga listrik yang mengalir per satuan waktu (joule/detik). Arus listrik yang mengalir dalam rangkaian dengan hambatan listrik menimbulkan kerja. Peranti mengkonversi kerja ini ke dalam berbagai bentuk yang berguna, seperti panas (seperti pada pemanas listrik), cahaya (seperti pada bola lampu), energi kinetik (motor listrik), dan suara (*loudspeaker*). Listrik dapat diperoleh dari pembangkit listrik atau penyimpan energi seperti baterai perkalian arus dan tegangan efektif dalam rangkaian AC dinyatakan dalam *voltampere* (VA) atau *kilovoltampere* (KVA). Satu KVA sama dengan 1.000 VA. Daya yang berguna atau daya nyata diukur dalam watt dan diperoleh jika voltampere dari rangkaian dikalikan dengan faktor yang disebut dengan faktor daya [6]. Maka dalam rangkaian AC satu *phase* adalah :

$$P = V \times I \times \cos \theta \dots\dots\dots(2.1)$$

$$\cos \theta = P / V \dots\dots\dots(2.2)$$

Keterangan :

P = Daya (Watt)

V = Tegangan (Volt)

I = Arus (Ampere)

Cos  $\theta$  = Cos phi

## 2.4 Energi Listrik

Energi Listrik yang dibutuhkan bagi peralatan listrik untuk menggerakkan motor, lampu penerangan, memanaskan, mendinginkan ataupun untuk menggerakkan kembali suatu peralatan mekanik untuk menghasilkan bentuk *energy* yang lain. “Satuan daya sama dengan 1 *joule/sec* sering disebut sebagai watt dan satuan *energy* dinyatakan dalam watt, yaitu watt-hour atau Wh [7].

$$1 Wh = 1 J/s \times 3600 s = 3600 J \dots\dots\dots(2.3)$$

$$1 KWh = 1000 Wh = 3600 kJ \dots\dots\dots(2.4)$$

Keterangan :

Wh = Watt – hour

KWh = Kilowatt – hour

kJ = Kilojoule

J = Joule

Pengertian energi listrik adalah kemampuan untuk melakukan atau menghasilkan usaha listrik (kemampuan yang diperlukan untuk memindahkan muatan dari satu titik ke titik yang lain), dilambangkan dengan W.

## 2.5 Aplikasi Berbasis Website

Aplikasi berbasis *website* merupakan aplikasi yang dapat diakses melalui *web browser* yang mendukung interaksi pengguna melalui antar muka yang terletak pada *web server*. Adapun uraian dari *web server* dan *web browser* yaitu:

### a. Web Server

*Web server* memberikan layanan berbasis data dan berfungsi menerima permintaan dari HTTP atau HTTPS pada *klien* yang dikenal (*web browser*) dan untuk mengirimkan kembali yang hasilnya dalam bentuk beberapa halaman *web* dan pada umumnya akan berbentuk dokumen HTML.

## b. *Web browser*

Untuk mengakses aplikasi berbasis *website*, diperlukan sebuah *web browser*. Dengan *web browser*, seorang pengguna bisa mengakses informasi yang disediakan pada banyak *website* secara cepat dan mudah. Menurut Hans S. Limantara (2009:2) beberapa *web browser* yang banyak digunakan antara lain [8]:

1. *Mozilla Firefox* adalah *web browser* yang cukup populer digunakan sebagian pengguna internet.
2. *Google Chrome* adalah *web browser* yang dirancang oleh *Google INC*, yang terkenal dengan mesin pencariinya.
3. *Microsoft Internet Explorer* adalah sebuah *web* dan *software* yang tidak bebas namun gratis dari *Microsoft*, dan diikutkan dalam setiap peluncuran sistem operasi *Microsoft Windows*.
4. *Opera* adalah *browser* populer yang mudah digunakan, serta perangkat lunak untuk membaca dan mengirim surat.

Aplikasi berbasis *website* memiliki beberapa kelebihan dan kelemahan. Adapun uraian dari kelebihan dan kelemahan adalah sebagai berikut.

Kelebihan dari aplikasi berbasis *website* ini adalah :

1. Dapat diakses kapanpun dan dimana pun selama ada internet.
2. Dapat diakses hanya dengan menggunakan *web browser*, tidak perlu menginstal aplikasi client khusus.
3. Memudahkan dalam penyimpanan data di *database*.

Kekurangan dari aplikasi berbasis *website* ini adalah :

1. Antarmuka yang dapat dibuat terbatas sesuai spesifikasi standar untuk membuat dokumen web dan keterbatasan kemampuan *web browser* untuk menampilkannya.
2. Terbatasnya kecepatan internet mungkin membuat respon aplikasi menjadi lambat.
3. Tingkat keamanan yang lebih rentan untuk diakses oleh orang lain atau pihak yang tidak berhak.

## 2.6 MySQL

*MySQL* adalah sebuah sistem manajemen *database* yang bersifat *open source*, yang bisa didownload oleh siapa saja. *MySQL* merupakan *database server* yang berhubungan erat dengan *PHP* dan turunan dari salah satu konsep utama dalam *database* sejak lama yaitu *Structured Query Language (SQL)*. *SQL* adalah sebuah konsep yang dirancang khusus untuk komunikasi *database* terutama untuk pemilihan/seleksi dan pemasukan data, yang memungkinkan data dikerjakan secara otomatis. *MySQL* dapat dikatakan lebih unggul dibandingkan *database server* lainnya dalam *query* data. Hal ini terbukti untuk *query* yang dilakukan oleh *singer user*, kecepatan *query MySQL* bisa sepuluh kali lebih cepat dari *PostgresSQL* dan lima kali lebih cepat dari *Interbase* [9]. Berikut merupakan beberapa kelebihan dari *MySQL* yaitu :

1. *Portability*, bersifat stabil tanpa kendala.
2. *Multiusers*, dapat digunakan untuk menangani beberapa *user* dalam waktu bersamaan tanpa mengalami masalah.
3. *Security*, memiliki beberapa lapisan keamanan seperti *hostname* dan izin akses *user* dengan sistem perizinan yang mendetail serta kata sandi terenkripsi.

## 2.7 PHP

Menurut Arif (2011, p.43) *PHP (PHP Hypertext Preprocessor)* adalah bahasa *server-side scripting* yang menyatu dengan *HTML* untuk membuat halaman *website* yang dinamis [10]. Karena *PHP* merupakan *server-side scripting* maka sintaks dan perintah-perintah *PHP* akan dieksekusi di *server* kemudian hasilnya dikirimkan ke *browser* dalam format *HTML*. Dengan demikian kode program yang ditulis didalam *PHP* tidak akan terlihat oleh *user* sehingga keamanan *web* lebih terjamin. *PHP* dirancang untuk membentuk halaman *website* yang dinamis, yaitu halaman yang dapat membentuk suatu tampilan berdasarkan permintaan terkini, seperti menampilkan isi basis data ke halaman *web*.

*PHP* termasuk dalam *open source product*, sehingga *source code PHP* dapat diubah dan didistribusikan secara bebas. Salah satu keunggulan yang dimiliki *PHP* adalah kemampuan untuk melakukan koneksi ke berbagai macam

*software* sistem manajemen basis data. *PHP* juga dapat berjalan lintas *platform* yaitu dapat digunakan dengan sistem operasi (*Windows dan Linux*) dan *web server* apapun. Contoh pemrograman php yang terkoneksi pada database:

```
<?php
$host = 'localhost';
$user = 'root';
$password = '';
$database='belajar';
mysql_connect($host,$user,$password);
//mysql_select_db("belajar");
mysql_select_db($database); ?>
```

## **2.8 HTML**

*HTML* atau *Hypertext Markup Language* adalah bahasa standar untuk membuat halaman-halaman *web*. Model kerja *HTML* diawali dengan permintaan suatu halaman *web* oleh browser. Berdasarkan *URL (Uniform Resource Located)* atau dikenal dengan sebuah alamat internet, *browser* mendapatkan alamat dari *web server*, mengidentifikasi halaman yang dikehendaki dan menyimpan segala informasi yang dibutuhkan oleh *web server*. Selanjutnya *web server* akan mencari *file* yang diminta dan memberikan isinya ke *web server* atau (yang biasa disebut *browser* saja) *browser* yang mendapatkan isinya segera melakukan proses penerjemahan kode *HTML* dan menampilkan kelayar *user*. Contoh pengkodean *HTML* dapat dilihat sebagai berikut:

```
<html>
<head><title>Selamat Datang</title></head>
<body></body></html>
```

## **2.9 Perangkat Keras (*Hardware*)**

### **2.9.1 *Arduino Mega 2560***

Komponen utama didalam papan *arduino* adalah sebuah 8 bit dengan merk *ATmega* yang dibuat oleh *Atmel Corporation*. Berbagai papan *arduino* menggunakan tipe *ATmega* yang berbeda-beda tergantung dari spesifikasinya, sebagai contoh *Arduino Uno* menggunakan *ATmega328* sedangkan *Arduino Mega2560* yang lebih canggih menggunakan *ATmega2560* [11]. Bentuk fisik *Arduino Mega2560* dapat dilihat pada Gambar 2.1.





Gambar 2.1 *Arduino mega2560*

Adapun data teknis board *Arduino Mega2560* adalah seperti yang tertera pada tabel 2.1

Tabel 2.1. Data teknik *Board Arduino Mega 2560*[9]

Mikrokontroler	<i>ATmega2560</i>
Tegangan Pengoperasian	5V
Tegangan <i>Input</i> (Rekomendasi)	7-12V
Batas Tegangan Input	6-20V
Pin <i>I/O Digital</i>	54 buah (6 diantaranya dapat di gunakan sebagai output PWM)
Pin Digital PWM	6 pin
Pin Input Analog	16 pin
Arus DC Tiap Pin <i>I/O</i>	20 Ma
Arus DC untuk pin 3.3V	50 Ma
<i>Flash Memory</i>	256 KB ( <i>ATmega2560</i> )
<i>SRAM</i>	Sekitar 8 KB digunakan untuk <i>bootloader</i>
<i>EEPROM</i>	8 KB ( <i>ATmega2560</i> )

Arduino mega2560 memiliki 54 buah digital pin yang dapat digunakan sebagai input atau output. Pin-pin tersebut bekerja pada tegangan 5 volt dan setiap pin menyediakan atau menerima arus 20mA serta memiliki tahanan *pull-up* sekitar 20-50 kohm. Arduino uno memiliki 16 pin analog, yang menyediakan 10 bit dengan resolusi ( yaitu 1024 nilai yang berbeda). Adapun fungsi dari masing-masing pin arduino mega2560 adalah sebagai berikut :

1. *Serial*, memiliki 4 *serial* yaitu *serial 0* : pin 0 (RX) dan pin 1 (TX), *serial 1* : pin 19 (RX) dan pin 18 (TX) serta *serial 3* : pin 14 (RX) dan pin 15 (TX). RX yang digunakan untuk menerima RX dan mengirim TX data serial.
2. *External Interrupts*, memiliki 6 buah pin *interrupts* yaitu pin 2 (untuk *interrupts 0*), pin 3 (untuk *interrupts 1*), pin 18 (untuk *interrupts 5*), pin 19 (untuk *interrupts 4*), pin 20 (untuk *interrupts 3*), serta pin 21 (untuk *interrupts 2*).
3. *PWM*, pin 2 hingga pin 13 serta pin 44 hingga pin 46, menyediakan output *PWM* 8-bit dengan menggunakan fungsi *analogWrite( )*.
4. *SPI*, pin 50 (MOSI), pin 51 (MISO), pin 52 (SCK), dan pin 53 (SS) mendukung komunikasi *SPI* dengan menggunakan *SPI Library*
5. *LED*, pin 13 terhubung *built-in Led* yang dikendalikan oleh digital pin nomor 13.
6. *I2C*, pin 20 (SDA) dan pin 21 (SCL) yang mendukung komunikasi *I2C* (TWI) dengan menggunakan *Wire Library*.
7. *Aref*, sebagai referensi tegangan untuk *input analog*.
8. *Reset*, hubungkan ke *LOW* untuk melakukan *reset* terhadap mikrokontroler.

### 2.9.2 Modul Relay

*Relay* adalah saklar (*switch*) yang dioperasikan secara listrik. *Relay* juga biasa disebut komponen elektromekanikal yang terdiri dari dua komponen yaitu bagian *coil (electromagnet)* dan seperangkat kontak saklar. Komponen *relay* menggunakan prinsip elektromagnetik sebagai penggerak kontak saklar, sehingga dengan menggunakan arus listrik yang kecil dapat menghantar arus listrik yang memiliki tegangan lebih tinggi guna mengurangi beban kerja pada komponen.

Beberapa fungsi *relay* yang telah umum diaplikasikan kedalam elektronika diantaranya adalah :

1. Untuk menjalankan fungsi logika.
2. Untuk memberikan fungsi penundaan waktu.

3. Untuk mengendalikan sirkuit tegangan tinggi dengan bantuan dari sinyal tegangan rendah.
4. Melindungi motor atau komponen lainnya dari kelebihan tegangan atau korsleting.



Gambar 2.2 Hubungan modul *relay* dengan *arduino mega 2560*

Hubungan antara modul relay dan arduino seperti terlihat pada Gambar 2.2 yaitu:

1. *GND* dihubungkan ke *ground*.
2. *IN1*, merupakan control relay saluran pertama, dihubungkan ke pin *digital arduino*.
3. *IN2*, merupakan control relay saluran kedua, dihubungkan ke pin *digital arduino*.
4. *VCC* dihubungkan ke tegangan 5v.

### 2.9.3 Modul *PZEM-004T*

Modul *PZEM-004T* adalah sebuah modul sensor multifungsi yang berfungsi untuk mengukur daya, tegangan, arus dan energi yang terdapat pada sebuah aliran listrik. Modul ini sudah dilengkapi sensor tegangan dan sensor arus (*CT*) yang sudah terintegrasi. Dalam penggunaannya, alat ini khusus untuk penggunaan dalam ruangan (*indoor*) dan beban yang terpasang tidak diperbolehkan melebihi daya yang sudah ditetapkan [12]. Bentuk fisik dari modul *PZEM-004T*, dapat dilihat pada Gambar 2.3



Gambar 2.3 Modul *PZEM-004T*

#### 2.9.4 *Buck Converter*

*Buck converter* adalah konverter yang menghasilkan tegangan keluaran yang lebih kecil dari tegangan masukannya. Tegangan keluaran yang dihasilkan mempunyai polaritas yang sama dengan tegangan masukannya. *Buck converter* biasa disebut juga sebagai *step-down converter* [13]. Gambar 2.4 merupakan bentuk fisik dari *buck converter*.



Gambar 2.4 *Buck converter*

#### 2.9.5 *Modul Wifi ESP8266*

*ESP8266* adalah sebuah *chip* yang sudah lengkap dimana didalamnya sudah termasuk *processor*, memori dan juga akses ke *GPIO*. Hal ini menyebabkan *ESP8266* dapat secara langsung menggantikan *arduino* dan ditambah lagi dengan kemampuannya untuk mensupport koneksi *wifi* secara lansung [14]. Bentuk fisik dari *ESP8266* dapat dilihat pada Gambar 2.5.



Gambar 2.5 *ESP8266*

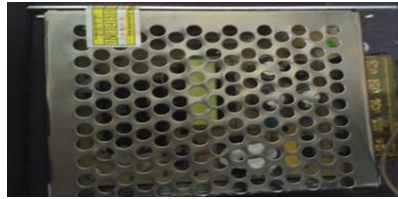
Pada umumnya, *ESP8266* dapat diprogram dengan :

1. Melalui *AT Command via serial* komunikasi *UART*
2. Pemrograman ke mikrokontroller yang ada di *ESP8266* menggunakan *arduino IDE* dengan *core* yang sudah terinstal *ESP8266*.

#### 2.9.6 *Power Supply*

*Power supply* merupakan alat yang digunakan sebagai sumber tegangan komponen maupun alat yang digunakan dalam rumah pintar ini. *Power supply*

yang di gunakan untuk rumah pintar ini, merupakan *power supply* dengan spesifikasi AC 110 V-260 V 50 Hz- 60 Hz dan DC 12 V / 5A. Gambar 2.6 menunjukkan bentuk fisik dari *power supply*.



Gambar 2.6 *Power supply*

### 2.9.7 *Sensor Passive Infrared Receiver (PIR)*

Sensor PIR adalah sensor yang digunakan untuk mendeteksi keberadaan manusia. Jika seseorang yang bergerak pada area sensor PIR, sensor tersebut langsung bisa menangkap bias sinar inframerah pasif yang terpancar dari tubuh manusia serta memiliki ukuran yang tidak sama dengan lingkungannya. Hal ini menjadikan material dalam *pyroelectric* langsung bereaksi dan menghasilkan arus listrik yang timbul dari energi panas yang sebelumnya dihasilkan oleh sinar inframerah. Lalu alat lainnya yaitu circuit amplifier menjadikan arus tersebut semakin bertambah kuat, kemudian arus itu dibandingkan lagi dengan komparator yang membuat output dapat dihasilkan.

*Output* pada sensor PIR hanya memberikan dua jenis logika yaitu high dan low. High untuk sistem yang mendeteksi adanya gerakan sedangkan low untuk kondisi sensor PIR tidak mendeteksi. Apabila manusia ada dibagian depan sensor PIR tetapi hanya berdiam diri saja, sensor PIR bias menghitung ukuran panjang gelombang yang muncul dari tubuh manusia itu adalah konstan. Ukuran panjang gelombang ini membuat energi panas yang ada dikondisikan sama dengan keadaan yang berada disekitarnya, sehingga sensor PIR tidak akan menimbulkan reaksi apapun juga Gambar 2.7 merupakan gambar dari sensor PIR



Gambar 2.7 *Sensor PIR*

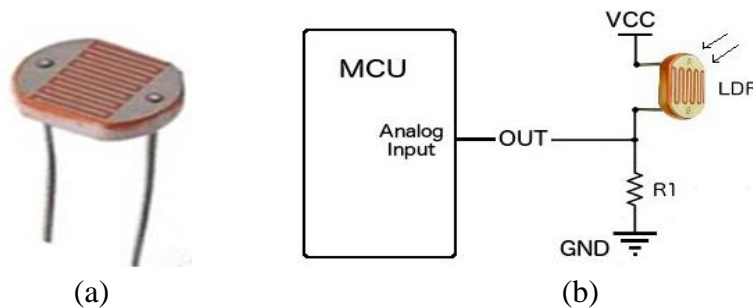
Sensor PIR memiliki karakterisasi sebagai berikut :

1. Tegangan *Catu Daya* : 4.7 – 12 VDC
2. Jangkauan deteksi sensor : 5 meter pada sudut 0 derajat
3. *Output* sensor tegangan *High* : 5 VDC
4. *Output* lebar pulsa : 0.5 s

### 2.9.8 Sensor LDR (*Light Dependent Resistor*)

Sensor *LDR* merupakan suatu jenis resistor yang peka terhadap cahaya, nilai resistansi *LDR* akan berubah-ubah sesuai dengan intensitas cahaya yang diterima. Jika *LDR* tidak terkena cahaya nilai tahanan akan menjadi besar (sekitar 10MΩ) dan jika terkena cahaya nilai tahanan akan menjadi kecil (sekitar 1MΩ).

Prinsip kerja dari sensor ini adalah mengubah *energy* dari foton menjadi electron, umumnya satu foton dapat membangkitkan satu electron. Bentuk dan simbol *LDR* dapat dilihat pada Gambar 2.8.



Gambar 2.8 Sensor *LDR*

(a) Bentuk fisik Sensor *LDR*, (b) Skematik rangkaian Sensor *LDR*

Untuk menghitung tegangan keluaran pada rangkaian *LDR* digunakan persamaan berikut.

$$V_o = \frac{LDR}{LDR + R_1} V_{cc} \dots \dots \dots (2.5)$$

dimana:

$V_o$  = Tegangan keluaran (Volt)

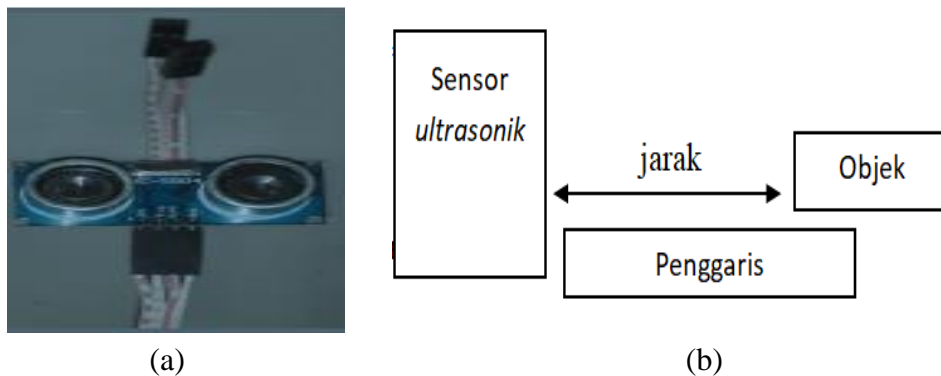
$LDR$  = Resistansi *LDR* (Ohm)

$R_1$  = Resistor (Ohm)

$V_{cc}$  = Tegangan masuk (Volt)

### 2.9.9 Sensor *Ultrasonik SRF04*

Modul *HC-SR04* merupakan modul sensor *ultrasonik* yang memiliki fungsi utama sebagai pengukur jarak. Modul ini terdiri atas sepasang *transduser* dengan empat pin, yaitu pin suplai tegangan (*Vcc*), *pin trigger*, *pin echo*, dan *pin ground*. Modul akan memulai pengukuran saat diberi sinyal pulsa *trigger* sepanjang 10  $\mu$ s, di mana *transmitter* akan mengirimkan gelombang *ultrasonik* yang akan diterima kembali oleh *receiver* saat gelombang tersebut mengenai objek dan memantul. Adapun diagram skematik dan bentuk fisik sensor *ultrasonik* dapat dilihat pada Gambar 2.9.



Gambar 2.9 Sensor *ultrasonik*

(a) Bentuk fisik Sensor *Ultrasonik*, (b) Pengujian Sensor *Ultrasonik*

Keterangan :

Objek = Kedalaman Air

Pengukuran jarak berbasis *ultrasonik* dapat dilakukan dengan dua metode yaitu

1. Perhitungan waktu tempuh

Pada medium rambat udara, gelombang *ultrasonik* memiliki kecepatan tempuh 340 m/s. Berdasarkan nilai tersebut, pengukuran jarak dapat dilakukan dengan mengamati waktu tempuh gelombang dari *transmitter* hingga diterima oleh *receiver*. Mengingat jarak yang ditempuh gelombang bersifat bolak-balik, maka perhitungan jarak metode ini berdasarkan *datasheet* sensor adalah:

$$Distance = \frac{340 \times t}{2} \text{ cm} \dots \dots \dots (2.6)$$

2. Perhitungan jumlah pulsa

*Transmitter* akan mengeluarkan deretan pulsa *burst* dengan nilai tertentu saat diberikan pulsa *trigger*. Untuk setiap satu sentimeter, jumlah pulsa yang dihasilkan adalah 29 pulsa, sehingga jarak tempuh total dapat dihitung berdasarkan jumlah pulsa yang diterima oleh *receiver*. Seperti halnya metode pertama, jalan tempuh deretan pulsa bersifat bolak-balik sehingga perhitungannya berdasarkan *datasheet* menjadi:

$$Distance = \frac{pulsecount}{58} cm.....(2.7)$$

Keterangan :

Distance = Jarak ( cm)

t = Waktu (*second*)

*Pulsecount* = Banyak Pulsa

### 2.9.10 Buzzer

*Buzzer* adalah sebuah komponen elektronika yang berfungsi untuk mengubah getaran listrik menjadi getaran suara. Pada dasarnya prinsip kerja *buzzer* hampir sama dengan loudspeaker, jadi *buzzer* juga terdiri dari kumparan yang terpasang pada diafragma dan kemudian kumparan tersebut dialiri arus sehingga menjadi elektromagnet, kumparan tadi akan tertarik ke dalam atau keluar, tergantung dari arah arus dan polaritas magnetnya, karena kumparan dipasang pada diafragma maka setiap gerakan kumparan akan menggerakkan diafragma secara bolak-balik sehingga membuat udara bergetar yang akan menghasilkan suara. Bentuk fisik dari *buzzer* dapat dilihat pada Gambar 2.10.



Gambar 2.10 *Buzzer*

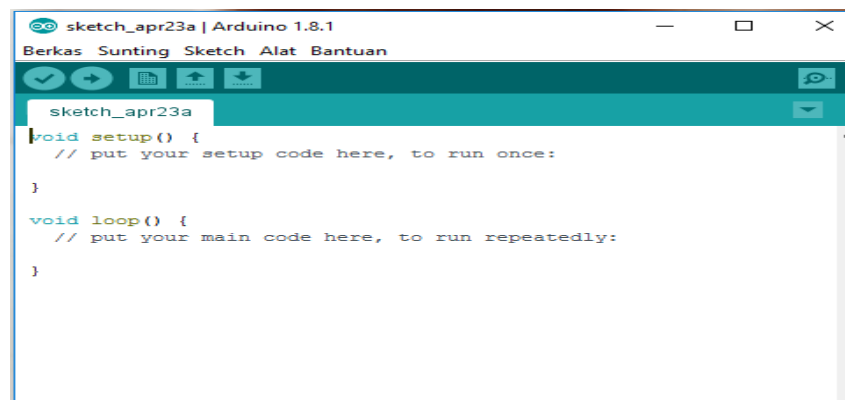
## 2.10 Perangkat Lunak (*Software*)



### 2.10.1 Software Arduino IDE Version 1.8.1

*IDE (Integrated Development Enviroment)* merupakan perangkat lunak yang telah disiapkan oleh *arduino* bagi para perancang untuk melakukan berbagai proses yang berkaitan dengan pemrograman *arduino*. Perangkat lunak disediakan secara gratis dan bisa didapatkan secara resmi *arduino* yang bersifat *open-source*. *IDE* ini juga mendukung berbagai system operasi seperti *Windows*, *Mac*, dan *Linux*. *Arduino IDE* dibuat dari bahasa pemrograman *JAVA* dan juga dilengkapi dengan library *C/C++* yang biasa disebut wiring yang membuat operasi *input* dan *output* lebih mudah[17].

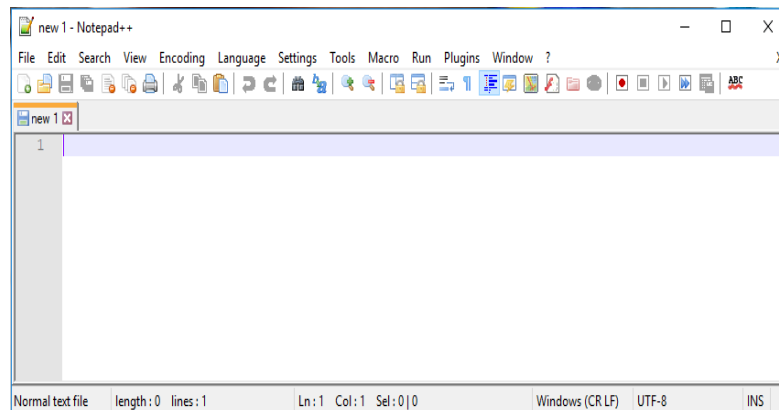
Pada *software arduino IDE*, terdapat semacam *message box*, berwarna hitam yang berfungsi menampilkan status, seperti pesan *error*, *compile*, dan *upload program*. Di bagian kanan paling bawah *software arduinoIDE*, menunjukkan *board* yang terkonfigurasi beserta *COM ports* yang digunakan. Gambar 2.11 dibawah ini merupakan tampilan dari *software arduino IDE*.



