

STOPKONTAK PINTAR BERBASIS ANDROID

PROYEK AKHIR

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



Disusun Oleh :

K'K Angrainy	NIRM	0031514
Steven Novaldy F	NIRM	0031529

**POLITEKNIK MANUFAKTUR NEGERI
BANGKA BELITUNG
TAHUN 2018**

LEMBAR PENGESAHAN

STOPKONTAK PINTAR BERBASIS ANDROID

Oleh :

K'K Anggrainy	NIRM	0031514
Steven Novaldy F	NIRM	0031529

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

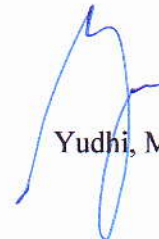
Menyetujui,

Pembimbing 1



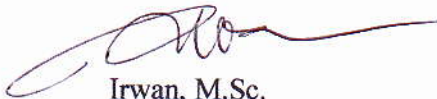
Indra Dwisaputra, M.T.

Pembimbing 2



Yudhi, M.T.

Penguji 1



Irwan, M.Sc.

Penguji 2



Aan Febriansyah, M.T.

Penguji 3



Ocsirendi, M.T.

PERNYATAAN BUKAN PLAGIAT

Yang bertanda tangan di bawah ini:

Nama Mahasiswa 1 : K'K Anggrainy

NIRM : 0031514

Nama Mahasiswa 2 : Steven Novaldy F

NIRM : 0031529

Dengan Judul : Stopkontak Pintar Berbasis Android


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

Sungailiat, 10 Mei 2018


Nama Mahasiswa

Tanda Tangan

1. K'K Anggrainy


.....

2. Steven Novaldy F


.....

ABSTRAK

Dengan banyaknya kebutuhan pemakaian listrik, pengguna perlu mengetahui pemakaian daya setiap beban listrik yang digunakan. Biasanya untuk mengetahui pemakaian daya listrik diperlukan alat pengukur daya listrik (wattmeter). Dengan berkembangnya teknologi, Smartphone yang sedang diminati oleh kebanyakan masyarakat dapat dimanfaatkan sebagai media untuk mengetahui pemakaian daya listrik tersebut. Alat ini merupakan sebuah sistem yang dibuat untuk mempermudah pengguna dalam pengontrolan stopkontak yaitu melalui sebuah perangkat android yang dapat diakses melalui hubungan bluetooth. Alat ini juga memiliki kemampuan untuk mengetahui nilai daya listrik (dalam satuan VA) dan biaya yang telah terpakai. Untuk mengetahui nilai daya real-time tersebut digunakan sensor PZEM-004T. Dan dalam pengontrolan on dan off stopkontak tersebut digunakan relay. Sedangkan untuk hubungan dari alat dan smartphone digunakan perangkat bluetooth HC-05. Semua perangkat tersebut diolah oleh mikrokontroler Arduino Uno R3. Berdasarkan hasil pengujian alat tersebut ketika dibandingkan dengan alat ukur sebenarnya dapat disimpulkan bahwa alat tersebut memiliki rata-rata error dibawah 3%

Kata kunci : Stopkontak, arduino, bluetooth, sensor, daya, android.

ABSTRACT

With amount of requirement electrical power, user must know power consumption of every power load used. Generally to know consumption of power load required electrical power measurement (wattmeter). With development of technology, Smartphone which interest by most of people can be used as media to know consumption of electrical power load. This device is a created system to simplify user in control of stopcontact through an android device which access by bluetooth connection. This device too have competence to know value of electrical power(in VA) and cost which has been used. To know value of electrical power used a sensor PZEM-004T. And to controlled on and off the stopcontact used a relay. Meanwhile to connection from device and smartphone used a bluetooth HC-05. All of components processed by microcontroller Arduino Uno R3. Based on result of examination this device when compared with real measuring instrument can conclude that devuce have error avverage under 3%.

Keywords: Stopcontact, arduino, bluetooth, sensor, power, android.

KATA PENGANTAR

Assalamualaikum wa rahmatullahi wabarakatuh.

Puji dan syukur penulis ucapkan kepada Allah SWT, karena berkat rahmat dan ridho-Nya penulis dapat menyelesaikan makalah proyek akhir ini tepat pada waktunya. Proyek akhir “Stopkontak Pintar Berbasis Android” merupakan salah satu syarat setiap kelompok kerja proyek akhir untuk memenuhi persyaratan pendidikan Diploma III di Politeknik Manufaktur Negeri Bangka Belitung .

Dalam menyelesaikan makalah proyek akhir ini, penulis mendapat bimbingan dan dorongan dari berbagai pihak, baik secara dukungan maupun material. Untuk itu, pada kesempatan ini dengan segala kerendahan dan ketulusan hati penulis menyampaikan rasa terima kasih yang sebesar-besarnya kepada:

1. Keluarga tercinta, khususnya kedua orang tua tercinta yang selalu sabar membimbing, mendoakan, dan memberikan memotivasi serta menasehati penulis. Terima kasih untuk setiap peluh keringat serta kasih sayang yang tidak ternilai
2. Bapak Sugeng Ariyono, M.Eng. Ph.D selaku Direktur Politeknik Manufaktur Negeri Bangka Belitung
3. Bapak Indra Dwisaputra, M.T. selaku dosen pembimbing 1 yang selama ini banyak membantu memberi masukan selama menjadi pembimbing kami
4. Bapak Yudhi, M.T. selaku dosen pembimbing 2 yang selama ini banyak membantu memberi masukan selama menjadi pembimbing kami
5. Bapak Eko Sulistyono, M.T. selaku Kepala Prodi DIII Teknik Elektronika Politeknik Manufaktur Negeri Bangka Belitung
6. Segenap dosen dan staf pengajar Politeknik Manufaktur Negeri Bangka Belitung
7. Rekan tim proyek akhir dan teman-teman seangkatan

8. Pihak–pihak lain yang telah memberikan bantuan secara langsung maupun tidak langsung dalam pembuatan tugas akhir ini yang tidak dapat disebutkan satu persatu

Penulis menyadari bahwa penulisan makalah proyek akhir ini masih terdapat banyak kekurangan baik dari segi penulisan maupun isi makalah. Oleh karena itu, penulis mengharapkan segala petunjuk, kritik, dan saran yang membangun dari pembaca agar dapat menunjang pengembangan dan perbaikan penulisan selanjutnya. Akhir kata, penulis mohon maaf atas kekurangan dalam penulisan makalah ini dan penulis dengan senang hati menerima saran dan kritik yang membangun dari pembaca.

Semoga makalah ini dapat berguna untuk menambah wawasan dan wacana bagi rekan-rekan mahasiswa.

Waalaikumsalam warahmarullahi wabarakatuh.

Sungailiat, 12 Mei 2018

Penulis

DAFTAR ISI

	Halaman
LEMBAR PENGESAHAN	ii
PERNYATAAN BUKAN PLAGIAT	iii
ABSTRAK	iv
ABSTRACT	v
KATA PENGANTAR	vi
DAFTAR ISI	viii
DAFTAR TABEL	ix
DAFTAR GAMBAR	xii
DAFTAR LAMPIRAN	xiv
BAB I PENDAHULUAN	1
1.1 Latar Belakang Masalah.....	1
1.2 Rumusan dan Batasan Masalah.....	2
1.2.1 Rumusan Masalah	2
1.2.2 Batasan Masalah.....	2
1.3 Tujuan	2
BAB II DASAR TEORI	3
2.1 Deskripsi Umum Energi Listrik dan Daya.....	3
2.7 App Inventor 2	6
2.8 Android	8
BAB III METODE PENELITIAN	13
3.1. <i>Flow Chart</i> Pembuatan Alat	13
3.2. Pengumpulan Data	14
3.3 Desain Peralatan.....	14
3.4 Pembuatan <i>Hardware</i>	15
3.5 Pembuatan Program	15
3.5.1 Pembuatan Program Arduino UNO R3.....	15
3.5.2 Pembuatan Aplikasi Android (Stopkontak).....	16

3.6 Uji Coba	16
3.7 Perbaikan.....	16
3.8 Kesimpulan	17
BAB IV PEMBAHASAN	18
4.1 Diagram Blok	18
4.2 Modul Sensor PZEM-004T.....	20
4.3 <i>Bluetooth</i> HC-05	25
4.3.1 Pengujian <i>Bluetooth</i> HC-05	26
4.4 Modul <i>Relay 2 Channel</i>	27
4.4.1 Pengujian Modul <i>Relay 2 Channel</i>	28
4.5 <i>Real Time Clock</i> DS3231	29
4.5.1 Pengujian <i>Real Time Clock</i> DS3231	30
4.6 Aplikasi Android (Stopkontak).....	31
4.6.1 Pengujian Aplikasi Android (Stopkontak)	35
4.7 <i>Program</i> Arduino UNO R3.....	36
4.8 Pengujian Stopkontak	36
4.8.1 Pengujian Keseluruhan Alat dan Aplikasi Stopkontak	37
BAB V PENUTUP	40
5.1. Kesimpulan	40
5.2. Saran.....	40
DAFTAR PUSTAKA	41
LAMPIRAN	43

DAFTAR TABEL

Tabel	Halaman
4.1 Hasil Pengujian Tegangan pada modul sensor PZEM-004T	22
4.2 Hasil Pengujian Arus pada modul sensor PZEM-004T	23
4.3 Hasil Pengujian Daya pada modul sensor PZEM-004T	24
4.4 Hasil Pengujian <i>Bluetooth</i> HC-05	27
4.5 Hasil Pengujian modul <i>Relay 2 Channel</i>	29
4.6 Hasil Pengujian Keseluruhan Alat dan Aplikasi	37

DAFTAR GAMBAR

Gambar 2.1 App Inventor 2	7
Gambar 2.2 Android.....	8
Gambar 3.1 <i>Flow Chart</i> Metode Penelitian	13
Gambar 4.1 Diagram Blok Stopkontak Pintar Berbasis Android	18
Gambar 4.2 Skematik Rangkaian Keseluruhan	19
Gambar 4.3 Hasil Pengukuran Tegangan pada modul sensor PZEM-004T	22
Gambar 4.4 Hasil Pengukuran Arus pada modul sensor PZEM-004T	23
Gambar 4.5 Hasil Pengukuran Daya pada modul sensor PZEM-004T	24
Gambar 4.6 Skematik Pemasangan Pin pada <i>Bluetooth</i> HC-05	25
Gambar 4.7 Skematik Pemasangan Pin pada Modul <i>Relay 2 Channel</i>	28
Gambar 4.8 Skematik Pemasangan Pin pada <i>Real Time Clock</i> DS3231	29
Gambar 4.9 Hasil Pengujian <i>Real Time Clock</i> DS3231.....	30
Gambar 4.10 <i>Layout</i> pada <i>Screen</i> 1	32
Gambar 4.11 <i>Layout</i> pada <i>Screen</i> 2	32
Gambar 4.12 <i>Layout</i> pada <i>Screen</i> 3	33
Gambar 4.13 <i>Layout</i> pada <i>Screen</i> 1 dan 2 di Android	34
Gambar 4.14 <i>Layout</i> pada <i>Screen</i> 3 di Android.....	34
Gambar 4.15 Hasil Perakitan Komponen pada Stopkontak.....	37
Gambar 4.16 Hasil Pengujian Stopkontak dengan beban Lampu 45 <i>Watt</i>	38
Gambar 4.17 Hasil Pengujian Stopkontak dengan beban Kipas Angin.....	39

DAFTAR LAMPIRAN

Lampiran 1: Daftar Riwayat Hidup

Lampiran 2: Program Arduino

Lampiran 3: Program Android

Lampiran 4: *Data Sheet PZEM-004T*

BAB I

PENDAHULUAN

1.1 Latar Belakang Masalah

Energi listrik merupakan energi yang sangat vital dalam operasional perusahaan maupun rumah tangga. Hampir semua peralatan dan sumber penerangan yang digunakan berasal dari energi listrik. Namun dalam pemakaiannya banyak dikeluhkan adanya kerugian pada penggunaan energi listrik ini. Hal ini terbukti dari cepat habisnya pulsa listrik Kwh Prabayar. Terlebih apabila ini terjadi dilingkungan kantor atau perusahaan maka akan mengakibatkan anggaran biaya yang cukup besar. Untuk mengefisiensikan pemakaian listrik, maka perlu dilakukan manajemen pemakaian listrik untuk penghematan energi.

Sebelumnya [1] telah dibuat alat dalam jurnal yang berjudul “Rancang Bangun Sistem Proteksi dan Monitoring Penggunaan Daya Listrik pada Beban Skala Rumah Tangga Berbasis Mikrokontroler ATmega328P, di mana alat tersebut menggunakan sensor arus ACS712 dan sensor tegangan ZMPT101B. Alat tersebut dapat mengukur dan menampilkan daya dengan sebuah komunikasi serial dengan modul GSM SIM900. Namun alat tersebut melakukan monitoring kepada keseluruhan daya rumah tangga. Peneliti lainnya [2] membuat alat ukur daya listrik berupa *power* meter digital berbasis mikrokontroler AVR ATmega8535 menggunakan perangkat keras berupa sensor arus ACS712. Alat ini dapat mengukur dan menampilkan tegangan, arus dan daya listrik suatu peralatan elektronik rumah tangga yang hanya ditampilkan melalui LCD.

Oleh karena itu, perlu adanya sistem monitoring yang dapat mengecek penggunaan daya yang digunakan oleh setiap beban listrik yang digunakan. Dengan ini penulis memodifikasi stopkontak yang sudah ada menjadi stopkontak yang bisa memonitoring penggunaan daya listrik dan biaya penggunaan listrik yang dikontrol melalui aplikasi yang berbasis android secara langsung. penulis mempunyai ide dengan membuat alat yang berjudul “Stopkontak Pintar Berbasis

Android” mampu mengecek lebih detail setiap daya yang digunakan pada beban listrik dan biaya dari beban listrik tersebut melalui android. Sehingga pengguna dapat mengetahui penggunaan daya pada setiap beban dengan mudah melalui Android.

1.2 Rumusan dan Batasan Masalah

1.2.1 Rumusan Masalah

Rumusan masalah dari Stopkontak Pintar Berbasis Android ini sebagai berikut :

- a. Bagaimana cara menerapkan komunikasi data dari Arduino ke Android ataupun sebaliknya melalui Bluetooth.
- b. Bagaimana cara pengambilan data dari sensor arus dan sensor tegangan.

1.2.2 Batasan Masalah

Dalam penyusunan proyek akhir ini akan dibatasi pada masalah berikut :

- a. Karena menggunakan komunikasi *Bluetooth*, pengguna hanya bisa mengakses alat tersebut dalam radius kurang lebih 10 meter.
- b. Pembacaan biaya hanya bisa dicek pada konsumsi daya non-subsidi pada daya sebesar 900 VA sampai 3300 VA.
- c. Alat ini memiliki keterbatasan ketika membaca arus dibawah 0.1 Ampere.

1.3 Tujuan

Adapun tujuan pembuatan proyek akhir ini adalah sebagai berikut:

1. Pengguna dapat mengontrol penggunaan dua buah Stopkontak melalui Android.
2. Pengguna dapat mengecek penggunaan beban listrik yang telah digunakan melalui Android.

