



TUGAS AKHIR - KI141502

SISTEM PERINGATAN DINI LIMIT PENGGUNAAN LISTRIK BERBASIS ARDUINO

MUHAMMAD TAUFIQ HIDAYAT
NRP 5109 100 093

Dosen Pembimbing I
Dr. Radityo Anggoro, S.Kom., M.Sc.
Dosen Pembimbing II
Ir. Muchammad Husni, M.Kom

JURUSAN TEKNIK INFORMATIKA
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember
Surabaya 2016



TUGAS AKHIR - KI141502

SISTEM PERINGATAN DINI LIMIT PENGGUNAAN LISTRIK BERBASIS ARDUINO

**MUHAMMAD TAUFIQ HIDAYAT
NRP 5109 100 093**

**Dosen Pembimbing I
Dr. Radityo Anggoro, S.Kom., M.Sc.**

**Dosen Pembimbing II
Ir. Muchammad Husni, M.Kom**

**JURUSAN TEKNIK INFORMATIKA
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember
Surabaya 2016**

[Halaman ini sengaja dikosongkan]



FINAL PROJECT - KI141502

EARLY WARNING SYSTEM LIMIT USE OF ELECTRICITY BASED ON ARDUINO MICROCONTROLLER

**MUHAMMAD TAUFIQ HIDAYAT
NRP 5109 100 093**

**Supervisor I
Dr. Radityo Anggoro, S.Kom., M.Sc.**

**Supervisor II
Ir. Muchammad Husni, M.Kom**

**DEPARTMENT OF INFORMATICS
FACULTY OF INFORMATION TECHNOLOGY
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA 2016**

[Halaman ini sengaja dikosongkan]

LEMBAR PENGESAHAN

**Sistem Peringatan Dini Limit Penggunaan Listrik Berbasis
Arduino**

TUGAS AKHIR

Diajukan Untuk Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer
pada
Bidang Studi Komputasi Berbasis Jaringan
Program Studi S-1 Jurusan Teknik Informatika
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember

Oleh

MUHAMMAD TAUFIQ HIDAYAT

NRP : 5109 100 093

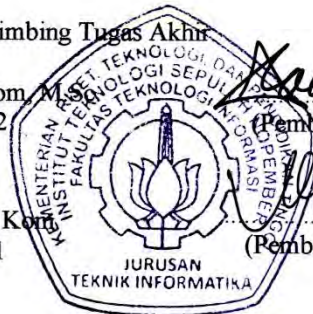
Disetujui oleh Dosen Pembimbing Tugas Akhir

Dr. Radityo Anggoro S.Kom, M.Kom
NIP.198410162008121002

Radityo
(Pembimbing 1)

Ir. Muchammad Husni, M.Kom
NIP.196002211984031001

Muchammad Husni
(Pembimbing 2)



**SURABAYA
JUNI, 2016**

[Halaman ini sengaja dikosongkan]

Sistem Peringatan Dini Limit Penggunaan Listrik Berbasis Arduino

Nama Mahasiswa : MUHAMMAD TAUFIQ HIDAYAT
NRP : 5109100093
Jurusan : Teknik Informatika FTIF-ITS
Dosen Pembimbing 1 : Dr. Radityo Anggoro, S.Kom., M.Sc.
Dosen Pembimbing 2 : Ir. Muchammad Husni, M.Kom

Abstrak

Penggunaan listrik yang tidak terkontrol dapat berdampak pada pemborosan energi dan tentu saja pemborosan uang. seringkali pengguna listrik lupa untuk mematikan atau mencabut sambungan listrik dari stop kontak yang membuat arus listrik terus mengalir, yang mana sebenarnya tidak diperlukan

Untuk itulah dibuat sebuah sistem yang dapat mengendalikan dan membatasi penggunaan listrik serta mampu untuk mengirimkan pesan peringatan ke pengguna apabila penggunaan listrik telah melewati kuota tertentu

Sistem Peringatan Dini Limit Penggunaan Listrik ini terdiri dari mikrokontroler Arduino sebagai unit pemrosesan utama pada sistem. Mikrokontroler Arduino yang dikombinasikan dengan modul GSM shield berperan sebagai penerima pesan perintah dan juga pengirim pesan peringatan dari perangkat smartphone android pengguna.

Kata kunci: Android, Arduino, GSM Shield, Mikrokontroler, Smartphone

[Halaman ini sengaja dikosongkan]

EARLY WARNING SYSTEM LIMIT USE OF ELECTRICITY BASED ON ARDUINO MICROCONTROLLER

Nama Mahasiswa : MUHAMMAD TAUFIQ HIDAYAT
NRP : 5109100093
Jurusan : Teknik Informatika FTIF-ITS
Dosen Pembimbing 1 : Dr. Radityo Anggoro, S.Kom., M.Sc.
Dosen Pembimbing 2 : Ir. Muchammad Husni, M.Kom

Abstract

Uncontrolled use of electricity could have an impact on energy waste and of course a waste of money. electricity users often forget to turn off or unplug the power cord from the wall socket that makes electric current continues to flow, which is not actually needed.

For that created a system that can control and limit the use of electricity. As well as being able to send a warning message to the user when the use of electricity has passed a certain quota.

Early Warning System Power Usage Limit consists of an Arduino microcontroller as the main processing unit in the system. Arduino microcontroller combined with a GSM shield module acts as the receiver of the message sender command and warning messages from the device android smartphone users

Keywords: Android, Arduino, GSM Shield, Microcontroller, Smartphone

[Halaman ini sengaja dikosongkan]

KATA PENGANTAR

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

Puji syukur Alhamdulillah kepada Allah Yang Maha Kuasa atas segala karunia dan rahmat-Nya penulis dapat menyelesaikan Tugas Akhir yang berjudul:

“Sistem Peringatan Dini Limit Penggunaan Listrik Berbasis Arduino”

Melalui lembar ini, penulis ingin menyampaikan ucapan terima kasih dan penghormatan yang sebesar-besarnya kepada:

1. Dosen-dosen pembimbing dan dosen-dosen pengajar Teknik Informatika ITS
2. Orangtua saya
3. Saudara-saudara saya
4. Teman-teman saya
5. Serta pihak-pihak lain yang namanya tidak dapat penulis sebutkan satu -persatu.

Bagaimanapun juga penulis telah berusaha sebaik-baiknya dalam menyusun Tugas Akhir ini, namun penulis mohon maaf apabila terdapat kekurangan, kesalahan maupun kelalaian yang telah penulis lakukan. Kritik dan saran yang membangun dapat disampaikan sebagai bahan perbaikan selanjutnya.