Gambar 2.11 Tampilan dari *software arduino IDE*

### 2.10.2 Notepad ++

*Notepad ++* adalah salah satu text editor yang paling banyak digunakan untuk menuliskan sebuah program. selain karena ringan dan mudah digunakan. *Notepad ++* menggunakan komponen scintilla untuk dapat menampilkan dan menyunting teks dan berkas kode sumber berbagai bahasa pemrograman yang berjalan diatas disistem operasi *Microsoft Windows*. Beberapa daftar bahasa program yang didukung oleh *Notepad ++* adalah *C*, *C++*, *Java*, *C#*, *XML*, *HTML*, *PHP*, *Javascript* dan lainnya. Gambar 2.12 tampilan awal *notepad ++*.



Gambar 2.12 Tampilan awal *notepad ++*

### 2.10.3 XAMPP

*XAMPP* merupakan singkatan dari :

*X* : *multiplatform bisa Windows, Linux, Mac OS, maupun Solaris*

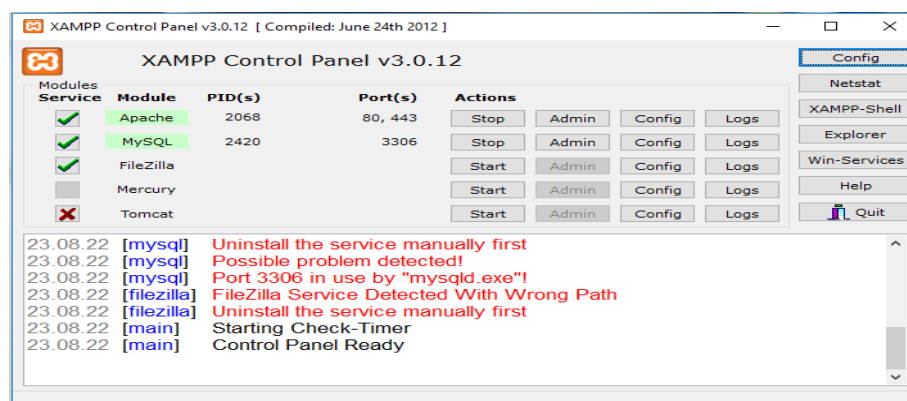
*A* : *Apache HTTP Server*

*M* : *MySQL database server*

*P* : *PHP Scripting Language*

*P* : *Perl Scripting Language*

*XAMPP* adalah tool yang menyediakan paket perangkat lunak ke dalam satu buah paket. Dalam paketnya sudah terdapat *Apache (web server)*, *MySQL(database)*, *PHP(server-side scripting)*, *Perl*, *FileZilla FTP server*, *PhpMyAdmin* dan berbagai pustaka bantu lainnya. Dengan meninstal *XAMPP* maka tidak perlu lagi melakukan instalasi dan konfigurasi *web server Apache*, *PHP* dan *MySQL* secara manual. *Xampp* akan menginstal dan mengonfigurasikannya secara otomatis[15]. Tampilan awal *XAMPP* dapat dilihat dari Gambar 2.13.

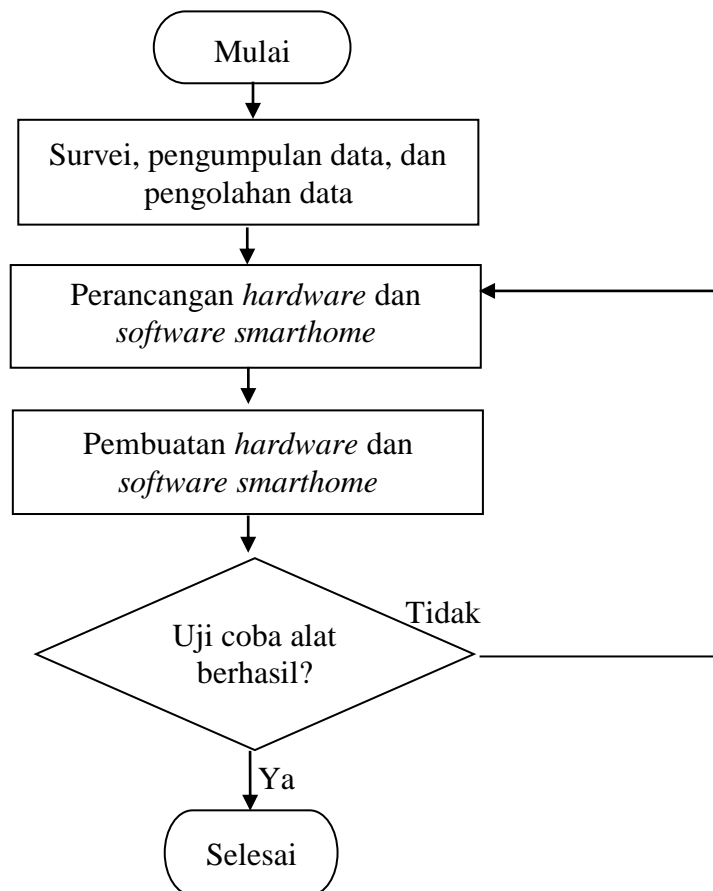


Gambar 2.13 Tampilan awal *XAMPP*

### BAB III METODE PELAKSANAAN

#### 3.1 *Flow Chart* Pembuatan Alat

Ada beberapa metode dalam pelaksanaan pembuatan proyek akhir ini, yaitu survei dan pengumpulan data, perancangan *hardware* dan *software*, pembuatan *hardware* dan *software*, serta uji coba. Metode-metode pelaksanaan proyek akhir ini bertujuan agar memudahkan pembuatan dan ketika melakukan tindakan menjadi lebih terarah dan terkontrol dalam pembuatan proyek akhir sehingga target-target yang diharapkan dapat tercapai. Adapun pelaksanaan pembuatan proyek akhir ini sesuai dengan *flowchart* pada Gambar 3.1 dibawah ini.



Gambar 3.1 *Flowchart* Metode Pelaksanaan Proyek Akhir

Dari gambar di atas dapat dijelaskan metode penelitian proyek akhir ini meliputi beberapa data:

### **3.2 Survei, Pengumpulan Data dan Pengolahan Data**

Survei dilakukan untuk mendapatkan data secara langsung. Pengumpulan data berfungsi untuk mendapatkan daftar peralatan dan komponen yang digunakan dalam pembuatan proyek akhir ini. Setelah itu, pengolahan data untuk mengetahui prinsip kerja dari komponen yang digunakan seperti sensor *PIR*, *ultrasonik*, *LDR*, *esp8266*. dan cara menghitung manajemen pemakaian energi listrik. Pada studi ini, pengumpulan data didapatkan dengan dua cara, yaitu:

#### **3.2.1 Metode Pengumpulan Data Primer**

Pengumpulan data dengan cara primer yaitu sumber data yang didapat langsung dari narasumber. Pengumpulan data dengan cara ini adalah bimbingan dengan dosen pembimbing. Metode ini untuk mendukung metode pemecahan masalah, dari pembimbing dan pihak-pihak lain agar tujuan yang diharapkan dapat tercapai.

#### **3.2.2 Metode Pengumpulan Data Sekunder**

Pengumpulan data dengan cara sekunder yaitu sumber data yang didapat secara tidak langsung dari narasumber. Pengumpulan data dengan cara ini adalah metode studi pustaka, yang mana metode ini adalah proses mengumpulkan data-data yang diinginkan dari berbagai referensi untuk menunjang dalam dalam sistem pembuatan proyek akhir ini.

Adapun referensi proyek akhir ini diambil dari beberapa jurnal, diantaranya adalah:

- [1] Mansykur Fauzan, Prasetiyowati, “Aplikasi Rumah Pintar (Smarthome) Pengendali Peralatan Elektronik Rumah Tangga Berbasis Web”, jurnal Teknologi Informaso dan Ilmu teknologi Vol 3, No. 1 Maret 2016, hlm 51-58.
- [2] Burange, A. W., & Cheung, H.(2012). Review of Internet of Things in Development of Smart Cities with Data Management & Privacy.

- [3] Epiyana & Tampubolon, Hetty Agustina (2017). MANAJEMEN PEMAKAIAN ENERGI LISTRIK UNTUK RUMAH PINTAR.
- [4] Rozaq Abdul, Yulita Noor, “Efesinsi Energi Smarthome (Ruamah Pintar) Berbasis Remote Relay dan LDR(Light Dependent Resistan” , jurnal Simestri, vol 8No 1 April 2017,
- [5] Pengertian dan Rumus Daya Listrik beserta satuanya [Online], diakses 15 Mei 2018, available:www.berpendidikan.com/2015/10/pengertian-dan-Rumus-Daya Listrik-beserta-satuanya.html?m=1

### **3.3 Rancangan dan Pembuatan *Hardware Smarthome***

Jika tahap pengumpulan data telah dikerjakan, maka tahap selanjutnya adalah perancangan desain *hardware* yang bertujuan agar mempermudah dan mempercepat proses pembuatan *hardware*, dan untuk mencapai target dengan baik. Rancangan untuk *smarthome* di desain menggunakan *microsoft visio* untuk merancang konstruksi dari *smarthome* dan *box panel* yang digunakan untuk meletakkan komponen-komponen kontrol dari *smarthome*. Setelah tahap rancangan selesai, maka dilanjutkan dengan tahapan pembuatan *hardware smarthome* dan *box panel* sesuai dengan hasil rancang *hardware* sebelumnya.

### **3.4 Rancangan dan Pembuatan *Software Smarthome***

Setelah perancangan dan pembuatan *hardware* selesai, maka langkah selanjutnya adalah merancang dan membuat *software smarthome* yang berfungsi sebagai sistem kontrol dari alat yang akan dibuat pada proyek akhir ini. *Software* yang digunakan dalam proyek akhir ini adalah sebagai berikut :

1. *Software ArduinoIDE Version 1.8.1* digunakan untuk mengontrol sistem kerja *smarthome*.
2. *Software Notepad ++* digunakan untuk menulis dan editor program php.
3. *Software Xampp Version 3.0.12* digunakan untuk pembuatan *database*.

### **3.5 Uji Coba *Smarthome***

Proses pengujian dilakukan sebagai tolak ukur apakah *smarthome* bekerja sesuai dengan fungsi yang diinginkan. Dengan demikian, setiap kekurangan dan

kesalahan dalam pembuatannya dapat dievaluasi sedini mungkin. Jika diperlukan, maka apabila sistem kerja *smarthome* tidak sesuai dengan yang diinginkan, proses pembuatan dapat diulang kembali mulai perancangan *hardware* dan pemrograman *software*. Uji coba yang dilakukan yaitu terhadap *power supply*, *Arduino Mega 2560*, modul *relay*, *Esp8266*, *Modul PZEM-004T*, sensor *PIR*, sensor *LDR*, sensor *ultrasonic*, dan *buck converter*. Setelah uji coba setiap komponen selesai, maka dilakukan uji coba antara *hardware* dan *software smarthome* secara keseluruhan yaitu uji coba terhadap kontrol *on/off* lampu dan monitoring pembacaan daya yang digunakan pada halaman *web*.

### **3.7 Analisa Data**

Jika alat yang sudah di uji coba sudah selesai dan masih terdapat kekurangan maka dilakukanlah analisis data untuk mendapatkan hasil yang sesuai dengan tuntutan supaya alat bisa berfungsi dengan baik.

### **3.8 Pembuatan Laporan**

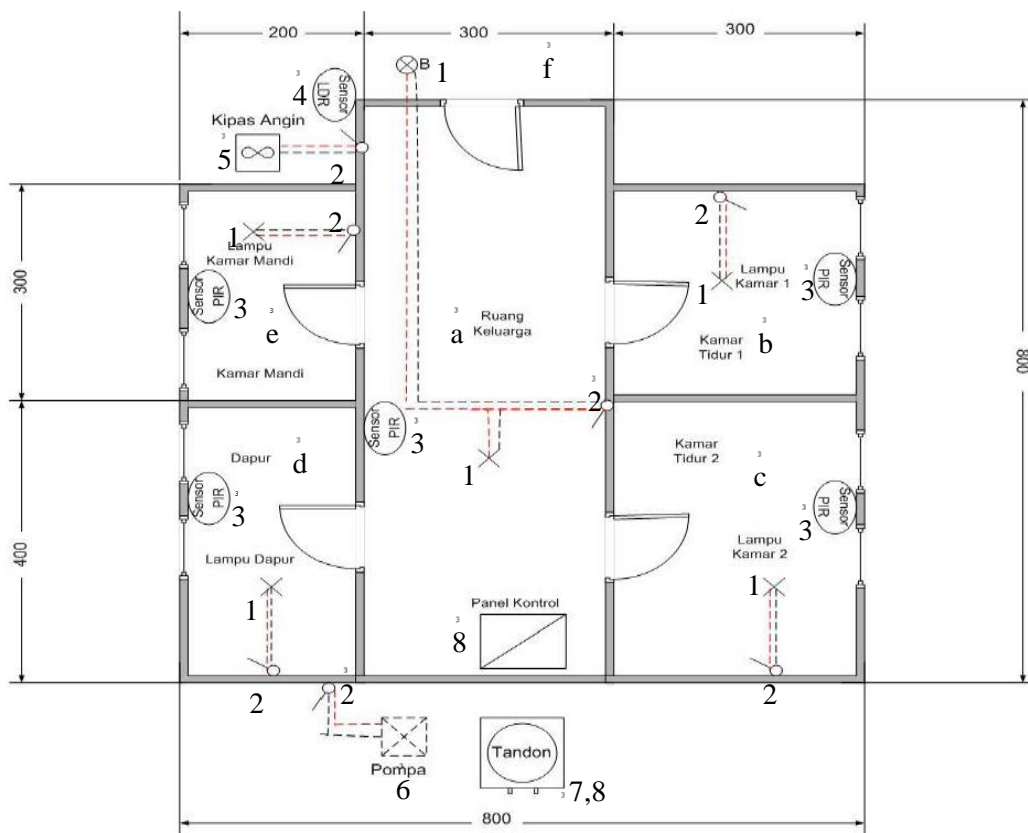
Tahapan ini adalah tahap terakhir apabila pembuatan alat telah memenuhi tuntutan yang ingin dicapai. Laporan berisi tahapan pembuatan alat dari awal hingga akhir secara mendetail.

## BAB IV PEMBAHASAN

Pada bab ini akan dijelaskan tentang proses pembuatan alat. Dalam proses pembuatan alat ada beberapa tahapan, yaitu: rancangan dan pembuatan *hardware*, pembuatan program, uji coba serta analisa.

### 4.1. Perancangan dan Pembuatan Konstruksi

Perancangan *Smarthome* yang dibuat memiliki ukuran keseluruhan dengan panjang 800mm x lebar 800mm x tinggi 400mm. Kerangka bagian-bagian rumah dibuat menggunakan kayu dengan ukuran 3cm x 3cm dan triplek dengan ketebalan 5mm untuk bagian dinding dan 6mm bagian atap serta lantai. Rancangan denah 2D dari rumah pintar dapat dilihat pada Gambar 4.1 dibawah ini.



Gambar 4.1 Rancangan denah 2D dari rumah pintar

Simbol:

a : Ruang Keluarga    b : Kamar 1    c : Kamar 2

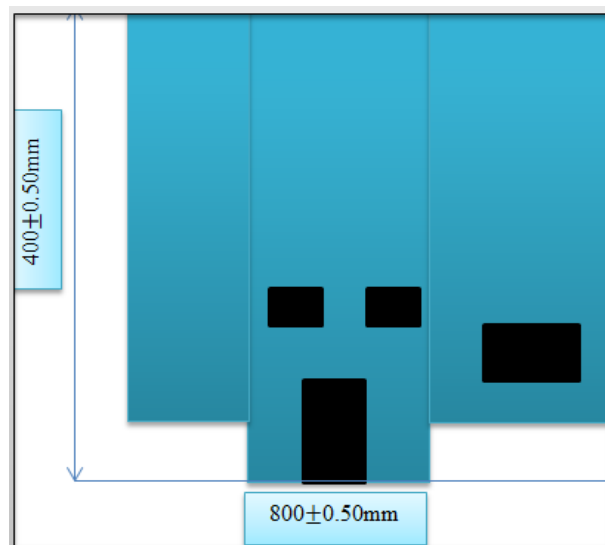
d : Dapur                    e : WC                    f : Teras

1 : Lampu    2 : Saklar    3 : Sensor PIR    4 : Sensor LDR

5 : Kipas    6 : Pompa    7 : Tandon    8 : Sensor Ultrasonik

#### 4.1.1 Perancangan *Smarthome*

Konstruksi *smarthome* didesain menggunakan *solidworks* dengan ukuran keseluruhan 80cm x 80cm x 40 cm. Konstruksi rumah dibuat menggunakan kayu dengan ukuran 3cm<sup>2</sup> x 3cm<sup>2</sup> dan triplek dengan ketebalan 5 mm untuk bagian kerangka dan 6 mm bagian atap serta lantai. Gambar 4.2 merupakan gambar tampak depan dari *smarthome*.



Gambar 4.2 Desain tampak depan *smarthome*

#### 4.1.2 Pembuatan *Smarthome*

Adapun langkah-langkah dalam membuat *smarthome* ini yaitu :

- membeli dan mempersiapkan peralatan yang akan digunakan
- mengukur dan memotong bagian-bagian *smarthome* sesuai dengan rancangan *smarthome*
- Assembly* bagian-bagian yang telah dipotong menggunakan kayu dan paku.
- mencecet dan memasang komponen atau alat yang dibutuhkan



#### 4.1.2 Hasil *Smarthome*

*Smarthome* yang telah dibuat memiliki 5 ruangan, 1 buah tandon dan sebuah kipas angin. Hasil pembuat konstruksi dari *smarthome* ini dapat dilihat maka hasil dari pembuatan kontruksi dapat dilihat pada Gambar 4.3, dimana bagian gambar (a) merupakan bagian tampak atas dari *smarthome*, gambar (b) merupakan bagian tampak depan dari *smarthome* dan gambar (c) merupakan bagian tampak belakang dari *smarthome*.



(a)



(b)



(c)

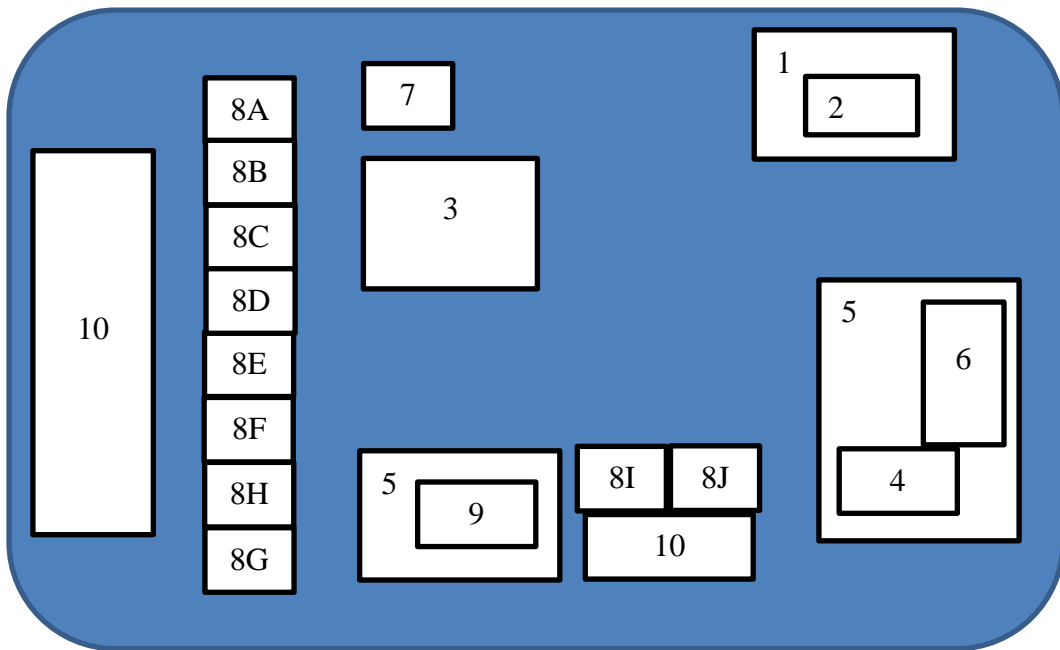
Gambar 4.3 Hasil *smarthome*  
(a) Tampak atas *smarthome*, (b) Tampak depan *smarthome*  
(c) Tampak belakang *smarthome*

#### 4.2 *Box Panel*

##### a. Perancangan *Box Panel*

*Box panel* merupakan kontrol utama dari *smarthome*, yang di dalamnya terdapat beberapa komponen yang akan digunakan. *Box panel* yang digunakan

adalah wadah makan berbahan plastik yang dimodifikasi sedemikian rupa agar sesuai dengan kebutuhan pada proyek akhir ini yang memiliki ukuran panjang 26cm, lebar 15cm, dan tinggi 10cm. Rancangan *box panel* dapat dilihat pada Gambar 4.4 dibawah ini.



Gambar 4.4 Perancangan *box panel*

Keterangan :

1. *Power supply*
2. Modul PZEM-004T
3. Arduino Mega 2560
4. *Pin Header*
5. PCB
6. *Buck Converter*
7. Modul *ESP8266*
8. Modul Relay 8 channel (A-J)
9. Resistor 10K dan 1K
10. Terminal Blok

b. *Pembuatan Box Panel*

*Box panel* kontrol dibuat dari kombinasi wadah makanan berbahan *plastic* yang dimodifikasi sedemikian rupa sehingga sesuai dengan desain dan kebutuhan.

Langkah-langkah dalam pembuatan box panel adalah sebagai berikut:

1. Mengebor bagian untuk penempatan alat
2. Memasang dan *wiring* alat

c. Hasil dari *Box Panel*

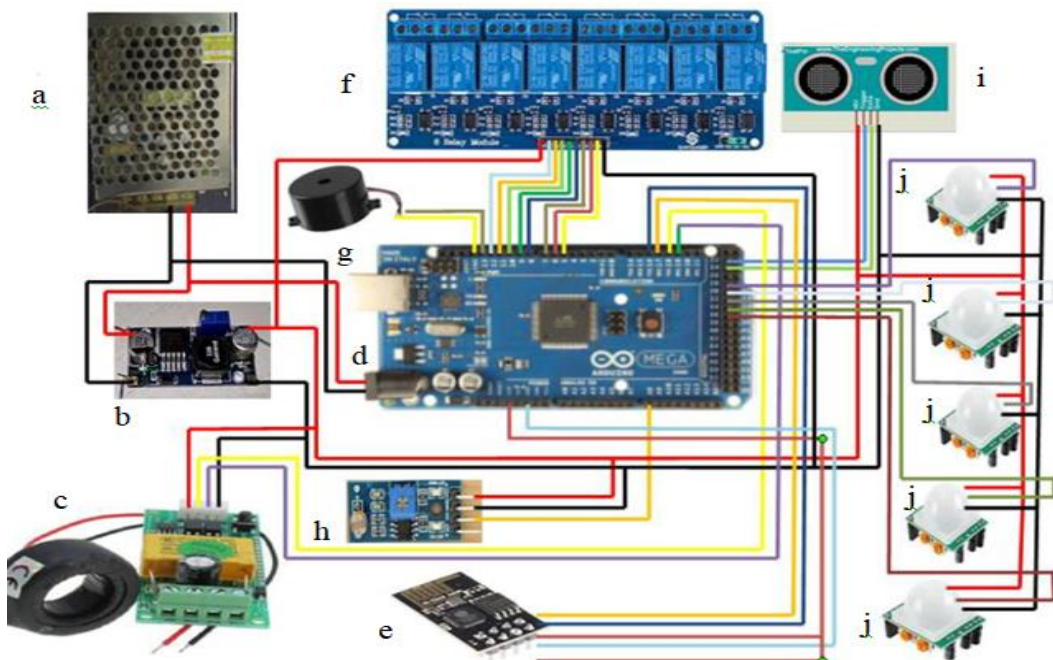
Berdasarkan rancangan yang telah dibuat, maka tahap selanjutnya adalah pemasangan komponen pada box dengan cara mengebor box tersebut. Adapun box panel yang telah selesai pemasangan komponen dapat dilihat pada Gambar 4.5 dibawah ini.



Gambar 4.5 Hasil dari Box Panel

### 4.3 Bagian *Hardware* Elektrik

Perancangan Sistem *hardware* elektrik dapat dilihat pada pada rangkaian Gambar 4.6 yang merupakan pemasangan komponen-komponen pada arduino.



Gambar 4.6 Pemasangan komponen-komponen pada arduino

Simbol:

a : Power Supply                      b : Buck Converter    c : Modul PZEM-004T  
d : Arduino Mega 2560    e : ESP 8266                      f : Modul Relay  
g : Buzzer                              h : Sensor PIR                      i : Sensor Ultrasonik  
j : Sensor LDR

Berdasarkan rangkaian yang ditunjukkan pada gambar 4.6, menunjukkan bahwa ada banyak komponen yang digunakan dalam proyek akhi ini diantaranya adalah *arduino mega 2560*, *esp8266*, *buck converter*, *power supply*, sensor *PIR*, sensor *LDR*, dan sensor *ultrasonik*, modul *relay*, modul *PZEM-004T*, dan *buzzer*. Namun, disini yang akan dijabarkan adalah komponen-komponen yang menjadi input dan output serta komunikasi nya saja.

#### 4.3.1 *Arduino Mega2560*

##### a. Perancangan *Arduino Mega2560*

*Arduino Mega 2560* berfungsi sebagai perangkat kontrol dalam mengontrol sistem kerja *smarthome*. Ada banyak komponen yang digunakan dalam pengontrolan *smarthome* ini, sehingga menggunakan *arduino mega2560* yang memiliki banyak pin-pin. Adapun pin-pin yang digunakan pada *arduino mega2560* dapat dilihat pada Tabel 4.1 dibawah.

Tabel 4.1 *Pin* pada *arduino mega2560*

No	Komponen	Pin pada <i>Arduino Mega2560</i>
1	Sensor <i>PIR</i>	A3, A4, A6, A5, A7
2	Modul <i>Relay</i>	D52, D50, D48, D46, D42
3	Sensor <i>LDR</i>	A0
4	Sensor <i>Ultrasonik</i>	A1
5	<i>Buck Converter</i>	A2
6	<i>ESP8266</i>	RX0, TX0
7	Sensor <i>PZMET-004</i>	A8

##### b. Pembuatan *Arduino Mega2560*

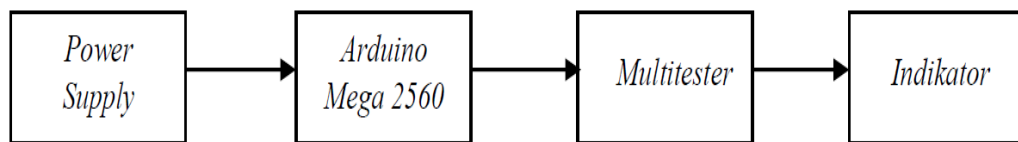
*Arduino* yang digunakan adalah *arduino mega2560* yang siap pakai, hal tersebut dilakukan karena untuk menghemat waktu dalam pengerjaan proyek akhir. Bentuk fisik dari *arduino mega2560* dapat dilihat pada Gambar 4.7 dibawah ini.