BAB II

DASAR TEORI

2.1 Deskripsi Umum Energi Listrik dan Daya

Energi merupakan kemampuan untuk melakukan kerja, energi merupakan kerja tersimpan. Pengertian ini tidaklah jauh beda dengan ilmu fisika yaitu sebagai kemampuan melakukan usaha .[11]

Energi listrik adalah energi yang berkaitan dengan akumulasi arus elektron, dinyatakan dalam *watt*-jam atau kilo *watt*-jam. Energi listrik tersebut diukur menggunakan *watthour* meter. Berupa tipe meter yang dikenal digunakan dirumah untuk menentukan rekening listrik bulanan. Hubungan dari *watthourmeter* sama dengan *wattmeter*, elemen arus dihubungkan secara seri dan elemen potensial dihubungkan paralel dengan rangkaian. Tetapi *watthourmeter* mempunyai elemen pemutar yang kepesatan berbanding lurus dengan daya yang digunakan pada saat itu. Maka jumlah perputaran dalam periode waktu tertentu berbanding lurus sedangkan energi yang digunakan pada saat itu. Elemen pemutar menggerakkan deretan roda gigi yang mencatat *kilowatthour* yang dipakai pada sederetan piringan angka. Maka banyaknya energi, dalam *kilowatt* yang digunakan selama periode tertentu adalah pembacaan piringan pada akhir periode dikurangi dengan pembacaan piringan pada awal periode. [3]

Sesuai perkembangan jaman, kilowatthourmeter sudah banyak dikembangkan misalnya kilowatthourmeter digital. Untuk kilowatthourmeter jenis digital adalah kilowatthourmeter yang menggunakan rangkaian elektronik sebagai penghitungnya. Prosesor sinyal digital digunakan untuk menghitung daya dengan pemberian parameter seperti tegangan dan arus, juga terdapat tegangan referensi pada prosesor sinyal digital tersebut. [14]

Temy Nusa dalam jurnal Teknik Elektro dan Komputer Juli 2018, Universitas Sam Ratulangi, Manado yang berjudul yang berjudul Sistem Monitoring Konsumsi Energi Listrik secara *Real-Time* Berbasis Mikrokontroler mengatakan bahwa daya listrik didefinisikan sebagai laju hantaran energi listrik dalam rangkaian listrik. Daya listrik dibagi menjadi tiga, yaitu daya aktif, daya reaktif dan daya semu. Daya dengan satuan *Joule*/detik atau watt disebut sebagai daya aktif. Simbolnya adalah P. Daya aktif adalah daya sebenarnya yang dihamburkan atau dipakai oleh beban. Daya aktif dihitung dengan persamaan : [2]

$$P = V \cdot I \cdot \cos \Phi \quad \dots\dots\dots(2-1)$$

Keterangan = V = tegangan, *volt*
 I = arus, *ampere*
 Φ = sudut phi, derajat

Daya reaktif satuannya adalah VAR (*Voltampere – reactive*). Daya reaktif (Q) ini merupakan jumlah daya yang diperlukan untuk pembentukan medan magnet, daya reaktif juga dipahami sebagai daya yang tidak dihamburkan olehan atau dengan kata lain merupakan daya yang diserap namun dikembalikan ke sumbernya. Daya reaktif dapat dihitung dengan persamaan : [2]

$$P = V \cdot I \cdot \sin \Phi \quad \dots\dots\dots(2-2)$$

Keterangan = V = tegangan, *volt*
 I = arus, *ampere*
 Φ = sudut phi, derajat

Daya tampak merupakan hasil penjumlahan trigonometri daya aktif dan reaktif yang disimbolkan dengan S. Dengan satuannya adalah VA (*Voltampere*). Daya tampak dapat dihitung menggunakan persamaan : [2]. Halaman 5

$$S = V \cdot I \quad \dots\dots\dots(2-3)$$

Keterangan = V = tegangan, *volt*

I = arus, *ampere*

Daya adalah ukuran kecepatan kerja dilakukan atau kecepatan energi dikeluarkan, sehingga energi yang digunakan oleh alat listrik adalah laju penggunaan daya dikalikan dengan waktu selama alat tersebut digunakan. Bila diukur satu jam pemakaian, maka dapat dihitung dengan menggunakan persamaan : [3]

$$\text{Daya (per-jam)} = \frac{V \cdot I}{1000} \quad \dots\dots\dots(2-4)$$

Setelah melakukan wawancara dengan salah satu karyawan Perusahaan Listrik Negara, penulis mendapatkan info mengenai biaya pemakaian listrik per-KWh yaitu 1450 per-Kwh dengan konsumsi daya non-subsidi dari 900 VA sampai 3300 VA. Adapun biaya perhitungan listrik pada setiap beban yang digunakan dalam satuan detik sebagai berikut :

$$\text{Biaya(per-jam)} = \text{daya (per-jam)} \cdot 1450 \quad \dots\dots\dots(2-5)$$

$$\text{Biaya (per-detik)} = \frac{V \cdot I}{3600} \times 1,45 \quad \dots\dots\dots(2-6)$$

Keterangan = V = tegangan, *volt*

I = arus, *ampere*

Pada masa sekarang bahwa KWh meter analog sudah digantikan dengan KWh meter digital. Adapun sistem pembayaran KWh Meter digital yaitu dengan sistem pembayaran modern dengan membeli sebuah *voucher* elektronik, berisi besaran digital yang berfungsi sebagai pulsa, Secara otomatis sistem ini memutuskan tegangan rumah bila besaran tersebut mencapai nilai 0. KWh Meter digital dikontrol oleh sebuah mikrokontroler dengan tipe AVR90S8515 dan menggunakan sebuah

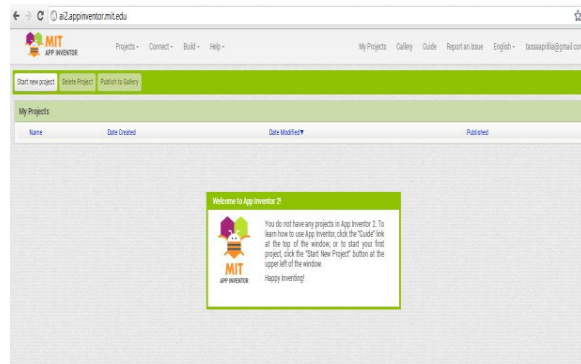
sensor digital tipe ADE7757 yang berfungsi untuk membaca tegangan dan arus serta untuk mengetahui besar energi yang digunakan pada instalasi rumah.

2.2 App Inventor 2

App Inventor 2 (AI2) merupakan IDE generasi kedua dari App Inventor yang dikelola oleh *Massachusetts Institute of Technology* (MIT). AI2 berbasis *cloud* yang diakses menggunakan internet *browser*. Masuk kategori dalam visual *programming*, AI2 menggunakan block puzzle yang disusun untuk menjadi rangkaian kode. [12]

App Inventor 2 adalah *Integrated Development Environment* (IDE) yang ditunjukkan bagi semua kalangan iuntuk mengembangkan aplikasi-aplikasi android tanpa harus berpengalaman dalam dunia pemograman sebelumnya. App Inventor 2 adalah *tools* atau aplikasi yang berfungsi untuk membuat berbagai jenis aplikasi Android dengan mudah dan cepat tanpa *coding*. Aplikasi ini merupakan aplikasi *open source* berbasis web yang awalnya disediakan oleh *Google* dan sekarang dikelola oleh *Massachusetts Institute of Technology* (MIT). [9].

App Inventor 2 adalah *tools* yang berbasis *Designer* dan *Block editor program* untuk mengembangkan aplikasi-aplikasi yang berjalan disistem operasi android. Aplikasi ini merupakan aplikasi yang inovatif untuk memperkenalkan pemograman komputer dan pembuatan aplikasi Android kepada pemula, bahkan bagi mereka yang belum pernah sekalipun mengenal dunia pemograman. Fitur *Block editor program* mampu mentransformasikan pengkodean bahasa pemograman berbasis teks ke dalam bahasa visual data bentuk blok-blok kode *program*. Pada fitur *Designer* anda dapat mendesain tampilan aplikasi dengan cara merancang komponen yang akan digunakan ke dalam *screen*. Selanjutnya setelah anda mendesain tampilan aplikasi tersebut, anda dapat memberikan beberapa blok-blok kode *program* yang di rancang dalam *Block Editor program*. [9]. Berikut adalah tampilan menu utama pada web APP Inventor 2 pada halaman 7.



Gambar 2.1 APP Inventor 2

Tampilan App Inventor 2 untuk pembuatan aplikasi Android yang dilakukan secara online untuk pembuatannya.

2.3 Android

Sulihati dan Andriyani pada jurnal Sains dan Teknologi April 2016, Universitas Tama Jagakarsa, Jakarta Selatan yang berjudul Aplikasi Akademik *Online* Berbasis *Mobile* Android mengatakan bahwa Android adalah aplikasi sistem operasi untuk telepon seluler yang berbasis *Linux*. Android menyediakan platform terbuka bagi para pengembang untuk menciptakan aplikasi mereka sendiri untuk digunakan oleh bermacam piranti bergerak. [13]

Android memang dirancang untuk dipasang pada perangkat-perangkat *mobile touchscreen* (*smartphone* dan *tablet*). Sehingga sistem operasi yang berada di dalam *smartphone* saat ini memang menyesuaikan dari spesifikasi kelas *low-end* hingga *high-end*. Sehingga perkembangan sistem Android memang cukup meningkat tajam. Android merupakan sistem operasi yang terbuka (*open source*) yang mana berarti jika pihak *Google* memperbolehkan dan membebaskan bagi pihak manapun untuk dapat mengembangkan sistem operasi tersebut. [10]

Sistem Android memiliki gudang aplikasi dan game yaitu *Google Play store*, yang mana disini anda bisa mengunduh serta menggunakan aplikasi atau game yang terdapat di *Google Play store* sepuasnya dengan menggunakan perangkat seluler

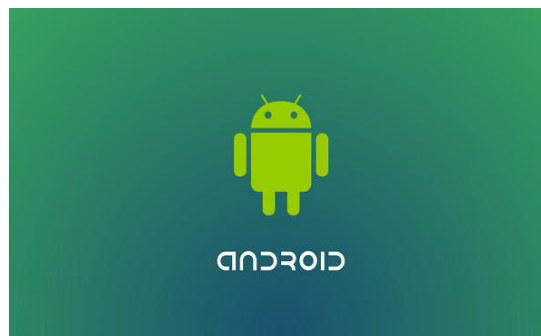
dengan sistem Android. Android menggunakan nama-nama makanan untuk membedakan versi sistem Android yang diluncurkannya. Android menggunakan huruf depan dari nama makanan tersebut sebagai penanda peningkatan versi sistemnya mulai dari *Cupcake* Android 1.5 (C), *Donuts* Android 1.6 (D), *Éclair* Android 2.0-2.1 (E) atau *Marshmallow* Android 6.0 (M). [10].

Sistem android memiliki kelebihan[13] berikut ini :

1. *Switching* dan *multitasking* yang lebih baik Android sangat mendukung multitasking aplikasi, kini hal tersebut kembali ditingkatkan.
2. Kapasitas yang lebih baik untuk beragam widget Kapabilitas terhadap beragam *widjet* dijanjikan bakal makin memanjakan para penggunanya.
3. Peningkatan kemampuan *copy-paste* Beberapa seri Android terdahulu memang sudah bisa melakukan *copy-paste*, namun beberapa pengguna masalah pemilihan teks yang agak sulit.

Selain memiliki kelebihan tersebut, sistem android memiliki beberapa kekurangan[13] berikut ini:

1. Koneksi internet yang terus menerus. Kebanyakan ponsel Android memerlukan koneksi internet yang simultan atau terus menerus aktif, itu artinya anda harus siap berlangganan paket GPRS
2. Iklan. Aplikasi di Ponsel Android memang bisa didapatkan dengan mudah dan gratis, namun konsekuensinya di setiap Aplikasi tersebut, akan selalu ada iklan yang terpampang.



Gambar 2.2 Android [10]

Android, Inc. didirikan di Palo Alto, California, pada Oktober 2003 oleh Andy Rubin (pendiri Danger), Rich Miner (pendiri Wildfire Communications, Inc.), Nick Sears (mantan VP T-Mobile), dan Chris White (kepala desain dan pengembangan antarmuka WebTV). Tujuan awal pengembangan Android adalah untuk mengembangkan sebuah sistem operasi canggih yang diperuntukkan bagi kamera digital, namun kemudian disadari bahwa pasar untuk perangkat tersebut tidak cukup besar, dan pengembangan Android lalu dialihkan bagi pasar telepon pintar untuk menyaingi *Symbian* dan *Windows Mobile*. Google mengakuisisi Android Inc. pada tanggal 17 Agustus 2005, menjadikannya sebagai anak perusahaan yang sepenuhnya dimiliki oleh Google.

Berikut adalah versi Android yang telah dirilis oleh Android Inc.

1. Android 1.0
Dirilis 23 September 2008 dengan menggunakan *device* HTC DREAM.
2. Android 1.1
Pada versi ini Android sudah dilengkapi dengan pembaharuan estis pada aplikasi, jam, alarm, *voice search*, pengirim pesan dan gmail.
3. Android 1.5 (*cupcake*)
Dirilis pada 27 April 2009 dengan menggunakan *kernel* Linux 2.6.27.
4. Android 1.6 (*donut*)
Menampilkan proses pencarian yang lebih baik, penggunaan baterai indikator dan kontrol *applet* VPN, adanya fitur galeri, kamera, *camcorder*, CDMA/EVDO, VPN, *Gestures*, dan *Text-to-speech engine*, kemampuan dial kontak serta teknologi *text to change speech*.
5. Android 2.0/2.1 (*Éclair*)
Pengoptimalan *hardware*, peningkatan Google Maps 3.1.2, perubahan UI dengan *browser* baru dan dukungan HTML5, dukungan *flash* untuk kamera 3,2 MP, *digital Zoom*, dan *Bluetooth 2.1*.

6. Android 2.2 (*froyo*)
Dukungan Adobe Flash 10.1, kecepatan kinerja 2-5 kali lebih cepat, integrasi V8 *Java Script engine* yang mempercepat kemampuan *rendering* pada *browser*, pemasangan aplikasi dalam *SD Card*, kemampuan *WiFi Hotspot* portabel, dan kemampuan *auto update* dalam *Android Market*.
7. Android 2.3 (*gingerbread*)
Peningkatan kemampuan *gaming*, *copy paste*, *User Interface*, dukungan format video VP8 dan WebM, efek audio baru (*reverb*, *equalization*, *headphone virtualization*, dan *bass boost*), dukungan *Near Field Communication* (NFC), dan dukungan jumlah kamera yang lebih dari satu.
8. Android 3.0/3.1 (*honeycomb*)
Android *Honeycomb* dirancang khusus untuk tablet. *Honeycomb* juga mendukung multi prosesor dan juga akselerasi perangkat keras (*hardware*) untuk grafis.
9. Android 4.0 (*ice cream sandwich*)
Menambahkan fitur baru membuka kunci dengan pengenalan wajah, pemantauan penggunaan data, kontak jaringan sosial, perangkat tambahan fotografi, mencari email secara *offline*, dan berbagi informasi dengan menggunakan NFC.
10. Android 4.1 (*Jelly Bean*)
Dirilis tahun 2012, Android *Jelly Bean* ini diperuntukkan untuk komputer tablet dan memungkinkan untuk digunakan pada sistem operasi PC atau Komputer.
11. Android 4.2 (*Jelly Bean API level 17*)
Dirilis pada 13 november 2012, versi ini melengkapi kekurangan maupun *bugs* yang sering terjadi pada JB 4.1, seperti perbaikan *bug* pada aplikasi *people*, penambahan tampilan nirkabel (*miracast*), perbaikan aksesibilitas, VPN yang selalu terhubung dan lain – lain.

12. Android 4.4 (*KitKat*)

Dirlis pada 31 Oktober 2013, versi ini memiliki antar muka terbaru dengan status bar dan navigasi transparan pada layar depan, *webviews* berbasis *Chromium*, mendukung media komunikasi Inframerah yang memungkinkan *devices* bisa menjadi *remote* untuk *smart TV*.

13. Android 5.0 (*Lollipop*)

Pada versi terbaru ini google membuat sebuah desain antar muka terbaru yang dinamakan "*Material Design*", serta mendukung 64bit *ART compiler*, dan menambahkan sistem keamanan yang bernama *factory reset protection* yang berfungsi ketika *smartphone* hilang, ia tidak bisa di-*reset* ulang tanpa memasukkan id dan *password* akun google

14. Android 6.0 *Marshmallow*

Kehadiran Android 6.0 *Marshmallow* untuk perangkat pintar semakin memperkuat apa yang telah dilakukan oleh Google selama ini melalui Android *Lollipop*. Melalui sistem operasi terbaru ini, Google membawa fitur-fitur canggih yang sangat berguna seperti modus *Doze* yang membuat baterai lebih awet, dukungan sensor pindai jari, dukungan *USB Type-C*, hingga percobaan *multi-window*. Yang tidak kalah pentingnya juga adalah kemampuan untuk mengadopsi penyimpanan eksternal layaknya internal.