Surabaya, Juni 2016

Muhammad Taufiq Hidayat

[Halaman ini sengaja dikosongkan]

DAFTAR ISI

LEMBAR PENGESAHAN	v
<i>Abstrak</i>	vii
<i>Abstract</i>	viii
KATA PENGANTAR	ix
DAFTAR ISI	x
DAFTAR GAMBAR	xii
DAFTAR TABEL	xiv
BAB I PENDAHULUAN	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah	2
1.3 Batasan Masalah.....	2
1.4 Tujuan.....	2
1.5 Metodologi	3
1.6 Sistematika Penulisan Laporan Tugas Akhir.....	4
BAB II DASAR TEORI	7
2.1 Arduino ATMEGA 328.....	7
2.2 Arduino GSM <i>Shield</i>	9
2.3 Sensor Arus ACS712	10
2.4 Sensor Tegangan ZMPT101B	11
2.5 <i>Relay</i>	12
2.6 Android.....	13
2.7 Android Studio	14
2.8 Arduino <i>Software</i> (IDE)	15
BAB III ANALISIS DAN PERANCANGAN SISTEM	19
3.1 Deskripsi Umum.....	19
3.2 Arsitektur Umum Sistem	20
3.3 Perancangan Perangkat Keras	21
3.4 Perancangan Diagram <i>Use Case</i>	23
3.5 Diagram Alir Aplikasi Sistem	24
3.5.1 Diagram Alir Proses Perhitungan Daya.....	24
3.5.2 Diagram Alir Proses Pengiriman Informasi Peringatan	26
3.5.3 Diagram Alir Proses Reset Kuota.....	28

3.5.4 Diagram Alir Proses Kontrol Listrik	30
3.6 Rancangan Antarmuka Sistem	32
3.6.1 Antarmuka Aplikasi Android	32
BAB IV IMPLEMENTASI.....	35
4.1 Lingkungan Implementasi	35
4.1.1 Lingkungan Implementasi Perangkat Keras.....	35
4.1.2 Lingkungan Implementasi Perangkat Lunak.....	36
4.2 Implementasi Perangkat Keras	36
4.3 Implementasi Proses.....	38
4.3.1 Implementasi Kondisi Awal Sistem	39
4.3.2 Implementasi Kondisi Sistem Telah Berjalan	41
4.4 Implementasi Sistem	50
4.4.1 Implementasi Sistem <i>Smart Home</i>	50
BAB V UJI COBA DAN ANALISA	53
5.1 Lingkungan Uji Coba	53
5.2 Skenario Uji Coba	54
5.2.1 Uji Coba Fungsionalitas Perangkat Android.....	54
5.2.2 Uji Coba Fungsionalitas Perangkat Arduino.....	61
5.2.3 Uji Coba Kalibrasi Sistem	72
BAB VI KESIMPULAN DAN SARAN.....	75
6.1 Kesimpulan.....	75
6.2 Saran.....	75
DAFTAR PUSTAKA	77
BIODATA PENULIS.....	79

DAFTAR GAMBAR

Gambar 2.1. Arduino ATMEGA 328.....	7
Gambar 2.2. Pin Arduino ATMEGA 328.....	8
Gambar 2.3. Arduino GSM <i>Shield</i>	9
Gambar 2.4. Sensor Arus ACS712.....	10
Gambar 2.5. Sensor Tegangan ZMPT101B	12
Gambar 2.6. Modul <i>Relay</i>	13
Gambar 2.7. Logo Android	14
Gambar 2.8. <i>Screenshot</i> Tampilan OS Android.....	14
Gambar 2.9. Logo Aplikasi Android studio	15
Gambar 2.10. Logo Aplikasi Arduino IDE	15
Gambar 2.11. Halaman Antarmuka Arduino IDE.....	16
Gambar 3.1. Skema Pemasangan Perangkat	19
Gambar 3.2. Keterangan Skema Pemasangan Perangkat	20
Gambar 3.3. Arsitektur Sistem	21
Gambar 3.4. Rancangan Perangkat Keras	22
Gambar 3.5. Diagram <i>Use Case</i>	24
Gambar 3.6. Diagram Alir Perhitungan Daya	25
Gambar 3.7. Diagram Alir Proses Pengiriman Informasi Peringatan.....	27
Gambar 3.8. Diagram Alir Proses Reset Kuota.....	29
Gambar 3.9. Diagram Alir Proses Kontrol Listrik	31
Gambar 3.10. Antar Muka Utama Program	32
Gambar 3.11. Antar Muka Pengaturan Reset Kuota	33
Gambar 3.12. Antar Muka Pengaturan Limit.....	34
Gambar 4.1. Rangkaian Perangkat Keras.....	37
Gambar 4.2. <i>Pseudocode</i> Inisialisasi Awal Arduino.....	40
Gambar 4.3. <i>Pseudocode</i> Proses Pembacaan Daya Listrik	42
Gambar 4.4. <i>Pseudocode</i> Memutuskan Sambungan Listrik....	43
Gambar 4.5. <i>Pseudocode</i> Menyalakan Sambungan Listrik....	44
Gambar 4.6. <i>Pseudocode</i> Cek Kuota.....	44
Gambar 4.7. <i>Pseudocode</i> Reset Kuota	45
Gambar 4.8. <i>Pseudocode</i> Set Limit.....	45

Gambar 4.9. <i>Pseudocode</i> Pengiriman Pesan Menyalakan Perangkat.....	46
Gambar 4.10. <i>Pseudocode</i> Pengiriman Pesan Mematikan Perangkat.....	47
Gambar 4.11. <i>Pseudocode</i> Pengiriman Pesan Cek Kuota.....	48
Gambar 4.12. <i>Pseudocode</i> Pengiriman Pesan Reset Kuota....	49
Gambar 4.13. <i>Pseudocode</i> Pengiriman Pesan Set Limit	50
Gambar 4.14. Rangkaian Smart Home Secara Keseluruhan ..	51
Gambar 5.1. Tampilan Menu Utama Aplikasi Android	55
Gambar 5.2. Tampilan Menu Reset Kuota.....	56
Gambar 5.3. Tampilan Menu Pengaturan Limit.....	56
Gambar 5.4. Tampilan Pesan Balasan Dari Perangkat	57
Gambar 5.5. Tampilan Pesan Balasan Dari Perangkat	57
Gambar 5.6. Tampilan Menu Pengaturan Limit.....	58
Gambar 5.7 Tampilan Peringatan Limit 1 kWh	59
Gambar 5.8 Tampilan Peringatan Limit 2 kWh	59
Gambar 5.9 Tampilan Peringatan Limit 3 kWh	60
Gambar 5.10. Tampilan Peringatan Limit 4 kWh	60
Gambar 5.11. Tampilan Peringatan Limit 5 kWh	61
Gambar 5.12. Pengujian dan Pengukuran Tegangan Masukan AC Pada Sensor Tegangan	62
Gambar 5.13. Pengujian dan Pengukuran Tegangan Keluaran <i>Trafo Step Down</i>	63
Gambar 5.14. <i>Pseudocode</i> Perhitungan dan Pengukuran Tegangan Arus AC.....	64
Gambar 5.15. Pengujian Dan Pengukuran Tegangan Keluaran Sensor Arus	65
Gambar 5.16. <i>Pseudocode</i> Proses Pembacaan Arus AC.....	66
Gambar 5.17. Respon Modem Pada Mode Teks.....	67
Gambar 5.18. Respon Modem Saat Mendapat Pesan Masuk	67
Gambar 5.19. Respon Modem Saat Mendapat Pesan Masuk	68
Gambar 5.20. Respon Modem Saat Mendapat Pesan Masuk	68
Gambar 5.21. Respon Modem Saat Mendapat Pesan Masuk	69
Gambar 5.22. Respon Modem Saat Mendapat Pesan Masuk	69
Gambar 5.23. <i>Pseudocode</i> Memproses Pesan yang Masuk ...	71

DAFTAR TABEL

Tabel 5.1 Uji Coba Pertama	72
Tabel 5.2 Uji Coba Kedua	72
Tabel 5.3 Uji Coba Ketiga.....	73

[Halaman ini sengaja dikosongkan]

BAB I

PENDAHULUAN

Pada bab ini dipaparkan mengenai garis besar Tugas Akhir, meliputi latar belakang, tujuan, rumusan permasalahan, batasan permasalahan, metodologi penyelesaian Tugas Akhir, dan sistematika penulisan.