Gambar 4.7 Bentuk fisik arduino mega 2560

c. Pengujian *Arduino Mega2560*

Untuk mengetahui apakah *board arduino* ini masih berfungsi dengan baik atau tidak dapat dilakukan dengan melakukan pengujian terhadap *port-port* dan pin-pinnya melalui suatu program sederhana. Program sederhana dibuat untuk melakukan pengukuran tegangan tiap-tiap pin dengan cara memberikan *logic 0 (low)* dan *logic 1 (high)*. Untuk mengetahui apakah rangkaian tersebut bekerja maka dipergunakan *LED* sebagai indikator. Gambar 4.8 merupakan blok diagram pengujian *arduino*.



Gambar 4.8 Blok diagram pengujian arduino

Pengujian pada *port arduino* dilakukan dengan 2 tahapan yaitu tahapan pertama pengujian *port-port arduino* dari pin D0-D13 diberikan *logic high* dan *logic low*. Adapun hasil uji pada tahapan pertama dapat dilihat dari Tabel 4.2 dibawah ini.

Tabel 4.2 Hasil pengukuran *active high* pin arduino mega2560

PIN	PIN <i>Arduino Mega2560</i>			Persentase Error
	Percobaan ke-1	Percobaan ke-2	Percobaan ke-3	
D0	5 V	5 V	5 V	0%
D1	5 V	5 V	5 V	0%

PIN	PIN Arduino Mega2560			Persentase Error
	Percobaan ke-1	Percobaan ke-2	Percobaan ke-3	
D2	5 V	5 V	5 V	0%
D3	5 V	5 V	5 V	0%
D4	5 V	5 V	5 V	0%
D5	5 V	5 V	5 V	0%
D6	5 V	5 V	5 V	0%
D7	5 V	5 V	5 V	0%
D8	5 V	5 V	5 V	0%
D9	5 V	5 V	5 V	0%
D10	5 V	5 V	5 V	0%
D11	5 V	5 V	5 V	0%
D12	5 V	5 V	5 V	0%
D13	5 V	5 V	5 V	0%

Tegangan yang diinginkan yaitu sebesar 5 Volt, namun dari data di atas didapat hasil pengukuran *active high* pin *Arduino Mega 2560* yaitu 5 Volt. Untuk menentukan persentase *error*nya dapat dilakukan dengan menggunakan formula sebagai berikut:

$$\left| \frac{\text{Nilai Pengukuran} - \text{Nilai Perhitungan}}{\text{Nilai Pengukuran}} \right| \times 100\% \dots \dots \dots (4.1)$$

Adapun persentase error dari *port-port arduino* adalah sebagai berikut.

$$\left| \frac{5V - 5V}{5V} \right| \times 100\% = 0\%$$

Nilai *output* yang keluar dari *Arduino Mega2560* tidak memiliki *error* sehingga mengindikasikan bahwa *board arduino* ini masih baik untuk digunakan.

Tabel 4.3 Hasil pengukuran *active low* pin *arduino mega 2560*

PIN	PIN Arduino Mega2560			Persentase Error
	Percobaan ke-1	Percobaan ke-2	Percobaan ke-3	
D0	0 V	0 V	0 V	0%
D1	0 V	0 V	0 V	0%
D2	0 V	0 V	0 V	0%
D3	0 V	0 V	0 V	0%

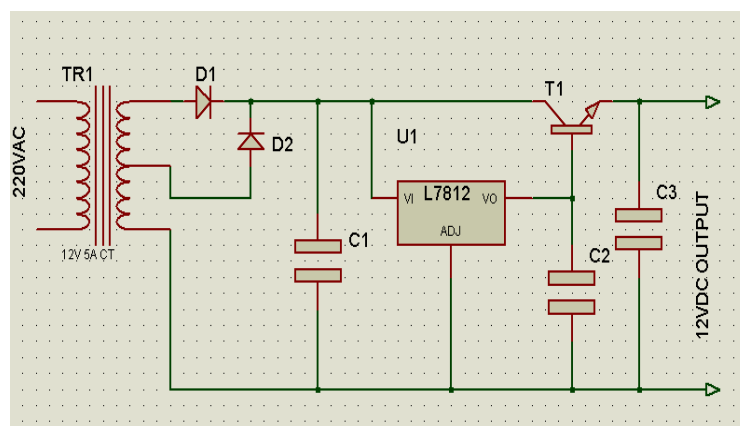
PIN	PIN Arduino Mega2560			Persentase Error
	Percobaan ke-1	Percobaan ke-2	Percobaan ke-3	
D4	0 V	0 V	0 V	0%
D5	0 V	0 V	0 V	0%
D6	0 V	0 V	0 V	0%
D7	0 V	0 V	0 V	0%
D8	0 V	0 V	0 V	0%
D9	0 V	0 V	0 V	0%
D10	0 V	0 V	0 V	0%
D11	0 V	0 V	0 V	0%
D12	0 V	0 V	0 V	0%
D13	0 V	0 V	0 V	0%

Hasil pengukuran *active low* pin *arduino mega2560* yaitu 0 volt dan errornya 0% yang mengindikasikan bahwa Arduino Mega 2560 baik untuk digunakan.

### 4.3.2 Power Supply

#### a. Perancangan Power Supply

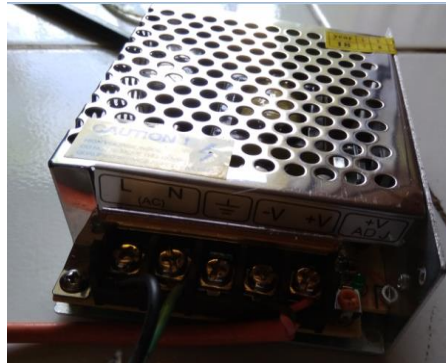
*Power supply* yang digunakan pada *smarthome* memiliki spesifikasi AC 110V-220V 50Hz-60Hz dan DC 12V/5A. Rangkaian skematik *power supply* dapat dilihat pada Gambar 4.9 di bawah ini.



Gambar 4.9 Rangkaian skematik *power supply*

b. Pembuatan *Power Supply*

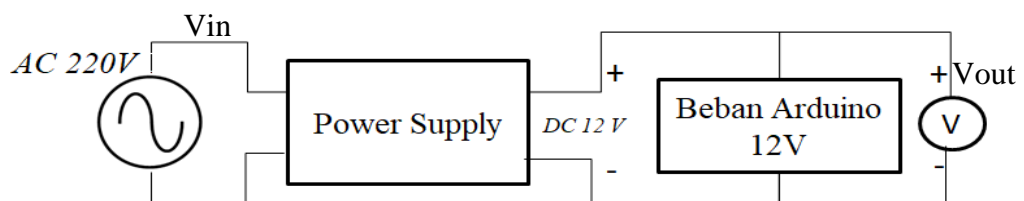
*Power Supply* yang digunakan adalah *power supply* siap pakai. Adapun bentuk fisik dari *power supply* dapat dilihat pada Gambar 4.10.



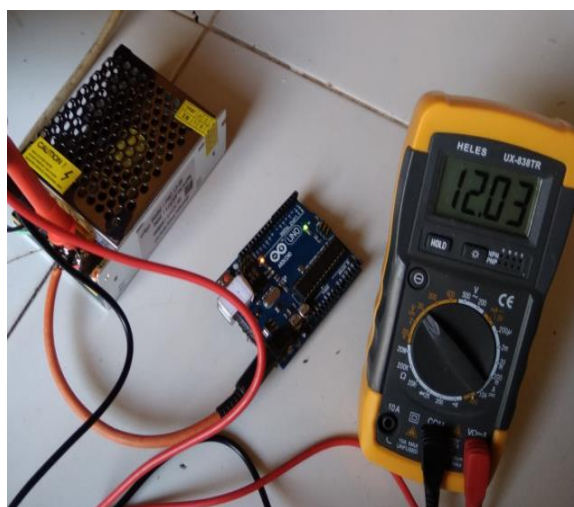
Gambar 4.10 Bentuk fisik dari *power supply*

c. Pengujian *Power Supply*

Pengujian yang dilakukan yaitu pengujian terhadap tegangan *output* menggunakan multimeter. Blok diagram pengujian dan hasil pengukuran *power supply* dapat dilihat pada gambar 4.11 dan gambar 4.12 di bawah ini.



Gambar 4.11 Blok diagram pengujian *power supply*



Gambar 4.12 Hasil pengukuran *power supply*



Hasil pengujian dari *power supply* dengan *Vout* sebagai tegangan yang terukur pada beban bisa dilihat pada tabel 4.4 dibawah ini.

Tabel 4.4 Pengujian *power supply*

No	Hasil Yang Diinginkan	Hasil dari Pengukuran	Presentase <i>Error</i>
1	12 V	12,03V	0,25%
2	12V	12,03V	0,25%
3	12V	12,03V	0,25%
4	12V	12,03V	0,25%

Dari data di atas didapat hasil pengukuran *output* tegangan yaitu 12,03 Volt. Untuk menentukan persentase *error power supply* dapat dilakukan dengan menggunakan formula sebagai berikut:

$$\left| \frac{\text{Nilai Pengukuran} - \text{Nilai Perhitungan}}{\text{Nilai Pengukuran}} \right| \times 100\% \dots \dots \dots (4.2)$$

Adapun persentase error dari *power supply* adalah sebagai berikut.

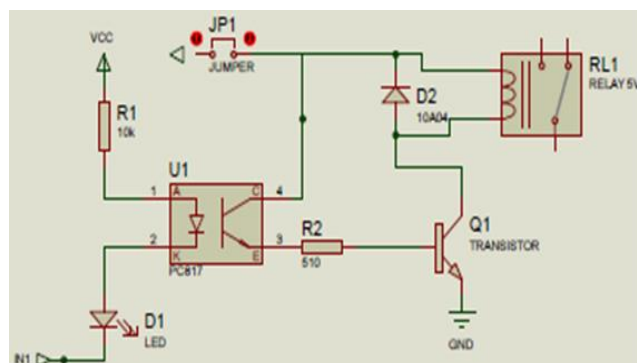
$$\left| \frac{12,03 \text{ V} - 12 \text{ V}}{12,03 \text{ V}} \right| \times 100\% = 0,25 \%$$

Persentase 0,25 % masih dalam jangkauan toleransi dan *power supply* dalam kondisi baik.

### 4.3.3 Modul *Relay*

#### a. Perancangan Modul *Relay*

Modul *relay* yang akan digunakan sebanyak enam buah. Rangkaian skematik modul *relay* dapat dilihat pada Gambar 4.13 berikut



Gambar 4.13 Rangkaian skematik relay

b. Pembuatan Modul *Relay*

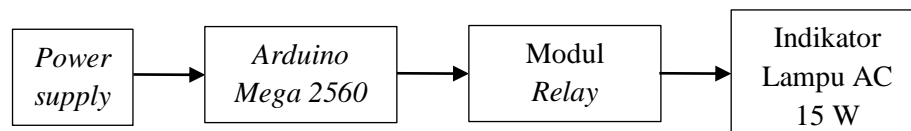
Modul *relay* yang digunakan adalah modul *relay* 8 channel yang terdiri dari 5 buah rangkaian *relay* yang digunakan untuk mengontrol kondisi tiap lampu pada tiap ruangan. Modul *relay* yang sudah dirangkai, dapat dilihat pada Gambar 4.14 di bawah ini.



Gambar 4.14 Modul *relay* yang sudah dirangkai

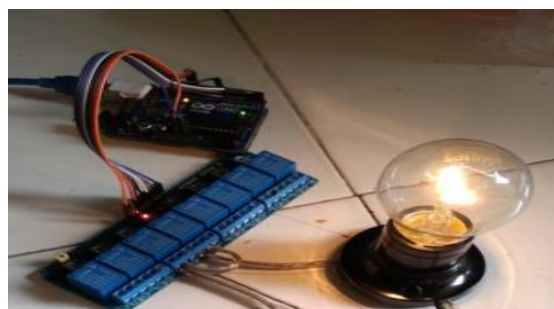
c. Pengujian Modul *Relay*

Pengujian yang dilakukan pada *relay* adalah dengan menguji konektifitas modul *relay* dengan indikator lampu dan untuk menguji kecocokan pin masukan *relay* modul dengan program. blok diagram pengujian *relay* dapat dilihat pada Gambar 4.15 di bawah ini



Gambar 4.15 Blok diagram pengujian modul *relay*

Berdasarkan hasil pengujian modul *relay* tersebut, lampu dapat *ON* dan *OFF* sehingga modul *relay* ini dipastikan dapat berfungsi dengan baik. Untuk lebih jelas, dapat dilihat pada Gambar 4.16 di bawah ini.

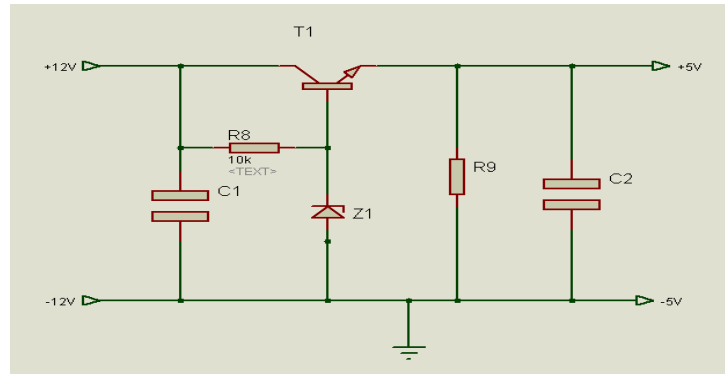


Gambar 4.16 Hasil pengujian modul *relay*

### 4.3.4 Buck Converter

#### a. Perancangan Buck Converter

*Buck converter* digunakan untuk menurunkan tegangan input dari 12 volt ke 5 volt. Gambar 4.17 merupakan rangkaian skematik *buck converter*, dapat dilihat di bawah ini.



Gambar 4.17 Rangkaian skematik *buck converter*

#### b. Pembuatan Buck Converter

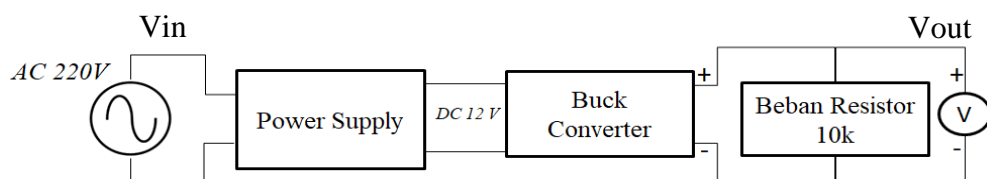
*Buck converter* mendapatkan sumber tegangan 12 volt dari *power supply*, kemudian selanjutnya keluarnya 5 volt yang bisa diatur polaritasnya. Gambar 4.18 merupakan *buck converter hardware*.



Gambar 4.18 *Buck converter*

#### c. Pengujian Buck Converter

Pengujian *buck converter* dilakukan adalah dengan mengukur tegangan output dari sebelumnya 12 volt menjadi 5 volt. Adapun blok diagram pengujian dan hasil pengukuran tegangan *output buck converter* dapat dilihat pada gambar 4.19 dan gambar 4.20.



Gambar 4.19 Blok diagram pengujian *buck converter*



Gambar 4.20 Pengukuran tegangan *output buck converter*

Adapun hasil pengukuran dari *buck Converter* dapat dilihat pada tabel 4.5 di bawah ini :

Tabel 4.5 Hasil pengukuran *buck converter*

No	<i>V Input (Power Supply)</i>	<i>V Output (Buck Converter)</i>	Hasil dari Pengukuran	Presentase Error
1	12 V	5 V	5 V	0%
2	12 V	5 V	5 V	0%
3	12 V	5 V	5 V	0%

Dari data di atas didapat hasil pengukuran *buck converter* yaitu 5 volt. Untuk menentukan persentase *error buck converter* dapat dilakukan dengan menggunakan formula sebagai berikut:

$$\left| \frac{\text{Nilai Terukur} - \text{Nilai Perhitungan}}{\text{Nilai Terukur}} \right| \times 100\% \dots \dots \dots (4.4)$$

Adapun persentase error dari *buck converter* adalah sebagai berikut.

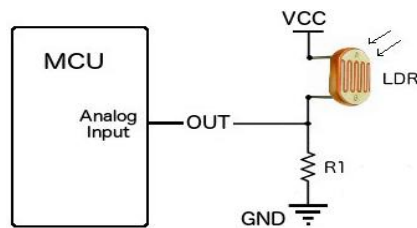
$$\left| \frac{5 V - 5 V}{5 V} \right| \times 100\% = 0\%$$

Persentase errornya yang 0% mengindikasikan bahwa *buck converter* baik untuk digunakan.

### 4.3.5 Sensor *LDR*

#### a. Perancangan Sensor *LDR*

Sensor *LDR* digunakan untuk mendeteksi cahaya, pada proyek akhir ini sensor *LDR* diletakkan diluar ruangan. Gambar 4.21 rangkaian skematik dari sensor *LDR*.



Gambar 4.21 Rangkaian skematik sensor *LDR*

b. Pembuatan Sensor *LDR*

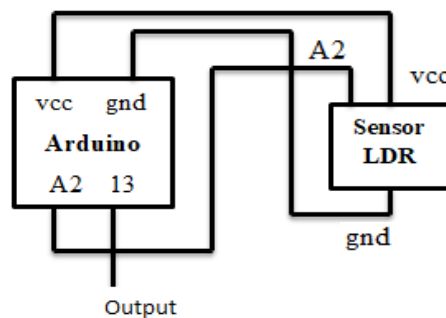
Sensor *LDR* yang digunakan adalah sensor *LDR* yang siap di pakai dengan tutup setengah lingkaran berbahan plastik. Bentuk fisik dari modul sensor *LDR* dapat dilihat pada gambar 4.22.



Gambar 4.22 Modul sensor *LDR*

c. Pengujian Sensor *LDR*

Pengujian dilakukan pada pagi hari dan menjelang malam dengan data ADC didapat berdasarkan nilai resistansi pada sensor *LDR* yang terhubung pada pin A2 Arduino. Pin A2 berfungsi untuk membaca nilai *ADC* sensor *LDR*. Gambar 4.23 merupakan skematik pengujian sensor *LDR*.



Gambar 4.23 Rangkaian pengujian sensor *LDR*

Pengujian sensor *LDR* untuk mendeteksi intensitas cahaya dengan membaca nilai *ADC* pada sensor *LDR* dengan menggunakan program sebagai berikut.

```

int LDR = A5;
int nilai_LDR;
const int r6 = 7;//teras
int LDR1;

void setup() {
  pinMode(r6, OUTPUT);}

void loop(){
  LDR_A ();
  Serial.println();
  Serial.print("Nilai ADC : ");
  Serial.println(nilai_LDR);
  Serial.println();
  Serial.println("Kondisi lampu : ");
  if (LDR1 > 500 ) {
    digitalWrite(r6, HIGH);
    Serial.print("led ON");}
  else {
    digitalWrite(r6, LOW );
    Serial.print("led OFF");
    Serial.println();
    Serial.println();
  }

void LDR_A (){
  int nilai_LDR;
  nilai_LDR = analogRead(LDR);
  nilai_LDR = LDR1;}

```

} inisialisasi pin yang digunakan  
 } inisialisasi input/output  
 } Membaca nilai pada Sensor LDR untuk lampu on  
 } Membaca nilai pada Sensor LDR untuk lampu off  
 } Membaca nilai adc pada sensor

Hasil pengujian sensor LDR untuk mendeteksi intensitas cahaya dapat dilihat pada tabel 4.6.

Tabel 4.6 Hasil pengujian sensor LDR

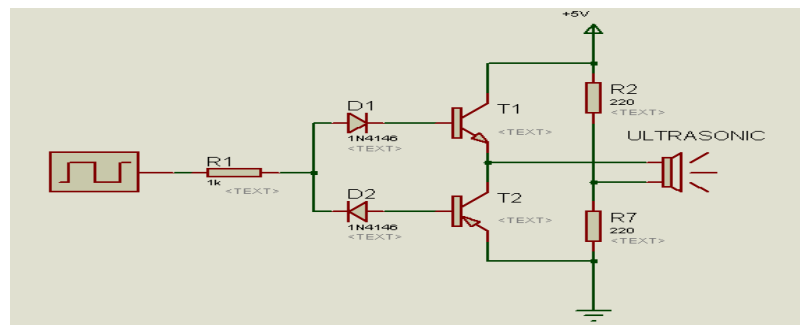
No	Pengujian Hari Ke-	Jam (WIB)	Cuaca	Nilai ADC
1	Hari Ke-1	05.30	Cerah	300-350
		06.00	Cerah	330-355
		17.30	Cerah	500-600
		18.00	Hujan	735-800
2	Hari Ke-2	05.30	Hujan	310-355
		06.00	Hujan	340-450
		17.30	Berawan	500-600
		18.00	Berawan	700-800
3	Hari Ke-3	05.30	Berawan	300-352
		06.00	Berawan	330-410
		17.30	Hujan	530-650
		18.00	Hujan	700-800

Berdasarkan hasil pengujian sensor didapat hasil keluaran yaitu ketika sensor terkena cahaya terang maka nilai *ADC* akan lebih kecil dibandingkan saat tidak terkena cahaya (gelap) hal ini menyebabkan nilai *ADC* saat pagi hari akan lebih kecil dibandingkan pada saat malam hari dikarenakan sinyal *ADC* pada LDR dipengaruhi intensitas cahaya yang diberikan.

#### 4.3.6 Sensor *Ultrasonik*

##### a. Perancangan Sensor *Ultrasonik*

Sensor *ultrasonik* digunakan sebagai sensor pendeteksi ketinggian air dalam tendon air. Gambar 4.24 merupakan rangkaian skematik dari sensor *ultrasonik*.



Gambar 4.24 Rangkaian skematik sensor *ultrasonik*

##### b. Pembuatan Sensor *Ultrasonik*

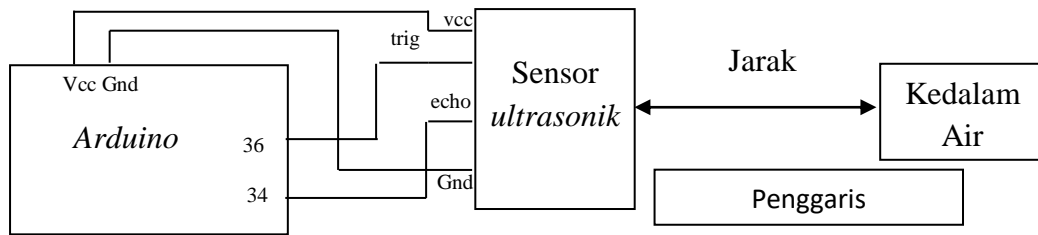
Sensor *ultrasonik* yang digunakan adalah sensor *ultrasonik* jenis *HC-SR04* yang siap pakai dengan inputan 5V. Bentuk fisik sensor *ultrasonik*, dapat dilihat pada gambar 4.25.



Gambar 4.25 Sensor *ultrasonik*

##### c. Pengujian Sensor *Ultrasonik*

Pengujian sensor *ultrasonik* adalah dengan menguji pendeteksian jarak sensor dibandingkan dengan pengukuran jarak menggunakan penggaris. Pada pengujian ini sensor diuji dengan objek berupa air yang berhuna untuk mengukur air dalam tendon. Skema pengujian sensor *ultrasonik* dapat dilihat pada Gambar 4.26.



Gambar 4.26 Skema pengujian sensor *ultrasonik*

Pengujian sensor ultrasonik dilakukan dengan membuat program untuk mengecek apakah sensor ultrasonik bisa atau tidak digunakan. Berikut adalah list program arduino untuk pengecekan.

<code>#define TRIG_PIN 36</code>	}	inisialisas
<code>#define ECHO_PIN 34</code>		i pin yang digunakan
<code>const uint8_BATAS_BAWAH=0;</code>	}	batas aktif sensor
<code>const uint8_BATAS_ATAS= 15;</code>		
<code>int sonar(void);</code>		
<code>cost int led7=6;</code>		
<code>void setup () {</code>	}	inisialisasi input/output
<code>  pinMode(ECHO_PIN, INPUT);</code>		
<code>  pinMode(TRIG_PIN, OUTPUT); }</code>		
<code>  Serial.println();</code>		
<code>  TANDON ();</code>		
<code>  Serial.println();</code>		
<code>  serial.println (); {</code>	}	Batasan on/off sensor
<code>  if (jarak&lt;BATAS_BAWAH){</code>		
<code>  digitalWrite(led7, HIGH);</code>		
<code>  }</code>		
<code>void TANDON ()</code>	}	Menampilkan jarak yang terukur sensor
<code>  { jarak=sonar();</code>		
<code>  Serial.print("Jarak dalam satuan cm:");</code>		
<code>  Serial.println();}</code>		
<code>  int sonar(void){</code>		
<code>  digitalWrite (TRIG_PIN, HIGH);</code>		
<code>  delayMicroseconds (10);</code>		
<code>  digitalWrite(TRIG_PIN, LOW);</code>		
<code>  int jarak=pulseIn(ECHO_PIN, HIGH);</code>		
<code>  return(jarak=jarak/58);</code>		
<code>  }</code>		

Setelah program selesai, maka pengujian dapat dilakukan dengan membandingkan nilai yang keluar pada *serial monitor software* arduino dengan nilai yang diukur menggunakan penggaris. Adapun hasil pengujian keduanya dapat dilihat pada Gambar 4.27 dan Tabel 4.7.





Gambar 4.27 Pengujian sensor ultrasonik pada tandon air

Tabel 4.7 Hasil perbandingan sensor ultrasonik dengan penggaris

No.	Hasil Sensor Ultrasonik	Hasi Penggaris	Persentase Error
1	9cm	9cm	0%
2	13cm	13cm	0%
3	18cm	18,3cm	1,63%
4	20cm	20,4cm	1,96%
5	24cm	24,4cm	1,96%
6	25cm	25,4cm	1,96%
7	26cm	26,4cm	1,96%
8	27cm	27,4cm	1,96%
	Rata-rata :		1,42%

Berdasarkan tabel diatas, nilai sensor ultrasonik tidak 100 % akurat. Dan untuk menentukan *persentase error* pada perbandingan hasil sensor ultrasonik dan penggaris (hasil yang terukur) dapat dihitung dengan menggunakan formula dibawah ini:

$$\left| \frac{\text{Nilai Terukur} - \text{Nilai Pengukuran}}{\text{Nilai Terukur}} \right| \times 100\% \dots\dots\dots(4.1)$$

Adapun *persentase error* sensor ultrasonik adalah:

- $\left| \frac{18,3 \text{ cm} - 18 \text{ cm}}{18,3 \text{ cm}} \right| \times 100\% = 1,63\%$
- $\left| \frac{20,4 \text{ cm} - 20 \text{ cm}}{20,4 \text{ cm}} \right| \times 100\% = 1,96\%$

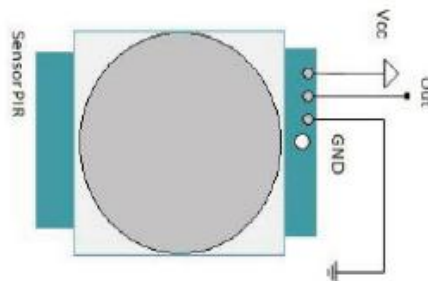
Berdasarkan hasil perhitungan diatas, hasil pengukuran jarak oleh sensor ultrasonik dengan *persentase error* berkisar dari 0-1,96 % atau rata-rata *error* nya

yaitu 1,42 %. *Error* yang terjadi bisa dikarenakan posisi sensor atau posisi benda yang dideteksi kurang pas. Selain itu, bisa juga dikarenakan waktu penerimaan gelombang suara yang lambat akan mempengaruhi hasil pengukuran.