15. Android 7.0 *Nougat*

Android 7.0 *Nougat* merupakan versi terbaru ketika artikel ini diturunkan. Pada sistem operasi ini Google masih lanjut untuk memberikan peningkatan terhadap pengalaman penggunaan agar lebih mudah. Fitur dukungan *multi-window* menjadi perhatian utama dari Android *Nougat* sehingga pengguna dapat menggunakan beberapa aplikasi secara bersamaan. Selain itu juga ada fitur yang membuat pembaruan sistem operasi lebih mulus dan tidak terasa saat perangkat sedang tidak digunakan.

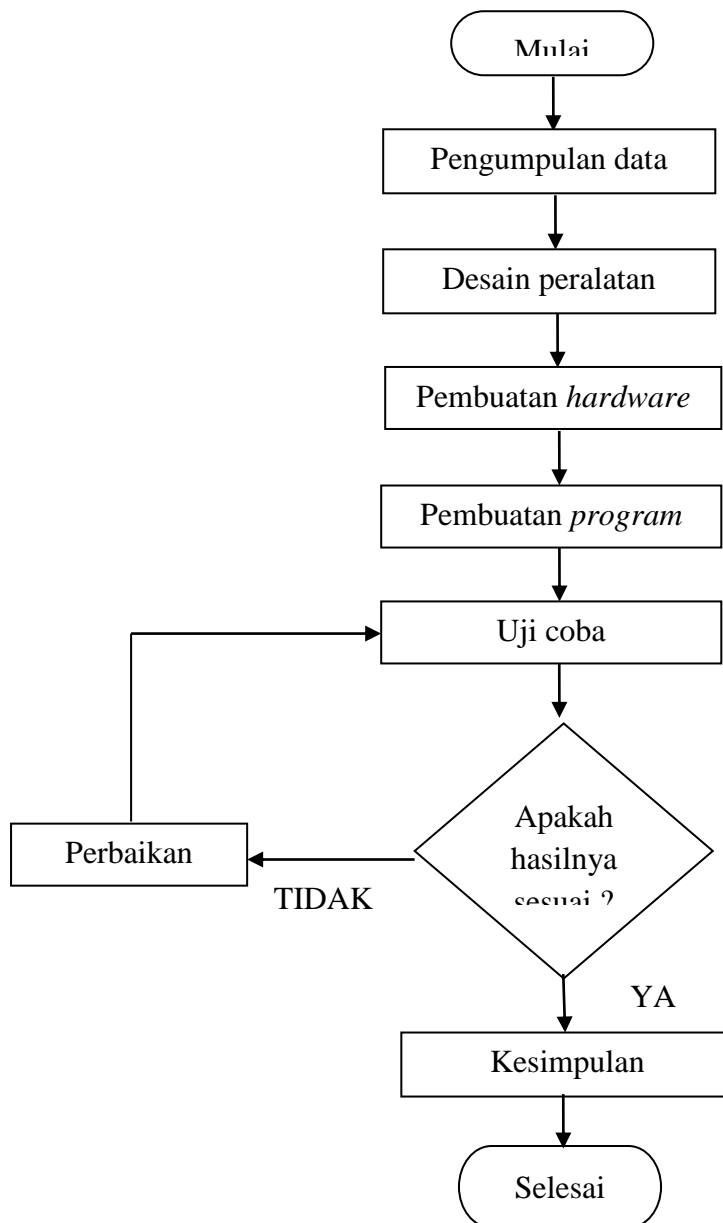
16. Android 8.0 *Oreo*

Android *Oreo* saat ini adalah versi android paling baru. Dirilis bertepatan dengan gerhana matahari di Amerika Serikat. Versi ini membawa beberapa pembaharuan seperti fitur *Notification Dots*, *Picture in Picture*, fitur *Autofill*, emoji yang baru dan masih banyak lagi. Ketika artikel ini diupdate sekarang *Oreo* sudah merilis versi 8.1, versi 8.1 ini tidak terlalu banyak membawa perubahan dalam sisi tampilan tapi dari sisi performa seperti *Project Treble* yang diharapkan nantinya semua perangkat Android lebih cepat mendapat pembaharuan.

BAB III METODE PENELITIAN

3.1 Flow Chart Pembuatan Alat

Berikut ini adalah *flow chart* pembuatan alat yang ditunjukkan pada gambar 3.1.



Gambar 3.1 *Flow Chart* Metode Penelitian

3.2 Pengumpulan Data

Pengumpulan data dilakukan untuk mencari data-data yang akan mendukung proses penelitian. Proses yang dilakukan pada tahap ini adalah dengan mencari berbagai referensi baik dari *searching* di internet, buku-buku elektronika dan juga konsultasi dengan dosen pembimbing. Adapun data-data yang dikumpulkan adalah data-data dari Modul sensor PZEM-004T (tegangan, arus, dan tegangan), *Real Time Clock DS3231*, *relay 2 channel*, dan *Bluetooth HC-05*.

3.3 Desain Peralatan

Untuk menghasilkan alat yang sesuai dengan spesifikasi yang diinginkan dalam pembuatan proyek akhir ini, maka dibutuhkan penganalisaan dan perencanaan peralatan yang akan dibuat tersebut meliputi:

1. Perencanaan *software* yang akan digunakan :

Analisa *software* yang akan digunakan merupakan tahap awal dalam pengerjaan Stopkontak Pintar Berbasis Android. Hasil analisa ini akan menentukan apakah *software* yang digunakan telah memenuhi *standard* atau tidak.

2. Perencanaan biaya pembuatan alat :

Perencanaan biaya pembuatan alat dilakukan berdasarkan kebutuhan yang dapat diketahui berdasarkan data yang didapat dari hasil *survey* informasi tentang alat, konsultasi dengan dosen, penentuan peralatan yang akan digunakan, harga barang atau komponen dan biaya perakitan.

3. Perencanaan konstruksi peralatan :

Perencanaan konstruksi peralatan (*hardware*) bertujuan untuk membuat suatu gambaran dan rancangan mengenai peralatan yang akan dibuat. Hasil dari rancangan inilah yang selanjutnya akan terus dikembangkan sampai menjadi suatu alat dengan rancangan yang lebih efisien baik dalam hal pengkonstruksian peralatan maupun dari keefisienan penggunaan komponen.

Setelah melewati tahap-tahap analisa dan perencanaan ini, selanjutnya adalah merealisasikan hasil perencanaan tersebut yaitu membeli dan melengkapi alat dan komponen yang diperlukan dalam pembuatan proyek akhir ini. Peralatan yang dibeli meliputi peralatan untuk *hardware*.

3.4 Pembuatan *Hardware*

Setelah desain peralatan selesai maka tahap awal pembuatan *hardware* yaitu:

1. Pembuatan konstruksi *box* stopkontak
2. Pemasangan komponen-komponen yang digunakan

Perancangan *hardware* Stopkontak adalah proses pembuatan desain Stopkontak. Pembuatan desain ini mencakup konstruksi Stopkontak yang akan dibuat. Proses pembuatan desain dibuat secara bertahap dan dimulai dari perancangan masing-masing *part* atau bagian dari Stopkontak. Ukuran atau dimensi dari Stopkontak yang akan dibuat yaitu 18,5 cm x 11,5 cm x 6 cm.

3.5 Pembuatan Program

Pembuatan program menggunakan *software* Arduino dan Program aplikasi Android yang dilakukan secara online.

3.5.1 Pembuatan Program Arduino UNO R3

Perencanaan program pada Arduino UNO R3 merupakan langkah awal dalam mendesain atau merancang bagaimana *system* pengontrolan yang akan dibuat. Pembuatan program Arduino UNO R3 ini dibuat untuk melihat alur pemograman pada Stopkontak. Pembuatan program yang dilakukan meliputi inisial variabel, pembacaan nilai pada sensor-sensor yang digunakan, pembuatan pemograman mengontrol Stopkontak, pembuatan pemograman pengiriman dan penerimaan data lewat *Bluetooth*, pembuatan logika pemograman, pembuatan pemograman menghitung data yang didapatkan, pembuatan pemograman penyimpanan data, pembuatan pemograman waktu, dan pembuatan pemograman secara keseluruhan.

3.5.2 Pembuatan Aplikasi Android (Stopkontak)

Setelah mendapatkan *hardware* Stopkontak serta *hardware* elektrik yang tepat, maka selanjutnya adalah pembuatan *software* yang akan diterapkan pada Stopkontak seperti pada aplikasi yang akan digunakan, pemrograman mikrokontroler Arduino UNO R3 dan penerapan sensor serta metode pengontrolan lain yang akan digunakan agar Stopkontak nantinya dapat berfungsi sesuai dengan target yang diinginkan. Pada aplikasi yang akan dibuat yaitu untuk meminta data yang akan dikirim ke *chip* Arduino UNO R3 menggunakan komunikasi *Bluetooth* yang terhubung antara *Bluetooth* Android dengan *Bluetooth* HC-05 pada Arduino UNO R3. Perancangan aplikasi yang akan dibuat antara lain :

- a. Aplikasi *button* untuk mengontrol Stopkontak
- b. Aplikasi dapat menampilkan penggunaan daya dan biaya yang akan ditampilkan pada label di aplikasi.

3.6 Uji Coba

Uji coba penggunaan alat akan dilakukan secara langsung dengan mengacu pada ketentuan-ketentuan dan rancangan awal yang telah dilakukan sebelumnya. Pengujian ini dilakukan untuk melihat fungsi dari peralatan, keefektifan penggunaan, serta melakukan analisa terhadap kehandalan alat yang digunakan. Dari hasil uji coba ini lah dapat diketahui apakah proses kerja dan fungsi alat telah sesuai dengan instruksi yang telah dirancang. Jika sudah sesuai berarti proses pembuatan Stopkontak Pintar Berbasis Android telah selesai. Apabila hasil yang didapatkan tidak sesuai dengan yang diharapkan maka dilakukan perbaikan alat sampai didapatkan hasil yang diinginkan.

3.7 Perbaikan

Tahap ini dilakukan apabila hasil yang diinginkan tidak sesuai dengan yang diharapkan Langkah awal yang dilakukanya itu analisa kerusakan lalu diteruskan dengan melakukan perbaikan pada kerusakan yang terjadi.

3.8 Kesimpulan

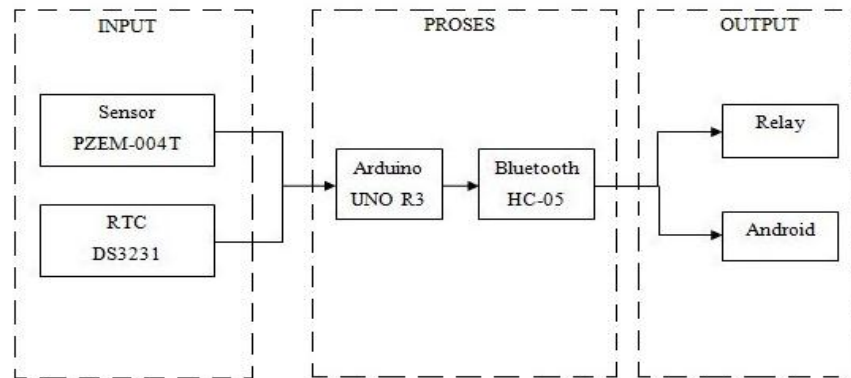
Setelah melewati tahap uji coba dan dinyatakan sesuai dengan perencanaan, maka peralatan yang penulis buat akan dianalisa dengan melakukan pengamatan terhadap kelayakan *system* kerja dari alat tersebut. Hasil pengamatan ini akan menentukan kelebihan dan kekurangan alat yang telah dibuat, serta mencari solusi untuk menutupi kekurangan-kekurangan tersebut hingga menjadi suatu kelebihan. Dari hasil inilah yang kemudian akan dibuat laporan beserta kesimpulannya.

BAB IV PEMBAHASAN

Pada bab ini akan dibahas mengenai proses serta metode yang digunakan dalam pembuatan alat proyek akhir dengan judul “ Stopkontak Pintar Berbasis Android ” yaitu sebagai berikut.

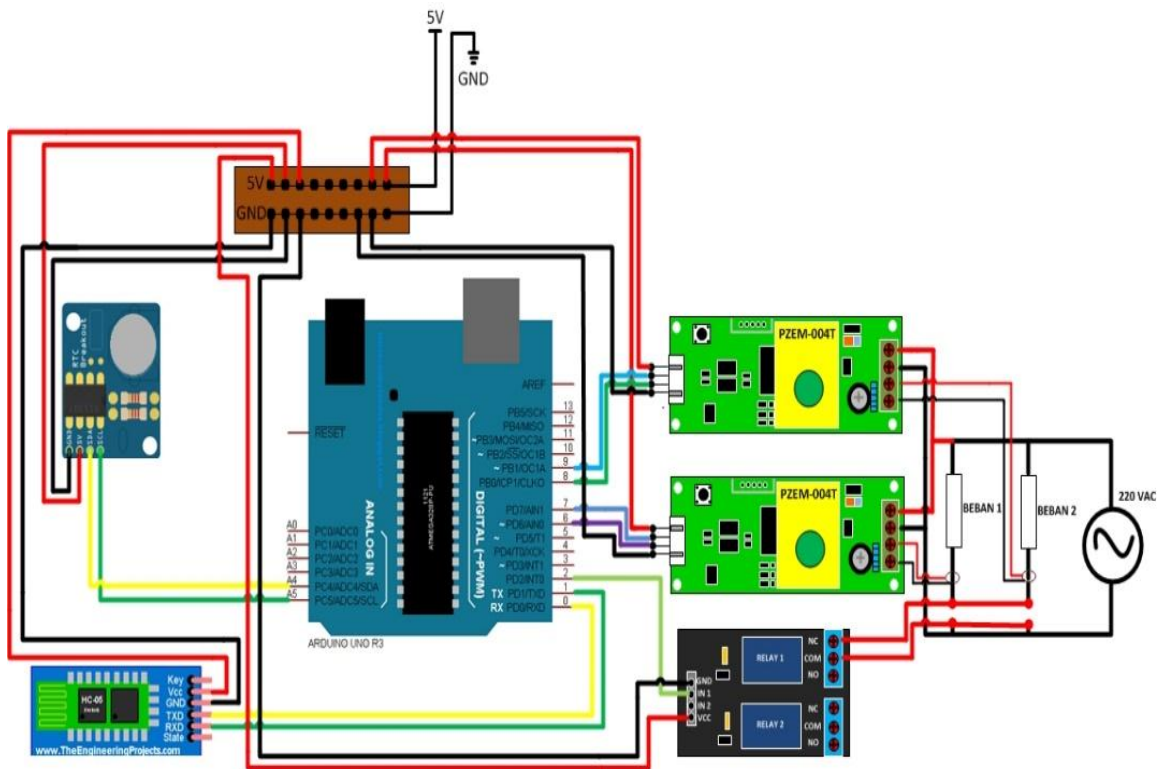
4.1 Diagram Blok

Berikut merupakan diagram blok dari Stopkontak Pintar Berbasis Android.



Gambar 4.1 Diagram Blok Stopkontak Pintar Berbasis Android

Pada gambar 4.1 ini *input* terdiri dari modul sensor PZEM-004T sebagai mendeteksi tegangan, arus, dan daya, input selanjutnya yaitu *real time clock* DS3231 sebagai pemberi hari dan jam. Pada proses terdiri dari Arduino UNO R3 sebagai tempat pengolahan semua data yang didapat dari *input*, sedangkan untuk *Bluetooth* HC-05 sebagai komunikasi untuk mengirim data yang diproses dari Arduino untuk dikirimkan ke Android melalui komunikasi *Bluetooth*. Pada *output* terdapat relay sebagai pengontrolan untuk *ON* atau *OFF*, sedangkan pada Android sebagai aplikasi untuk mengontrol dan mengecek daya dan biaya dari setiap beban yang digunakan.



Gambar 4.2 Skematik Rangkaian Keseluruhan

Pada gambar 4.2 ini proses dan sistem kerjanya sama dengan gambar pada 4.1 yang memiliki Arduino UNO R3 sebagai tempat pengolahan data, *real time clock* DS3231 sebagai pemberi hari dan jam, *Bluetooth* HC-05 ini sebagai komunikasi untuk mengirim data dari Arduino UNO R3 ke Android. Dua buah modul sensor PZEM-004T sebagai pendeteksi tegangan, arus, dan daya. Modul *relay 2 channel* sebagai pengontrolan untuk *ON* atau *OFF*, terdapat dua buah beban yang positif tersebut dari sumber 220 volt dan negatif dari beban diambil dari *relay*. Sumber dari semua komponen elektrikal didapatkan dari adaptor 5 volt.