1.1 Latar Belakang

Penggunaan listrik pada era modern seperti sekarang ini sudah bukan menjadi hal yang unik, hampir semua perangkat penunjang pekerjaan manusia semuanya menggunakan listrik untuk pengoperasiannya contohnya saja perangkat komputer dan ponsel. Dapat dibayangkan betapa terbatasnya kegiatan manusia jika ia tidak memiliki akses ke sebuah sumber listrik seperti misalnya saat terjadi pemadaman listrik, atau aliran listrik terputus akibat belum membayar tagihan atau pulsa listrik yang dimiliki sudah habis atau belum dibayar.

Penggunaan listrik prabayar sudah sering ditemui pada zaman sekarang, akan tetapi karena kekurangannya seperti yang sudah disebutkan sebelumnya, masyarakat merasa enggan untuk menggunakan listrik prabayar ini dan juga mengesampingkan kelebihanannya.

Proyek ini dibuat agar dapat membantu memonitor penggunaan listrik, sehingga sang pengguna dapat membeli/membayar sejumlah pulsa listrik sebelum aliran listrik di tempatnya padam karena kehabisan pulsa.

1.2 Rumusan Masalah

Rumusan masalah yang diangkat dalam Tugas Akhir ini sebagai berikut:

1. Bagaimana mikrokontroler dapat menghitung jumlah daya listrik yang telah digunakan ?
2. Bagaimana Arduino dapat membaca dan mengirim perintah melalui SMS ?
3. Bagaimana memutuskan arus listrik pada keadaan yang telah ditentukan ?
4. Bagaimana memonitor penggunaan listrik ?

1.3 Batasan Masalah

Beberapa batasan Tugas Akhir ini adalah sebagai berikut:
Sistem ini menggunakan beberapa perangkat keras, yaitu :

- Satu mikrokontroler Arduino uno
- Satu arduino GSM Shield
- *Relay*
- Sensor tegangan
- Sensor arus
- Sistem ini dioperasikan dengan bantuan aplikasi Android *base*

1.4 Tujuan

Tujuan tugas akhir ini adalah untuk sistem peringatan dini saat sisa daya listrik yang dimiliki mendekati habis, sehingga dapat meminimalisir kemungkinan listrik padam dikarenakan kehabisan daya.

Manfaat tugas akhir ini antara lain adalah untuk memudahkan pengguna untuk dapat meng-otomatisasi informasi sisa pulsa listrik. Sehingga sewaktu pengguna tidak berada di tempat rangkaian terpasang dan daya listrik mendekati habis, pengguna tetap mendapat informasi tentang pulsa listrik tersebut. Sehingga

ketika sang pengguna akan pulang ke rumah, ia dapatampir ke sebuah tempat untuk membeli sejumlah pulsa listrik.

1.5 Metodologi

Langkah-langkah yang ditempuh dalam pengerjaan Tugas Akhir ini adalah:

1. Studi Literatur

Pada tahap ini dilakukan pengumpulan informasi mengenai hal-hal pendukung pengerjaan Tugas Akhir. Pengumpulan ini dimaksudkan untuk melakukan analisis dan perancangan sistem. Adapun literatur yang dipakai adalah:

- a. Teori konsep Android Studio
- b. Teori konsep Arduino IDE
- c. Teori penggunaan Arduino pada Arduino IDE
- d. Teori penggunaan Android pada Android Studio

2. Analisis dan Desain Perangkat Sistem

Pada tahapan ini, penulis melakukan analisis dan desain aplikasi. Analisis dan perancangan aplikasi dilakukan untuk merumuskan spesifikasi kebutuhan aplikasi. Adapun tahap desain dilakukan untuk memodelkan hasil analisis dalam bentuk desain arsitektur, desain antarmuka dan desain-desain pendukung lain dalam pembangunan aplikasi.

3. Implementasi

Pada tahap ini dilakukan implementasi perangkat lunak ke dalam bentuk kode program.

4. Pengujian dan Evaluasi

Tahapan ini digunakan untuk melakukan pengujian dan evaluasi pada aplikasi. Tahapan ini bertujuan untuk mengetahui kesalahan-kesalahan yang ada pada aplikasi sehingga dapat dilakukan perbaikan terhadap aplikasi itu sendiri. Pengujian berdasarkan pengujian pengguna.

5. Penyusunan Buku Tugas Akhir.

Pada tahap ini dilakukan penyusunan laporan yang menjelaskan dasar teori dan metode yang digunakan dalam Tugas Akhir ini serta hasil dari implementasi aplikasi perangkat lunak yang telah dibuat.

1.6 Sistematika Penulisan Laporan Tugas Akhir

Buku Tugas Akhir ini bertujuan untuk mendapatkan gambaran dari pengerjaan Tugas Akhir ini. Selain itu, diharapkan dapat berguna untuk pembaca yang tertarik untuk melakukan pengembangan lebih lanjut. Secara garis besar, buku Tugas Akhir terdiri atas beberapa bagian seperti berikut ini:

Bab I Pendahuluan

Bab yang berisi mengenai latar belakang, tujuan, dan manfaat dari pembuatan Tugas Akhir. Selain itu permasalahan, batasan masalah, metodologi yang digunakan, dan sistematika penulisan juga merupakan bagian dari bab ini.

Bab II Dasar Teori

Bab ini berisi penjelasan secara detail mengenai dasar-dasar penunjang dan teori-teori yang digunakan untuk mendukung pembuatan Tugas Akhir ini.

Bab III Perancangan Perangkat Lunak

Bab ini berisi tentang desain sistem yang disajikan dalam bentuk *pseudocode*.

Bab IV Implementasi

Bab ini membahas implementasi dari desain yang telah dibuat pada bab sebelumnya. Penjelasan berupa *code* yang digunakan untuk proses implementasi.

Bab V Uji Coba Dan Evaluasi

Bab ini menjelaskan kemampuan perangkat lunak dengan melakukan pengujian kebenaran dan pengujian kinerja dari sistem yang telah dibuat.

Bab VI Kesimpulan Dan Saran

Bab ini merupakan bab terakhir yang menyampaikan kesimpulan dari hasil uji coba yang dilakukan dan saran untuk pengembangan perangkat lunak ke depannya.

[Halaman ini sengaja dikosongkan]

BAB II DASAR TEORI

Pada bab ini membahas mengenai teori-teori yang berkaitan dengan pengimplementasian perangkat lunak. Penjelasan ini bertujuan untuk memberikan gambaran secara umum terhadap sistem yang dibuat dan berguna sebagai penunjang dalam pengembangan perangkat lunak.