#### 4.3.7 Sensor *PIR*

##### a. Perancangan Sensor *PIR*

Sensor *PIR* digunakan untuk kontrol lampu otomatis sebagai pendeteksi objek yang bergerak. Didalam proyek akhir ini terdapat 5 sensor *PIR*. Adapun rangkaian skematik dari sensor *PIR* dapat dilihat pada Gambar 4.28.



Gambar 4.28 Rangkaian skematik sensor *PIR*

##### b. Pembuatan Sensor *PIR*

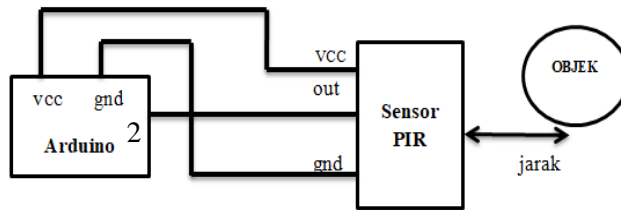
Sensor *PIR* yang digunakan adalah sensor *PIR* siap pakai yang memiliki tegangan inputan 5V. Gambar 4.29 merupakan Sensor *PIR* yang sudah dirangkai.



Gambar 4.29 Sensor *PIR* yang sudah dirangkai

##### c. Pengujian Sensor *PIR*

Pengujian sensor *PIR* dilakukan dengan objek manusia, hewan, tumbuhan, batu, dan kertas. Gambar 4.30 yang merupakan skema rangkaian pengujian.



Gambar 4.30 Rangkaian pengujian sensor *PIR*

Pengujian sensor PIR untuk mendeteksi benda yang bergerak indikator apabila sensor PIR aktif maka lampu AC akan menyala dan dengan menggunakan program berikut ini.

```
int ledPin = 13; int inputPin = 2;
int pirState = LOW;
void setup() {
  pinMode(ledPin, OUTPUT);
  pinMode(inputPin, INPUT);
  Serial.begin(9600);}
void loop(){
  data = digitalRead(inputPin);
  if(data == HIGH){
    digitalWrite(ledPin, HIGH);
    if(pirState == LOW){
      Serial.println("ada objek!");
      pirState = HIGH;}}else {
    digitalWrite(ledPin, LOW); // LED padam
    if(pirState == HIGH){
      Serial.println("tidak ada objek!");
      pirState = LOW;}}
```

} inisialisasi pin yang digunakan

} Rumus Fungsi sensor PIR

Pada pengujian sensor PIR ini akan mengukur konektifitas sensor pada tiap ruangan. Pengujian ruangan pada sensor PIR dapat dilihat pada gambar 4.31, tabel 4.8, dan tabel 4.9.



Gambar 4.31 Pengujian ruangan dengan sensor PIR:

(a) Saat lampu kamar 2 off, (b) Saat lampu kamar 2 on

Keterangan:

1 = lampu

2 = Sensor PIR

3 = Objek (Manusia)

Tabel 4.8 Hasil pengujian sensor PIR dengan berbagai objek

No	Objek	Jarak dan Kondisi Lampu			
		Diagonal	Lurus	Samping	Lampu
1	Manusia	37cm	37cm	26cm	ON
2	Hewan	27cm	26cm	15cm	ON
3	Tumbuhan	21cm	20cm	16cm	OFF
4	Batu	21cm	20cm	16cm	OFF
5	Kertas	21cm	20cm	16cm	OFF

Tabel 4.9 Hasil pengujian sensor PIR

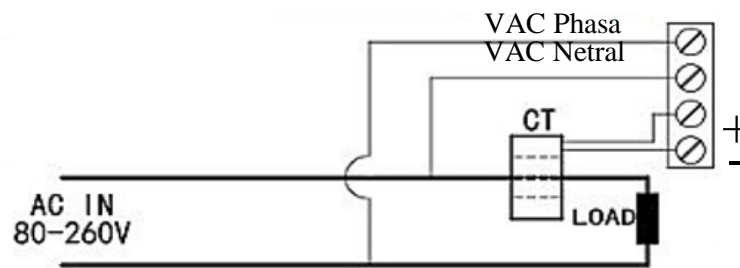
Sensor PIR di	Objek Manusia	Respon Waktu (s)	Jarak dan Kondisi Lampu			
			Diagonal	Lurus	Samping	Lampu
Ruang Keluarga	Ada	5	37cm	37cm	26cm	ON
Kamar Tidur1	Ada	10	27cm	26cm	15cm	ON
Kamar Tidur2	Ada	10	35cm	27cm	23cm	ON
Kamar Mandi	Ada	10	30cm	16cm	20cm	ON
Dapur	Ada	10	21cm	20cm	16cm	ON

Berdasarkan tabel diatas sensor hanya bisa mendeteksi manusia dan hewan pada saat objek bergerak. Respon atau sensitivitas dari sensor PIR berbeda-beda, sehingga menyebabkan indikator lampu AC menyala tidak secara langsung. Sensivitas sensor PIR dapat diatur dengan memutar sensitivitas searah jarum jam. Pendeteksian sensor PIR juga bisa berpengaruh tergantung pemasangan sensor itu sendiri.

### 4.3.8 Modul *PZEM-004T*

#### a. Perancangan Modul *PZEM-004T*

Modul *PZEM-004T* adalah sensor yang dapat mengukur nilai arus, nilai tegangan, nilai daya dan nilai  $\cos \phi$ . Pada modul *pzem-004t* memiliki 4 pin yang terhubung ke arduino yaitu pin RX dihubungkan dengan pin RX2 pada arduino, pin TX dihubungkan dengan pin TX2 arduino, pin VCC dihubungkan ke pin 5 volt arduino, dan pin GND dihubungkan ke pin GND arduino. Gambar 4.32 merupakan rangkian skematik dari modul *PZEM-004T* yang diperoleh dari data sheet.



Gambar 4.32 Rangkaian skematik modul *PZEM-004T*

#### b. Pembuatan Modul *PZEM-004T*

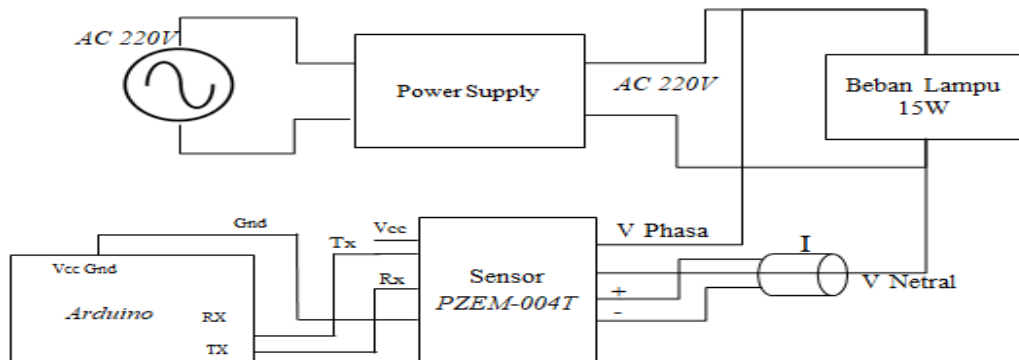
Modul *PZEM-004T* yang digunakan adalah modul yang siap pakai, hal ini dikarenakan untuk menghemat penggunaan waktu penyelesaian tugas akhir. Modul *PZEM-004T* yang sudah dirangkai, dapat dilihat pada gambar 4.33 dibawah ini.



Gambar 4.33 Modul *PZEM-004T* yang sudah dirangkai

#### c. Pengujian Modul *PZEM-004T*

Pengujian modul *pzem-004t* dilakukan berdasarkan gambar 4.34 yang merupakan skema rangkaian pengujian.



Gambar 4.34 Rangkaian pengujian sensor *PZEM-004T*

Keterangan:

$I$  = Klem Ampere sensor *PZEM-004T*

$V$  = Voltmeter sensor *PZEM-004T* (V phasa dan V netral)

Pengujian modul *pzem-004t* untuk membaca nilai arus, nilai daya, nilai dan nilai tegangan dilakukan dengan upload program yang ada dibawah ini.

```

PZEM004T pzem(&Serial1);
IPAddress ip(192, 168, 1, 1);
void setup() {
  Serial.begin(9600);
  pzem.setAddress(ip);
}

void loop(){
  float v; float iM;
  float cos; DAYA(); delay (10);
}

void DAYA() {
  float v = pzem.voltage(ip);
  if (v < 0.0) v = 0.0;
  Serial.print("Tegangan : ");
  Serial.print(v);
  Serial.println("V ");

  float iM = pzem.current(ip);
  if (iM >= 0.0) {
    Serial.print("Arus : ");
    Serial.print(iM);
    Serial.println("A "); }

  float p = pzem.power(ip);
  if (p >= 0.0) {
    Serial.print("Daya : ");
    Serial.print(p);
  }
}

```

} inialisasi pin yang digunakan

} Menampilkan nilai tegangan

} Menampilkan nilai arus

} Menampilkan nilai daya

```

Serial.println("W "); }
float cOS = p / (v * iM);
if (cOS >= 0.0) {
Serial.print("Cos Phi : ");
Serial.println(cOS);}}

```

} Menampilkan  
nilai  $\cos \theta$

Setelah rangkaian skematik dan program sudah benar, maka pengujian dilakukan dengan sebuah lampu yang menjadi beban lampu dan hasil pembacaan nilai arus, nilai tegangan, dan nilai daya dapat dilihat di serial monitor dari software arduino dan alat ukur wattmeter. Hasil pengujian modul PZEM-004T dapat dilihat pada gambar 4.35 dan tabel 4.10.



Gambar 4.35 Pembacaan daya dengan *wattmeter*

Tabel 4.10 Pengujian Modul PZEM-004T

Beban	Tegangan		Arus		Daya	
	Pengukuran	Sensor	Pengukuran	Sensor	Pengukuran	Sensor
1 Lampu 15 W	217,6V	217,6V	0,10A	0,10A	20W	20W
2 Lampu 15 W	217,6V	217,6V	0,16A	0,16A	34W	35,4W
3 Lampu 15 W	217,6V	217,6V	0,23A	0,24A	50W	51,1W
4 Lampu 15 W	217,6V	217,6V	0,3A	0,3A	64W	66W
5 Lampu 15 W	217,6V	217,6V	0,38A	0,39A	83W	85W
Rata-rata	217,6V	217,6V	1,17A	1,19A	50,3W	51,5W

Berdasarkan tabel diatas, terdapat perbedaan data *serial monitor* dan *wattmeter* pada modul *pzem-004t*. Untuk menentukan persentase *error* modul *pzem-004t* dapat dilakukan dengan menggunakan formula sebagai berikut:

$$\left| \frac{\text{Nilai Terukur} - \text{Nilai Pengukuran}}{\text{Nilai Terukur}} \right| \times 100\% \dots \dots \dots (4.2)$$

Adapun *persentase error* sensor pzem-004t adalah:

- Perhitungan Daya rata-rata

$$\left| \frac{20 \text{ W} - 15 \text{ W}}{20 \text{ W}} \right| \times 100\% = 2,5\%$$

- Perhitungan Arus rata-rata

$$\left| \frac{1,19\text{A} - 1,17\text{A}}{1,19\text{A}} \right| \times 100\% = 1,68\%$$

- Perhitungan Tegangan yang terukur dengan tegangan yang digunakan

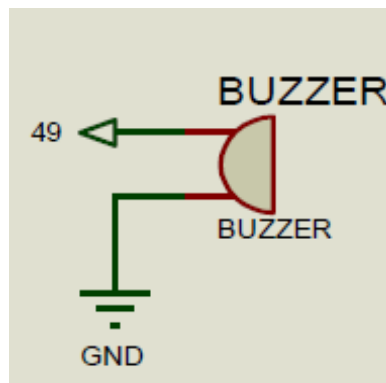
$$\left| \frac{220\text{V} - 217,6\text{V}}{220\text{V}} \right| \times 100\% = 1,09\%$$

Pada modul PZEM-004T memiliki persentase error daya 2,5%, arus 1,68% dan tegangan 1,09%.

### 4.3.9 Buzzer

#### a. Perancangan Buzzer

*Buzzer* digunakan sebagai peringatan apabila pemakaian daya sudah overload. Prinsip kerja dari *buzzer* sendiri apabila diberikan sumber tegangan pada pin positif maka *buzzer* akan mengeluarkan suara yang menandakan bahwa *buzzer* telah aktif. Rangkaian skematik *buzzer* dapat dilihat pada Gambar 4.36.



Gambar 4.36 Rangkaian skematik *buzzer*

#### b. Pembuatan Buzzer

*Buzzer* dirangkai berdasarkan rangkaian skematik yang telah dirancang sebelumnya. *Buzzer* yang sudah dirangkai dapat dilihat pada Gambar 4.37 di bawah ini.





Gambar 4.37 *Buzzer*

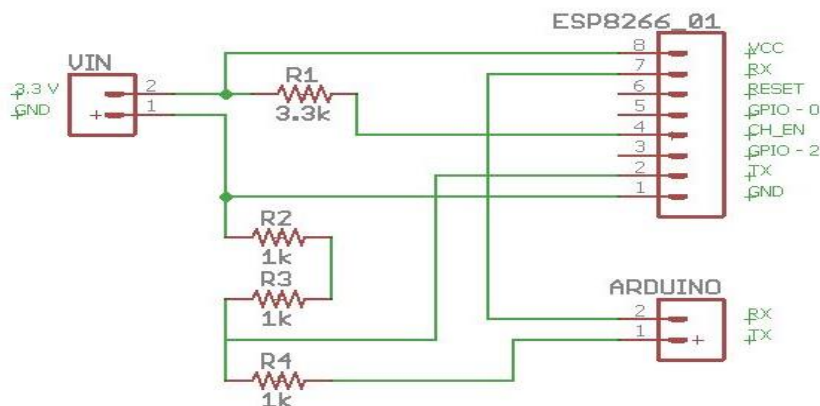
c. Pengujian *Buzzer*

Pengujian yang dilakukan adalah pengujian terhadap bunyi yang dikeluarkan. Caranya adalah dengan menghubungkan kaki pertama *buzzer* pada tegangan positif dan kaki kedua pada tegangan negatif atau ground (0V). Jika buzzer dalam kondisi baik, maka buzzer akan mengeluarkan bunyi.

**4.3.10 Modul *WiFi ESP8266***

a. Perancangan Modul *WiFi ESP8266*

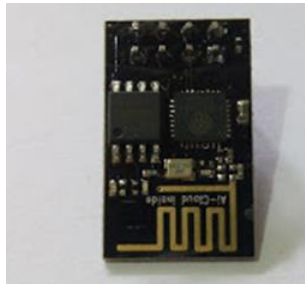
Modul *wireless ESP8266* merupakan modul *low-cost WiFi* yang digunakan sebagai komunikasi antara sistem kontrol dengan sistem *internet of things*. Gambar 4.38 merupakan rangkaian skematik dari *ESP8266* yang dirangkian dengan resistor yang memiliki resistansi sebesar 3,3K sebanyak 1 buah dan 3 buah resistor dengan dengan resistansi 1K. Fungsi dari resistor tersebut adalah untuk menstabilkan tegangan yang dikeluarkan arduino sebesar 3,3 volt.



Gambar 4.38 Rangkaian skematik modul *wifi ESP8266*

b. Pembuatan Modul *WiFi ESP8266*

Modul *WiFi ESP8266* yang digunakan adalah *ESP8266 Tipe 01* yang siap pakai dan dirangkaian dengan rangkaian pembagi tegangan. Adapun bentuk dari *ESP8266*, dapat dilihat pada Gambar 4.39 dibawah ini.



Gambar 4.39 Modul *wifi ESP8266*

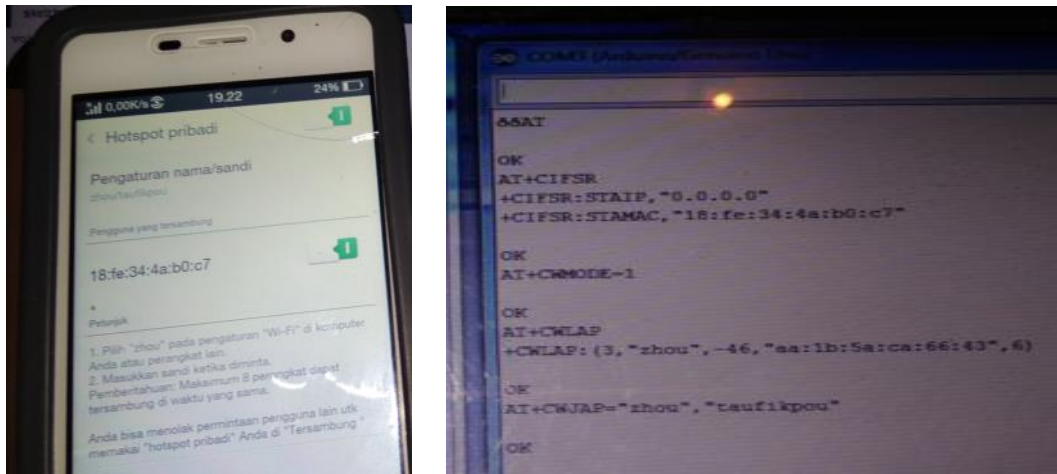
c. Pengujian Modul *WiFi ESP8266*

Pengujian pada modul *WiFi ESP8266* dilakukan adalah dengan mengkoneksikan *esp8266* dengan jaringan internet menggunakan program kosong yang di upload dengan perintah pada modul *WiFi ESP8266 (at command)*. Disini jaringan internet yang digunakan berasal dari hotspot smartphone.

Langkah-langkah pengujian koneksi modul *WiFi ESP8266* dengan *hotspot* dengan perintah *at command* menggunakan *hotspot* adalah sebagai berikut:

1. Rangkaian *ESP8266* dengan *arduino mega2560* sesuai rangkaian skematik
2. Buka *software IDE Arduino*, pasang USB *arduino mega2560* kemudian download program kosong.
3. Buka *serial monitor* pada *IDE Arduino*
4. Ketikkan "AT", AT berfungsi untuk mengecek modul terus berjalan. Jika OK lanjut ketahap selanjutnya.
5. Ketikkan "AT+CIFSR", berfungsi untuk mengetahui alamat ip lokal dari komputer yang digunakan. Jika alamat ip lokal telah tampil
6. Ketikkan "AT+CWMODE=1", berfungsi untuk menerima pancaran sinyal dari *hotspot*.
7. Ketikkan "AT+CMLAP", berfungsi untuk melihat *wifi* atau *hotspot* yang tersedia serta *connect* ke *access point*.

Hasil pengujian koneksi antara modul *WiFi ESP8266* dengan *smartphone* dapat dilihat pada gambar 4.40 dan tabel 4.11.



(a)

(b)

Gambar 4.40 Koneksi modul *wifi ESP8266* dengan *hotspot*

(a) Koneksi *esp8266* ke *hotspot* pada *smartphone*, (b) Tampilan *serial monitor esp8266* yang terhubung ke *hotspot*

Pada gambar 4.41 (a) merupakan terkoneksiya *esp8266* ke *hotspot* dari *smartphone* yaitu \*, sedangkan gambar 4.41 (b) merupakan *serial monitor* yang menampilkan *password* dan *user* yang terhubung dengan *hotspot* dengan menggunakan perintah `AT+CWJAP"zhou", "taufikpou"`.

Tabel 4.11 Pengujian Koneksi ESP8266 dengan *hotspot*

No	Objek	Jarak dan Koneksi			
		Jarak 1	Jarak 2	Jarak 3	Koneksi
1	<i>esp8266 1</i>	1 M	1,5 M	2 M	ON
2	<i>esp8266 2</i>	1 M	1,5 M	2 M	ON
3	<i>esp8266 3</i>	1 M	1,5 M	2 M	OFF
4	<i>esp8266 4</i>	1 M	1,5 M	2 M	ON
5	<i>esp8266 5</i>	1 M	1,5 M	2 M	OFF
6	<i>esp8266 1</i>	1 M	1,5 M	2 M	ON

Berdasarkan tabel diatas konektifitas dari *esp8266* dipengaruhi oleh baik atau tidaknya kondisi *esp8266* yang digunakan hal ini dikarenakan *esp8266* mudah rusak pada saat tegangan input yang diberikan melebihi tegangan komponen sehingga disarankan menggunakan rangkaian pengaman dalam menggunakan *esp8266*.

#### 4.4 *Assembly* Komponen Keseluruhan

Setelah pembuatan konstruksi dan pengujian komponen selesai, tahap selanjutnya adalah *Assembly* Komponen keseluruhan komponen menjadi satu dan diletakan pada ruangan-ruangan atau kamar-kamar dari smarthome. Gambar 4.41 merupakan hasil dari proses perakitan, dapat dilihat dibawah ini.



Gambar 4.41 Tampak atas hasil perakitan *hardware*

#### 4.5 Pembuatan Program

Proses pembuatan program menggunakan *software Integrated Development Environment (IDE) arduino* dan *Notepad ++*.

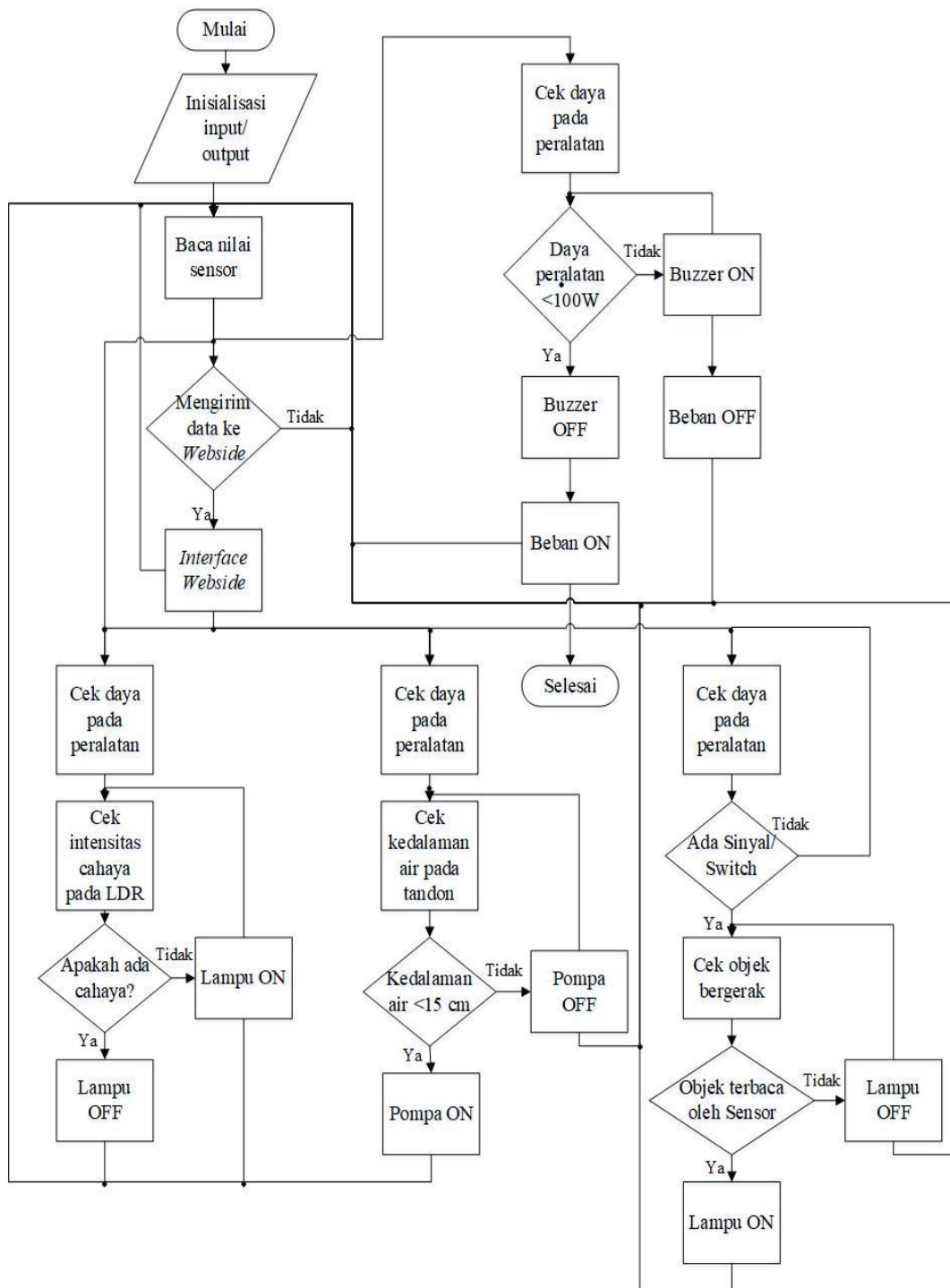
##### 4.5.1 Sistem Kontrol Pada *Smarthome*

*Software* yang digunakan pada proyek akhir ini adalah *Arduino IDE Version 1.8.1*. Adapun hal-hal yang dilakukan dalam membuat program kontrol smarthome yaitu:

1. Menyiapkan library untuk komponen-komponen yang library nya belum terinstall otomatis pada software arduino, seperti library ESP8266, library modul PZEM-004T, library sensor PIR, library sensor LDR, dan library sensor ultrasonik HC SR04.
2. Mulai membuat program untuk menjalankan setiap komponen satu persatu sesuai dengan fungsinya dalam sistem kerja smarthome.
3. Ketika program setiap komponen selesai dibuat, selanjutnya menggabungkan program-program tersebut menjadi satu kesatuan system kerja *smarthome*

4. secara keseluruhan sehingga fungsi smarthome sesuai dengan tujuan dibuatnya smarthome ini.

Adapun *flowchart* sistem kerja *smarthome* yang akan dibuat programnya dapat dilihat pada Gambar 4.42 berikut.