4.2 Modul Sensor PZEM-004T

PZEM-004T ini digunakan sebagai sensor yang berguna untuk mengecek tegangan, arus, dan daya yang didapat dari setiap beban yang digunakan. Modul ini sudah dilengkapi sensor tegangan dan sensor arus (CT) yang sudah terintegrasi. Dalam alat ini khusus untuk penggunaan dalam ruangan (*indoor*) dan beban yang terpasang tidak diperbolehkan melebihi daya yang sudah ditetapkan.

Modul PZEM-004T dibundel dengan kumparan trafo arus diameter 33mm. Bagian utama dari modul PZEM-004T adalah chip SD3004 dari SDIC Microelectronics Co., Ltd. Modul ini dilengkapi dengan antarmuka komunikasi data serial TTL melalui *port serial* dapat dibaca dan mengatur parameter yang relevan.[4]

Berikut adalah *list* program pengujian.

```
float v1,v2,i1,i2,daya1,daya2,p,p2;
```

```
void Val()
```

```
{
```

```
  v1 = pzem.voltage(ip);
```

```
  if (v1 < 0.0) v1 = 0.0;
```

```
  Serial.print(v1);Serial.print("V; ");
```

Pembacaan Nilai
Teangan (V1)

```
  i1 = pzem.current(ip);
```

```
  if(i1 >= 0.0){ Serial.print(i1);Serial.print("A; "); }
```

Pem-
bacaan
Nilai
Arus
(A1)

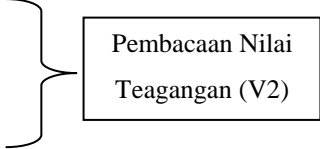
```
  daya1= v1*i1;
```

```
  Serial.println((String)"daya 1 : " + daya1);
```

```
}
```

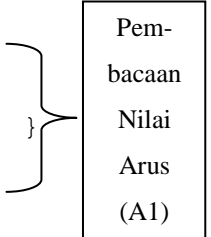
Pembacaan
Nilai Daya
(VA1)


```
void Va2 ()
{
  v2 = pzem1.voltage(ip);
  if (v2 < 0.0) v2 = 0.0;
  Serial.print(v2);Serial.print("V2; ");
```



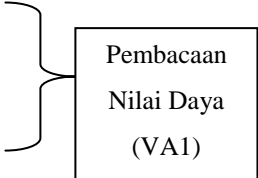
Pembacaan Nilai
Teangan (V2)

```
  i2 = pzem1.current(ip);
  if(i2 >= 0.0){ Serial.print(i2);Serial.print("A2; "); }
```



Pembacaan
Nilai
Arus
(A1)

```
  daya2= v2*i2;
  Serial.println((String)"daya 2 : " + daya2);
}
```



Pembacaan
Nilai Daya
(VA1)

```
void setup() {
  Serial.begin(9600);
  pzem.setAddress(ip);
  pzem1.setAddress(ip);
}
```

```
void loop() {
  Va1();
  Va2();
}
```



Pemanggilan
fungsi untuk
pembacaan
nilai

Berikut adalah tabel pengujian tegangan, arus dan daya pada modul sensor PZEM-004T.

Tabel 4.1 Hasil pengujian tegangan pada modul sensor PZEM-004T

No	Beban	Arduino	Pengukuran	Eror
		Tegangan (V)	Tegangan (V)	
1	Solder	230,0-231,1	229,7-230,8	0,13 %
2	Rice cooker (mode cook)	230,0-231,1	229,7-230,8	0,13 %
3	Rice cooker (mode warm)	230,0-231,1	229,7-230,8	0,13 %
4	Lampu 45watt	230,0-231,1	229,7-230,8	0,13 %
5	Rata-rata			0.13 %

Berikut adalah gambar hasil pengukuran tegangan pada modul sensor PZEM-004T.



Gambar 4.3 Hasil pengukuran Tegangan pada modul sensor PZEM-004T

Dari hasil pengujian tegangan dengan modul sensor PZEM-004T ini menggunakan beban solder, *rice cooker* dengan *mode cook*, *rice cooker* dengan *mode warm* dan lampu 45 watt, dari hasil pengujian tersebut didapatkan eror dengan nilai sebesar 0.13 % sama dengan hasil rata-ratanya. Dari hasil pengujian dan analisa dapat disimpulkan bahwa modul sensor PZEM-004T yang digunakan untuk pengecekan tegangan sesuai dengan perancangan yang diinginkan.

Tabel 4.2 Hasil pengujian arus pada modul sensor PZEM-004T

No	Beban	Arduino	Pengukuran	Eror
		Arus (A)	Arus (A)	
1	Solder	0,26	0,267	2,62 %
		0,25	0,259	3,47 %
		0,27	0,271	0,37 %
2	Rice cooker (mode cook)	2,09	2,123	1,55 %
		2,11	2,152	1,95 %
		2,09	2,136	2,15 %
3	Rice cooker (mode warm)	0,21	0,217	3,22 %
		0,21	0,215	2,32 %
		0,22	0,225	2,22 %
4	Lampu 45watt	0,31	0,32	3,12 %
		0,33	0,336	1,78 %
		0,32	0,328	2,43 %
5	Rata-rata			2,267 %

Berikut adalah gambar hasil pengukuran arus dengan beban solder pada modul sensor PZEM-004T.



Gambar 4.4 Hasil pengukuran Arus pada modul sensor PZEM-004T

Dari hasil pengujian arus dengan modul sensor PZEM-004T ini menggunakan beban solder, *rice cooker* dengan *mode cook*, *rice cooker* dengan *mode warm* dan lampu 45 watt, dari hasil pengujian tersebut didapatkan eror terbesar dengan nilai sebesar 3.47 % dan dengan hasil rata-ratanya yaitu 2.267 %. Dari hasil pengujian dan analisa dapat disimpulkan bahwa modul sensor PZEM-

004T yang digunakan untuk pengecekan arus sesuai dengan perancangan yang diinginkan.

Tabel 4.3 Hasil pengujian daya pada modul sensor PZEM-004T

No	Beban	Arduino	Pengukuran	Error
		Daya (VA)	Daya (VA)	
1	Solder	59,94	61.47	2,48 %
		57,64	59.63	3,33 %
		62.24	62.39	0.24 %
2	<i>Rice cooker (mode cook)</i>	481.85	488.82	1,42 %
		486.46	494.5	1,82 %
		481.85	489.9	1,64 %
3	<i>Rice cooker (mode warm)</i>	48.41	49.91	3.00 %
		48.41	49.45	3.02 %
		50.72	51.75	1,99 %
4	Lampu 45watt	45.91	46.42	1.1 %
		48.20	47.84	0.75 %
		48.20	48.07	0.27 %
		Rata-rata		1,755 %

Berikut adalah gambar hasil pengukuran daya dengan beban solder pada modul sensor PZEM-004T.



Gambar 4.5 Hasil pengukuran daya pada modul sensor PZEM-004T

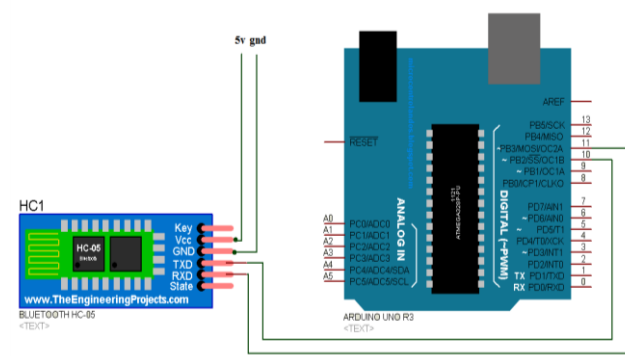
Dari hasil pengujian arus dengan modul sensor PZEM-004T ini menggunakan beban solder, *rice cooker* dengan *mode cook*, *rice cooker* dengan

mode warm dan lampu 45 *watt*, dari hasil pengujian tersebut didapatkan eror terbesar dengan nilai sebesar 3,02 % dan dengan hasil rata-ratanya yaitu 1,755 %. Dari hasil pengujian dan analisa dapat disimpulkan bahwa modul sensor PZEM-004T yang digunakan untuk pengecekan daya sesuai dengan perancangan yang diinginkan.

4.3 Bluetooth HC-05

Modul *Bluetooth* HC-05 adalah module komunikasi nirkabel *via Bluetooth* yang dimana beroperasi pada frekuensi 2.4GHz dengan pilihan dua mode konektivitas. mode 1 berperan sebagai *slave* atau *receiver* data saja, mode 2 berperan sebagai *master* atau dapat bertindak sebagai *transceiver*. Dengan menggunakan modul *Bluetooth* HC-05 ini dapat menggunakan koneksi *Bluetooth* ke Arduino dari HP atau *device Bluetooth* lainnya. Modul ini dapat bekerja pada data *stream* 9600 hingga 115200bps dengan menggunakan tegangan 3.3V hingga 6V. Pengaplikasian komponen ini sangat cocok pada project elektronika dengan komunikasi nirkabel atau *wireless*. Aplikasi yang dimaksud antara lain aplikasi sistem kendali, monitoring, maupun gabungan keduanya. [5]

Bluetooth digunakan untuk mengirim serta menerima data instruksi yang dikirimkan oleh aplikasi Stopkontak pada Android. Dengan menggunakan modul *Bluetooth* HC-05 ini dapat menggunakan koneksi *Bluetooth* ke Arduino dari HP atau *device Bluetooth* lainnya. Berikut adalah gambar skematik pemasangan pin rangkaian *Bluetooth* HC-05 pada Arduino UNO R3 yang dibuat menggunakan *software* ISIS Proteus.



Gambar 4.6 Skematik Pemasangan Pin pada *Bluetooth* HC-05

4.3.1 Pengujian *Bluetooth* HC-05

Langkah awal yang dilakukan dalam pengujian modul *Bluetooth* HC-05 yang digunakan yaitu harus disetting terlebih dahulu menggunakan perintah *AT Command* lewat *Serial Monitor* menggunakan Arduino UNO R3. Langkah yang dilakukan untuk masuk ke mode *AT Command* yaitu hubungkan pin *Bluetooth* seperti pada koneksi yang sudah dibuat, hubungkan pin *key* VCC 5V. Tekan tombol yang ada pada *Bluetooth*, kemudian pasang catu daya tunggu selama 3 detik hingga LED pada *Bluetooth* HC-05 berkedip pelan. Selanjutnya masuk ke *Serial Monitor* pada Arduino, perintah yang ditulis yaitu :

- a. *AT+NAME* : Untuk memberikan nama *Bluetooth*.
- b. *AT+VERSION?* : Untuk mengetahui versi *Bluetooth*.
- c. *AT+UART=9600,0* : Untuk mengatur *baudrate Bluetooth* ke 9600.
- d. *AT+ROLE=0* : Untuk mengeset *Bluetooth* sebagai *SLAVE*.
- e. *AT+ORGL* : Untuk kembali ke pengaturan pabrik.
- f. *AT+CMODE=0* : Untuk mengatur jumlah *device* yang terhubung ke *Bluetooth*.
- g. *AT+PSWD=1234* : Untuk mengatur *password*.

Berikut adalah *list* program pengujian *Bluetooth* HC-05.

```
Void setup()  
{  
    Serial.begin(9600);Bluetooth.begin(38400);  
}  
void loop()  
{  
    if (bluetooth.available()){  
        Serial.write(bluetooth.read());}  
    if (Serial.available()){  
        bluetooth.write(Serial.read());}  
}
```



Mengecek
ketersediaan
Bluetooth

Adapun hasil pengujian yang dilakukan pada *Bluetooth* HC-05 untuk melihat terdeteksi atau tidaknya data yang dikirimkan aplikasi Stopkontak dapat dilihat pada tabel 4.4 berikut.

Tabel 4.4 Hasil Pengujian Bluetooth HC-05

No	Jarak	Status	
		Tanpa Penghalang	Dengan Penghalang
1	1 Meter	Ya	Ya
2	2 Meter	Ya	Ya
3	3 Meter	Ya	Ya
4	4 Meter	Ya	Ya
5	5 Meter	Ya	Ya
6	6 Meter	Ya	Ya
7	7 Meter	Ya	Ya
8	8 Meter	Ya	Ya
9	9 Meter	Ya	Ya
10	10 Meter	Ya	Ya

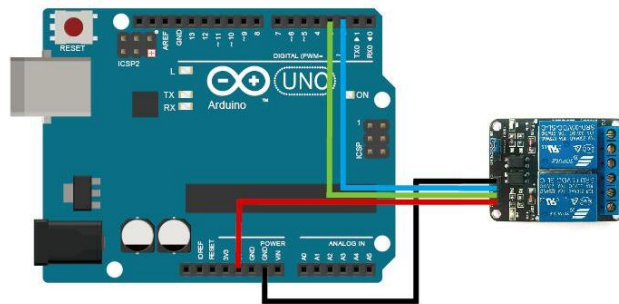
Dari pengujian yang dilakukan dapat dianalisa bahwa *Bluetooth* HC-05 yang digunakan dapat menerima data yang dikirim oleh aplikasi pada Android hingga jarak 10 meter dengan baik pada ruangan tanpa penghalang dan dengan penghalang, dan jika melebihi dari range tersebut maka kualitas konektivitas akan semakin kurang maksimal. Android dapat terhubung dan mengirim data dengan baik pada modul *Bluetooth* HC-05 yang digunakan pada Stopkontak. Dari hasil pengujian dan analisa dapat disimpulkan bahwa *Bluetooth* yang digunakan sesuai dengan perancangan yang diinginkan.

4.4 Modul *Relay 2 Channel*

Modul *relay 2 channel* ini adalah salah satu bagian utama untuk mengontrol *ON* atau *OFF* pada peralatan listrik berdaya besar untuk *system* Stopkontak yang akan dibuat.

Relay 2 channel ini memerlukan arus sebesar sekurang-kurangnya 15-20mA untuk mengontrol masing-masing *channel*. Disertai dengan relay *high-current* sehingga dapat menghubungkan perangkat dengan tegangan AC250V 10A. *Relay* ini menggunakan tegangan rendah 5V, sehingga langsung

dihubungkan pada *system* mikrokontroler. Pin pengendali pada Relay dapat dihubungkan dengan *port* mikrokontroler mana saja, sehingga membuat pemrogram dapat leluasa menentukan pin mikrokontroler yang digunakan sebagai pengendali. *Driver* bertipe “*active high*” atau kumparan *relay* akan aktif saat pin pengendali diberi logika “1”. [7]



Gambar 4.7 Skematik Pemasangan Pin pada Modul *Relay 2 Channel*

4.4.1 Pengujian Modul *Relay 2 Channel*

Berikut adalah *list* program Arduino untuk pengecekan modul *Relay 2 Channel* menggunakan komunikasi serial.

Berikut adalah *list* program modul *Relay 2 Channel*.

```
void setup() {
  Serial.begin(9600);pinMode(5,OUTPUT);pinMode(4,OUTPUT);
}
void loop() {
  if(Serial.available(>0)
  {
    str = Serial.read();
    if(str=='A')    digitalWrite(5,1);
    else if (str == 'B') digitalWrite(4,1);
    else if (str == 'C') digitalWrite(5,0);
    else if (str == 'D') digitalWrite(4,0);
  }
}
```

Pengecekan *relay* dengan memasukkan karakter di *serial*

Berikut adalah tabel dari hasil pengujian modul *Relay 2 Channel*.