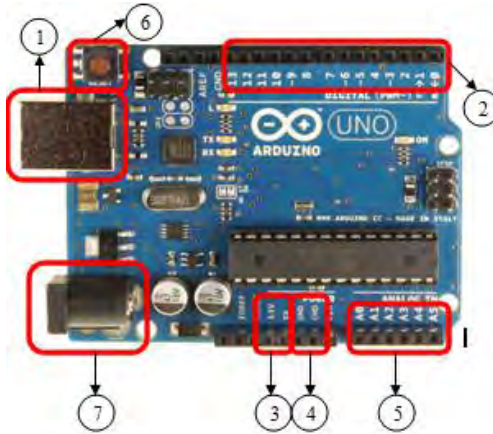
2.1 Arduino ATMEGA 328

Arduino ATMEGA 328 adalah kit elektronik atau papan rangkaian elektronik yang dapat ditanamkan sebuah program agar rangkaian elektronik tersebut dapat membaca masukan, memproses masukan tersebut dan kemudian menghasilkan keluaran sesuai yang diinginkan. Contoh pada Gambar 2.1.



Gambar 2.1. Arduino ATMEGA 328

Komponen-komponen yang ada pada Arduino Uno revision 3 beserta penjelasannya disajikan pada Gambar 2.2. *USB Connection* (1) adalah bagian dari Arduino Uno yang digunakan untuk menghubungkan laptop atau komputer dengan mikrokontroler Arduino Uno dengan sebuah kabel USB A-B. *USB Connection* berfungsi untuk mengunggah program dari laptop atau komputer ke mikrokontroler [1].



Gambar 2.2. Pin Arduino ATMEGA 328

Pin Digital I/O (2) memiliki fungsi untuk menghubungkan komponen-komponen digital pada Arduino. Terdapat 14 pin digital pada Arduino, 6 diantaranya adalah digital *pwm output*. VCC Pin (3) adalah pin yang memiliki fungsi untuk menghubungkan pin *vcc* dari komponen (seperti sensor) ke mikrokontroler Arduino. Ground Pin (4) adalah Pin yang memiliki fungsi untuk menghubungkan pin *ground (GND)* dari komponen (seperti rangkaian sensor) ke mikrokontroler Arduino. Analog input (5) adalah Pin yang memiliki fungsi untuk menghubungkan pin *analog* pada mikrokontroler Arduino dengan inputan dari komponen. Reset button (6) adalah sebuah tombol untuk mengirim sinyal *reset* terhadap program yang telah ditanamkan pada mikrokontroler. *Power adaptor* (7) berfungsi untuk memberikan daya pada mikrokontroler Arduino ketika mikrokontroler sedang tidak terhubung dengan laptop atau computer.

2.2 Arduino GSM Shield

Arduino GSM Shield adalah modul tambahan pada arduino yg sudah dilengkapi dengan Shield GSM/GPRS. Shield ini berfungsi untuk menghubungkan Arduino ke internet menggunakan jaringan GPRS, dengan cara memasukan kartu GSM yang sudah dipaket data ke operator GSM tersebut [2]. Bentuk lebih jelasnya tentang GSM Shield ditunjukkan pada Gambar 2.3.

Fitur Arduino GSM *Shield* adalah:

- *Quad-Band* 850/ 900/ 1800/ 1900 MHz
- *GPRS Multi-slot class* 10/8
- *GPRS Mobile station class* B
- *Compliant to GSM phase* 2/2+
- *Class* 4 (2 W @850/ 900 MHz)
- *Class* 1 (1 W @ 1800/1900MHz)
- *Low Power Consumption*: 1.5mA (*sleep mode*)
- *Control* via AT Commands (GSM 07.07 ,07.05 and SIMCOM enhanced AT Commands)
- *Operation Temperature*: -40°C to +85 °C

Dalam sistem ini GSM Shield digunakan sebagai media penerima dan pengirim pesan melalui SMS yang dikirimkan oleh perangkat Android yang kemudian pesan tersebut diolah oleh perangkat mikrokontroler Arduino.



Gambar 2.3. Arduino GSM Shield

2.3 Sensor Arus ACS712

ACS712 adalah sensor arus yang bekerja berdasarkan efek medan. Sensor arus ini dapat digunakan untuk mengukur arus AC atau DC. Modul sensor ini telah dilengkapi dengan rangkaian penguat operasional, sehingga sensitivitas pengukuran arusnya meningkat dan dapat mengukur perubahan arus yang kecil. Sensor ini digunakan pada aplikasi-aplikasi di bidang industri, komersial, maupun komunikasi. Contoh aplikasinya antara lain untuk sensor kontrol motor, deteksi dan manajemen penggunaan daya, sensor untuk catu daya tersaklar, sensor proteksi terhadap arus lebih, dan lain sebagainya (Gambar 2.4).



Gambar 2.4. Sensor Arus ACS712

Spesifikasi Sensor Arus ACS712 :

1. Berbasis ACS712 dengan fitur:

- Waktu kenaikan perubahan luaran = 5 μ s.
 - Lebar frekuensi sampai dengan 80 kHz.
 - Total kesalahan luaran 1,5% pada suhu kerja $T_A = 25^\circ\text{C}$.
 - Tahanan konduktor internal 1,2 $\text{m}\Omega$.
 - Tegangan isolasi minimum 2,1 kVRMS antara pin1-4 dan pin 5-8.
 - Sensitivitas luaran 185 mV/A.
 - Mampu mengukur arus AC atau DC hingga 5 A.
 - Tegangan luaran proporsional terhadap masukan arus AC atau DC.
2. Tegangan kerja 5 VDC.
 3. Dilengkapi dengan penguat operasional untuk menambah sensitivitas keluaran

Sensor ini memiliki pembacaan dengan ketepatan yang tinggi, karena di dalamnya terdapat rangkaian offset rendah linier medan dengan satu lintasan yang terbuat dari tembaga. Cara kerja sensor ini adalah arus yang dibaca mengalir melalui kabel tembaga yang terdapat didalamnya yang menghasilkan medan magnet yang ditangkap oleh IC medan terintegrasi dan diubah menjadi tegangan proporsional. Ketelitian dalam pembacaan sensor dioptimalkan dengan cara pemasangan komponen yang ada di dalamnya antara penghantar yang menghasilkan medan magnet dengan transducer medan secara berdekatan.

Hambatan dalam penghantar sensor sebesar 1,2 $\text{m}\Omega$ dengan daya yang rendah. Jalur terminal konduktif secara kelistrikan diisolasi dari sensor timah mengarah (pin 5 sampai pin 8). Hal ini menjadikan sensor arus ACS712 dapat digunakan pada aplikasi-aplikasi yang membutuhkan isolasi listrik tanpa menggunakan opto-isolator atau teknik isolasi lainnya yang mahal. IC ACS712 tipe 5A IC ini mempunyai sensitivitas sebesar 185mV/A. Saat arus yang mengalir 0A IC ini mempunyai output tegangan 2,5V. Nilai tegangan akan bertambah berbanding lurus dengan nilai arus.