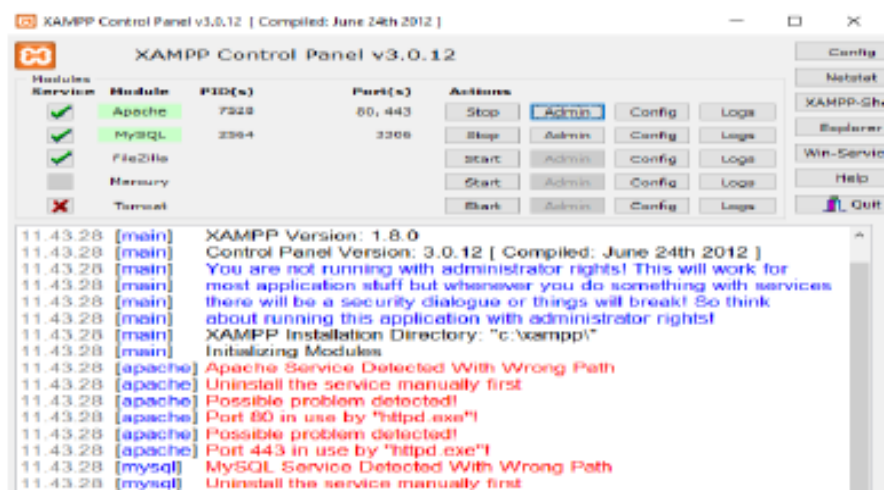
Gambar 4.42 *Flow chart* sistem kerja *smarthome*

## 4.5.2 Pembuatan Aplikasi Website

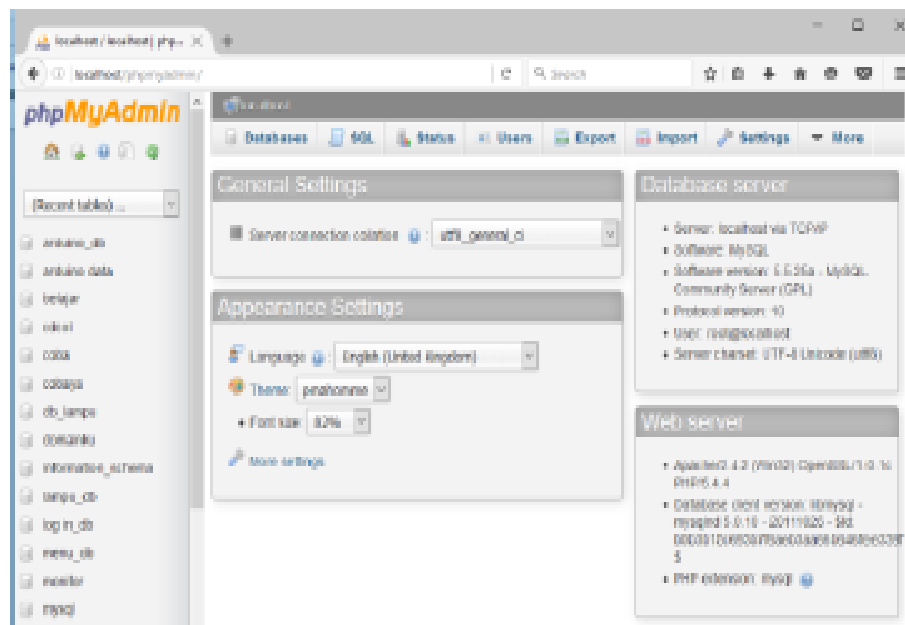
Ada beberapa tahapan proses dalam pembuatan halaman *website* yaitu:

### a. Database

*Database* digunakan untuk menyimpan berbagai data, mulai dari data pengguna (*username*, *password* dan lainnya) maupun data aplikasi. Xampp merupakan *web server* berbasis yang difungsikan sebagai wadah pembuatan database. Gambar 4.43 merupakan tampilan awal dari xampp, sedangkan Gambar 4.44 merupakan halaman pembuatan database.



Gambar 4.43 Tampilan awal XAMPP



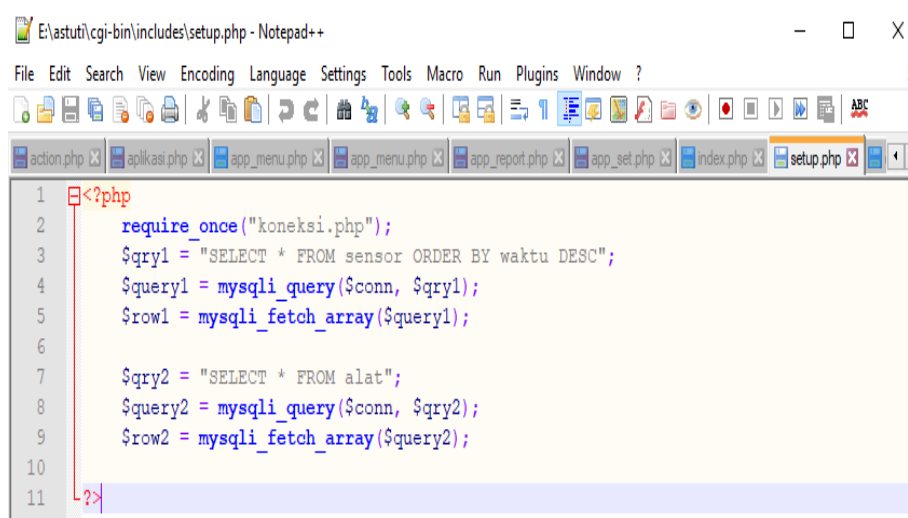
Gambar 4.44 Halaman pembuatan database

Program untuk menghubungkan antara database dengan aplikasi website dapat dilihat dibawah ini.

```
<?php
    $servername = "localhost";
    $username = "labandroid_powermeter";
    $password = "powermeterpwd";
    $dbname = "labandroid_powermeter";
// Create connection
$conn = mysqli_connect
($servername, $username, $password, $dbname);
if(!$conn){
die("KONEKSI GAGAL : ". mysqli_connect_error());}?>
```

#### b. Halaman Website

Setelah pembuatan database tahap selanjutnya adalah pembuatan dari website itu sendiri. Pembuatan *website* menggunakan bahasa pemrograman php yang pengkodeanya ditulis pada *notepad++*. Gambar 4.45 merupakan tampilan pada *notepad ++*.

The image shows a screenshot of the Notepad++ application window. The title bar reads "E:\astuti\cgi-bin\includes\setup.php - Notepad++". The menu bar includes File, Edit, Search, View, Encoding, Language, Settings, Tools, Macro, Run, Plugins, Window, and ?. The toolbar contains various icons for file operations and editing. The main text area shows PHP code with line numbers 1 through 11. The code includes a database connection setup and two SQL queries. Line 1: <?php. Line 2: require\_once("koneksi.php");. Line 3: \$qry1 = "SELECT \* FROM sensor ORDER BY waktu DESC";. Line 4: \$query1 = mysqli\_query(\$conn, \$qry1);. Line 5: \$row1 = mysqli\_fetch\_array(\$query1);. Line 6: (blank). Line 7: \$qry2 = "SELECT \* FROM alat";. Line 8: \$query2 = mysqli\_query(\$conn, \$qry2);. Line 9: \$row2 = mysqli\_fetch\_array(\$query2);. Line 10: (blank). Line 11: ?>. The code is color-coded: PHP tags are red, keywords are blue, and strings are black.

Gambar 4.45 Tampilan notepad ++

Program untuk menghubungkan antara *database* dengan aplikasi *website* dapat dilihat dilampiran.

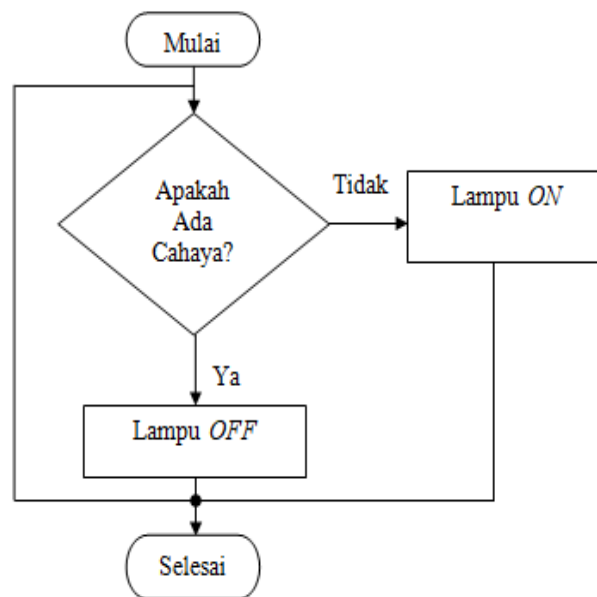
#### 4.5.2 Uji Coba Sistem Keseluruhan

Uji coba sistem keseluruhan dilakukan dengan beberapa uji coba yaitu pengujian pertama adalah pengujian lampu depan dengan sensor LDR yang bertujuan untuk mengontrol lampu depan dalam pengaktifan atau mematikan

lampu, agar dilakukan penghematan daya listrik, pengujian kedua adalah pengujian lampu setiap ruangan dengan sensor *PIR* yang bertujuan mendeteksi objek yang bergerak dalam ruangan dan pengujian ketiga adalah pengujian kontrol pompa dengan sensor ultrasonik bertujuan untuk mengaktifkan pompa dengan mendeteksi kedalaman air dalam tandon. Detail pengujiannya dapat di lihat pada penjelasan dibawah ini.

#### 1. Pengujian lampu depan dengan sensor *LDR*

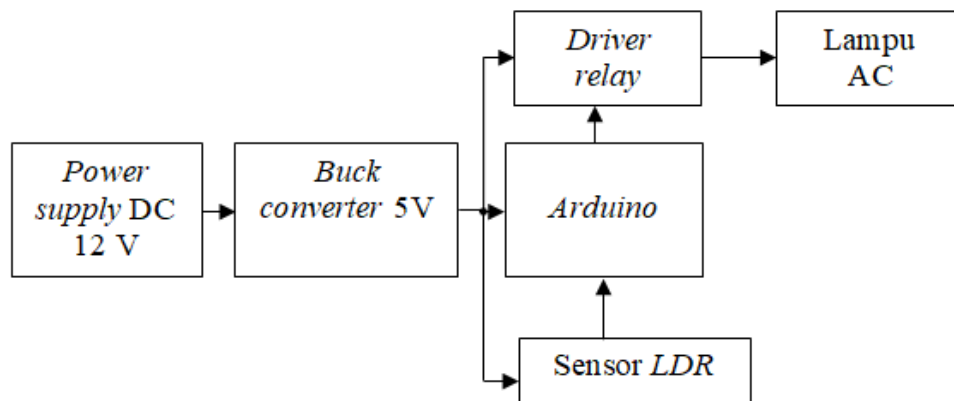
Lampu depan yang berada dirumah pintar dikontrol otomatis menggunakan sensor *LDR*. Sensor *LDR* yang digunakan sebanyak 1 buah yang diletakan di luar ruangan yang dilindungi oleh benda berbahan plastik yang berbentuk setengah lingkaran dan alas berbahan triplek. Adapun Flowchart Program *LDR* dapat dilihat pada Gambar 4.46.



Gambar 4.46 Flowchart program *LDR*

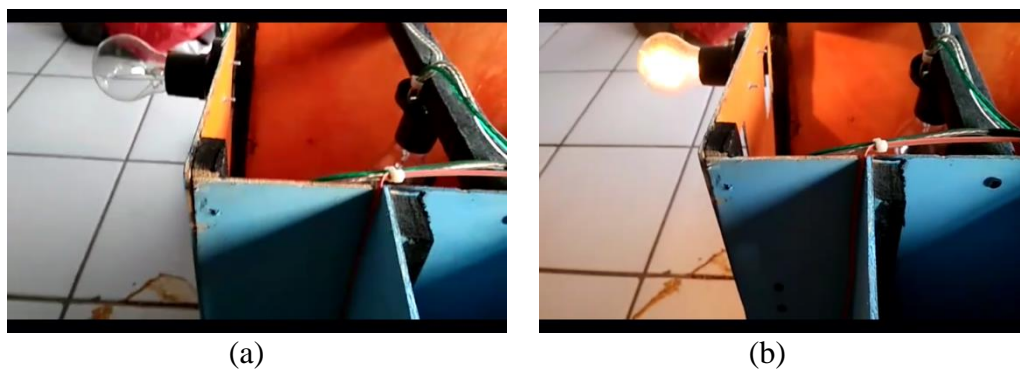
Sensor *LDR* berpengaruh terhadap intensitas cahaya pada lingkungan sehingga sensor ini digunakan pada Lampu depan yang akan menyala ketika nilai *ADC LDR*  $\geq 500$  (saat gelap) dan lampu padam ketika nilai *ADC LDR*  $< 500$  (saat terang). Gambar 4.47 adalah blok diagram pengujian lampu dengan menggunakan *sensor LDR*.





Gambar 4.47 Blok Diagram pengujian sensor *LDR*

Hasil pengujian sensor *LDR* dapat dilihat pada gambar 4.48 tabel 4.12 dibawah ini dengan kondisi waktu yang berbeda beda.



Gambar 4.48 Pengujian sensor *LDR*

(a) Pada saat jam 05.30 kondisi lampu OFF, (b) Pada saat jam 17.30 kondisi lampu ON

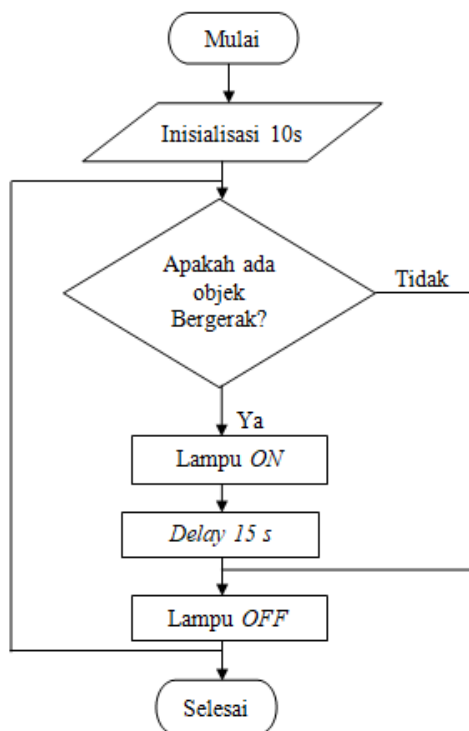
Tabel 4.12 Hasil pengujian sensor *LDR*

NO	Jam (WIB)	Nilai <i>ADC</i>	Kondisi Lampu
1	05.30	300-350	OFF
2	06.00	330-355	OFF
3	17.30	500-600	ON
4	18.00	700-800	ON

Berdasarkan hasil pengujian sensor didapat pada intensitas cahaya yang berbeda-beda akan mempengaruhi nilai *ADC* pada sensor. Berdasarkan pengujian nilai *ADC LDR*  $\geq 500$  (saat gelap) terdapat pada waktu 17.30- 05.00 dan saat terang terdapat pada waktu 05.05- 17.00, sehingga pada lampu depan waktu nilai *ADC LDR* yang efektif untuk mengaktifkan sensor *LDR*. Oleh karena itu, *settingan* sensor lampu depan yang akan menyala ketika nilai *ADC LDR*  $\geq 500$  (saat gelap) dan lampu padam ketika nilai *ADC LDR*  $< 500$  (saat terang).

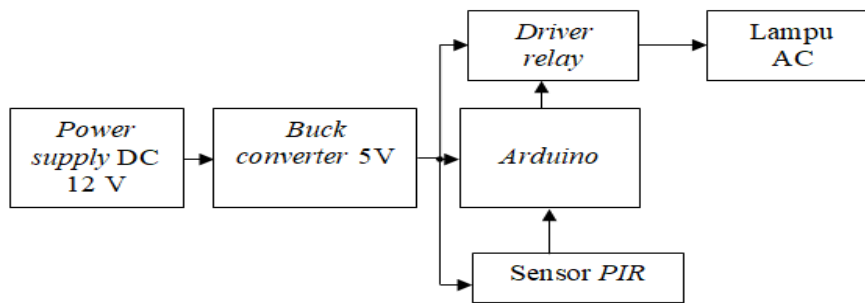
## 2. Pengujian lampu ruangan dengan sensor *PIR*

Lampu yang berada dirumah dalam pintar dikontrol otomatis menggunakan sensor *PIR*. Sensor *PIR* yang digunakan sebanyak 5 buah yang diletakan di dalam ruangan. Gambar 4.49 adalah flowchart dalam *Sensor PIR*.



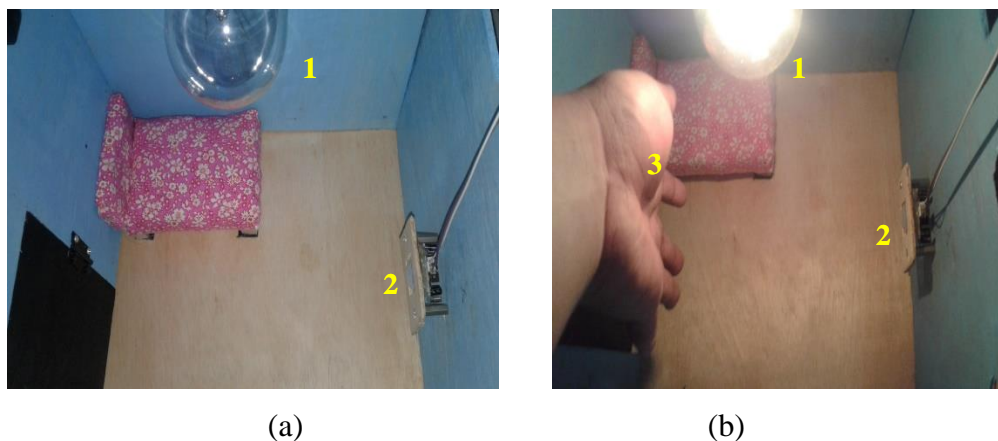
Gambar 4.49 Flowchart sensor *PIR*

Lampu akan menyala ketika terdapat objek bergerak yang terdeteksi oleh sensor *PIR*. Sensor ini digunakan dikarenakan sensor akan aktif hanya pada manusia dan hewan yang bergerak. Berikut ini adalah gambar blok diagram pengujian lampu dengan menggunakan *sensor PIR* pada gambar 4.50.



Gambar 4.50 Blok diagram pengujian sensor *LDR*

Hasil pengujian sensor LDR dapat dilihat pada gambar 4.51 dibawah ini dengan objek manusia dan tabel 4.13.



Gambar 4.51 Pengujian ruangan dengan sensor PIR:  
 (a) Saat Lampu kamar 2 off, (b) Saat lampu kamar 2 on

Keterangan:

1 = lampu

2 = Sensor PIR

3 = Objek (Manusia)

Tabel 4.13 Pengujian lampu

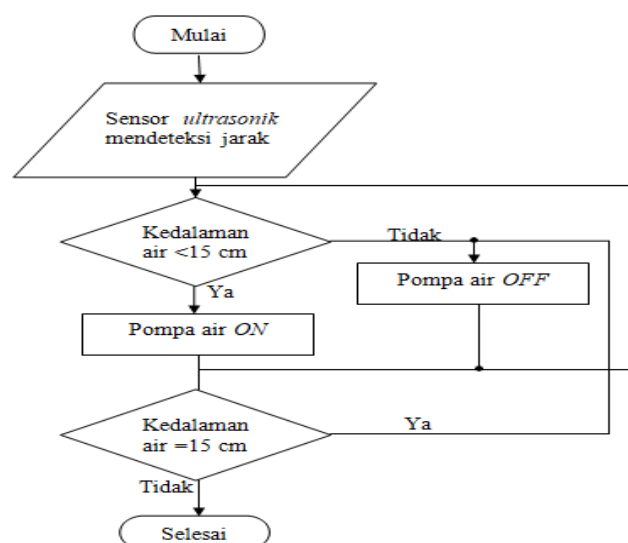
No	Lampu	PIR	Kondisi Lampu	
			ON	OFF
1	Ruang Keluarga	Ada objek	✓	-
2	Ruang Tamu	Ada objek	✓	-

No	Lampu	PIR	Kondisi Lampu	
			ON	OFF
3	Kamar Mandi	Ada objek	-	✓
4	Dapur	Ada objek	-	✓
5	Kamar 1	Ada objek	✓	-
6	Kamar 2	Ada objek	✓	-

Pada bagian ini dilakukan pengujian ruangan dengan sensor PIR. Pada pengujian sensor PIR. Sensor PIR dapat mengurangi penggunaan daya listrik karena sensor hanya akan aktif dan tidak aktif dengan kondisi sesuai dengan settingan sensor. Pada pendeteksian sensor PIR memiliki durasi yang cukup lama. Hal ini dikarenakan kondisi sensor PIR yang digunakan kurang stabil dalam mendeteksi ada atau tidaknya objek pada ruangan. Oleh karena itu kurang disarankan menggunakan sensor PIR dalam mengaktifkan beban secara *real time*.

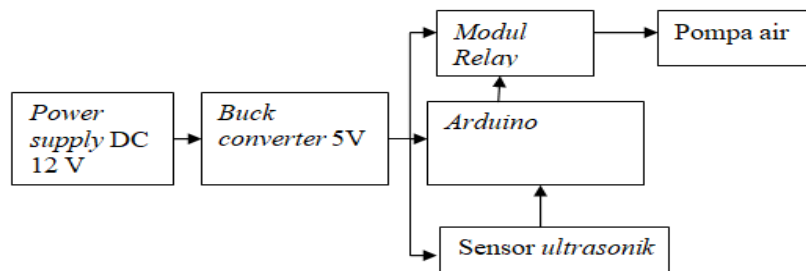
### 3. Pengujian Kontrol Pompa dengan Sensor Ultrasonik

Pompa air dikontrol secara otomatis menggunakan sensor *ultrasonik*. Sensor *ultrasonik* akan aktif pada saat ketinggian air < 15 cm. Berikut ini adalah gambar 4.52 flowchart dalam *sensor ultrasonik*.



Gambar 4.52 Flowchat sensor ultrasonik

Pada smarthome sensor ini digunakan untuk mengaktifkan pompa air yang akan menyala ketika air yang terdeteksi oleh sensor *ultrasonik* dalam tandon air dengan batas maksimum sensor ini pada jarak 15 cm. Gambar 4.53 adalah blok diagram pengujian lampu dengan menggunakan *sensor PIR*.



Gambar 4.53 Blok Diagram Pengujian

Hasil pengujian pada tandon air dapat dilihat pada gambar 4.54 dan tabel 4.14 pengujian sensor ultrasonik.



Gambar 4.54 Pengujian sensor ultrasonik

Tabel 4.14 Hasil pengujian sensor Ultrasonik

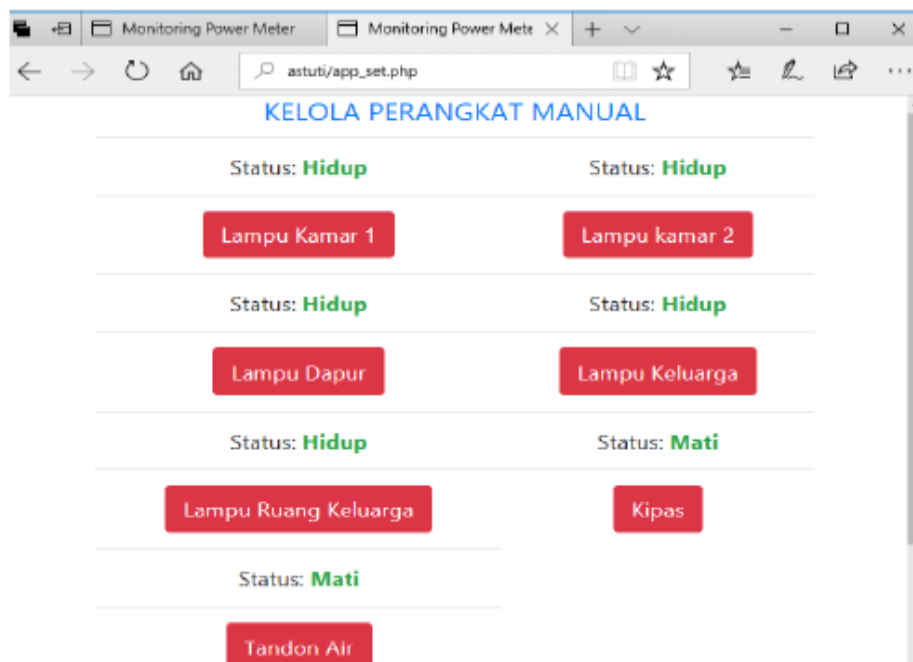
No.	Hasil Sensor Ultrasonik	Kondisi Sensor
1	9cm	ON
2	13cm	ON
3	15cm	OFF
4	20cm	OFF
5	25cm	OFF
6	28cm	OFF

Pada pengujian ini berfungsi untuk mengetahui aktif atau tidaknya pompa air digunakan untuk mengalirkan air yang masuk ke tandon dengan batasan jarak yang terbaca pada sensor ultrasonik. Sensor ini akan aktif apabila ketinggian pada dalam tandon kurang dari 15 cm, dan saat kedalaman air mencapai 15cm pompa

akan tidak aktif. Pada saat pengujian sensor ultrasonik terdapat *error* yang terjadi dikarenakan posisi sensor yang mendedeksi air pada dalam tandon.

#### 4.6.1 Pengujian Dengan Website

Pada bagian ini dilakukan pengujian sistem kontrol dan monitoring melalui *website*, untuk menghidupkan atau mematikan dapat dilakukan dengan mengirim intruksi kepada ESP8266. Pengujian dilakukan dengan cara melakukan pengontrolan pada beban dan pemantauan daya yang terpakai secara langsung. Pengujian ini dapat dilihat pada gambar 4.55 gambar 4.56, dan gambar 4.57.



Gambar 4.55 Pengujian kontrol pada website



(a)

(b)

Gambar 4.56 Pengujian lampu pada website

(a) Saat lampu kamar 1 off, (b) Saat lampu kamar 1 on

Tegangan	225 Volt
Arus	0,4091 Ampere
Daya	92,065 Watt
Orang	Tidak ada Objek terdeteksi
Status Perangkat	
Lampu Kamar 1	Hidup
Lampu Kamar 2	Hidup
Lampu Dapur	Hidup
Lampu Kamar Mandi	Hidup
Lampu Ruang Keluarga	Hidup
Kipas Angin	Mati
Tandon Air	Mati

Gambar 4.57 Hasil monitoring peralatan listrik

Kecepatan pengiriman intruksi tergantung pada kestabilan jaringan internet, apabila jaringan internet stabil maka proses pengiriman instruksi akan cepat, dan sebaliknya apabila jaringan tidak stabil maka proses pengiriman intruksi akan mengalami kendala.

#### 4.6.2 Penggunaan Energi Listrik Harian

##### 1. Perhitungan Energi Listrik Secara Manual

Dalam pengujian ini, diasumsikan penggunaan lampu, kipas angin dan pompa air untuk pemakaian secara manual seperti pada Tabel 4.15 dibawah ini.

Tabel 4.15 Perhitungan Pemakaian Manual

No	Ruangan	Daya (W)	Pemakaian (h)	Daya*Jam(Wh)
1	Kamar 1	15	4	60
2	Kamar 2	15	4	60
3	Dapur	15	12	180
4	Kamar Mandi	15	12	180
5	Ruang Keluarga	15	12	180
6	Lampu Depan	15	12	180
7	Kipas Angin	25	10	250
8	Pompa Air	18	4	72
Total		133	70	1.162

Jadi, total pemakaian daya peralatan listrik selama satu hari adalah:

$$\text{Daya total} = 1.162 \text{ Wh}$$

Daya yang digunakan selama 1 bulan (30 hari) adalah:

$$\text{Daya selama 1 bulan} = \text{Daya total} \times 30 \text{ hari}$$

$$1.162 \text{ Wh} \times 30 = 34.860 \text{ Wh}$$

$$34.860 \text{ Wh} \div 1000 = 34,86 \text{ KWh}$$

Berdasarkan ketetapan dari PLN, untuk daya pemakaian 1300 VA harga listrik per  $\text{KWh} = \text{Rp. } 1467,28, -$ .