Tabel 4.5 Hasil Pengujian modul *Relay 2 Channel*

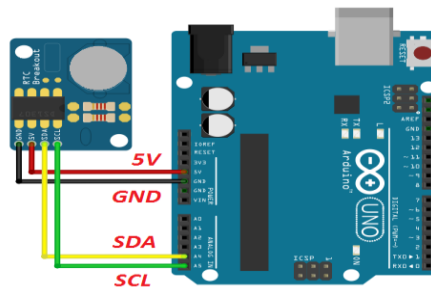
No	<i>Input Serial</i>	Kondisi
1	A	Relay pin 5 (tidak aktif)
2	B	Relay pin 4 (tidak aktif)
3	C	Relay pin 5 (aktif)
4	D	Relay pin 4 (aktif)

Pada hasil pengujian modul *Relay 2 Channel* yaitu terdiri dari *input serial* dan kondisi, di mana pada *input A* dan *input B* pada kondisi *Relay pin 5* dan *Relay pin 4* adalah kondisi tidak aktif, sedangkan pada *input C* dan *input D* pada kondisi *Relay pin 5* dan *Relay pin 4* adalah kondisi aktif. Dapat dianalisa bahwa modul *Relay 2 Channel* yang digunakan berjalan dengan baik dan sesuai dengan perencanaan yang diinginkan.

4.5 Real Time Clock DS3231

Real time clock DS3231 ini memiliki akurasi dan presisi yang sangat tinggi dalam mencacah waktu dengan menggunakan IC RTC DS3231 yaitu salah satu jenis RTC yang dapat digunakan bersama *controller* Arduino. RTC DS3231 menggunakan *battery* 3V dan 6 pin *interface*. Modul ini dilengkapi dengan IC AT24C32 yang memberikan EEPROM tambahan sebesar 4 KB (32.768 bit). [8]

Real Time Clock sebagai salah satu *chip* IC yang berfungsi sebagai menghitung waktu dan tanggal atau hari sebagai pemilihan untuk pengecekan biaya yang digunakan dan data akan dikirim serta ditampilkan pada aplikasi Stopkontak.



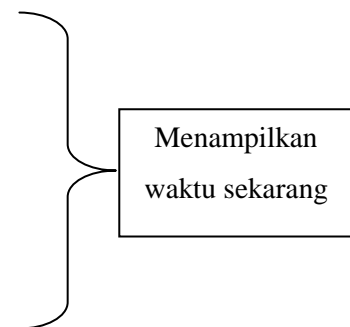
4.8 Skematik Pemasangan Pin pada *Real Time Clock* DS3231

4.5.1 Pengujian *Real Time Clock* DS3231

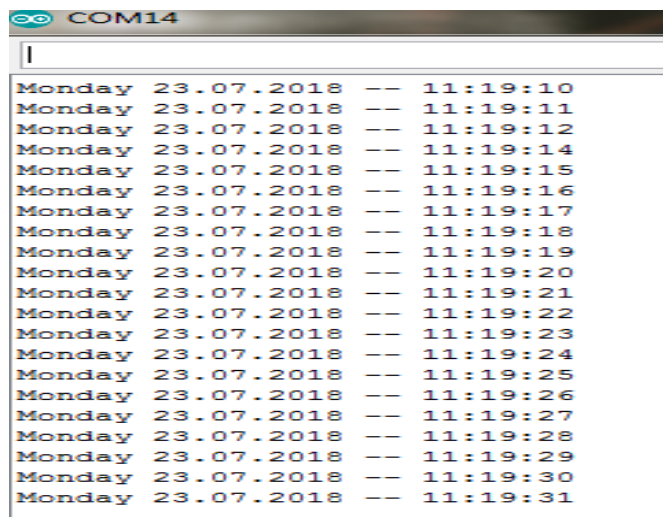
Langkah yang harus dilakukan untuk mengetahui fungsi kerja dari *real time clock* ini dengan menampilkan hari, tanggal dan waktu pada hasil *serial monitor* di Arduino UNO R3.

Berikut adalah *list* program hasil pengujian dari *Real Time Clock*.

```
Void Setup() {  
    Serial.begin(9600);  
}  
  
void loop() {  
    Serial.print(rtc.getDOWStr());  
    Serial.print(" ");  
    Serial.print(rtc.getDateStr());  
    Serial.print(" -- ");  
    Serial.println(rtc.getTimeStr());  
    delay (1000);  
}
```



Berikut adalah hasil pengujian *Real Time Clock* DS3231.



4.9 Hasil Pengujian *Real Time Clock* DS3231

Dari hasil pengujian *Real Time Clock* DS3231 tersebut bahwa hari, tanggal dan waktu yang diinginkan, berjalan dengan baik. Hasil waktu yang ditampilkan

yaitu berubah dalam hitungan detik. Dari hasil pengujian dan analisa dapat disimpulkan bahwa *Real Time Clock DS3231* yang digunakan sesuai dengan perancangan yang diinginkan.

4.6 Aplikasi Android (Stopkontak)

Pembuatan aplikasi Android dilakukan menggunakan *software* App Inventor secara *online* menggunakan PC/Komputer. Pada saat akan menggunakan, maka App Inventor 2 akan meminta data email *user*, dimana yang nantinya *server* app inventor akan menyimpan data yang berupa *project user* yang sudah dikerjakan. Ketika *user* akan mengakses *project* data yang dikerjakan sebelumnya secara otomatis sehingga *user* hanya perlu masuk ke *home* App Inventor 2, kemudian kembali pada *project* atau aplikasi yang telah dibuat.

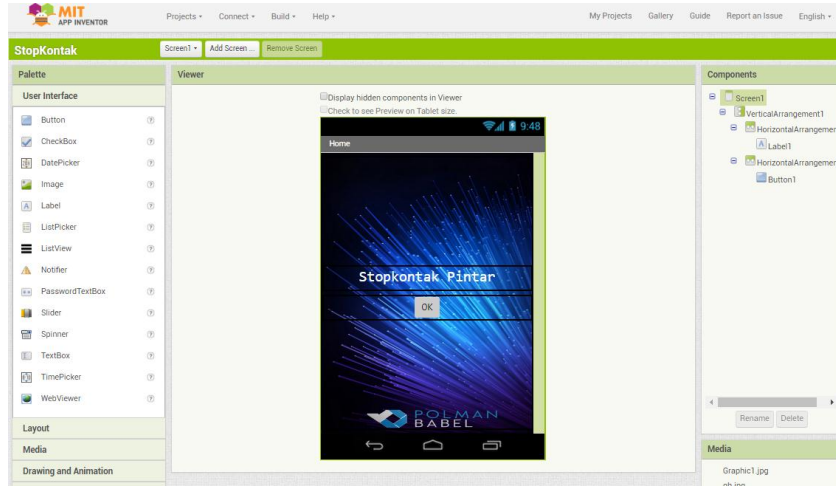
Pada pembuatan aplikasi menggunakan App Inventor 2 dibagi menjadi dua komponen yaitu :

1. *Designer* yaitu bagian untuk membuat tampilan aplikasi beserta komponen apa saja yang akan digunakan.
2. *Block Editor* yaitu bagian yang berisi program yang akan *user* telah buat sesuai dengan kinerja aplikasi yang diinginkan dan untuk *merakit program visual* dari aplikasi tersebut.

Berikut adalah langkah-langkah membuat aplikasi Stopkontak pada Android menggunakan App Inventor.

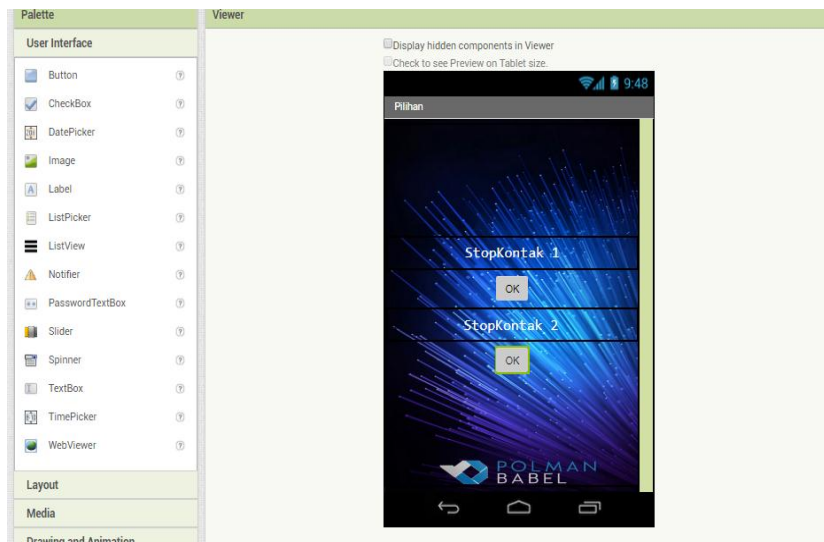
- a. *User login* pada aplikasi App Inventor 2 menggunakan *account Google user* melalui website [http:// ai2.appinventor.mit.edu/](http://ai2.appinventor.mit.edu/)
- b. Pilih Membuat *project* baru dengan cara klik *start new project*, kemudian ketikkan nama *project* dan klik OK.
- c. Kemudian membuat desain *layout* aplikasi Stopkontak beserta komponen-komponen yang akan digunakan pada komponen *Designer*. Berikut adalah gambar *layout* pada *Screen 1* pada gambar 4.10, *Screen 2* pada gambar 4.11, dan *Screen 3* pada gambar 4.12 yang telah dibuat. Halaman 29.

Layout pada Screen 1



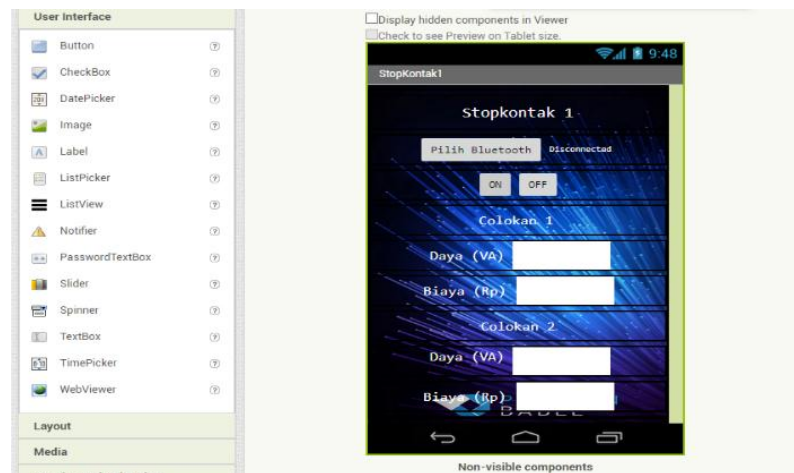
Gambar 4.10 Layout pada Screen 1

Layout pada Screen 2



Gambar 4.11 Layout pada Screen 2

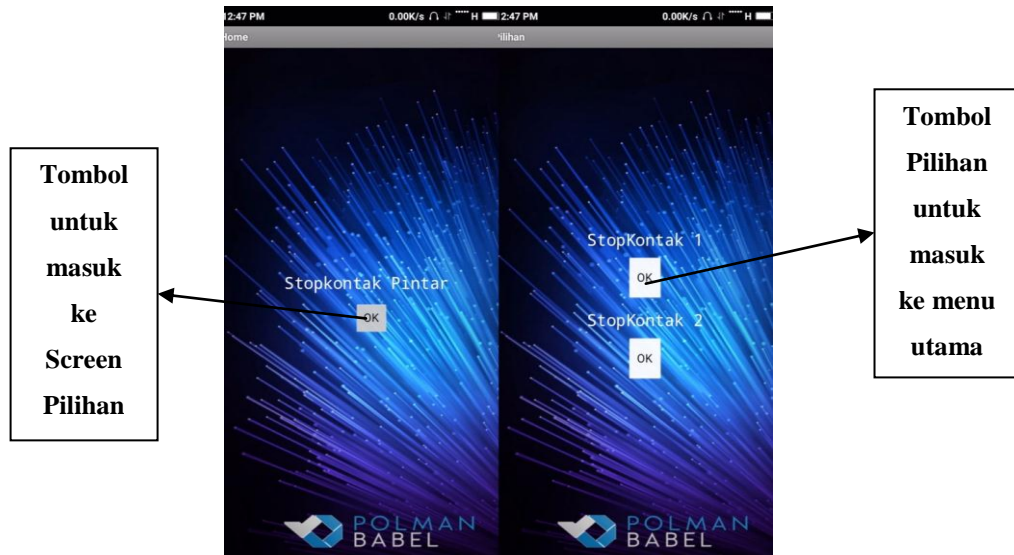
Layout pada Screen 3



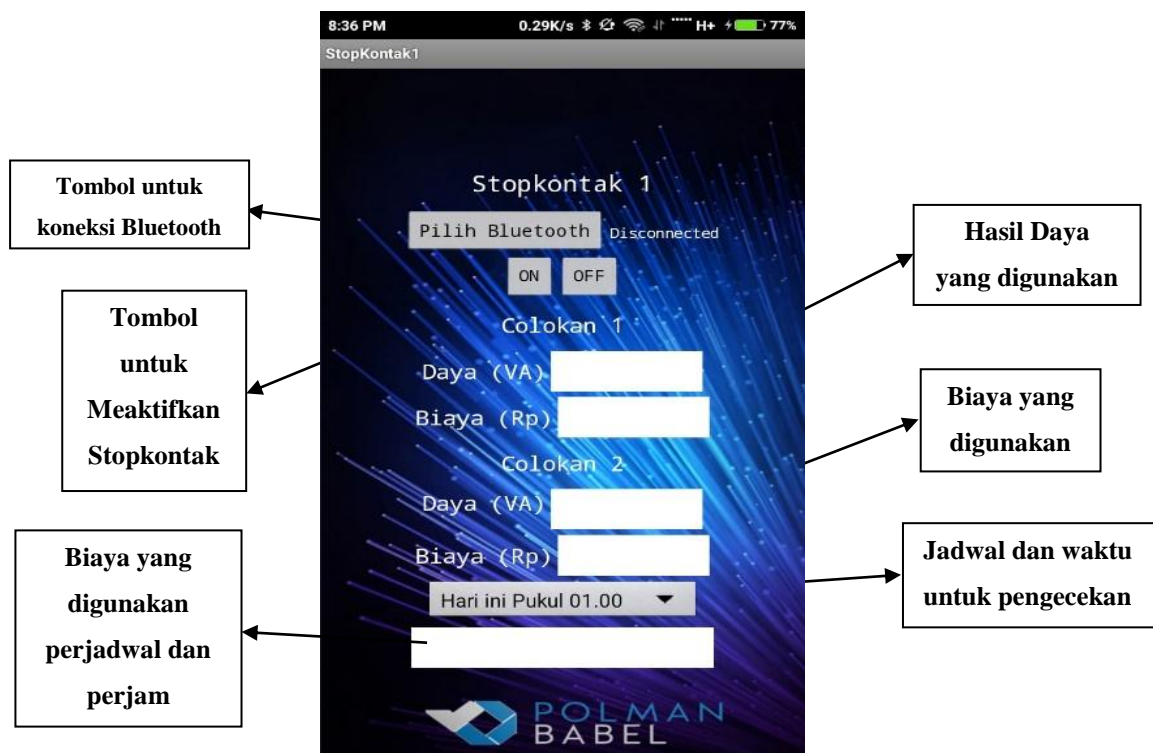
Gambar 4.12 Layout pada Screen 3

- d. Selanjutnya adalah membuat program *visual* pada bagian *Block Editor*. Pada Aplikasi yang dibuat terdapat empat *Screen* yang digunakan. *Screen 1* dimaksudkan sebagai tampilan pembuka pada aplikasi Stopkontak untuk masuk ke pilihan dengan memilih antara Stopkontak 1 atau Stopkontak 2 pada screen 2. Sedangkan pada Screen 3 dan Screen 4 secara umum berisi tentang *Block Editor program* koneksi *Bluetooth*, cara mengontrol Stopkontak dan program *Bluetooth* untuk mengirim dan menerima data. Daftar *Block Editor program* dilampirkan pada lampiran.
- e. Pada *Block Editor program* di *Screen 1* dan *Screen 2*, komponen serta fitur yang digunakan yaitu *button* dan *label*. *Screen 3* dan *Screen 4* komponen serta fitur yang digunakan yaitu *button*, *label*, *list picker*, *Bluetoothclient*, dan *clock*.
- f. Langkah terakhir adalah dengan melakukan *Build* aplikasi yang sudah dibuat dengan cara *klik Build* kemudian pilih App (*save .apk to my computer*) maka aplikasi yang dibuatkan secara otomatis terunduh pada PC/Komputer. Selanjutnya adalah memindahkan aplikasi tersebut ke *Smartphone* Android untuk di instal.