2.4 Sensor Tegangan ZMPT101B

Sensor tegangan ZMPT101B Adalah sebuah sensor yang berfungsi untuk mengukur tegangan AC 220 V dengan tegangan keluaran DC yang dapat dihubungkan dengan mikrokontroler secara langsung (Gambar 2.5). Sensor tegangan ZMPT101B dilengkapi dengan trimpot multiturun untuk mendapatkan ketepatan pengukuran yang paling baik. Sensor arus diperlukan dalam sistem ini, karena mengingat keluaran tegangan listrik tidak stabil [3].



Gambar 2.5. Sensor Tegangan ZMPT101B

2.5 Relay

Relay adalah komponen elektronika berupa saklar elektronik yang digerakkan oleh arus listrik. Secara prinsip *relay* merupakan tuas saklar dengan lilitan kawat pada batang besi (*solenoid*) di dekatnya. Ketika solenoid dialiri arus listrik, tuas akan tertarik karena adanya gaya magnet yang terjadi pada solenoid sehingga kontak saklar akan menutup (Gambar 2.6).

Pada saat arus dihentikan, gaya magnet akan hilang, tuas akan kembali ke posisi semula dan kontak saklar kembali terbuka.

Relay biasanya digunakan untuk menggerakkan arus/tegangan yang besar (misalnya peralatan listrik 4 ampere AC 220 V) dengan memakai arus/tegangan yang kecil (misalnya 0.1 ampere 12 Volt DC). *Relay* yang paling sederhana ialah *relay* elektromekanis yang memberikan pergerakan mekanis saat mendapatkan energi listrik. Secara sederhana *relay* elektromekanis ini didefinisikan sebagai berikut :

- 1) Alat yang menggunakan gaya elektromagnetik untuk menutup (atau membuka) kontak saklar.
- 2) Saklar yang digerakkan (secara mekanis) oleh daya/energi listrik.

Dalam pemakaian biasanya *relay* yang digerakkan dengan arus DC dilengkapi dengan sebuah dioda yang di-parallel dengan lilitannya dan dipasang terbalik yaitu anoda pada tegangan (-) dan katoda pada tegangan (+). Ini bertujuan untuk mengantisipasi sentakan listrik yang terjadi pada saat *relay* berganti posisi dari *on* ke *off* agar tidak merusak komponen di sekitarnya.

NO (*Normally Open*) dan NC (*Normally Close*) adalah penamaan kondisi atau keadaan switch saat switch belum dipasang atau belum in-service atau belum ada aksi dari parameter yang dideteksinya. Selain NO (*Normally Open*) dan NC (*Normally Close*) ada istilah lain untuk dunia per-switch-an, NE (*Normally Energize*) dan ND (*Normally De-energize*) adalah istilah lain tersebut.



Gambar 2.6. Modul Relay

2.6 *Android*

Android adalah sistem operasi yang berbasis Linux untuk telepon seluler seperti *smartphone* dan tablet. Logo Android sendiri didefinisikan sebagai robot hijau seperti yang ditunjukkan pada Gambar 2.7. Sistem operasi Android menyediakan *platform* terbuka bagi para pengembang untuk menciptakan aplikasi mereka sendiri untuk digunakan oleh bermacam peranti bergerak [4].



Gambar 2.7. Logo Android

Tampilan OS Android di telepon seluler pun sangat bagus untuk ukuran di zaman ini, seperti contoh Gambar 2.8.



Gambar 2.8. Screenshot Tampilan OS Android

2.7 Android Studio

Android Studio adalah sebuah IDE untuk pengembangan aplikasi di platform Android. Sama seperti kombinasi antara Eclipse dan *Android Developer Tools* (ADT), Android Studio juga dapat di-download di situs resmi Android (Gambar 2.9).



Gambar 2.9. Logo Aplikasi Android Studio

2.8 *Arduino Software* (IDE)

Arduino Software (IDE) merupakan perangkat lunak yang digunakan untuk menulis kode sumber dan dapat diunggah langsung ke *board* Arduino. Tampilan Logo Arduino IDE tertera pada Gambar 2.10. *Arduino Software* (IDE) merupakan perangkat lunak yang digunakan untuk menulis *code* dan dapat diunggah langsung ke *board* Arduino.

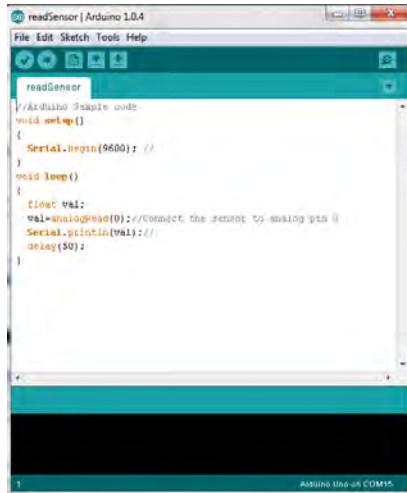


Gambar 2.10. Logo Aplikasi Arduino IDE

Perangkat lunak untuk Arduino, lazim disebut “sketches“, sebagai sketsa untuk menciptakan program untuk ditanamkan di Arduino. *Software* untuk menciptakan dijanlakan di sebuah PC (Windows, Linux dan Mac OS X) berupa lingkungan pengembangan terpadu (IDE) untuk Arduino. IDE memungkinkan Anda untuk menulis dan mengedit kode dan mengubah kode ini ke dalam instruksi yang dapat dimengerti oleh hardware Arduino [5]. IDE juga melakukan transfer instruksi tersebut ke papan sirkuit dengan proses yang disebut “upload“. *Arduino software* ini banyak digunakan karena gratis dan mudah untuk dioperasikan.

Pada Gambar 2.11 menunjukkan terdapat beberapa fitur yang disediakan Arduino. Fitur-fitur tersebut terdapat pada *toolbar* dan *menubar* halaman IDE. IDE mampu memeriksa kebenaran sketsa yang dibuat pengguna. IDE akan juga akan menunjukkan kesalahan dalam penulisan kode program. Untuk melakukan pemeriksaan kode program dilakukan dengan menekan *icon* pada *toolbar* pojok kiri. Pemeriksaan kode program dilakukan juga sebelum kode diunggah ke mikrokontroler Arduino.

Bahasa pemrograman yang digunakan untuk penulisan kode adalah C/C++ yang telah dimodifikasi dan disederhanakan. IDE dapat mengunggah kode program yang ditulis ke dalam mikrokontroler Arduino, sehingga mikrokontroler dapat menjalankan fungsi sesuai yang direncanakan *programmer*.



Gambar 2.11. Halaman Antarmuka Arduino IDE

Dalam sistem yang akan dibuat, perangkat lunak ini berguna untuk memprogram Arduino supaya dapat berfungsi

[Halaman ini sengaja dikosongkan]

BAB III

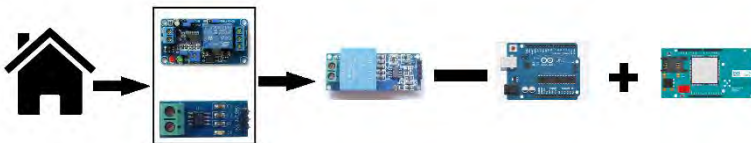
ANALISIS DAN PERANCANGAN SISTEM

Pada bab ini akan dijelaskan mengenai hal-hal yang berkaitan dengan perancangan sistem yang akan dibuat. Perancangan tersebut mencakup deskripsi umum aplikasi, arsitektur sistem, model fungsional aplikasi, spesifikasi kebutuhan aplikasi, *use case* aplikasi, serta diagram alir aplikasi.