Jadi biaya yang harus dilakukan selama satu bulan adalah:

$$\text{Biaya selama 1 bulan} = \text{Daya selama 1 bulan} \times \text{harga listrik per KWh}$$

$$34,86 \text{ Kwh} \times \text{Rp. } 1467,28, - = \text{Rp. } 51.149,38, -$$

## 2. Perhitungan Energi Listrik Secara Otomatis

Penggunaan energi listrik dengan sistem otomatis ada 2 macam, yaitu penggunaan pada beban listrik dan pada alat. Adapun penggunaan daya pada beban listrik dan alat dapat dilihat pada Tabel 4.16 dan Tabel 4.17

Tabel 4.16 Perhitungan Otomatis

No	Ruangan	Daya (W)	Pemakaian (h)	Daya*Jam(Wh)
1	Kamar 1	15	4	60
2	Kamar 2	15	4	60
3	Dapur	15	8	120
4	Kamar Mandi	15	6	90
5	Ruang Keluarga	15	5	75
6	Lampu Depan	15	11	160
7	Kipas Angin	25	8	200
8	Pompa Air	18	3,7	66,6
Total		133	49,7	836,6

Jadi, total pemakaian daya peralatan listrik selama satu hari adalah:

$$\text{Daya total peralatan} = 836,6 \text{ Wh}$$

Daya yang digunakan selama 1 bulan (30 hari) adalah:

$$\text{Daya selama 1 bulan} = \text{Daya total} \times 30 \text{ hari}$$

$$836,6 \text{ Wh} \times 30 = 25.098 \text{ Wh}$$



$$25.098 \text{ Wh} \div 1000 = 25,098 \text{ KWh}$$

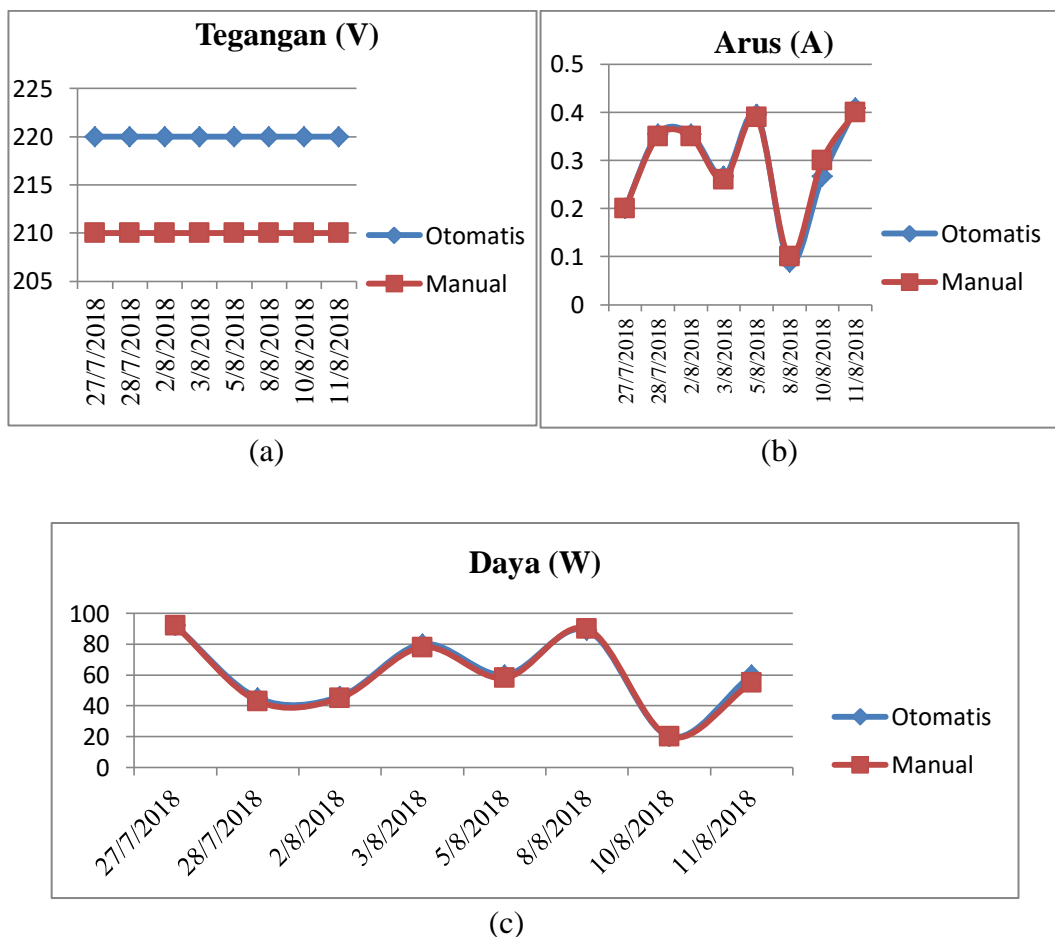
Berdasarkan ketetapan dari PLN, untuk daya pemakaian 1300 VA harga listrik per KWh= Rp.1467,28,-.

Jadi biaya yang harus dilakukan selama satu bulan adalah:

$$\text{Biaya selama 1 bulan} = \text{Daya selama 1 bulan} \times \text{harga listrik per KWh}$$

$$25,098 \text{ KWh} \times \text{Rp. 1467,28,-} = \text{Rp. 36.825,79,-}$$

Hasil database penggunaan listrik pada *webside* ini dapat dilihat pada gambar 4.58 grafik *monitoring* penggunaan energi listrik, gambar 4.59 hasil dari *data base* pada *website*, gambar 4.60 pembacaan daya dengan *wattmeter*, dan tabel 4.17.



Gambar 4.58 Grafik *Monitoring* Penggunaan Energi Listrik

(a) Grafik *Monitoring* Tegangan Listrik, (b) Grafik *Monitoring* Arus Listrik,

(c) Grafik *Monitoring* penggunaan daya listrik

istuti/app\_report.php Search

### LAPORAN POWER METER

Waktu	Tegangan (V)	Arus (A)	Daya	Orang	Tandon Air
27 July 2018 04:40:50	220 volt	0.4091 Amper	92,065 Watt	Tidak terdeteksi	Kosong
28 July 2018 12:04:50	220 volt	0.2 Amper	45 Watt	Tidak terdeteksi	Kosong
02 Agust 2018 09:04:50	220 volt	0.355 Amper	80 Watt	Tidak terdeteksi	Kosong
03 Agust 2018 15:04:50	220 volt	0.267 Amper	60 Watt	Tidak terdeteksi	Kosong
05 Agust 2018 12:04:50	220 volt	0.3955 Amper	89 Watt	Tidak terdeteksi	Kosong
08 Agust 2018 18:04:50	220 volt	0,088 Amper	20 Watt	Tidak terdeteksi	Kosong
10 Agust 2018 15:04:50	220 volt	0.267 Amper	60 Watt	Tidak terdeteksi	Kosong
11 Agust 2018 04:40:50	220 volt	0.4091 Amper	92,065 Watt	Tidak terdeteksi	Kosong
12 Agust 2018 12:04:50	220 volt	0.2 Amper	45 Watt	Tidak terdeteksi	Kosong

Kembali Activate Windows

Gambar 4.59 Hasil dari *Data Base* pada *Website*

Tabel 4.17 Hasil *Monitoring* Penggunaan Energi Listrik

Waktu	Tegangan (V)	Arus(A)	Daya (W)	Orang	Tandon Air
27 July 2018 04.40.50	220 Volt	0,4091 Amper	92,065 Watt	Tidak Terdeteksi	Kosong
28 July 2018 12.04.50	220 Volt	0,2 Amper	45 Watt	Tidak Terdeteksi	Kosong
02 Agust 2018 09:04:50	220 Volt	0,355 Amper	80 Watt	Tidak Terdeteksi	Kosong

Waktu	Tegangan (V)	Arus(A)	Daya (W)	Orang	Tandon Air
03 Agust 2018 15:04:50	220 Volt	0,267 Amper	60 Watt	Tidak Terdeteksi	Kosong
05 Agust 2018 12:04:50	220 Volt	0,3955 Amper	89 Watt	Tidak Terdeteksi	Kosong
08 Agust 2018 18:04:50	220 Volt	0,088 Amper	20 Watt	Tidak Terdeteksi	Kosong
10 Agust 2018 15:04:50	220 Volt	0,267 Amper	60 Watt	Tidak Terdeteksi	Kosong
11 Agust 2018 04:40:50	220 Volt	0,4091 Amper	92,065 Watt	Tidak Terdeteksi	Kosong
02 Agust 2018 12:04:50	220 Volt	0,2 Amper	45 Watt	Tidak Terdeteksi	Kosong



(a)



(b)

Gambar 4.60 Pembacaan Daya Dengan *Wattmeter*:

(a) 3 lampu dan pompa ON, (b) 5 lampu dan pompa ON

#### 4.7 Analisa

Berdasarkan hasil pengujian pemakaian energi listrik pada sistem kontrol manual, untuk penggunaan normal maupun penggunaan karena human error dibandingkan dengan sistem kontrol otomatis dapat dilihat pada Tabel 4.18.

Tabel 4.18 Perbandingan Perhitungan Secara Amanual dan Otomatis

No	Pengujian	Daya/hari	Daya/bulan	Biaya/bulan
1	Manual	1.160Wh	34.860Wh	Rp51.149,38,-.
2	Otomatis	836,6Wh	25.098Wh	Rp36.825,79,-.

Perhitungan *persentase error* penghematan pemakaian energi listrik ketika menggunakan sistem kontrol otomatis dibandingkan sistem kontrol manual dapat dihitung menggunakan *formula* dibawah ini:

$$\left| \frac{\text{Biaya Manual} - \text{Biaya otomatis}}{\text{Biaya Manual}} \right| \times 100\% \dots\dots\dots(4.3)$$

Adapun *persentase* pemakaian daya secara manual dan otomatis

- $\left| \frac{34.860 \text{ Wh} - 25.098 \text{ Wh}}{34.860 \text{ Wh}} \right| \times 100\% = 8,40\%$

Berdasarkan hasil perhitungan *presentase error* di peroleh penghematan energi yang cukup signifikan antara pemakaian secara manual dan otomatis yaitu hampir 8,40%. Untuk itu, diperlukan sistem kontrol otomatis untuk meminimalisir yang sering terjadi karena kelalaian pengguna energi listrik.

## **BAB V**

### **PENUTUP**

#### **5.1 Kesimpulan**

Berdasarkan hasil analisa serta pengujian yang dilakukan dapat diambil kesimpulan bahwa:

1. Berdasarkan hasil pengujian sistem, pada sensor PIR memiliki durasi yang cukup lama yaitu sekitar 5-8 detik. Hal ini dikarenakan kondisi sensor PIR yang digunakan kurang maksimal dan tidak stabil dalam mendeteksi ada atau tidaknya objek yang bergerak pada ruangan.
2. Pada hasil pengujian ke website bisa dilakukan dengan baik dalam pengontrolan beban dan mengirim data berupa nilai tegangan, arus, daya, nilai ADC dan kedalaman air dengan lamanya durasi waktu yang pengiriman Modul *WiFi ESP8266* sekitar 8 detik.
3. Alat Prototype Smarthome dengan Konsep Internet of Thing (IoT) berbasis Website ini dapat menghemat penggunaan daya sebesar 8,40% daripada pemakaian secara manual dengan *error* sebesar 2,5%.

#### **5.2 Saran**

1. Dalam menggunakan modul *WiFi ESP8266* memiliki durasi waktu yang lama dalam penguploadan data yang yaitu sebesar 8 detik dan dipengaruhi oleh jaringan internet yang lancar. Sehingga disarankan menggunakan *provider* yang konektivitas internetnya tinggi sehingga tidak menghambat sistem kerjanya jaringan internet yang lancar.
2. Pada pendeteksi objek yang bergerak menggunakan sensor PIR kurang efektif karena waktu respon sensor terhadap gerak cukup lama. Diharapkan ada pengembangan sensor PIR, sehingga waktu respon sensor lebih cepat.
3. Alat ini memiliki tampilan website yang masih sederhana seperti latar background pada website yang berwarna putih dan menu-menu yang

ditampilkan hanya terdapat tiga bagian yaitu bagian menu utama, bagian tampilan monitoring dan bagian tampilan kontrol. Maka dari itu disarankan dalam pembuatan tampilan website lebih baik dalam segi background serta menu-menu yang di tampilkan lebih beragam.

## DAFTAR PUSTAKA

- [1] Menghitung Biaya Energi Listrik Rumah/Kantor [Online], diakses pada 15 Mei 2018, Available:<http://ardra.biz/sain-teknologi-terapan/menghitung-biaya-energi-listrik-rumahkantor/>
- [2] Mansykur Fauzan, Prasetyowati, “Aplikasi Rumah Pintar (Smarthome) Pengendali Peralatan Elektronik Rumah Tangga Berbasis Web”, jurnal Teknologi Informaso dan Ilmu teknologi Vol 3, No. 1 Maret 2016, hlm 51-58.
- [3] Rozaq Abdul, Yulita Noor, “Efesinsi Energi Smarthome (Ruamah Pintar) Berbasis Remote Relay dan LDR(Light Dependent Resistan” , jurnal Simestri, vol 8No 1 April 2017,
- [4] Burange, A. W., & Cheung, H.(2012). Review of Internet of Things in Development of Smart Cities with Data Management & Privacy.
- [5] Kevin Ashton, “That ‘Internet of Things’Thing”, Jurnal RFID, 2009
- [6] Pengertian dan Rumus Daya Listrik beserta satuanya [Online], diakses 15 Mei 2018, available:[www.berpendidikan.com/2015/10/pengertian-dan-Rumus-Daya-Listrik-beserta-satuanya.html?m=1](http://www.berpendidikan.com/2015/10/pengertian-dan-Rumus-Daya-Listrik-beserta-satuanya.html?m=1)
- [7] Eddi Kurniawan, Cucu Suhery, Dedi Triyanto, “ Sistem Penerangan Rumah Otomatis Dengan Sensor Cahaya Berbasis Mikrokontroler” *Jurnal Coding Sistem Komputer Universitas Tanjungpura*, vol. 01, no. 2, pp. 1-10, 2013.

- [8] Limantara, Hans S.2009 Jelajah Dunia Maya Dengan Cepat dan Mudah. Jakarta:PT.Elex Media Komputino
- [9] I.G.B.R.Putra, "Implementasi MySQL Cluster Pada Basis Data Terdistribusi," Jurnal Elektronik Ilmu Komputer Universitas Undayan, Vol. Volume 1, pp.11-20,2012,
- [10] Ritoga Pahmi, 2015. "Pengertian Bahasa Pemrogram PHP menurut para ahli"[Online] diakses pada 18 juni 2018, Available :<http://bangpahmi.com/Pengertian-Bahasa-Pemrogram-PHP-menurut-para-ahli/>
- [11] Djuandi F, "Pengnalan arduino"[Online] diakses pada 09 Mei 2018, available:[tobuku.com.docs>arduino-pengnalan-arduino.pdf](http://tobuku.com/docs/arduino-pengnalan-arduino.pdf)
- [12] Habibi, FN. 2017 "Alat Montoring Pemakian Energi Listrik Berbasis Android Menggunakan Modul PZEM-004T"[Online] diakses pada 18 Agustus 2018, available:[sngbr.polinema.ac.id>download.pdf](http://sngbr.polinema.ac.id/download.pdf)
- [13] Buck Converter[Online] diakses pada 19 Mei 2018, available : [http://wikipedia.org/wiki/buck\\_converter](http://wikipedia.org/wiki/buck_converter)
- [14] Apa itu modul ESP8266.2017, available:<http://www.nyebarilmu.com/apa-itu-modul-esp8266/>
- [15] Epiyana & Tampubolon, Hetty Agustina (2017). MANAJEMEN PEMAKAIAN ENERGI LISTRIK UNTUK RUMAH PINTAR.



## DAFTAR RIWAYAT HIDUP

### 1. Data Pribadi

Nama Lengkap : Muhammad Taufik  
Tempat & Tanggal Lahir : Pangkalpinang, 16 Agustus 1997  
Alamat : Jalan Bukit Betung II, Sungailiat  
No. HP : 0831988668810  
Email : muhammadtaufikpolman@gmail.com  
Jenis kelamin : Laki-laki



### 2. Riwayat Pendidikan

TK Pembina Sungailiat	Lulus 2003
SDN Negeri 10 Sungailiat	Lulus 2009
SMP Negeri 2 Sungailiat	Lulus 2012
SMA Negeri 1 Sungailiat	Lulus 2015
D III POLMAN NEGERI BABEL	2015-Sekarang

### 3. Pengalaman Kerja

Praktik Kerja Lapangan di Politeknik Negeri Samarinda	September 2017 s/d Januari 2018
-------------------------------------------------------	---------------------------------

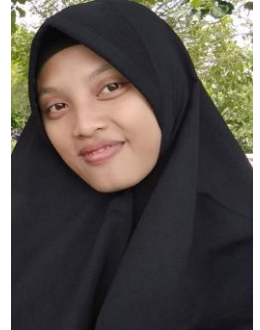
Sungailiat, 06 Agustus 2018

Muhammad Taufik

## DAFTAR RIWAYAT HIDUP

### 1. Data Pribadi

Nama Lengkap : Ria Astuti  
Tempat & Tanggal Lahir : Riau, 21 Agustus 1997  
Alamat : Jalan Raya Pangkalpinang  
Muntok , Bakam  
No. HP : 085379083090  
Email : riaastuti1921@gmail.com  
Jenis kelamin : Perempuan



### 2. Riwayat Pendidikan

SD Negeri 3 Mangka Lulus 2009  
SMP Negeri 1 Bakam Lulus 2012  
SMA Negeri 1 Sungailiat Lulus 2015  
D III POLMAN NEGERI BABEL 2015-Sekarang

### 3. Pengalaman Kerja

Praktik Kerja Lapangan di PT.Arkha Jayanti Persada	September 2017 s/d Januari 2018
-------------------------------------------------------	------------------------------------

Sungailiat, 06 Agustus 2018

Ria Astuti

## Kontrol Perangkat Manual

```
<?php
```

```
require_once("includes/setup.php")
$action = $_GET['case'];
$qry="SELECT * FROM alat";
$query=mysqli_query($conn, $qry);
$row=mysqli_fetch_array($query);
if($action=="Lampu Kamar 1"){
if($row2['lampu Kamar 1']=="1"){
$qry = "UPDATE alat SET Lampu Kamar 1=0";
    }else{
        $qry = "UPDATE alat SET Kamar 1=1";
    }
}
if($action=="Lampu Kamar 2"){
if($row2['Lampu Kamar 2']=="1"){
$qry = "UPDATE alat SET Lampu Kamar 2=0";
    }else{
        $qry = "UPDATE alat SET Lampu Kamar 2=1";
    }
}
if($action=="Lampu Dapur"){
if($row2['lampu Dapur']=="1"){
$qry = "UPDATE alat SET Lampu Dapur=0";
    }else{
        $qry = "UPDATE alat SET Lampu Dapur=1";
    }
}
if($action=="Lampu Keluarga"){
if($row2['Lampu Keluarga']=="1"){
```

```
$qry = "UPDATE alat SET Lampu Keluarga=0";
    }else{ $qry = "UPDATE alat SET Lampu Keluarga=1";
    }}

if($action=="Tandon"){
if($row2['Tandon']=="1"){
$qry = "UPDATE alat SET Tandon=0";
    }else{
    $qry = "UPDATE alat SET Tandon=1";
    }}

if($action=="Kipas"){
if($row2['Kipas']=="1"){
    $qry = "UPDATE alat SET Kipas=0";
    }else{
    $qry = "UPDATE alat SET Kipas=1";
    }}

$query = mysqli_query($conn, $qry);
header("location:app_set.php");
```

?>

## Pengaplikasian

```
<?php
    require_once("includes/setup.php");?>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">

<head>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />

<meta name="viewport" content="width=device-width, initial-scale=0.9">

<link rel="stylesheet" href="css/bootstrap.min.css">

<link rel="stylesheet" href="css/btn-circle.css">

<!-- <script src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.4/jquery.min.js"></script> -->

<script src="js/jquery.min.js"></script>

<script src="js/bootstrap.min.js"></script>

<title>IOT</title>

<div class="text-center"><h2 class="text-primary"><b>Floor</b></h2></div>

</head>

<body class="bg-success">

<br>

<br>

<div class="text-right">

    <button type="button" class="btn btn-circle btn-ml <?php if($row['status']==1) echo
"btn-success glyphicon glyphicon-ok-circle"; else echo "btn-danger glyphicon glyphicon-
remove-circle"; ?>"></button>

    <br>

    <h5><b>Direction</b></h5>

</div>

<div class="container">
```

```

        <table class="table">

<tbody>

<tr>

<td class="text-center">

<h3><b>Status</b></h3>

<b><h5><?php if($row['status']==="1") echo "<b class='text-success'>Normal</b>"; else echo
"<b class='text-success'>Out Of Service</b>";?></h5></b></td></tr>

<tr><br>

<tr>

        <td class="text-center">

        <h3><b>Reservation Mode</b></h3>

<b><h5><?php if($row['rev_mode']==="1") echo "<b class='text-success'>Active</b>"; else
echo "<b class='text-danger'>Inactive</b>";?></h5></b></td></tr>

<br><tr>

        <td class="text-center">

        <h3><b>Fire Service</b></h3>

<b><h5><?php if($row['fire_serv']==="1") echo "<b class='text-success'>Active</b>"; else echo
"<b class='text-danger'>Inactive</b>";?></h5></b></td></tr>

</tbody>

</table>

</div>

</body>

</html>

```

## Tampilan Menu

```
<?php
    require_once("includes/setup.php");
    date_default_timezone_set("Asia/Bangkok");
    $tanggal = date("d-M-Y");
    $jam = date("H:i");
?>
<html lang="en">
<head>
<!-- Required meta tags -->
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1, maximum-scale=1, user-
scalable=no">
<?php
    echo "<meta http-equiv='refresh' content='5; url=$_SERVER[PHP_SELF]'>";
?>
<!-- Bootstrap CSS -->
<link rel="stylesheet" href="css/bootstrap.min.css">
<script src="js/bootstrap.min.js"></script>
<title>Monitoring Power Meter</title>
</head>
<body>
<div class="text-center">
<h5 class="text-primary">MONITORING POWER METER</h5>
</div>
<div class="container">
```

```

<table class="table">
<tr class="bg-success">
<td colspan="2">
<b>Waktu &emsp;&emsp;&emsp;<?php echo $jam; echo "&emsp;&emsp;"; echo $tanggal;
?></b>
</td></tr>
<tr><td>
    Tegangan
</td>
<td>
<b class="text-success"><?php echo $row1['sensorvolt']; ?> Volt</b></td></tr>
<tr><td>
    Arus
</td>
<td>
<b class="text-success"><?php echo $row1['sensorarus']; ?> Ampere</b></td></tr>
<tr><td>
    Daya
</td>
<td>
<b class="text-success"><?php echo $row1['sensordaya']; ?> Watt</b>
</td></tr>
<tr><td>
    Orang
</td>
<td>
<?php if($row1['ldr']>"900") echo "<b class='text-success'>Ada</b>"; else echo "<b class='text-
success'>Tidak ada Objek Terdeteksi</b>";?></td></tr>

```



```

<tr class="bg-warning">
<td class="text-center" colspan="2"><b>Status Perangkat</b></td></tr>
<tr><td>
    Lampu Kamar 1
</td>
<td>
<?php if($row2['Lampu Kamar 1']=="1") echo "<b class='text-success'>Hidup</b>"; else echo
"<b class='text-danger'>Mati</b>";?></td></tr>
<tr><td>
    Lampu Kamar 2
</td>
<td>
<?php if($row2['Lampu Kamar 2']=="1") echo "<b class='text-success'>Hidup</b>"; else echo
"<b class='text-danger'>Mati</b>";?></td></tr>
<tr><td>
    Lampu Dapur
</td>
<td>
<?php if($row2['Lampu Dapur']=="1") echo "<b class='text-success'>Hidup</b>"; else echo "<b
class='text-danger'>Mati</b>";?>
</td></tr>
<tr><td>
    Lampu Ruang Keluarga
</td>
<td>
<?php if($row2['Lampu Ruang Keluarga']=="1") echo "<b class='text-success'>Hidup</b>";
else echo "<b class='text-danger'>Mati</b>";?>
</td></tr>
<tr><td>

```

Kipas

</td>

<td>

```
<?php if($row2['Kipas']==="1") echo "<b class='text-success'>Hidup</b>"; else echo "<b class='text-danger'>Mati</b>";?></td></tr><tr><td>
```

Tandon

</td>

<td>

```
<?php if($row2[' Tandon ']==="1") echo "<b class='text-success'>Hidup</b>"; else echo "<b class='text-danger'>Mati</b>";?></td></tr>
```

</table>

</div>

<div class="container text-center">

```
<a href="app_report.php"><button type="button" class="btn btn-primary btn-lg">Laporan</button></a>
```

&emsp;

```
<a href="app_set.php"><button type="button" class="btn btn-primary btn-lg">Perangkat</button></a>
```

</div>

</body>

</html>

## Menu Laporan Monitoring

```
<?php
    require_once("includes/setup.php");

    $qry = "SELECT *, DATE_FORMAT(waktu,'%d-%m-%Y %H:%i:%S') AS jam,
DATE(waktu) AS tanggal FROM sensor ORDER BY waktu DESC LIMIT 10";

    $query = mysqli_query($conn, $qry);?>

<html lang="en">

<head>

<!-- Required meta tags -->

<meta charset="utf-8">

<meta name="viewport" content="width=device-width, initial-scale=1, maximum-scale=1, user-
scalable=no">

<!-- Bootstrap CSS -->

<link rel="stylesheet" href="css/bootstrap.min.css">

<script src="js/bootstrap.min.js"></script>

<title>MONITORING POWER METER</title>

</head>

<body>

<div class="text-center">

<h5 class="text-primary">LAPORAN POWER METER</h5>

</div>

<div class="container">

<table class="table table-striped table-bordered">

<thead>

<tr class="bg-success">

<th class="text-center">Waktu</th>

<th class="text-center">Tegangan (V)</th>

<th class="text-center">Arus (A)</th>
```

```

<th class="text-center">Orang</th>
        <th class="text-center">Tandon</th>
</tr></thead><tbody>
<?php
    while ($row=mysqli_fetch_array($query)) {
        echo "<tr>";
        echo "<td class='text-center'>$row[jam]</td>";
        echo "<td class='text-center'>$row[sensorvolt]</td>";
        echo "<td class='text-center'>$row[sensorarus]</td>";
        if($row['ldr']>"900") echo "<td class='text-center'>Ada</td>";else echo "<td class='text-
center'>Tidak Ada</td>";
            echo "<td class='text-center'>$row[sensorultrasonik]</td>";

        echo "</tr>";
    }
?>
</tbody></table>
</div>
<div class="container">
<a href="app_menu.php"><button type="button" class="btn btn-primary btn-
block">Kembali</button></a>
</div></body></html>

```

## Menu Kontrol Perangkat

```
<?php
    require_once("includes/setup.php");?>
<html lang="en">
<head>
<!-- Required meta tags -->
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1, maximum-scale=1, user-
scalable=no">
<!-- Bootstrap CSS -->
<link rel="stylesheet" href="css/bootstrap.min.css">
<script src="js/bootstrap.min.js"></script>
<title>Monitoring Power Meter</title>
</head>
<body>
<div class="text-center">
<h5 class="text-primary">KELOLA PERANGKAT MANUAL</h5>
</div>
<div class="container">
<table class="table text-center">
<tr><td>
        Status: <?php if($row2['Lampu Kamar 1']==="1") echo "<b class='text-
success'>Hidup</b>"; else echo "<b class='text-danger'>Mati</b>";?>
</td>
<td>
        Status: <?php if($row2['Lampu Kamar 1']==="1") echo "<b class='text-
success'>Hidup</b>"; else echo "<b class='text-danger'>Mati</b>";?>
</td></tr>
</td></tr>
```

```

<tr><td>

<a href="action.php?case=Lampu Kamar 1"><button type="button" class="btn <?php
if($row2['Lampu Kamar 1']==="1") echo "btn-success"; else echo "btn-danger";?> btn-
md">Lampu Kamar 1<br></button></a>

</td><td>

<a href="action.php?case=Lampu Kamar 2"><button type="button" class="btn <?php
if($row2['Lampu Kamar 2']==="1") echo "btn-success"; else echo "btn-danger";?> btn-
md">Lampu Kamar 2<br></button></a>

</td></tr>

<tr><td>

        Status: <?php if($row2['Lampu Dapur']==="1") echo "<b class='text-
success'>Hidup</b>"; else echo "<b class='text-danger'>Mati</b>";?>

</td>

<td>

        Status: <?php if($row2['Lampu Ruang Keluarga']==="1") echo "<b class='text-
success'>Hidup</b>"; else echo "<b class='text-danger'>Mati</b>";?>

</td></tr>

<tr><td>

<a href="action.php?case=Lampu Dapur"><button type="button" class="btn <?php
if($row2['Lampu Dapur']==="1") echo "btn-success"; else echo "btn-danger";?> btn-md">Lampu
Dapur<br></button></a></td>

<td>

<a href="action.php?case=Lampu Ruang Keluarga"><button type="button" class="btn <?php
if($row2['Lampu Ruang Keluarga']==="1") echo "btn-success"; else echo "btn-danger";?> btn-
md">Lampu Ruang Keluarga<br></button></a>

</td></tr>

<tr><td>

        Status: <?php if($row2['Kipas']==="1") echo "<b class='text-success'>Hidup</b>"; else
echo "<b class='text-danger'>Mati</b>";?>

</td><td>

        Status: <?php if($row2['Tandon']==="1") echo "<b class='text-success'>Hidup</b>"; else
echo "<b class='text-danger'>Mati</b>";?>