Berikut adalah gambar aplikasi Stopkontak yang telah dibuat pada android.



Gambar 4.13 Layout pada Screen 1 dan Screen 2 di Android



Gambar 4.14 Layout pada Screen 3 di Android

Pada menu utama Stopkontak 1 ini adalah sebagai tempat mengontrol beban yang akan digunakan, setelah itu terdapat label yang bertuliskan colokan 1 dan colokan 2 terdiri dari daya (VA) dan biaya (Rp) adalah sebagai tempat yang menampilkan daya dan biaya yang digunakan oleh pengguna Stopkontak.

4.6.1 Pengujian Aplikasi Android (Stopkontak)

Pengujian aplikasi dilakukan pada Android dengan versi *Marshmallow* 6.0.1. Tujuan Pengujian aplikasi untuk menguji apakah aplikasi yang dibuat dapat berjalan sesuai dengan perencanaan program yang telah dibuat yaitu dengan mengirim perintah ke Arduino UNO R3 dan menerima data dari Arduino UNO R3 melalui komunikasi *Bluetooth*. Adapun langkah-langkah yang dilakukan dalam pengujian atau percobaan aplikasi Android (Stopkontak) yaitu sebagai berikut.

- a. Nyalakan *Bluetooth* pada perangkat Android
- b. Kemudian buka aplikasi Stopkontak, pada *Screen 1* klik tombol OK untuk masuk ke tampilan pilihan pada *Screen 2*.
- c. Pada *Screen 2* akan muncul pilihan Stopkontak 1 dan Stopkontak 2, kemudian pilih salah satu untuk masuk ke tampilan utama pada *Screen 3*.
- d. Pada *Screen 3*, klik tombol pilih *Bluetooth* untuk mengkoneksikan Bluetooth pada aplikasi Stopkontak 1 atau Stopkontak 2, kemudian pilih *Bluetooth* yang sesuai dengan nama modul *Bluetooth*, setelah dipilih maka akan kembali ke *Screen 3* apabila sudah terkoneksi dengan baik pada modul *Bluetooth*.
- e. Pilih tombol *ON* atau *OFF* pada menu Stopkontak 1 atau Stopkontak 2, kemudian ketika memilih tombol *ON* maka beban yang digunakan akan menyala serta pada tampilan daya satuan VA dan biaya satuan rupiah akan menampilkan hasil data yang dikirim melalui Arduino UNO R3. Pada saat memilih tombol *OFF* maka beban yang digunakan akan padam serta daya dan biayanya akan menampilkan hasil data penggunaan yang terakhir
- f. Ketika pengguna mau mengecek biaya yang digunakan terhitung pada penggunaan hari ini, kemarin dan kemarin lusa dalam waktu 24 jam. Maka

pilih hari dan waktu. Setelah dipilih maka data akan dikirim dari Arduino UNO R3 ke aplikasi Stopkontak.

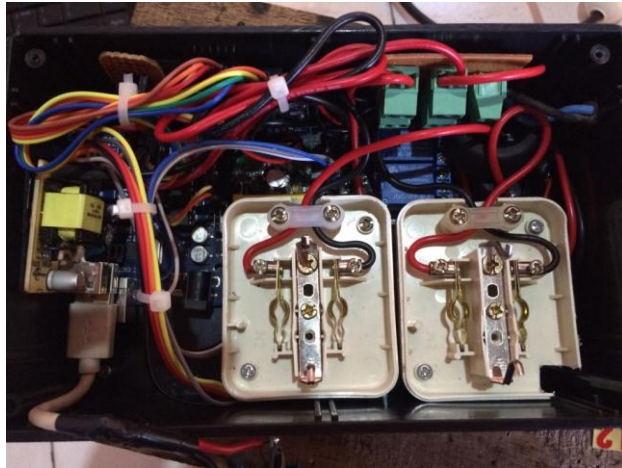
4.7 Program Arduino UNO R3

Pembuatan *program* dilakukan menggunakan *software* Arduino. Pemograman Stopkontak dibagi menjadi beberapa tahapan sesuai dengan perancangan yang telah dibuat yaitu sebagai berikut :

- a. Pemograman *Bluetooth* HC-05 untuk mengirim dan menerima data dari aplikasi Android (Stopkontak).
- b. Pemograman Modul *Relay 2 Channel* untuk mengontrol *ON/OFF* pada Stopkontak.
- c. Pemograman modul sensor PZEM-004T untuk mendeteksi nilai tegangan, arus dan daya yang didapatkan pada setiap beban yang digunakan.
- d. Pemograman perhitungan daya hasil dari data arus dikalikan data tegangan, kemudian biaya yang digunakan pada setiap daya yang digunakan .
- e. Pemograman *Eeprom* Arduino sebagai penyimpanan data daya, biaya serta waktu.
- f. Pemograman *Real Time Clock* untuk menampilkan waktu dan hari.
- g. Pemograman Stopkontak secara keseluruhan.

4.8 Pengujian Stopkontak

Setelah bagian *hardware* dan *software* dari Stopkontak telah dibuat, proses selanjutnya adalah pengujian terhadap *system* yang telah dibuat pada Stopkontak. Pengujian *system* ini dilakukan untuk menguji serta melihat bagaimana hasil yang akan didapatkan setelah melakukan perancangan, pembuatan , dan pengujian pada masing-masing *hardware* maupun *software*. Berikut adalah hasil perakitan komponen Stopkontak yaitu sebagai berikut. Halaman 33.



Gambar 4.15 Hasil Perakitan Komponen pada Stopkontak

4.8.1 Pengujian Keseluruhan Alat dan Aplikasi Stopkontak

Berikut ini adalah tabel 4.6 pengujian keseluruhan alat dan aplikasi.

Tabel 4.6 Hasil Pengujian Keseluruhan Alat dan Aplikasi Stopkontak 28 Juli 2018

No	Beban	Daya (VA)	Biaya (Rp)	Jadwal	Waktu (WIB)
1	Lampu 45 Watt	43.85	61.52	28 Juli 2018	01.00-02.00
2	Lampu 45 Watt	43.61	61.35	28 Juli 2018	02.00-03.00
3	Lampu 45 Watt	43.35	61.27	28 Juli 2018	03.00-04.00
4	Lampu 45 Watt	44.12	62.14	28 Juli 2018	04.00-05.00
5	Lampu 45 Watt	43.78	61.46	28 Juli 2018	05.00-06.00
6	Kipas Angin (mode 3) 50 Watt	48.64	69.88	28 Juli 2018	09.00-10.00
7	Kipas Angin (mode 3) 50 Watt	48.53	69.64	28 Juli 2018	10.00-11.00
8	Kipas Angin (mode 3) 50 Watt	47.89	68.75	28 Juli 2018	11.00-12.00
9	Kipas Angin (mode 3) 50 Watt	47.95	68.83	28 Juli 2018	12.00-13.00
10	Kipas Angin (mode 3) 50 Watt	48.70	69.92	28 Juli 2018	13.00-14.00

Berikut adalah gambar hasil dari pengujian beban lampu 45 watt dan kipas angin (mode 3) 50 watt yang dilakukan dengan aplikasi Stopkontak.



Gambar 4.16 Hasil Pengujian Stopkontak dengan beban Lampu 45 Watt

Dari hasil pengujian dengan beban lampu 45 watt tersebut, pada tabel 4.6 yang diuji pada tanggal 28 Juli 2018 tersebut, tercantum pada Stopkontak 1 yang digunakan dalam kondisi *ON*, pada saat pengujian pada colokan 1 tercantum daya sebesar 43.61 VA dan biaya sebesar 55.30 rupiah yang di cek pada kisaran pukul 02.00-03.00 WIB pada aplikasi Stopkontak. Pada pengecekan jadwal penyimpanan biaya yang digunakan yaitu pada pukul 01.00-02.00 WIB pada gambar 4.16 yaitu sebesar 61.52 rupiah dan pada tabel 4.6. Maka dari itu pengontrolan dan pengecekan beban yang digunakan pada Stopkontak dan aplikasi Stopkontak berjalan dengan baik.



Gambar 4.17 Hasil Pengujian Stopkontak dengan beban Kipas Angin

Dari hasil pengujian dengan beban kipas angin (*mode 3*) 50 watt tersebut, pada tabel 4.6 yang diuji pada tanggal 28 Juli 2018 tersebut, tercantum pada Stopkontak 1 yang digunakan dalam kondisi *ON*, pada saat pengujian pada colokan 1 tercantum daya sebesar 47.89 VA dan biaya sebesar 30.66 rupiah yang di cek pada kisaran pukul 11.00-12.00 WIB pada aplikasi Stopkontak. Pada pengecekan jadwal penyimpanan biaya yang digunakan yaitu pada pukul 09.00-10.00 WIB pada gambar 4.16 yaitu sebesar 69.88 rupiah dan pada tabel 4.6. Maka dari itu pengontrolan dan pengecekan beban yang digunakan pada Stopkontak dan aplikasi Stopkontak berjalan dengan baik.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Dari hasil perancangan alat dan pengujian sistem kerja alat dapat ditarik kesimpulan sebagai berikut :

- a. Dari hasil pengujian sensor PZEM-004T didapat bahwa PZEM-004T mampu membaca nilai arus, tegangan, dan daya dengan *error* kurang dari 3%.
- b. Dari hasil pengujian *Bluetooth* HC-05 didapat bahwa HC-05 mampu menjadi perantara dalam komunikasi data dengan jarak kurang lebih 10 meter.
- c. Dari hasil pengujian modul *Relay 2 Channel* didapat bahwa *relay* yang digunakan tersebut dapat aktif apabila diberi logika *LOW*. Sedangkan jika diberi logika *HIGH* maka *relay* tidak aktif.
- d. Dari hasil pengujian *Real Time Clock* DS3231 didapat bahwa waktu yang ditampilkan sesuai dengan waktu sebenarnya

5.2 Saran

Dengan memperhatikan kelemahan dan kekurangan dari proyek akhir ini maka perlu adanya beberapa saran sebagai berikut:

- a. Sebaiknya alat ini dikembangkan dengan perangkat yang memiliki jangkauan akses yang lebih luas dari perangkat *Bluetooth*.
- b. Perlunya pengembangan yang lebih luas dan variatif sesuai dengan kebutuhan dan dipatenkan agar dipabrik-kan secara massal.

DAFTAR PUSTAKA

- [1]. Hilman HR Jufri (2012). Rancang Bangun Alat Ukur Daya Arus Bolak-Balik Berbasis Mikrokontroler ATmega8535, Universitas Sumatera Utara, Medan. Diakses pada tanggal 25 Juli 2018
- [2]. Temy Nusa (2015). Sistem Monitoring Konsumsi Energi Listrik secara *Real-Time* Berbasis Mikrokontroler, Universitas Sam Ratulangi, Manado. Diakses pada tanggal 25 Juli 2018.
- [3]. Eugene C Lister *Chief Electrical Engineer (Ret.) Stanley Consultants, Inc* (1993). Energi Listrik dan Daya, Mesin dan Rangkaian Listrik edisi Keenam, *The institute of Electrical and Electronik Engineer, Inc. (Life Senior Member)*. Diakses pada tanggal 20 Juli 2018
- [4]. Modul sensor PZEM-004T, <https://innovatorsguru.com/ac-digital-multifunction-meter-using-pzem-004t/>. Diakses pada tanggal 22 Juli 2018
- [5]. Modul *Bluetooth* HC-05, <https://www.nyebarilmu.com/tutorial-arduino-module-bluetooth-hc-05/>. Diakses pada tanggal 15 April 2018
- [6]. Arduino UNO R3, <http://ecadio.com/mengenal-dan-belajar-arduino-uno-r3>. Diakses pada tanggal 07 April 2018
- [7]. Modul *Relay 2 Channel*, <https://www.indo-ware.com/produk-2788-modul-relay-2-indoware.html>. Diakses pada tanggal 07 April 2018
- [8]. *Real Time Clock* DS 3231, <https://www.indo-ware.com/produk-3048-ds3231-at24c32-clock-module.html>. Diakses pada tanggal 12 Juli 2018
- [9]. App Inventor, <http://shopansoft.com/mengenal-aplikasi-app-inventor-2/>. Diakses pada tanggal 16 Juli 2018
- [10]. Sistem Android, <https://www.nesabamedia.com/pengertian-android-beserta-kelebihan-dan-kekurangannya>. Diakses pada tanggal 15 April 2018
- [11]. Ditjeng Marsudi (2006). Operasi Sistem Tenaga Listrik, Edisi Pertama, Graha Ilmu, Yogyakarta. Diakses pada tanggal 12 Agustus 2018

- [12]. Endar Suprih Wihidayat (2017). Pengembangan Aplikasi Android menggunakan *Integrated Development Environment* (Ide) App Inventor 2, Universitas Sebelas Maret, Yogyakarta. Diakses pada tanggal 12 Agustus 2018
- [13]. Sulihati dan Andriyani (2016). Aplikasi Akademik *Online* Berbasis *Mobile Android*, Universitas Tama Jagakarsa, Jakarta Selatan. Diakses pada tanggal 12 Agustus 2018
- [14]. Deonugrah Pratama dan Riski Novriandy (2014). Alat Penghitung Pemakaian Listrik via *SMS*, Politeknik Manufaktur Negeri Bangka Belitung.

LAMPIRAN

LAMPIRAN 1
DAFTAR RIWAYAT HIDUP

DAFTAR RIWAYAT HIDUP

1. Data Pribadi

Nama lengkap : Steven Novaldy Fernando
Tempat dan tanggal lahir : Muntok, 02 November 1997
Alamat rumah : Kp. Tanjung Sawah, Muntok
Bangka Barat
Telp : -
Hp : 085810563273
Email : snovaldy54@gmail.com
Jenis kelamin : Laki-laki
Agama : Islam



2. Riwayat Pendidikan

SD N 6 Muntok	2003-2009
SMP N 1 Muntok	2009-2012
SMK N 1 Muntok	2012-2015
DIII Politeknik Manufaktur Negeri Bangka Belitung	2015-2018

3. Pendidikan Non Formal

PT. Rire Sanjaya Sakti	2017-2018
------------------------	-----------

Sungailiat, 14 Juli 2018



Steven Novaldy Fernando

DAFTAR RIWAYAT HIDUP

1. Data Pribadi

Nama lengkap : K'K Anggrainy
Tempat dan tanggal lahir : Muara Enim, 17 Oktober 1996
Alamat rumah : Jl. Riau, Karya Makmur
Pemali, Bangka
Telp : -
Hp : 082183950270
Email : kkangrainy@gmail.com
Jenis kelamin : Perempuan
Agama : Islam



2. Riwayat Pendidikan

SD N 1 Sungailiat	2002-2008
SMP Setia Budi Sungailiat	2008-2011
SMA Setia Budi Sungailiat	2011-2014
DIII Politeknik Manufaktur Negeri Bangka Belitung	2015-2018