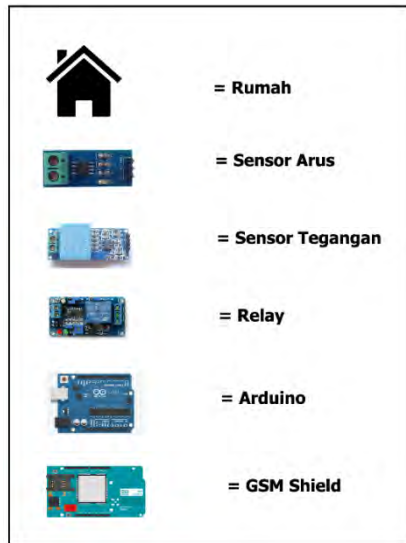
3.1 Deskripsi umum

Dalam satu rumah, akan dipasang perangkat kWh meter yang berisi sensor arus & sensor tegangan. Kemudian ditambahkan juga modul *relay* sebagai penyambung dan pemutus aliran listrik. Kemudian modul modul tersebut di hubungkan dengan mikrokontroller Arduino (Gambar 3.1-3.2).

Arduino dalam sistem ini bertugas sebagai unit pemrosesan utama (CPU) yang mengontrol *relay* dan mengolah hasil pembacaan dari sensor tegangan dan sensor arus. Lalu Arduino tersebut ditambahkan modul GSM yang bertugas sebagai media komunikasi dengan pengguna atau pemilik rumah.



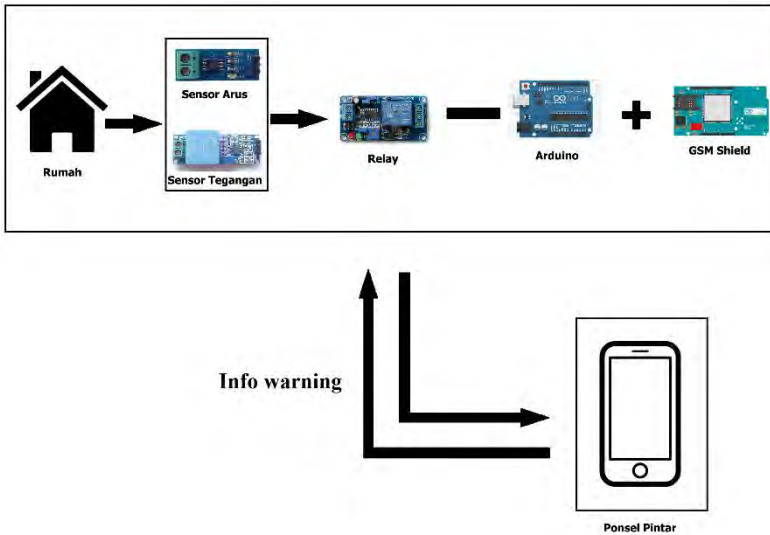
Gambar 3.1. Skema Pemasangan Perangkat



Gambar 3. 2. Keterangan Skema Pemasangan Perangkat

3.2 Arsitektur Umum Sistem

Media komunikasi ini digunakan untuk mengirim peringatan apabila melewati nilai tertentu ke pengguna. Selain itu modul GSM ini digunakan untuk menerima perintah yang dikirim oleh pengguna melalui aplikasi android (Gambar 3.3).



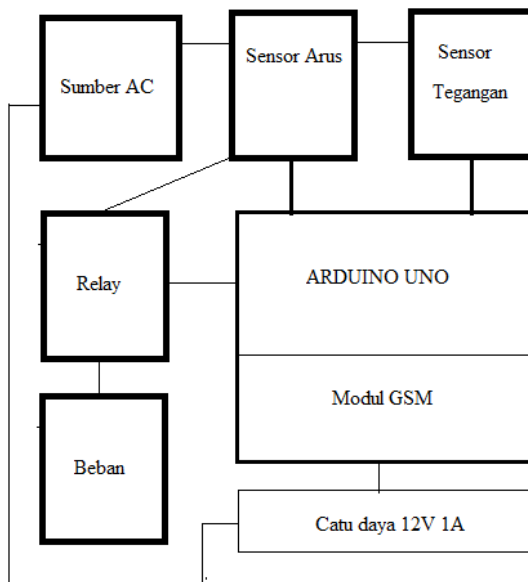
Gambar 3.3. Arsitektur Sistem

3.3 Perancangan Perangkat Keras

Pada sistem ini, perangkat keras yang digunakan adalah sebagai berikut:

- 1 buah mikrokontroler Arduino Uno Atmega 328
- 1 buah modul GSM
- 1 buah sensor arus ACS712 5A
- 1 buah sensor tegangan AC
- 1 buah *relay*
- 1 buah *smartphone* Android
- 1 buah *Sim Card*
- 1 buah catu daya DC 12 V

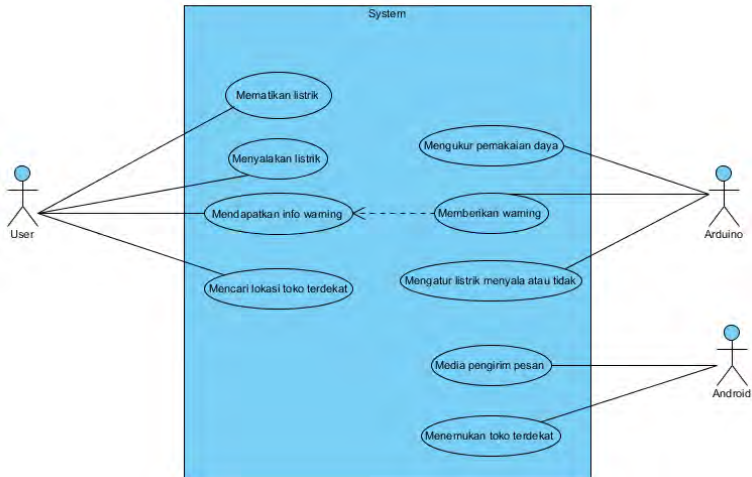
Perangkat-perangkat tersebut dirangkai seperti Gambar 3.4. Dengan mikrokontroler Arduino sebagai pusat pemrosesan. Modul GSM dipasang ke mikrokontroler Arduino dengan cara menancapkan besi port dengan lubang yang sesuai dan ditenagai oleh *power supply*. Modul ini digunakan Arduino untuk berkomunikasi dengan perangkat lain melalui jaringan GSM. Kemudian sensor arus, sensor tegangan, dan modul *relay* dihubungkan ke rangkaian Arduino dan GSM *shield*. Sensor tegangan dirangkai secara paralel dengan sumber listrik, sedangkan sensor daya dan modul *relay* dirangkai secara seri dengan sumber listrik. Hasil keluaran dari *relay* dihubungkan ke beban. Rangkaian tersebut dapat dilihat pada Gambar 3.4.



Gambar 3.4. Rancangan Perangkat Keras

3.4 Perancangan Diagram *Use Case*

Use Case diagram menggambarkan fungsionalitas sistem beserta aktor yang terlibat.