```

```
</td></tr>
```

```
<tr><td>
```

```
<a href="action.php?case=Kipas"><button type="button" class="btn <?php  
if($row2['Kipas']=="1") echo "btn-success"; else echo "btn-danger";?> btn-  
md">Kipas<br></button></a>
```

```
</td>
```

```
<td>
```

```
<a href="action.php?case=Tandon"><button type="button" class="btn <?php  
if($row2['Tandon']=="1") echo "btn-success"; else echo "btn-danger";?> btn-  
md">Tandon<br></button></a>
```

```
</td></tr>
```

```
</table>
```

```
<table class="table">
```

```
<tr><td>
```

Tegangan

```
</td><td>
```

```
<b class="text-success"><?php echo $row1['sensorvolt'];?> Volt</b>
```

```
</td></tr>
```

```
<tr><td>
```

Arus

```
</td><td>
```

```
<b class="text-success"><?php echo $row1['sensorarus']; ?> Ampere</b>
```

```
</td></tr>
```

```
<tr><td>
```

Orang

```
</td>
```

```
<td>
```

```
<?php if($row1['ldr']>"900") echo "<b class='text-success'>Ada</b>"; else echo "<b class='text-  
success'>Tidak ada</b>";?>
```

```
</td></tr>
```

```
<tr><td>
```

```
    Daya
```

```
</td>
```

```
<td>
```

```
<b class="text-success"><?php echo $row1['sensordaya']; ?> Watt</b>
```

```
</td></tr>
```

```
</table></div>
```

```
<div class="container">
```

```
<a href="app_menu.php"><button type="button" class="btn btn-primary btn-  
block">Kembali</button></a>
```

```
</div></body></html>
```



## Index

```
<?php
//header("location:aplikasi.php?action=home");
header("location:app_menu.php");?>
```

### Koneksi Ke Database

```
<?php
require_once("koneksi.php");
$qry1 = "SELECT * FROM sensor ORDER BY waktu DESC";
$query1 = mysqli_query($conn, $qry1);
$row1 = mysqli_fetch_array($query1);
$qry2 = "SELECT * FROM alat";
$query2 = mysqli_query($conn, $qry2);
$row2 = mysqli_fetch_array($query2);?>
```

### Koneksi Ke Database

```
<?php
$servername = "localhost";
$username = "labandroid_powermeter";
$password = "powermeterpwd";
$dbname = "labandroid_powermeter";

// Create connection
$conn = mysqli_connect($servername, $username, $password, $dbname);
if(!$conn){
    die("KONEKSI GAGAL : ". mysqli_connect_error());
}?>
```

## Data base

```
-- phpMyAdmin SQL Dump
-- version 4.7.7
-- https://www.phpmyadmin.net/
--
-- Host: localhost:3306
-- Generation Time: Jul 18, 2018 at 11:34 AM
-- Server version: 10.1.32-MariaDB
-- PHP Version: 5.6.30

SET SQL_MODE = "NO_AUTO_VALUE_ON_ZERO";
SET AUTOCOMMIT = 0;
START TRANSACTION;
SET time_zone = "+00:00";

/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
/*!40101 SET NAMES utf8mb4 */;

-- Database: `labandroid_powermeter`
-----

-- Table structure for table `alat`

CREATE TABLE `alat` (
  `id` int(4) NOT NULL,
  `Lampu Kamar 1` int(2) NOT NULL,
  `Lampu Kamar 2` int(2) NOT NULL
  `Lampu Dapur` int(2) NOT NULL,
  `Lampu Ruang Keluarga` int(2) NOT NULL
  `Tandon 1` int(2) NOT NULL,
```

```

`Kipas` int(2) NOT NULL
) ENGINE=MyISAM DEFAULT CHARSET=latin1;
--
-- Dumping data for table `alat`
--
INSERT INTO `alat` (`id`, `Lampu Kamar 1`, `Lampu Kamar 2`, `Lampu Dapur`, `Lampu
Ruang Keluarga`, `tandon`, `kipas`) VALUES
(0, 1, 0);
-----
--
-- Table structure for table `sensor`
--
CREATE TABLE `sensor` (
  `id` double NOT NULL,
  `sensorvolt` int(10) NOT NULL,
  `sensorarus` int(10) NOT NULL,
  `sensordaya` int(10) NOT NULL,
  `ldr` int(10) NOT NULL,
  `ultrasonik` int(10) NOT NULL,
  `pir` int(10) NOT NULL,
  `waktu` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE
CURRENT_TIMESTAMP
) ENGINE=MyISAM DEFAULT CHARSET=latin1;
--
-- Dumping data for table `sensor`
--
INSERT INTO `sensor` (`id`, `sensorvolt`, `sensorarus`, `ldr`, `ultrasonik`, `pir`, `waktu`)
VALUES

```

```
-- Indexes for dumped tables
-- Indexes for table `alat`
--
ALTER TABLE `alat`
  ADD PRIMARY KEY (`id`);

--
-- Indexes for table `sensor`
--
ALTER TABLE `sensor`
  ADD PRIMARY KEY (`id`);

--
-- AUTO_INCREMENT for dumped tables
--
--
-- AUTO_INCREMENT for table `sensor`
--
ALTER TABLE `sensor`
  MODIFY `id` double NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=2;
COMMIT;
/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;
/*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;
/*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;
```

## Program Arduino

```
#include <PZEM004T.h>

PZEM004T pzem(&Serial1);
IPAddress ip(192, 168, 1, 1);

#define TRIG_PIN 24
#define ECHO_PIN 22

const uint8_t BATAS_BAWAH = 9;
const uint16_t BATAS_ATAS = 25;

const int p1 = 52;//manual Ruang keluarga
const int p2 = 50;//manual kamar 1
const int p3 = 48;//manual kamar 2
const int p4 = 46;//manual wc
const int p5 = 44;//manual dapur

const int r1 = 12;//Ruang keluarga
const int r2 = 11;//kamar 1
const int r3 = 10;//kamar 2
const int r4 = 9;//wc
const int r5 = 8;//dapur
const int r6 = 7;//teras
```

```
const int led7 = 6;//tandon ar
const int led8 = 5;//beban kipas angin
const int bz = 4;//buzzer
```

```
const int PIR1 = 36;
const int PIR2 = 34;
const int PIR3 = 32;
const int PIR4 = 30;
const int PIR5 = 28;
```

```
int LDR = A11;
int nilai_LDR;
int b = 0;
int c = 0;
int d = 0;
int e = 0;
int f = 0;
int g = 0;
int h = 0;
int i = 0;
```

```
int jarak;
int counter=0;
int mode=0;
```

```

float v;

float im;

float COS;

float p;

byte S1, S2, S3, S4, S5, S6, S7;

void LDR_A ()
{
    nilai_LDR = analogRead(LDR);

    if (nilai_LDR >= 520 ) {digitalWrite(r6, LOW );} // Lampu Depan
    else {digitalWrite(r6, HIGH );}

    Serial.print("Kondisi lampu : ");
    Serial.println(nilai_LDR);
}

void DAYA() {
    v = pzem.voltage(ip);
    if(v<=0)v=0;
    if (v < 0.0) v = 0.0;
    Serial.print("Tegangan : "); Serial.print(v); Serial.println("V ");
    serial();
    im = pzem.current(ip);
}

```

```
if(im<=0)im=0;
if (im >= 0.0) {
    Serial.print("Arus : ");
    Serial.print(im);
    Serial.println("A ");
}
serial();
p = pzem.power(ip);
if(p<=0)p=0;
if (p >= 0.0) {
    Serial.print("Daya : ");
    Serial.print(p);
    Serial.println("W ");
}
serial();
COS = p / (v * im);
if(COS>=1)COS=1;
else if(COS<=0)COS=0;
if (COS >= 0.0) {
    Serial.print("Cos Phi : ");
    Serial.println(COS);
    Serial.println();
}
serial();
if (p > 100) {digitalWrite(bz, LOW );Serial.println("buzzer on");}
```



```
else {digitalWrite(bz, HIGH);Serial.println("buzzer off");}
}
```

```
void pb(){
  b = digitalRead(p1);
  c = digitalRead(p2);
  d = digitalRead(p3);
  e = digitalRead(p4);
  f = digitalRead(p5);

  if (b == LOW || S1 == 1) {
    digitalWrite(r1, LOW);

  } else if(b == HIGH || S1 == 0){
    digitalWrite(r1, HIGH);

  }

  if (c == LOW || S2 == 1) {
    digitalWrite(r2, LOW);
  } else if(c == HIGH || S2 == 0){
    digitalWrite(r2, HIGH);
  }

  if (d == LOW || S3 == 1) {
```

```

    digitalWrite(r3, LOW);
} else if (d == HIGH || S3 == 0){
    digitalWrite(r3, HIGH);
}

if (e == LOW || S4 == 1) {
    digitalWrite(r4, LOW);
} else if(e == HIGH || S4 == 0){
    digitalWrite(r4, HIGH);
}

if (f == LOW || S5 == 1) {
    digitalWrite(r5, LOW);
} else if(e == HIGH || S5 == 0) {
    digitalWrite(r5, HIGH);
}
}

void TANDON()
{
    jarak = sonar();
    jarak = 24 - jarak;
    if(jarak<=0)jarak=0;

    Serial.print("Jarak dalam satuan cm: ");

```

```
    Serial.println(jarak);

    Serial.println();
}

int sonar(void)
{
    digitalWrite(TRIG_PIN, HIGH);
    delayMicroseconds(10);
    digitalWrite(TRIG_PIN, LOW);

    int jarak = pulseIn(ECHO_PIN, HIGH);
    return (jarak = jarak / 58);
}

void setup() {
    Serial.begin(9600);
    Serial2.begin(4800);
    pzem.setAddress(ip);
    pinMode(ECHO_PIN, INPUT);
    pinMode(TRIG_PIN, OUTPUT);

    pinMode(r1, OUTPUT);
    pinMode(r2, OUTPUT);
    pinMode(r3, OUTPUT);
    pinMode(r4, OUTPUT);
    pinMode(r5, OUTPUT);
```

```
pinMode(r6, OUTPUT);  
pinMode(led7, OUTPUT);  
pinMode(led8, OUTPUT);
```

```
pinMode(bz, OUTPUT);  
digitalWrite(bz, LOW);
```

```
pinMode(p1, INPUT_PULLUP);  
pinMode(p2, INPUT_PULLUP);  
pinMode(p3, INPUT_PULLUP);  
pinMode(p4, INPUT_PULLUP);  
pinMode(p5, INPUT_PULLUP);
```

```
digitalWrite(r1, HIGH);  
digitalWrite(r2, HIGH);  
digitalWrite(r3, HIGH);  
digitalWrite(r4, HIGH);  
digitalWrite(r5, HIGH);
```

```
digitalWrite(led7, HIGH);  
digitalWrite(led8, HIGH);
```

```
}
```

```
void kirim(){  
  Serial2.print('A');
```

```
    delay(10);
    Serial2.print(v);
    delay(50);
    Serial2.print('B');
    delay(10);
    Serial2.print(im);
    delay(50);
    Serial2.print('C');
    delay(10);
    Serial2.print(p);
    delay(50);
    Serial2.print('D');
    delay(10);
    Serial2.print(COS);
    delay(50);
    Serial2.print('E');
    delay(10);
    Serial2.print(jarak);
    delay(100);
}

void serial(){
    if(Serial2.available()>0){
        char c = Serial2.read();
```

```
if(c=='A'){S1=1;mode=1;}
else if(c=='B'){S1=0;mode=0;}

if(c=='C'){S2=1;mode=1;}
else if(c=='D'){S2=0;mode=0;}

if(c=='E'){S3=1;mode=1;}
else if(c=='F'){S3=0;mode=0;}

if(c=='G'){S4=1;mode=1;}
else if(c=='H'){S4=0;mode=0;}

if(c=='I'){S5=1;mode=1;}
else if(c=='J'){S5=0;mode=0;}

if(c=='K'){S6=1;mode=1;}
else if(c=='L'){S6=0;mode=0;}

if (S6 == 0 || jarak > 15){digitalWrite(led7, HIGH);}
else if (S6 == 1) {digitalWrite(led7, LOW);}

if(c=='M'){S7=1;mode=1;}
else if(c=='N'){S7=0;mode=0;}

if (S7 == 0) {digitalWrite(led8, HIGH);}
```

```
    else if (S7 == 1) {digitalWrite(led8, LOW);}
  }
  pb();
}
```

```
void loop()
{
  serial();
  pb();
  LDR_A ();
  TANDON();
  DAYA();
  if(millis() - counter >= 1000){
    counter = millis();
    serial();
    kirim();
    delay(10);
  }
}
```

# Datasheet

## ➤ PIR Motion Detector Module



Motion detector module uses a motion detector IC and PCB mounted Fresnel lens.

- **General**

SB0061 is a pyroelectric sensor module which developed for human body detection. A PIR detector combined with a fresnel lens are mounted on a compact size PCB together with an analog IC, SB0061, and limited components to form the module. High level output of variable width is provided.

- **Features and Electrical Specification**

Compact size (28 x 38 mm)

Supply current : DC5V-20V(can design DC3V-24V)

Current drain : < 50uA

(Other choice : DC0.8V-4.5V; Current drain: 1.5mA-0.1mA)

Voltage Output: High/Low level signal : 3.3V

(Other choice: Open-Collector Output)

TTL output : High sensitivity

Delay time : 5s-18 minute

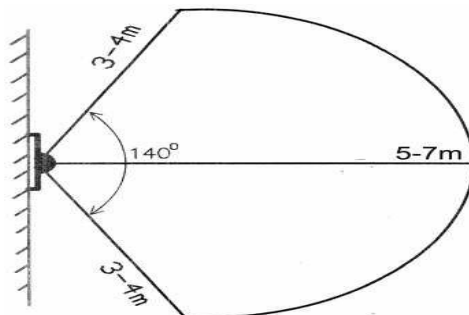
Blockade time : 0.5s-50s (acquiescently 0 seconds)

Operation Temperature: -15oC -70oC

Infrared sensor: dual element, low noise, high sensitivity

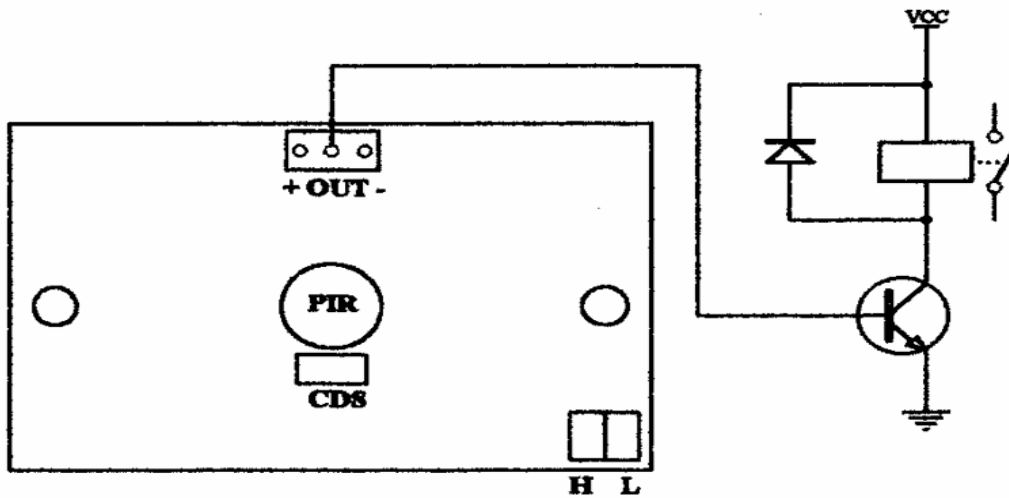
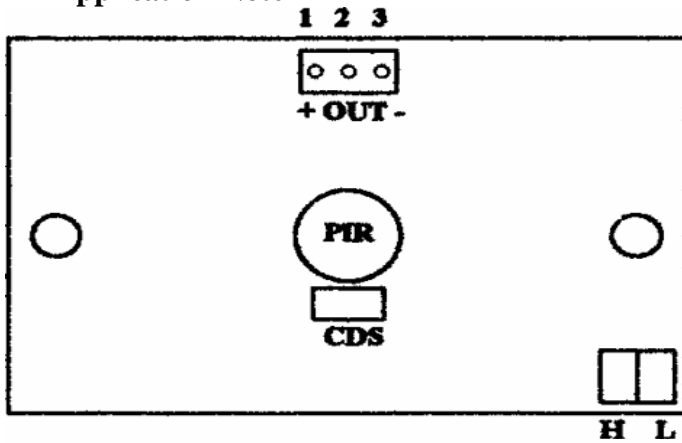
Light sensor : CdS photocell (can be add as customer requirement)

- **Lens information**





• Application Note



1. Power anode
2. Output: High level signal
3. Power cathode

H: Can be spring repeatedly

L: Can not be spring repeatedly

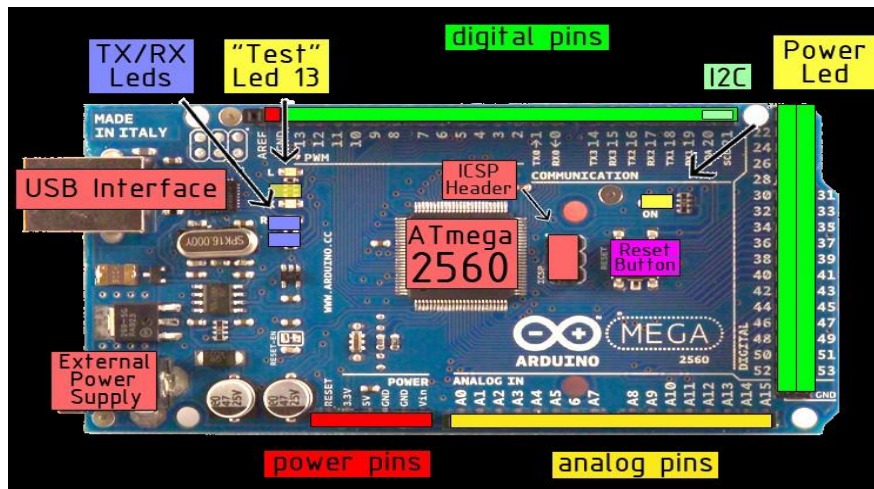
CDS: Photocell

**Note**

Due to the high sensitivity of PIR sensor device, it is not recommended to use the module in the following or similar condition.

- a) in rapid environmental changes
- b) in strong shock or vibration
- c) in a place where there are obstructing material (eg. glass) through which IR cannot pass within detection area.
- d) exposed to direct sun light
- e) exposed to direct wind from a heater or air condition

## ➤ Arduino Mega 2560



The Arduino Mega2560 can be powered via the USB connection or with an external power supply. The power source is selected automatically. External (non-USB) power can come either from an AC-to-DC adapter (wall-wart) or battery. The adapter can be connected by plugging a 2.1mm center-positive plug into the board's power jack. Leads from a battery can be inserted in the Gnd and Vin pin headers of the POWER connector. The board can operate on an external supply of 6 to 20 volts. If supplied with less than 7V, however, the 5V pin may supply less than five volts and the board may be unstable. If using more than 12V, the voltage regulator may overheat and damage the board. The recommended range is 7 to 12 volts. The Mega2560 differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it features the Atmega8U2 programmed as a USB-to-serial converter.

The power pins are as follows:

- **VIN.** The input voltage to the Arduino board when it's using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin.
- **5V.** The regulated power supply used to power the microcontroller and other components on the board. This can come either from VIN via an on-board regulator, or be supplied by USB or another regulated 5V supply.
- **3V3.** A 3.3 volt supply generated by the on-board regulator. Maximum current draw is 50 mA.
- **GND.** Ground pins.

The ATmega2560 has 256 KB of flash memory for storing code (of which 8 KB is used for the bootloader), 8 KB of SRAM and 4 KB of EEPROM (which can be read and written with the [EEPROM library](#)). Each of the 54 digital pins on the Mega can be used as an input or output, using [pinMode\(\)](#), [digitalWrite\(\)](#), and [digitalRead\(\)](#) functions. They operate at 5 volts. Each pin can provide or receive a maximum of 40 mA and has an internal pull-up resistor (disconnected by default) of 20-50 kOhms. In addition, some pins have specialized functions:

- **Serial: 0 (RX) and 1 (TX); Serial 1: 19 (RX) and 18 (TX); Serial 2: 17 (RX) and 16 (TX); Serial 3: 15 (RX) and 14 (TX).** Used to receive (RX) and transmit (TX) TTL serial data. Pins 0 and 1 are also connected to the corresponding pins of the ATmega8U2 USB-to-TTL Serial chip.

- **External Interrupts: 2 (interrupt 0), 3 (interrupt 1), 18 (interrupt 5), 19 (interrupt 4), 20 (interrupt 3), and 21 (interrupt 2).** These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value. See the [attachInterrupt\(\)](#) function for details.
- **PWM: 0 to 13.** Provide 8-bit PWM output with the [analogWrite\(\)](#) function.
- **SPI: 50 (MISO), 51 (MOSI), 52 (SCK), 53 (SS).** These pins support SPI communication, which, although provided by the underlying hardware, is not currently included in the Arduino language. The SPI pins are also broken out on the ICSP header, which is physically compatible with the Duemilanove and Diecimila.
- **LED: 13.** There is a built-in LED connected to digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off.
- **I2C: 20 (SDA) and 21 (SCL).** Support I2C (TWI) communication using the [Wire library](#) (documentation on the Wiring website). Note that these pins are not in the same location as the I2C pins on the Duemilanove.

The Mega2560 has 16 analog inputs, each of which provide 10 bits of resolution (i.e. 1024 different values). By default they measure from ground to 5 volts, though it is possible to change the upper end of their range using the AREF pin and [analogReference\(\)](#) function.

There are a couple of other pins on the board:

- **AREF.** Reference voltage for the analog inputs. Used with [analogReference\(\)](#).
- **Reset.** Bring this line LOW to reset the microcontroller. Typically used to add a reset button to shields which block the one on the board.

## ➤ Sensor LDR

### ▪ Light Dependent Resistor - LDR

Two cadmium sulphide (cds) photoconductive cells with spectral response similar to that of the human eye. The cell resistance falls with increasing light intensity. Applications include smoke detection, automatic lighting control, batch counting and burglar alarm systems.

### ▪ Applications

Photoconductive cells are used in many different types of circuits and applications.

### Analog Applications

- Camera Exposure Control
- Auto Slide Focus - dual cell
- Photocopy Machines - density of toner
- Colorimetric Test Equipment
- Densitometer
- Electronic Scales - dual cell
- Automatic Gain Control – modulated light source
- Automated Rear View Mirror

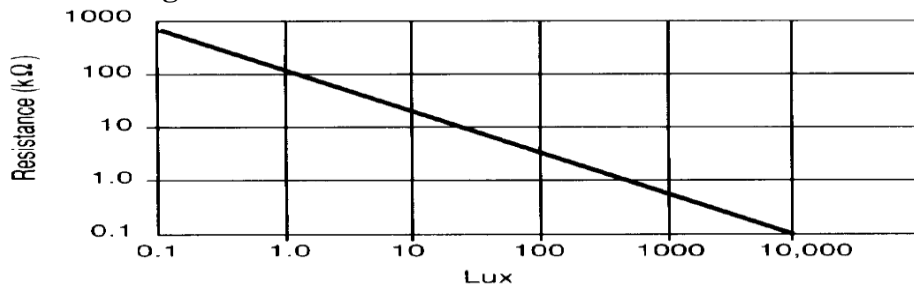
### Digital Applications

- Automatic Headlight Dimmer
- Night Light Control
- Oil Burner Flame Out
- Street Light Control
- Absence / Presence (beam breaker)
- Position Sensor

▪ **Sensitivity**

The sensitivity of a photodetector is the relationship between the light falling on the device and the resulting output signal. In the case of a photocell, one is dealing with the relationship between the incident light and the corresponding resistance of the cell.

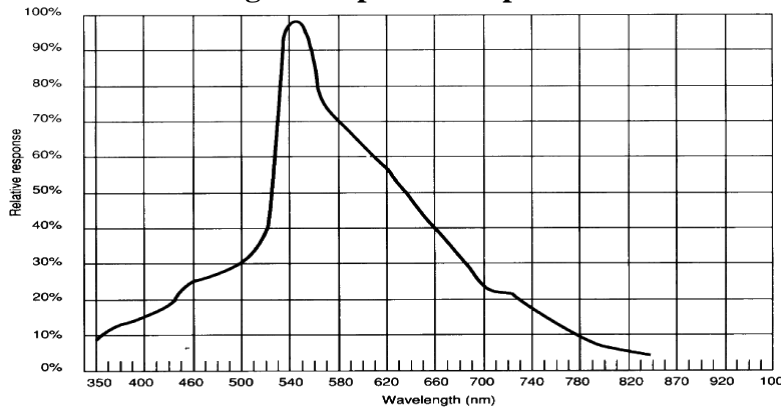
**Figure 2 Resistance As Function Of Illumination**



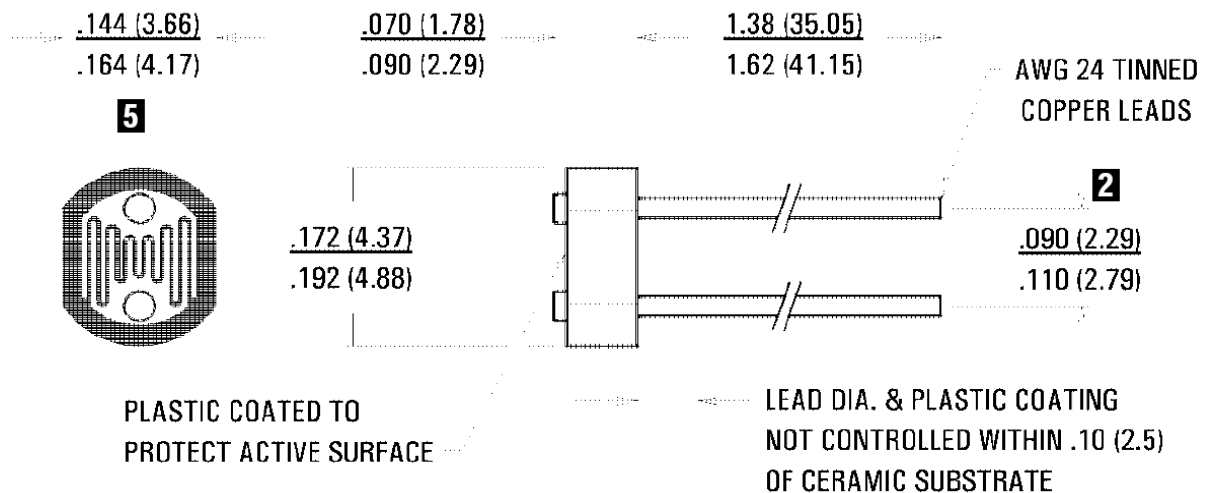
▪ **Spectral Response**

Like the human eye, the relative sensitivity of a photoconductive cell is dependent on the wavelength (color) of the incident light. Each photoconductor material type has its own unique spectral response curve or plot of the relative response of the photocell versus wavelength of

**Figure 3 Spectral response**

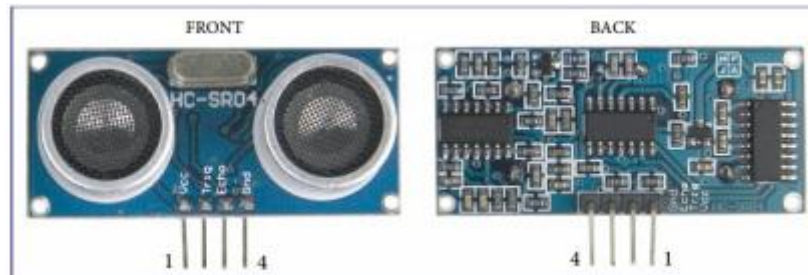


▪ **Dimensions**



➤ **HC-SR04 (Ultrasonic Sensor)**

HC-SR04 adalah Sensor Ultrasonik yang memiliki dua elemen, yaitu elemen Pendeteksi gelombang ultrasonik, dan juga sekaligus elemen Pembangkit gelombang ultrasonik. Sensor Ultrasonik adalah sensor yang dapat mendeteksi gelombang ultrasonik, yaitu gelombang suara yang memiliki frekuensi ultrasonik atau frekuensi di atas kisaran frekuensi pendengaran manusia.