3. Pengalaman Kerja

PT. FESTO	2017-2018
-----------	-----------

Sungailiat, 14 Juli 2018

K'K Anggrainy

LAMPIRAN 2

Program Arduino

```
#include <SoftwareSerial.h>
```

```
#include <EEPROMex.h>
```

```
#include "Arduino.h"
```

```
#include <DS3231.h>
```

```
#include <PZEM004T.h>
```

```
char hasil;
```

```
PZEM004T pzem(7,6);// (RX,TX) connect to TX,RX of PZEM
```

```
PZEM004T pzem1(9,8);
```

```
IPAddress ip(192,168,1,1);
```

```
DS3231 rtc(SDA, SCL);
```

```
int s,m,h,d;
```

```
byte addr=5;
```

```
float
```

```
j0,j1,j2,j3,j4,j5,j6,j7,j8,j9,j10,j11,j12,j13,j14,j15,j16,j17,j18,j19,j20,j21,j22,j23;
```

```
float
```

```
k0,k1,k2,k3,k4,k5,k6,k7,k8,k9,k10,k11,k12,k13,k14,k15,k16,k17,k18,k19,k20,k2
```

```
1,k22,k23;
```

```
float
kl0,kl1,kl2,kl3,kl4,kl5,kl6,kl7,kl8,kl9,kl10,kl11,kl12,kl13,kl14,kl15,kl16,kl17,kl1
8,kl19,kl20,kl21,kl22,kl23;

float datalogger;

float biayamatilampu;int jammatilampu=35;

float biaya1=0;float biaya2=0;

float v1,v2,i1,i2,daya1,daya2;

float biayatotal1,biayatotal2,biayatotalsimpan;

int dayamati,tanggalsimpan;

int relay1 = 2;

int relay2 = 3;

unsigned long lastMillis = 0;

////////////////////////////////////

//void tampilwaktu()

//{{

// Serial.print(rtc.getDOWStr());

// Serial.print(" ");

//

// Serial.print(rtc.getDateStr());

// Serial.print(" -- ");
```

```
//  
  
// Serial.println(rtc.getTimeStr());  
  
//  
  
//}  
  
////////////////////////////////////  
  
void Va1()  
  
{  
  
    v1 = pzem.voltage(ip);  
  
    if (v1 < 0.0) v1 = 0.0;  
  
    // Serial.print(v1);Serial.print("V; ");  
  
  
    i1 = pzem.current(ip);  
  
    // if(i1 >= 0.0){ Serial.print(i1);Serial.print("A; "); }  
  
}  
  
  
void Va2()  
  
{  
  
    v2 = pzem1.voltage(ip);  
  
    if (v2 < 0.0) v2 = 0.0;  
  
    // Serial.print(v2);Serial.print("V2; ");  
  
  
    i2 = pzem1.current(ip);
```

```

        // if(i2 >= 0.0){ Serial.print(i2);Serial.print("A2; "); }
    }

/////////////////////////////////////////////////////////////////

void daya()
{
    Va1();Va2();

    m=rtc.getTime().min;s=rtc.getTime().sec;

    daya1=v1*i1;daya2=v2*i2;

    // Serial.println((String)"daya1 a = " + daya1);

//

    // Serial.println((String)"daya2a = " + daya2);

if(dayamati==0){ daya1=0;daya2=0;}

    biaya1=daya1*1.450/720;biayatotal1 += (biaya1);

    biaya2=daya2*1.450/720;biayatotal2 += (biaya2);

    if(jammatilampu!=h){EEPROM.writeFloat(500,0);biayamatilampu=0;EEPROM.write(505,h);jammatilampu=h;}

    biayatotalsimpan=biayatotal1+biayatotal2+biayamatilampu;
}

```

////////////////////////////////////////////////////////////////

void bacaEEPROM()

{

j0= EEPROM.readFloat(5);j1= EEPROM.readFloat(9);j2=
EEPROM.readFloat(13);j3= EEPROM.readFloat(17);

j4= EEPROM.readFloat(21);j5= EEPROM.readFloat(25);j6=
EEPROM.readFloat(29);j7= EEPROM.readFloat(33);

j8= EEPROM.readFloat(37);j9= EEPROM.readFloat(41);j10=
EEPROM.readFloat(45);j11= EEPROM.readFloat(49);

j12= EEPROM.readFloat(53);j13= EEPROM.readFloat(57);j14=
EEPROM.readFloat(61);j15= EEPROM.readFloat(65);

j16= EEPROM.readFloat(69);j17= EEPROM.readFloat(73);j18=
EEPROM.readFloat(77);j19= EEPROM.readFloat(81);

j20= EEPROM.readFloat(85);j21= EEPROM.readFloat(89);j22=
EEPROM.readFloat(93);j23= EEPROM.readFloat(97);

k0= EEPROM.readFloat(105);k1= EEPROM.readFloat(109);k2=
EEPROM.readFloat(113);k3= EEPROM.readFloat(117);

k4= EEPROM.readFloat(121);k5= EEPROM.readFloat(125);k6=
EEPROM.readFloat(129);k7= EEPROM.readFloat(133);

k8= EEPROM.readFloat(137);k9= EEPROM.readFloat(141);k10=
EEPROM.readFloat(145);k11= EEPROM.readFloat(149);

k12= EEPROM.readFloat(153);k13= EEPROM.readFloat(157);k14=
EEPROM.readFloat(161);k15= EEPROM.readFloat(165);

k16= EEPROM.readFloat(169);k17= EEPROM.readFloat(173);k18=
EEPROM.readFloat(177);k19= EEPROM.readFloat(181);


```

k20= EEPROM.readFloat(185);k21= EEPROM.readFloat(189);k22=
EEPROM.readFloat(193);k23= EEPROM.readFloat(197);

k10= EEPROM.readFloat(205);k11= EEPROM.readFloat(209);k12=
EEPROM.readFloat(213);k13= EEPROM.readFloat(217);

k14= EEPROM.readFloat(221);k15= EEPROM.readFloat(225);k16=
EEPROM.readFloat(229);k17= EEPROM.readFloat(233);

k18= EEPROM.readFloat(237);k19= EEPROM.readFloat(241);k110=
EEPROM.readFloat(245);k111= EEPROM.readFloat(249);

k112= EEPROM.readFloat(253);k113= EEPROM.readFloat(257);k114=
EEPROM.readFloat(261);k115= EEPROM.readFloat(265);

k116= EEPROM.readFloat(269);k117= EEPROM.readFloat(273);k118=
EEPROM.readFloat(277);k119= EEPROM.readFloat(281);

k120= EEPROM.readFloat(285);k121= EEPROM.readFloat(289);k122=
EEPROM.readFloat(293);k123= EEPROM.readFloat(297);

}

////////////////////////////////////////////////////////////////

void simpanEEPROM()

{

s=rtc.getTime().sec; m=rtc.getTime().min; h=rtc.getTime().hour;
d=rtc.getTime().date;

if (s==59 || s==29){EEPROM.writeFloat(500,biayatotalsimpan);}

if ((h==0 & s==0) || tanggalsimpan!=d )
{ GantiHariEEPROM();tanggalsimpan=d;EEPROM.write(400,d);}

```

```
if (h==0){addr= 5;if ( s== 59 ){EEPROM.writeFloat(addr,biayatotalsimpan);}}

if (h==1){addr = 9 ;if ( s== 59 ){EEPROM.writeFloat(addr,
biayatotalsimpan);}}

if (h==2){addr = 13 ;if ( s== 59 ){EEPROM.writeFloat(addr,
biayatotalsimpan);}}

if (h==3){addr = 17 ;if ( s== 59 ){EEPROM.writeFloat(addr,
biayatotalsimpan);}}

if (h==4){addr = 21 ;if ( s== 59 ){EEPROM.writeFloat(addr,
biayatotalsimpan);}}

if (h==5){addr = 25 ;if ( s== 59 ){EEPROM.writeFloat(addr,
biayatotalsimpan);}}

if (h==6){addr = 29 ;if ( s== 59 ){EEPROM.writeFloat(addr,
biayatotalsimpan);}}

if (h==7){addr = 33 ;if ( s== 59 ){EEPROM.writeFloat(addr,
biayatotalsimpan);}}

if (h==8){addr = 37 ;if ( s== 59 ){EEPROM.writeFloat(addr,
biayatotalsimpan);}}

if (h==9){addr = 41 ;if ( s== 59 ){EEPROM.writeFloat(addr,
biayatotalsimpan);}}

if (h==10){addr = 45 ;if ( s== 59 ){EEPROM.writeFloat(addr,
biayatotalsimpan);}}

if (h==11){addr = 49 ;if ( s== 59 ){EEPROM.writeFloat(addr,
biayatotalsimpan);}}

if (h==12){addr = 53 ;if ( s== 59 ){EEPROM.writeFloat(addr,
biayatotalsimpan);}}
```

```
if (h==13){addr = 57 ;if ( s== 59 ){EEPROM.writeFloat(addr,
biayatotalsimpan);}}

if (h==14){addr = 61 ;if ( s== 59 ){EEPROM.writeFloat(addr,
biayatotalsimpan);}}

if (h==15){addr = 65 ;if ( s== 59 ){EEPROM.writeFloat(addr,
biayatotalsimpan);}}

if (h==16){addr = 69 ;if ( s== 59 ){EEPROM.writeFloat(addr,
biayatotalsimpan);}}

if (h==17){addr = 73 ;if ( s== 59 ){EEPROM.writeFloat(addr,
biayatotalsimpan);}}

if (h==18){addr = 77 ;if ( s== 59 ){EEPROM.writeFloat(addr,
biayatotalsimpan);}}

if (h==19){addr = 81 ;if ( s== 59 ){EEPROM.writeFloat(addr,
biayatotalsimpan);}}

if (h==20){addr = 85 ;if ( s== 59 ){EEPROM.writeFloat(addr,
biayatotalsimpan);}}

if (h==21){addr = 89 ;if ( s== 59 ){EEPROM.writeFloat(addr,
biayatotalsimpan);}}

if (h==22){addr = 93 ;if ( s== 59 ){EEPROM.writeFloat(addr,
biayatotalsimpan);}}

if (h==23){addr = 97 ;if ( s== 59 ){EEPROM.writeFloat(addr,
biayatotalsimpan);}}

}
```

```
void GantiHariEEPROM()
```

```
{
```

```
  bacaEEPROM();
```

```
    EEPROM.writeFloat(105,j0);EEPROM.writeFloat(109,j1);EEPROM.writeFloat(113,j2);EEPROM.writeFloat(117,j3);EEPROM.writeFloat(121,j4);
```

```
    EEPROM.writeFloat(125,j5);EEPROM.writeFloat(129,j6);EEPROM.writeFloat(133,j7);EEPROM.writeFloat(137,j8);EEPROM.writeFloat(141,j9);
```

```
    EEPROM.writeFloat(145,j10);EEPROM.writeFloat(149,j11);EEPROM.writeFloat(153,j12);EEPROM.writeFloat(157,j13);EEPROM.writeFloat(161,j14);
```

```
    EEPROM.writeFloat(165,j15);EEPROM.writeFloat(169,j16);EEPROM.writeFloat(173,j17);EEPROM.writeFloat(177,j18);EEPROM.writeFloat(181,j19);
```

```
    EEPROM.writeFloat(185,j20);EEPROM.writeFloat(189,j21);EEPROM.writeFloat(193,j22);EEPROM.writeFloat(197,j23);
```

```
  delay(1);
```

```
    EEPROM.writeFloat(205,k0);EEPROM.writeFloat(209,k1);EEPROM.writeFloat(213,k2);EEPROM.writeFloat(217,k3);EEPROM.writeFloat(221,k4);
```

```
    EEPROM.writeFloat(225,k5);EEPROM.writeFloat(229,k6);EEPROM.wri
```

```
EEPROM.writeFloat(233,k7);EEPROM.writeFloat(237,k8);EEPROM.writeFloat(241,k9);
```

```
EEPROM.writeFloat(245,k10);EEPROM.writeFloat(249,k11);EEPROM.writeFloat(253,k12);EEPROM.writeFloat(257,k13);EEPROM.writeFloat(261,k14);
```

```
EEPROM.writeFloat(265,k15);EEPROM.writeFloat(269,k16);EEPROM.writeFloat(273,k17);EEPROM.writeFloat(277,k18);EEPROM.writeFloat(281,k19);
```

```
EEPROM.writeFloat(285,k20);EEPROM.writeFloat(289,k21);EEPROM.writeFloat(293,k22);EEPROM.writeFloat(297,k23);
```

```
delay(1);
```

```
EEPROM.writeFloat(5,0);EEPROM.writeFloat(9,0);EEPROM.writeFloat(13,0);EEPROM.writeFloat(17,0);EEPROM.writeFloat(21,0);
```

```
EEPROM.writeFloat(25,0);EEPROM.writeFloat(29,0);EEPROM.writeFloat(33,0);EEPROM.writeFloat(37,0);EEPROM.writeFloat(41,0);
```

```
EEPROM.writeFloat(45,0);EEPROM.writeFloat(49,0);EEPROM.writeFloat(53,0);EEPROM.writeFloat(57,0);EEPROM.writeFloat(61,0);
```

```
EEPROM.writeFloat(65,0);EEPROM.writeFloat(69,0);EEPROM.writeFloat(73,0);EEPROM.writeFloat(77,0);EEPROM.writeFloat(81,0);
```

```

        EEPROM.writeFloat(85,0);EEPROM.writeFloat(89,0);EEPROM.writeFlo
        at(93,0);EEPROM.writeFloat(97,0);

    delay(1);
}

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

void setup() {

    // Open serial communications and wait for port to open:

        rtc.begin();

        //  rtc.setTime(23, 16, 50);

        //  rtc.setDate(23, 7, 2018);

    Serial.begin(38400);

    pzem.setAddress(ip);

    pzem1.setAddress(ip);

    pinMode(relay1, OUTPUT);

    digitalWrite(relay1, HIGH);

    dayamati=1;

    jammatilampu= EEPROM.read(505);

    biayamatilampu= EEPROM.readFloat(500);

    tanggalsimpan=EEPROM.read(400);

}

```

```

void loop()

{

    m=rtc.getTime().min;s=rtc.getTime().sec;h=rtc.getTime().hour;

// Serial.println(h);

    simpanEEPROM();

    bacaEEPROM();

    if (m==0 && s==0){biayatotal1=0;biayatotal2=0;}

    if (millis() - lastMillis > 5000) {

        lastMillis = millis();

        daya();

    }

    hasil=Serial.read();

    if(hasil=='a'){digitalWrite(relay1, HIGH);dayamati=1;}

    if(hasil=='b'){digitalWrite(relay1, LOW);dayamati=0;}

    if(hasil=='1'){datalogger=j0;}if(hasil=='2'){datalogger=j1;}if(hasil=='3'){d
    atalogger=j2;}

    if(hasil=='4'){datalogger=j3;}if(hasil=='5'){datalogger=j4;}if(hasil=='6'){d
    atalogger=j5;}

```

```
if(hasil=='7'){ datalogger=j6;}if(hasil=='8'){ datalogger=j7;}if(hasil=='9'){ d  
atalogger=j8;}
```

```
if(hasil=='!'){ datalogger=j9;}if(hasil=='?'){ datalogger=j10;}if(hasil=='>'){  
datalogger=j11;}
```

```
if(hasil=='<'){ datalogger=j12;}if(hasil=='')){ datalogger=j13;}if(hasil=='{'  
)}{ datalogger=j14;}
```

```
if(hasil=='['){ datalogger=j15;}if(hasil=='']){ datalogger=j16;}if(hasil=='=')  
{ datalogger=j17;}
```

```
if(hasil=='+'){ datalogger=j18;}if(hasil=='-'  
)){ datalogger=j19;}if(hasil=='*'){ datalogger=j20;}
```

```
if(hasil=='&'){ datalogger=j21;}if(hasil=='^'){ datalogger=j22;}if(hasil=='$'  
)}{ datalogger=j23;}
```

```
if(hasil=='c'){ datalogger=k0;}if(hasil=='d'){ datalogger=k1;}if(hasil=='e'){  
datalogger=k2;}
```

```
if(hasil=='f'){ datalogger=k3;}if(hasil=='g'){ datalogger=k4;}if(hasil=='h'){  
datalogger=k5;}
```

```
if(hasil=='i'){ datalogger=k6;}if(hasil=='j'){ datalogger=k7;}if(hasil=='k'){ d  
atalogger=k8;}
```



```
if(hasil=='l'){datalogger=k9;}if(hasil=='m'){datalogger=k10;}if(hasil=='n')  
{datalogger=k11;}
```

```
if(hasil=='o'){datalogger=k12;}if(hasil=='p'){datalogger=k13;}if(hasil=='q'  
)>{datalogger=k14;}
```

```
if(hasil=='r'){datalogger=k15;}if(hasil=='s'){datalogger=k16;}if(hasil=='t')  
{datalogger=k17;}
```

```
if(hasil=='u'){datalogger=k18;}if(hasil=='v'){datalogger=k19;}if(hasil=='  
w'){datalogger=k20;}
```

```
if(hasil=='x'){datalogger=k21;}if(hasil=='y'){datalogger=k22;}if(hasil=='z'  
)>{datalogger=k23;}
```

```
if(hasil=='%'){datalogger=k10;}if(hasil=='#'){datalogger=k11;}if(hasil=='  
@'){datalogger=k12;}
```

```
if(hasil=='~'){datalogger=k13;}if(hasil=='A'){datalogger=k14;}if(hasil=='B'  
)>{datalogger=k15;}
```

```
if(hasil=='C'){datalogger=k16;}if(hasil=='D'){datalogger=k17;}if(hasil=='E'  
)>{datalogger=k18;}
```

```
if(hasil=='F'){ datalogger=k19;}if(hasil=='G'){ datalogger=k110;}if(hasil=='H'){ datalogger=k111;}
```

```
if(hasil=='I'){ datalogger=k112;}if(hasil=='J'){ datalogger=k113;}if(hasil=='K'){ datalogger=k114;}
```

```
if(hasil=='L'){ datalogger=k115;}if(hasil=='M'){ datalogger=k116;}if(hasil=='N'){ datalogger=k117;}
```

```
if(hasil=='O'){ datalogger=k118;}if(hasil=='P'){ datalogger=k119;}if(hasil=='Q'){ datalogger=k120;}
```

```
if(hasil=='R'){ datalogger=k121;}if(hasil=='S'){ datalogger=k122;}if(hasil=='T'){ datalogger=k123;}
```

```
Serial.print(" | ");
```

```
Serial.print(daya1);
```

```
Serial.print(" | ");
```

```
Serial.print(biayatotal1);
```

```
Serial.print(" | ");
```

```
Serial.print(daya2);
```

```
Serial.print(" | ");
```

```
Serial.print(biayatotal2);
```

```
Serial.print(" | ");
```

```
Serial.print(datalogger);
```

```
Serial.print(" | ");  
delay(300);  
}
```