Gambar 3.5. Diagram *Use Case*

Gambar 3.5 menunjukkan bahwa sistem melibatkan 2 aktor lain selain pengguna. Sistem melibatkan mikrokontroler Arduino dan *smartphone* Android. User dapat mengatur proses kontrol listrik yang berfungsi mematikan dan menyalakan listrik, selain itu user juga akan mendapatkan info *warning*.

Lalu, mikrokontroler Arduino bertugas untuk mengukur pemakaian daya listrik dengan bantuan modul sensor arus dan sensor daya, juga bertugas untuk memberikan info *warning* kepada user, dan juga tak lupa untuk bertugas untuk mengatur menyala atau tidaknya aliran listrik yang mana hal ini dapat dilakukan dengan bantuan modul *relay*. Yang terakhir perangkat android berfungsi sebagai media pengirim pesan

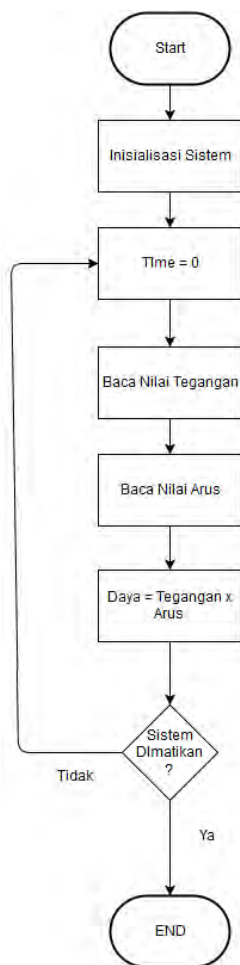
3.5 Diagram Alir Aplikasi Sistem

Alur setiap proses yang terdapat pada aplikasi digambarkan pada diagram alir, untuk memudahkan pemahaman secara garis besar proses yang dilakukan oleh sistem.

3.5.1 Diagram Alir Proses Perhitungan Daya

Pertama-tama dilakukan inisialisasi sistem, kemudian dilanjutkan dengan memastikan $time = 0$. Lalu sistem melakukan proses pembacaan nilai tegangan. Hal ini dapat dilakukan oleh arduino dan sensor tegangan. Kemudian, sistem melakukan pembacaan nilai arus. Hal ini dapat dilakukan oleh arduino dan sensor arus. Lalu didapatkan nilai daya (dalam satuan *Watt*) dari hasil perhitungan tegangan kali arus, maka didapatkan daya yang di konsumsi waktu itu. Kemudian dicari rata-rata penggunaan daya yang digunakan, dalam konteks ini penulis menggunakan waktu *sampling* selama 1 detik.

Ketika sudah melalui waktu *sampling* tersebut (1 detik) maka didapatkan nilai penggunaan listrik dalam satuan *watt second* yang dikonversi ke KWh. KWh didapatkan dengan membagi besarnya daya yang digunakan dengan lamanya waktu penggunaan. Waktu *sampling* yang digunakan harus dirubah menjadi satuan watt dan juga satuan daya harus dirubah menjadi satuan kilo. Perhitungan daya ini dilakukan terus menerus sampai sistem berhenti atau dimatikan, kemudian nilai tersebut diakumulasikan sehingga didapat total penggunaan daya selama sistem digunakan (Gambar 3.6).

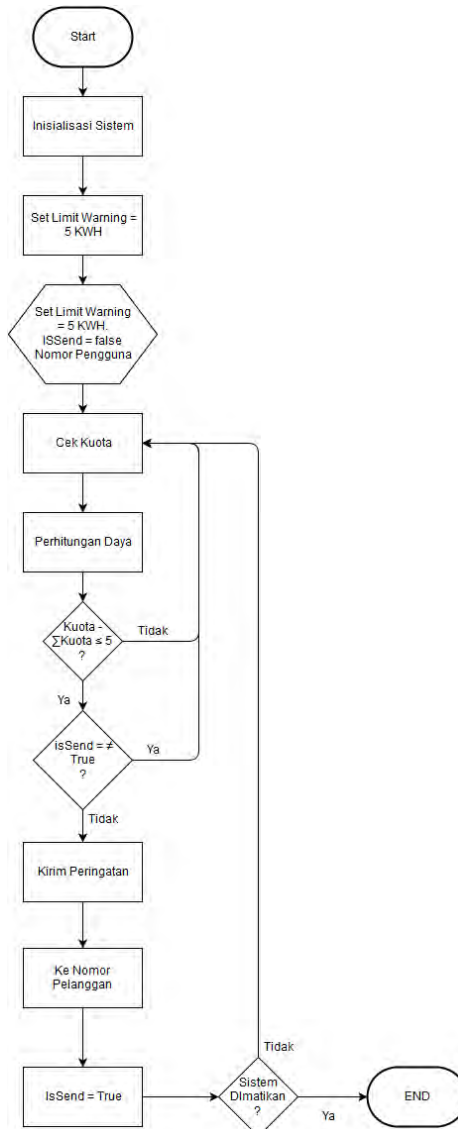


Gambar 3.6. Diagram Alir Perhitungan Daya

3.5.2 Diagram Alir Proses Pengiriman Informasi Peringatan

Proses pengiriman informasi peringatan ini diawali dengan proses inialisasi sistem, dimana sistem mengecek *port*, GSM *Shield*, limit batas warning sisa kWh, nomor pengguna (nomor yang menerima pesan peringatan), dan IS send = False (Mencegah pesan dikirim secara berkali-kali). Kemudian dilakukan cek kuota, yaitu proses membaca berapa sisa kuota listrik yang dimiliki sekarang. Setelah itu dilakukan perhitungan daya seperti yang sudah dijelaskan di sub bab 3.5.1. Setelah itu dilakukan perhitungan, apakah daya yang dikonsumsi tersebut telah mencapai atau kurang dari batas warning yang telah di set sebelumnya. Apabila belum mencapai batas warning, maka dilakukan sisa kuota secara terus menerus.

Jika telah mencapai limit yang sudah ditentukan, sistem akan memeriksa apakah telah dilakukan pengiriman pesan *warning*. Jika belum, maka sistem akan mengirimkan pesan *warning* ke nomor pengguna yang sudah ditentukan sebelumnya. Dan juga sebaliknya, Jika sudah pernah dikirimkan pesan sebelumnya maka sistem tidak akan mengirimkan pesan yang sama. Hal ini ditujukan agar sistem tidak mengirimkan pesan dalam jumlah yang berlebihan atau terus menerus saat limit penggunaan sudah terlampaui (Gambar 3.7).