**Gambar 1. Ultrasonic Sensor**

Fungsi Pin-pin HC-SR04

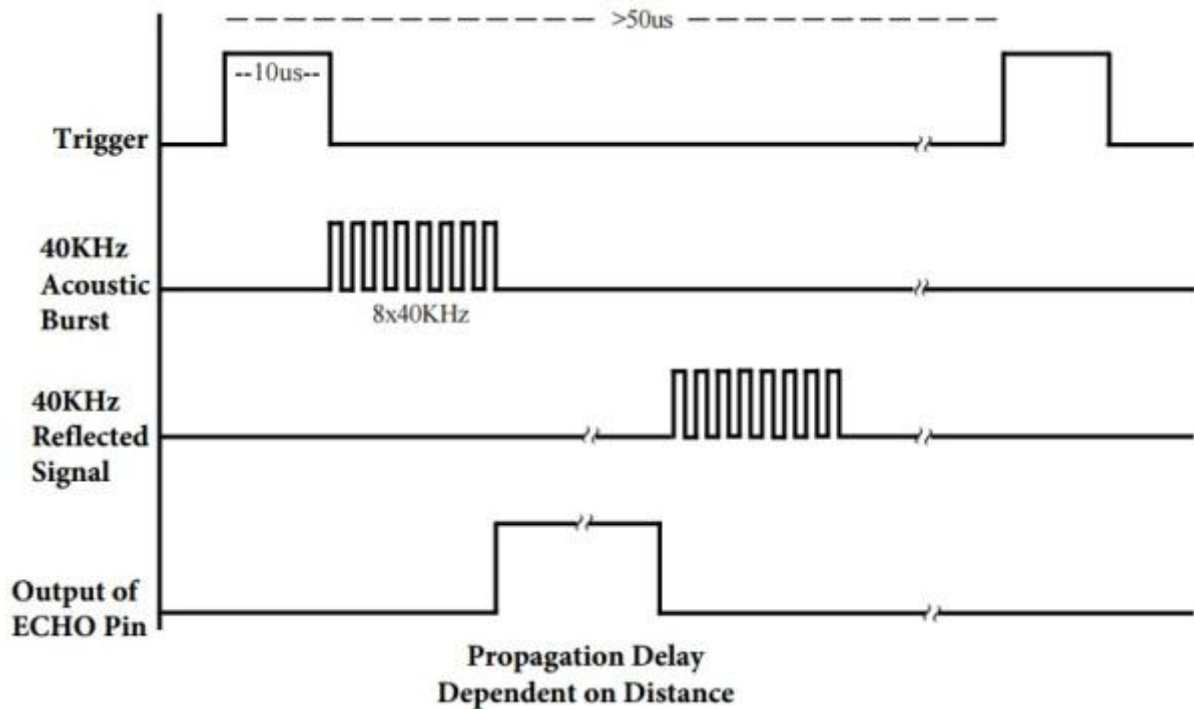
1. **VCC** = *5V Power Supply*. Pin sumber tegangan positif sensor.
2. **Trig** = *Trigger/Penyulut*. Pin ini yang digunakan untuk membangkitkan sinyal ultrasonik.
3. **Echo** = *Receive/Indikator*. Pin ini yang digunakan untuk mendeteksi sinyal pantulan ultrasonik.
4. **GND** = *Ground/0V Power Supply*. Pin sumber tegangan negatif sensor.

▪ **Karakteristik HC-SR04**

- Tegangan sumber operasi tunggal 5.0 V
- Konsumsi arus 15 mA
- Frekuensi operasi 40 KHz
- Minimum pendeteksi jarak 0.02 m (2 cm)
- Maksimum pendeteksian jarak 4 m
- Sudut pantul gelombang pengukuran 15 derajat
- Minimum waktu penyulutan 10 mikrodetik dengan pulsa berlevel TTL
- Pulsa deteksi berlevel TTL dengan durasi yang bersesuaian dengan jarak deteksi
- Dimensi 45 x 20 x 15 mm

▪ **Diagram Waktu HC-SR04**

HC-SR04 memerlukan sinyal logika '1' pada pin **Trig** dengan durasi waktu 10 mikrodetik (us) untuk mengaktifkan rentetan (*burst*) 8x40KHz gelombang ultrasonik pada elemen Pembangkitnya. Selanjutnya pin **Echo** akan berlogika '1' setelah rentetan 8x40 KHz tadi, dan otomatis akan berlogika '0' saat gelombang pantulan diterima oleh elemen Pendeteksi gelombang ultrasonik.



**Gambar 2.** Diagram Waktu HC-SR04.

## ➤ ESP8266

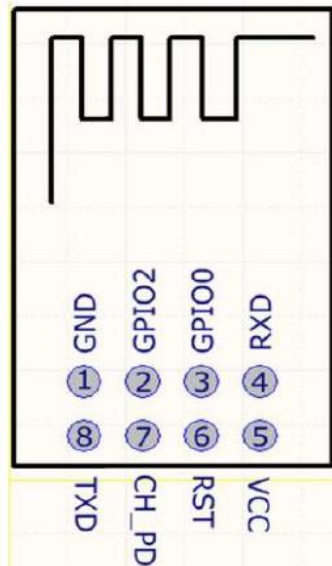
ESP8266 is among the most integrated WiFi chip in the industry; it integrates the antenpower amplifier, low noise receive amplifier, filters, power management modules, and the entire solution, including front-end module, is designed ESP8266EX also integrates an enhanced version of Tensilica's L106 Diamond series 32 SRAM, besides the Wi-Fi functionalities. ESP8266EX is often integrated with specific devices through its GPIOs; codes for such applications are Espressif Systems' Smart Connectivity Platform (ESCP) demonstrates sophisticated system fast sleep/wake context switching for energy signal processing, and spur cancellation and radio co LCD interference mitigation.

### ▪ Features

- 802.11 b/g/n
- Integrated low power 32-bit MCU
- Integrated 10-bit ADC
- Integrated TCP/IP protocol stack
- Integrated TR switch, balun, LNA, power amplifier and matching network
- Integrated PLL, regulators, and power management units
- Supports antenna diversity
- Wi-Fi 2.4 GHz, support WPA/WPA2
- Support STA/AP/STA+AP operation modes
- Support Smart Link Function for both Android and iOS devices
- Support Smart Link Function for both Android and iOS devices
- SDIO 2.0, (H) SPI, UART, I2C, I2S, IRDA, PWM,
- STBC, 1x1 MIMO, 2x1 MIMO
- A-MPDU & A-MSDU aggregation and 0.4s guard interval

- Deep sleep power <10uA, Power down leakage current < 5uA
- Wake up and transmit packets in < 2ms
- Standby power consumption of < 1.0mW (DTIM3)
- +20dBm output power in 802.11b mode
- Operating temperature range -40C ~ 125C

▪ **Pin Descriptions**



Gambar 1. Pin Descriptions

NO	PIN NAME	Function
1	GND	GND
2	GPIO2	GPIO,Internal Pull
3	GPIO0	GPIO,Internal Pull
4	RXD	UART0,data received pin RXD
5	VCC	3.3V power supply (VDD)
6	RST	1) External reset pin
		2) Can loft or external MCU
7	CH_PD	Chip enable pin. Active high
8	TXD	UART0, data send pinRXD

▪ **Parameter**

Parameters	Min	Typical	Max	Unit
Input frequency	2412		2484	MHz
Input impedance		50		$\Omega$
Input reflection			-10	dB
Output power of PA for 72.2Mbps	15.5	16.5	17.5	dBm
Output power of PA for 11b mode	19.5	20.5	21.5	dBm
Sensitivity				
DSSS, 1Mbps		-98		dBm
CCK, 11Mbps		-91		dBm
6Mbps (1/2 BPSK)		-93		dBm
54Mbps (3/4 64-QAM)		-75		dBm
HT20, MCS7 (65Mbps, 72.2Mbps)		-72		dBm
<b>Adjacent Channel Rejection</b>				
OFDM, 6Mbps		37		dB
OFDM, 54Mbps		21		dB
HT20, MCS0		37		dB
HT20, MCS7		20		dB

➤ ***Buck Converter***

**Features**

- 3.3-V, 5-V, 12-V, and Adjustable Output Versions
- Adjustable Version Output Voltage Range: 1.2-V to 37-V  $\pm$  4% Maximum Over Line and Load
- Conditions
- Available in TO-220 and TO-263 Packages
- 3-A Output Load Current
- Input Voltage Range Up to 40 V
- Requires Only 4 External Components
- Excellent Line and Load Regulation Specifications
- 150-kHz Fixed-Frequency Internal Oscillator
- TTL Shutdown Capability
- Low Power Standby Mode, IQ, Typically 80  $\mu$ A
- High Efficiency
- Uses Readily Available Standard Inductors
- Thermal Shutdown and Current-Limit Protection
- Create a Custom Design Using the LM2596 with the WEBENCH Power Designer

**Applications**

- Simple High-Efficiency Step-Down (Buck)Regulator
- On-Card Switching Regulators
- Positive to Negative Converter



## Electrical Characteristics – 3.3-V Version

Specifications are for  $T_J = 25^\circ\text{C}$  (unless otherwise noted)

PARAMETER		TEST CONDITIONS		MIN <sup>(1)</sup>	TYP <sup>(2)</sup>	MAX <sup>(1)</sup>	UNIT
<b>SYSTEM PARAMETERS<sup>(3)</sup></b> (see Figure 35 for test circuit)							
$V_{\text{OUT}}$	Output voltage	$4.75\text{ V} \leq V_{\text{IN}} \leq 40\text{ V}$ , $0.2\text{ A} \leq I_{\text{LOAD}} \leq 3\text{ A}$	$T_J = 25^\circ\text{C}$	3.168	3.3	3.432	V
			$-40^\circ\text{C} \leq T_J \leq 125^\circ\text{C}$	3.135		3.465	
$\eta$	Efficiency	$V_{\text{IN}} = 12\text{ V}$ , $I_{\text{LOAD}} = 3\text{ A}$		73%			

- (1) All room temperature limits are 100% production tested. All limits at temperature extremes are specified via correlation using standard Statistical Quality Control (SQC) methods. All limits are used to calculate Average Outgoing Quality Level (AOQL).
- (2) Typical numbers are at  $25^\circ\text{C}$  and represent the most likely norm.
- (3) External components such as the catch diode, inductor, input and output capacitors can affect switching regulator system performance. When the LM2596 is used as shown in Figure 35, system performance is shown in the test conditions column.

## Electrical Characteristics – 5-V Version

Specifications are for  $T_J = 25^\circ\text{C}$  (unless otherwise noted)

PARAMETER		TEST CONDITIONS		MIN <sup>(1)</sup>	TYP <sup>(2)</sup>	MAX <sup>(1)</sup>	UNIT
<b>SYSTEM PARAMETERS<sup>(3)</sup></b> (see Figure 35 for test circuit)							
$V_{\text{OUT}}$	Output voltage	$7\text{ V} \leq V_{\text{IN}} \leq 40\text{ V}$ , $0.2\text{ A} \leq I_{\text{LOAD}} \leq 3\text{ A}$	$T_J = 25^\circ\text{C}$	4.8	5	5.2	V
			$-40^\circ\text{C} \leq T_J \leq 125^\circ\text{C}$	4.75		5.25	
$\eta$	Efficiency	$V_{\text{IN}} = 12\text{ V}$ , $I_{\text{LOAD}} = 3\text{ A}$		80%			

- (1) All room temperature limits are 100% production tested. All limits at temperature extremes are specified via correlation using standard Statistical Quality Control (SQC) methods. All limits are used to calculate Average Outgoing Quality Level (AOQL).
- (2) Typical numbers are at  $25^\circ\text{C}$  and represent the most likely norm.
- (3) External components such as the catch diode, inductor, input and output capacitors can affect switching regulator system performance. When the LM2596 is used as shown in Figure 35, system performance is shown in the test conditions column.

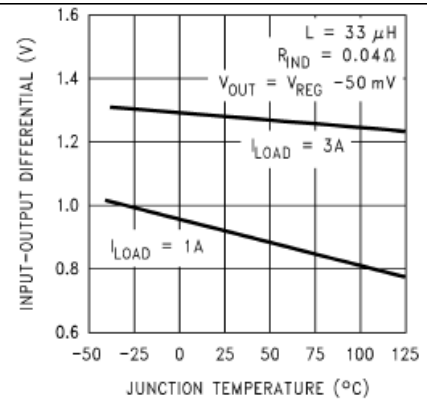
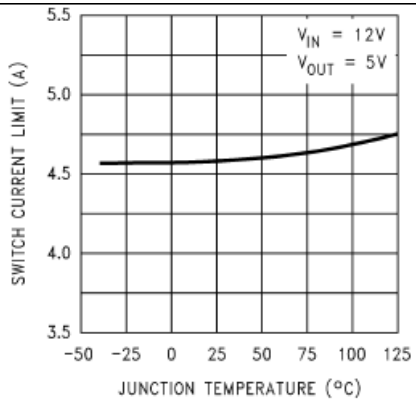
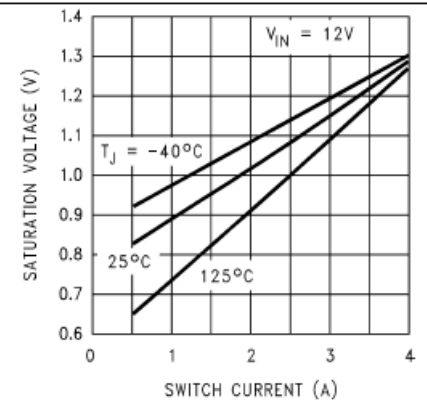
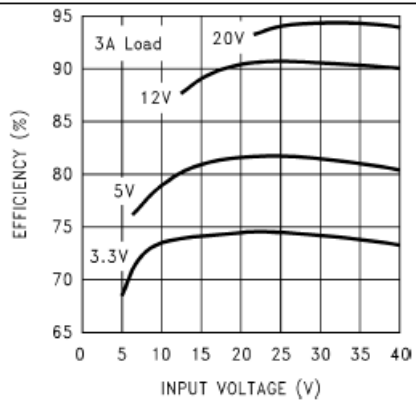
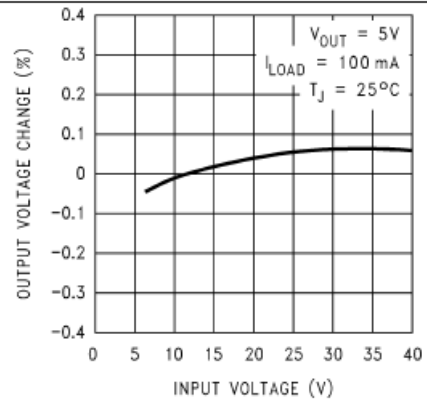
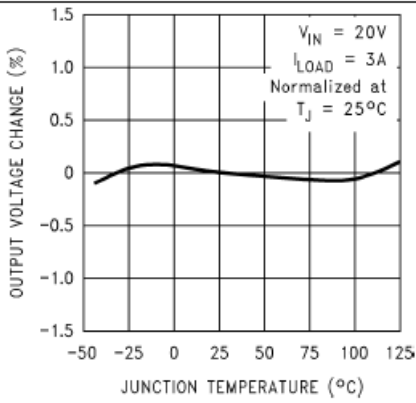
## Electrical Characteristics – 12-V Version

Specifications are for  $T_J = 25^\circ\text{C}$  (unless otherwise noted)

PARAMETER		TEST CONDITIONS		MIN <sup>(1)</sup>	TYP <sup>(2)</sup>	MAX <sup>(1)</sup>	UNIT
<b>SYSTEM PARAMETERS<sup>(3)</sup></b> (see Figure 35 for test circuit)							
$V_{\text{OUT}}$	Output voltage	$15\text{ V} \leq V_{\text{IN}} \leq 40\text{ V}$ , $0.2\text{ A} \leq I_{\text{LOAD}} \leq 3\text{ A}$	$T_J = 25^\circ\text{C}$	11.52	12	12.48	V
			$-40^\circ\text{C} \leq T_J \leq 125^\circ\text{C}$	11.4		12.6	
$\eta$	Efficiency	$V_{\text{IN}} = 25\text{ V}$ , $I_{\text{LOAD}} = 3\text{ A}$		90%			

- (1) All room temperature limits are 100% production tested. All limits at temperature extremes are specified via correlation using standard Statistical Quality Control (SQC) methods. All limits are used to calculate Average Outgoing Quality Level (AOQL).
- (2) Typical numbers are at  $25^\circ\text{C}$  and represent the most likely norm.
- (3) External components such as the catch diode, inductor, input and output capacitors can affect switching regulator system performance. When the LM2596 is used as shown in Figure 35, system performance is shown in the test conditions column.

## Typical Characteristics



## AC digital display Multifunction Meter

### Product Type: PZEM-004(V3.0)

#### A. Function

1. Electrical parameter measurement function (voltage, current, active power, energy).
2. Overload alarm function (over power alarm threshold the power flash and the buzzer beeping to alarm).
3. Power alarm threshold preset function ( can set power alarm threshold).
4. The reset function of energy key.
5. Store data when power off (store the accumulated energy before power off).
6. Bright red digital display function (display voltage, current, active power, energy).
7. Serial communication function (with TTL serial interface itself, can communicate with a variety of terminal through the pin board, read and set the parameters).

#### B. Front display and key

##### I. Display Interface

Display interface is formed by four bright red digital tubes, used to display the voltage, current, power, energy parameters.

##### II. Display Format

1. Power: Test Range: 0 ~ 22kW  
Within 0 ~ 10kW, the display format is 0.000 ~ 9.999;  
Within 10 ~ 22kW, the display format is 10.00 ~ 22.00.
2. Energy: Test Range: 0 ~ 9999kWh  
Within 0 ~ 10kWh, the display format is 0.000 ~ 9.999;  
Within 10 ~ 100kWh, the display format is 10.00 ~ 99.99;  
Within 100 ~ 1000kWh, the display format is 100.0 ~ 999.9;  
1000 ~ 9999kWh and above, the display format is 1000 ~ 9999.
3. Voltage: Test Range: 80 ~ 260VAC  
Display Format is 110.0 ~ 220.0.
4. Current: Test Range: 0 ~ 100A  
Display Format is 00.00 ~ 99.99.

##### III. Key

There is a key on the panel, it can be used to reset energy.

The method of reset energy: Long press the key for 5 seconds until the digital on energy display window flicker, then release the key. Short press the key again, then the energy data is cleared and quit the flickering state, now the reset operation is completed; if long press for 5 seconds again until no longer flicker, it means exit the reset state.

#### C. Wiring diagram

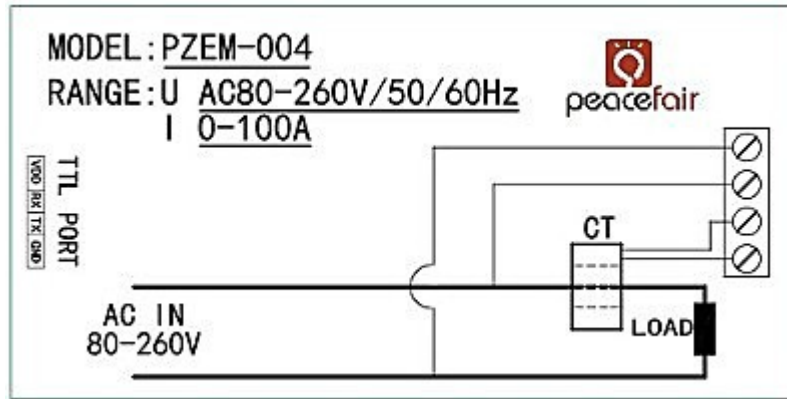


Figure 1 Wiring diagram

The wiring of this module is divided into two parts: the voltage and current test input terminal wiring and the serial communication wiring, as shown in Figure 1; according to the actual needs of the clients, with different TTL pin board to achieve communicate with different terminals.

#### D. Display Interface

The whole meter panel display window is formed by four windows, they are voltage, current, power and energy; the following are brief description of each parameter display:

##### 1. Voltage Display

Measure and display the current power frequency grid voltage.

##### 2. Current display

Measure and display the current load (appliances) current. There is supplementary instruction that the current test value is from the beginning of 10mA , but this module belongs to high power test equipment, if you care about the mA level current testing accuracy, it is not be recommended.

##### 3. Energy display

Measure and display the current accumulative power consumption. There is supplementary instruction that the minimum unit of the energy metering is 0.001kWh,which means it begins to accumulate from 1Wh, relatively speaking, the resolution is rather high, for the low-power(within 100W)load test, you can observe the accumulative process rather intuitively.

##### 4. Power display

Measure and display the current load power. There is supplementary instruction that the power test value is from the beginning of 0.001kW , which means it begins to test from 1W, but this module belongs to high power test equipment, if you have the requirement of the testing within 1W, it is not be recommended.

#### E. Serial communication

This module is equipped with TTL serial data communication interface, you can read and set the relevant parameters via the serial port; but if you want to communicate with a device which has USB or RS232 (such as computer), you need to be equipped with different TTL pin board (USB communication needs to be equipped with TTL to USB pin board; RS232 communication needs to be equipped with TTL to RS232 pin board), the specific connection type as shown in Figure 2. In the below table are the communication protocols of this module:

NO.	function	Head	Data1- Data5	Sum
1	voltage	B0	C0 A8 01 01 00 (Computer sends a request to read the voltage value)	1A

		A0	00 E6 02 00 00 (Meter reply the voltage value is 230.2V)	88
2	current	B1	C0 A8 01 01 00 (Computer sends a request to read the current value)	1B
		A1	00 11 20 00 00 (Meter reply the current value is 17.32A)	D2
3	Active power	B2	C0 A8 01 01 00 (Computer sends a request to read the active power value)	1C
		A2	08 98 00 00 00 (Meter reply the active power value is 2200w)	42
4	Read energy	B3	C0 A8 01 01 00 (Computer sends a request to read the energy value)	1D
		A3	01 86 9f 00 00 (Meter reply the energy value is 99999wh)	C9
5	Set the module address	B4	C0 A8 01 01 00 (Computer sends a request to set the address, the address is 192.168.1.1)	1E
		A4	00 00 00 00 00 (Meter reply the address was successfully set)	A4
6	Set the power alarm threshold	B5	C0 A8 01 01 14 (computer sends a request to set a power alarm threshold)	33
		A5	00 00 00 00 00 (Meter reply the power alarm threshold was successfully set)	A5

**Illustration of the communication protocol example:**

**1. Set the communication address: 192.168.1.1**

Send command: B4 C0 A8 01 01 00 1E

Reply data: A4 00 00 00 00 00 A4

**Note:** The above example illustrate that setting the communication address as 192.168.1.1 (the user can set their own address based on their preferences and needs), sending commands and replying data automatically are as shown above, the data are expressed in hexadecimal; the last byte of the sending and replying data are 1E and A4, belong to cumulative sum. At sending commands: B4 + C0 + A8 + 01 + 01 + 00 = 21E (use the hexadecimal addition), the cumulative sum data is 21E, take the last two bytes 1E to be used the cumulative sum data in sending commands; data in reply: A4 + 00 + 00 + 00 + 00 + 00 = A4 (use the hexadecimal addition),the cumulative sum data is A4,which is the cumulative sum data in reply.

The explanation of the cumulative sum is now finished, the following parameter examples are the same as this, there is no explanation any more.

**2. Set the power alarm threshold:20 KW**

Send command: B5 C0 A8 01 01 14 33

Reply data: A5 00 00 00 00 00 A5

**Note:** 14 in the sending command is the alarm value (14 is a hexadecimal data representation, which converted to decimal is 20). What you should note is the power alarm value of this module is based on KW units, which means the minimum alarm value is 1KW, the maximum value is 22KW.

**3. Read the current voltage**

Send command: B0 C0 A8 01 01 00 1A

Reply data: A0 00 E6 02 00 00 88

**Note:** Reply voltage data is D1D2D3 = 00 E6 02, 00 E6 represent the integer-bit of the voltage, 02 represent the decimal of the voltage, the decimal is one digit, converts 00 E6 to decimal is 230; converts 02 to decimal is 2, so the current voltage value is 230.2V.

#### **4. Read the current current**

Send command: B1 C0 A8 01 01 00 1B

Reply data: A1 00 11 20 00 00 D2

**Note:** Reply current data is D2D3 = 11 20, 11 represent the integer-bit of the current, 20 represent the decimal of the current, the current decimal is two digits, converts 11 to decimal is 17; converts 20 to decimal is 32, so the current current value is 17.32 A.

#### **5. Read the current power**

Send command: B2 C0 A8 01 01 00 1C

Reply data: A2 08 98 00 00 00 42

**Note:** Reply power data is D1D2 = 08 98, converts 08 98 to decimal is 2200, so the current voltage value is 2200W.

#### **6. Read the energy**

Send command: B3 C0 A8 01 01 00 1D

Reply data: A3 01 86 9F 00 00 C9

**Note:** Reply energy data is D1D2D3 = 01 86 9F, converts 01 86 9F to decimal is 99999, so the accumulated power is 99999Wh.

### **F. Illustration of the communication**

1. Connect hard wire according to the wiring diagram in figure 1 and 2.
2. After connect the wire, please choose the communication port, this module's upper computer software support communication port: COM2\COM3\COM4, you can check through device manager, if it is not the above communication port, you should amend it through port.

### **G. Precautions**

1. This module is suitable for indoor, please do not use outdoor.
2. Applied load should not exceed the rated power.
3. Wiring order can't be wrong.

### **H. Specification parameters**

1. Working voltage: 80 ~ 260VAC
2. Test voltage: 80 ~ 260VAC
3. Rated power: 100A/22000W
4. Operating frequency: 45-65Hz
5. Measurement accuracy: 1.0 grade