LAMPIRAN 3

Program Android

```
when Screen1 .BackPressed
do close application
```

```
when Button1 .Click
do open another screen screenName "Screen2"
```

```
when Screen2 .BackPressed
do close screen
```

```
when Button1 .Click
do open another screen screenName "StopKontak1"
```

```
when Button2 .Click
do open another screen screenName "StopKontak2"
```

```
initialize global data to create empty list
```

```
when StopKontak1 .BackPressed
do call BluetoothClient1 .Disconnect
close screen
```

```
when ListPicker1 .BeforePicking
do set ListPicker1 .Elements to BluetoothClient1 .AddressesAndNames
```

```
when ListPicker1 .AfterPicking
do if call BluetoothClient1 .Connect
address ListPicker1 .Selection
then set ListPicker1 .Elements to BluetoothClient1 .AddressesAndNames
if BluetoothClient1 .IsConnected
then set Label2 .Text to "Connected"
else set Label2 .Text to "Disconnected"
```

```
when ON .Click
do
  if BluetoothClient1 .IsConnected
  then
    call BluetoothClient1 .SendText
    text "a"
    set ON .TextColor to red
    set OFF .TextColor to black
```

```
when OFF .Click
do
  if BluetoothClient1 .IsConnected
  then
    call BluetoothClient1 .SendText
    text "b"
    set OFF .TextColor to red
    set ON .TextColor to black
```

```
when Clock1 .Timer
do
  if BluetoothClient1 .IsConnected and call BluetoothClient1 .BytesAvailableToReceive > 0
  then
    set global data to split text call BluetoothClient1 .ReceiveText
    numberOfBytes call BluetoothClient1 .BytesAvailableToReceive
    at 0
    if length of list list get global data >= 2
    then
      set Daya .Text to select list item list get global data
      index 2
    if length of list list get global data >= 3
    then
      set Blaya .Text to select list item list get global data
      index 3
    if length of list list get global data >= 4
    then
      set Daya2 .Text to select list item list get global data
      index 4
    if length of list list get global data >= 5
    then
      set Blaya2 .Text to select list item list get global data
      index 5
    if length of list list get global data >= 6
    then
      set Label19 .Text to select list item list get global data
      index 6
```

```
if BluetoothClient1 . IsConnected and get selection = Hari ini Pukul 08.00
then call BluetoothClient1 . SendText
    text " 8 "

if BluetoothClient1 . IsConnected and get selection = Hari ini Pukul 07.00
then call BluetoothClient1 . SendText
    text " 7 "

if BluetoothClient1 . IsConnected and get selection = Hari ini Pukul 08.00
then call BluetoothClient1 . SendText
    text " 8 "

if BluetoothClient1 . IsConnected and get selection = Hari ini Pukul 09.00
then call BluetoothClient1 . SendText
    text " 9 "

if BluetoothClient1 . IsConnected and get selection = Hari ini Pukul 10.00
then call BluetoothClient1 . SendText
    text " 10 "

if BluetoothClient1 . IsConnected and get selection = Hari ini Pukul 11.00
then call BluetoothClient1 . SendText
    text " ? "
```

```
when Spinner1 . AfterSelecting
    selection
do
if BluetoothClient1 . IsConnected and get selection = Hari ini Pukul 01.00
then call BluetoothClient1 . SendText
    text " 1 "

if BluetoothClient1 . IsConnected and get selection = Hari ini Pukul 02.00
then call BluetoothClient1 . SendText
    text " 2 "

if BluetoothClient1 . IsConnected and get selection = Hari ini Pukul 03.00
then call BluetoothClient1 . SendText
    text " 3 "

if BluetoothClient1 . IsConnected and get selection = Hari ini Pukul 04.00
then call BluetoothClient1 . SendText
    text " 4 "

if BluetoothClient1 . IsConnected and get selection = Hari ini Pukul 05.00
then call BluetoothClient1 . SendText
    text " 5 "
```

```
if BluetoothClient1 . IsConnected and get selection = Hari ini Pukul 12.00
then call BluetoothClient1 . SendText
      text >
if BluetoothClient1 . IsConnected and get selection = Hari ini Pukul 13.00
then call BluetoothClient1 . SendText
      text <
if BluetoothClient1 . IsConnected and get selection = Hari ini Pukul 14.00
then call BluetoothClient1 . SendText
      text 1
if BluetoothClient1 . IsConnected and get selection = Hari ini Pukul 15.00
then call BluetoothClient1 . SendText
      text 11
if BluetoothClient1 . IsConnected and get selection = Hari ini Pukul 16.00
then call BluetoothClient1 . SendText
      text 111
if BluetoothClient1 . IsConnected and get selection = Hari ini Pukul 17.00
then call BluetoothClient1 . SendText
      text 111
```

```
if BluetoothClient1 . IsConnected and get selection = Hari ini Pukul 24.00
then call BluetoothClient1 . SendText
      text 3
if BluetoothClient1 . IsConnected and get selection = Kemarin Pukul 01.00
then call BluetoothClient1 . SendText
      text 0
if BluetoothClient1 . IsConnected and get selection = Kemarin Pukul 02.00
then call BluetoothClient1 . SendText
      text 0
if BluetoothClient1 . IsConnected and get selection = Kemarin Pukul 03.00
then call BluetoothClient1 . SendText
      text e
if BluetoothClient1 . IsConnected and get selection = Kemarin Pukul 04.00
then call BluetoothClient1 . SendText
      text i
if BluetoothClient1 . IsConnected and get selection = Kemarin Pukul 05.00
then call BluetoothClient1 . SendText
      text g
```



```
if BluetoothClient1 . IsConnected and get selection = "Kemarin Pukul 08.00"
then call BluetoothClient1 . SendText
    text "h"
if BluetoothClient1 . IsConnected and get selection = "Kemarin Pukul 07.00"
then call BluetoothClient1 . SendText
    text "i"
if BluetoothClient1 . IsConnected and get selection = "Kemarin Pukul 08.00"
then call BluetoothClient1 . SendText
    text "j"
if BluetoothClient1 . IsConnected and get selection = "Kemarin Pukul 09.00"
then call BluetoothClient1 . SendText
    text "k"
if BluetoothClient1 . IsConnected and get selection = "Kemarin Pukul 10.00"
then call BluetoothClient1 . SendText
    text "l"
if BluetoothClient1 . IsConnected and get selection = "Kemarin Pukul 11.00"
then call BluetoothClient1 . SendText
    text "m"
```

```
if BluetoothClient1 . IsConnected and get selection = "Kemarin Pukul 08.00"
then call BluetoothClient1 . SendText
    text "h"
if BluetoothClient1 . IsConnected and get selection = "Kemarin Pukul 07.00"
then call BluetoothClient1 . SendText
    text "i"
if BluetoothClient1 . IsConnected and get selection = "Kemarin Pukul 08.00"
then call BluetoothClient1 . SendText
    text "j"
if BluetoothClient1 . IsConnected and get selection = "Kemarin Pukul 09.00"
then call BluetoothClient1 . SendText
    text "k"
if BluetoothClient1 . IsConnected and get selection = "Kemarin Pukul 10.00"
then call BluetoothClient1 . SendText
    text "l"
if BluetoothClient1 . IsConnected and get selection = "Kemarin Pukul 11.00"
then call BluetoothClient1 . SendText
    text "m"
```

```
if BluetoothClient1 . IsConnected and get selection = "Kemarin Pukul 12.00"
then call BluetoothClient1 . SendText
    text "n"
if BluetoothClient1 . IsConnected and get selection = "Kemarin Pukul 13.00"
then call BluetoothClient1 . SendText
    text "o"
if BluetoothClient1 . IsConnected and get selection = "Kemarin Pukul 14.00"
then call BluetoothClient1 . SendText
    text "p"
if BluetoothClient1 . IsConnected and get selection = "Kemarin Pukul 15.00"
then call BluetoothClient1 . SendText
    text "q"
if BluetoothClient1 . IsConnected and get selection = "Kemarin Pukul 16.00"
then call BluetoothClient1 . SendText
    text "r"
if BluetoothClient1 . IsConnected and get selection = "Kemarin Pukul 17.00"
then call BluetoothClient1 . SendText
    text "s"
```

```
if BluetoothClient1 . IsConnected and get selection = "Kemarin Pukul 18.00"
then call BluetoothClient1 . SendText
    text "t"
if BluetoothClient1 . IsConnected and get selection = "Kemarin Pukul 19.00"
then call BluetoothClient1 . SendText
    text "u"
if BluetoothClient1 . IsConnected and get selection = "Kemarin Pukul 20.00"
then call BluetoothClient1 . SendText
    text "v"
if BluetoothClient1 . IsConnected and get selection = "Kemarin Pukul 21.00"
then call BluetoothClient1 . SendText
    text "w"
if BluetoothClient1 . IsConnected and get selection = "Kemarin Pukul 22.00"
then call BluetoothClient1 . SendText
    text "x"
if BluetoothClient1 . IsConnected and get selection = "Kemarin Pukul 23.00"
then call BluetoothClient1 . SendText
    text "y"
```

```
if BluetoothClient1 . IsConnected and get selection = Kemarin Pukul 18.00
then call BluetoothClient1 . SendText
      text i
if BluetoothClient1 . IsConnected and get selection = Kemarin Pukul 19.00
then call BluetoothClient1 . SendText
      text U
if BluetoothClient1 . IsConnected and get selection = Kemarin Pukul 20.00
then call BluetoothClient1 . SendText
      text V
if BluetoothClient1 . IsConnected and get selection = Kemarin Pukul 21.00
then call BluetoothClient1 . SendText
      text W
if BluetoothClient1 . IsConnected and get selection = Kemarin Pukul 22.00
then call BluetoothClient1 . SendText
      text X
if BluetoothClient1 . IsConnected and get selection = Kemarin Pukul 23.00
then call BluetoothClient1 . SendText
      text Y
```

```
if BluetoothClient1 . IsConnected and get selection = Kemarin Pukul 24.00
then call BluetoothClient1 . SendText
      text Z
if BluetoothClient1 . IsConnected and get selection = Kemarin Lusa Pukul 01.00
then call BluetoothClient1 . SendText
      text S
if BluetoothClient1 . IsConnected and get selection = Kemarin Lusa Pukul 02.00
then call BluetoothClient1 . SendText
      text #
if BluetoothClient1 . IsConnected and get selection = Kemarin Lusa Pukul 03.00
then call BluetoothClient1 . SendText
      text @
if BluetoothClient1 . IsConnected and get selection = Kemarin Lusa Pukul 04.00
then call BluetoothClient1 . SendText
      text ~
if BluetoothClient1 . IsConnected and get selection = Kemarin Lusa Pukul 05.00
then call BluetoothClient1 . SendText
      text A
```

```
if BluetoothClient1.isConnected and get selection == 'Kemarin Lusa Pukul 08.00'
then call BluetoothClient1.SendText
      text 'B'
if BluetoothClient1.isConnected and get selection == 'Kemarin Lusa Pukul 07.00'
then call BluetoothClient1.SendText
      text 'C'
if BluetoothClient1.isConnected and get selection == 'Kemarin Lusa Pukul 08.00'
then call BluetoothClient1.SendText
      text 'D'
if BluetoothClient1.isConnected and get selection == 'Kemarin Lusa Pukul 09.00'
then call BluetoothClient1.SendText
      text 'E'
if BluetoothClient1.isConnected and get selection == 'Kemarin Lusa Pukul 10.00'
then call BluetoothClient1.SendText
      text 'F'
if BluetoothClient1.isConnected and get selection == 'Kemarin Lusa Pukul 11.00'
then call BluetoothClient1.SendText
      text 'G'
```

```
if BluetoothClient1.isConnected and get selection == 'Kemarin Lusa Pukul 12.00'
then call BluetoothClient1.SendText
      text 'H'
if BluetoothClient1.isConnected and get selection == 'Kemarin Lusa Pukul 13.00'
then call BluetoothClient1.SendText
      text 'I'
if BluetoothClient1.isConnected and get selection == 'Kemarin Lusa Pukul 14.00'
then call BluetoothClient1.SendText
      text 'J'
if BluetoothClient1.isConnected and get selection == 'Kemarin Lusa Pukul 15.00'
then call BluetoothClient1.SendText
      text 'K'
if BluetoothClient1.isConnected and get selection == 'Kemarin Lusa Pukul 16.00'
then call BluetoothClient1.SendText
      text 'L'
if BluetoothClient1.isConnected and get selection == 'Kemarin Lusa Pukul 17.00'
then call BluetoothClient1.SendText
      text 'M'
```

```
if BluetoothClient1 . IsConnected and get selection = "Kemarin Lusa Pukul 18.00"
then call BluetoothClient1 .SendText
    text "N"
if BluetoothClient1 . IsConnected and get selection = "Kemarin Lusa Pukul 19.00"
then call BluetoothClient1 .SendText
    text "O"
if BluetoothClient1 . IsConnected and get selection = "Kemarin Lusa Pukul 20.00"
then call BluetoothClient1 .SendText
    text "P"
if BluetoothClient1 . IsConnected and get selection = "Kemarin Lusa Pukul 21.00"
then call BluetoothClient1 .SendText
    text "Q"
if BluetoothClient1 . IsConnected and get selection = "Kemarin Lusa Pukul 22.00"
then call BluetoothClient1 .SendText
    text "R"
if BluetoothClient1 . IsConnected and get selection = "Kemarin Lusa Pukul 23.00"
then call BluetoothClient1 .SendText
    text "S"
```

```
if BluetoothClient1 . IsConnected and get selection = "Kemarin Lusa Pukul 24.00"
then call BluetoothClient1 .SendText
    text "T"
```

LAMPIRAN 4

Data Sheet PZEM-004T

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

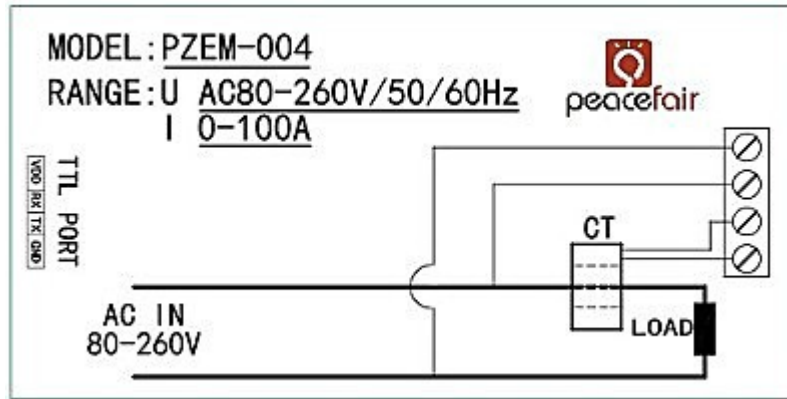


Figure 1 Wiring diagram

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