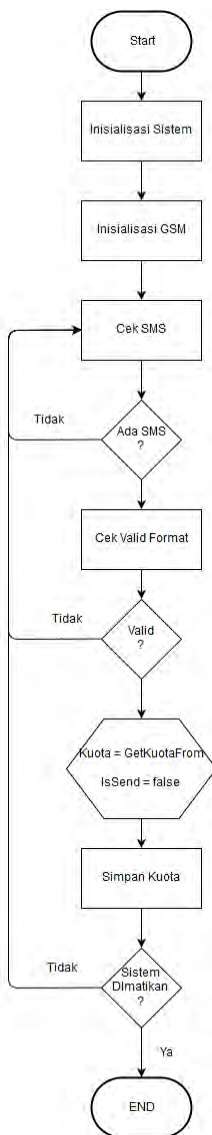


Gambar 3.7. Diagram Alir Proses Pengiriman Informasi Peringatan

3.5.3 Diagram Alir Proses Reset Kuota

Langkah pertama dalam proses reset kuota adalah melakukan inisialisasi sistem dan juga inisialisasi GSM. Kemudian sistem memeriksa apakah ada pesan yang masuk. Jika ada pesan masuk, maka sistem akan mengecek kebenaran format pesan yang masuk. Format *reset* kuota yang digunakan : `#*9#(besar kuota)#`. Jika format tidak valid, pesan itu akan diabaikan.

Jika format pesan valid, maka dari pesan tersebut diambil nilai besar kuota kemudian nilai tersebut ditambah ke nilai kuota yang tersimpan pada sistem (dalam hal ini memperbaharui nilai kuota yang tersimpan di sistem) kemudian status warning di set menjadi false. Jika sistem tidak dimatikan, maka sistem akan terus memeriksa pesan yang masuk. Proses ini berhenti ketika sistem dimatikan (Gambar 3.8).

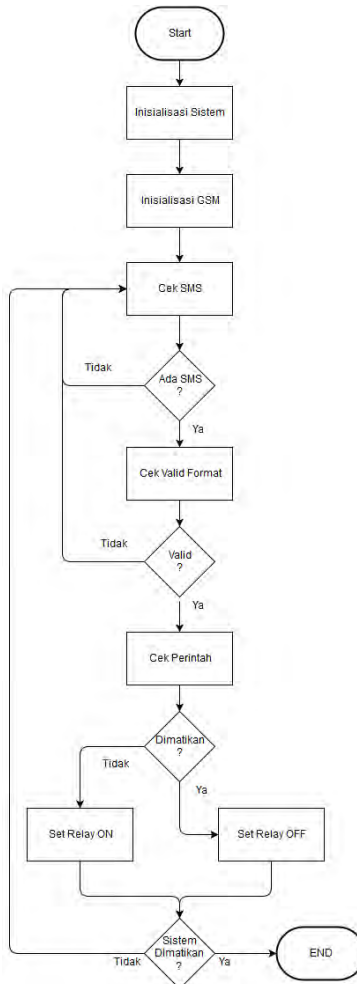


Gambar 3.8. Diagram Alir Proses Reset Kuota

3.5.4 Diagram Alir Proses Kontrol Listrik

Langkah dalam proses kontrol listrik diawali dengan proses inialisasi sistem, yang kemudian diikuti dengan proses inialisasi GSM. Kemudian sistem memeriksa apakah ada pesan masuk. Jika ada pesan masuk, sistem akan mengecek kebenaran format pesan yang masuk. Ada pun format yang digunakan adalah `#*0#` untuk mematikan, dan `#*1#` untuk menyalakan.

Jika valid, sistem akan memeriksa pesan yang dikirim. Jika perintahnya “0” maka *relay* akan memutuskan aliran listrik. Dan jika perintahnya “1” maka *relay* akan menyambung aliran listrik. Jika sistem tidak dimatikan, maka sistem akan terus memeriksa pesan yang masuk. Proses ini berhenti ketika sistem dimatikan (Gambar 3.9).



Gambar 3.9. Diagram Alir Proses Kontrol Listrik

3.6 Rancangan Antarmuka Sistem

Antarmuka merupakan perantara antara pengguna dengan sistem. Tampilan antarmuka sangat mempengaruhi penggunaan suatu sistem, oleh karena itu antarmuka harus dirancang sesederhana mungkin sehingga memudahkan pengguna dalam menggunakan sistem tersebut. Pada tahap ini akan dilakukan perancangan antarmuka sistem yang akan digunakan.

3.6.1 Antarmuka Aplikasi Android

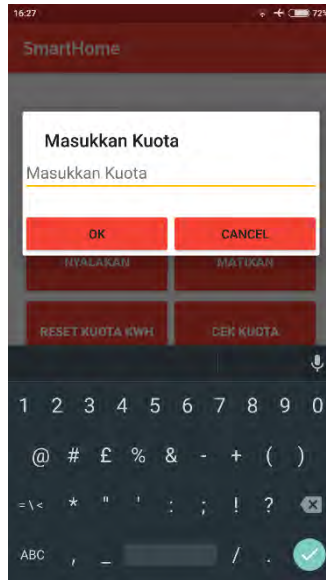


Gambar 3.10. Antar Muka Utama Program

Aplikasi android terdiri dari beberapa menu yang berguna untuk melakukan suatu fungsi tertentu. Menu-menu tersebut dapat dilihat pada Gambar 3.10, yaitu menu nyalakan yang berfungsi untuk mengirim perintah menyalakan perangkat, menu matikan

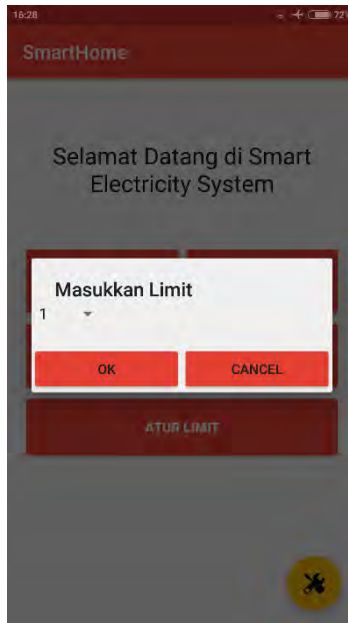
yang berfungsi untuk mengirim mematikan menyalakan perangkat, reset kuota yang berfungsi untuk mengirim perintah mengubah kuota pulsa listrik, cek kuota berfungsi untuk mengirim perintah untuk mengecek banyak kuota pulsa pada perangkat dan atur limit yang berfungsi untuk mengirim perintah untuk mengatur limit peringatan batas saldo kuota yang tersisa.

Untuk mengirimkan besar kuota pada menu reset kuota dapat dilihat pada antar muka Gambar 3.11.



Gambar 3.11. Antar Muka Pengaturan Reset Kuota

Untuk pengaturan limit peringatan kuota terendah dapat dilihat pada antar muka Gambar 3.12).



Gambar 3.12. Antar Muka Pengaturan Limit

BAB IV IMPLEMENTASI

Pada bab ini akan dibahas mengenai implementasi sistem pada perangkat keras dan perangkat lunak yang telah dirancang pada perancangan sistem yang telah dibahas pada bab 3.

4.1 Lingkungan Implementasi

Pada implementasi sistem diperlukan beberapa perangkat pendukung, yaitu :

4.1.1 Perangkat Keras

Perangkat keras yang digunakan dalam implementasi sistem adalah perangkat laptop, perangkat Android, mikrokontroler Arduino , modem SIM900 dan perangkat pendukung lainnya. Spesifikasi dari perangkat-perangkat tersebut adalah sebagai berikut:

- Laptop Asus A42JC, Intel Core i5 processor 430M (2.3 GHz, 667 MHz FSB, 3 MB L2 cache)
- Arduino Uno, Atmega328P dengan 32 KB *flash* memori dan 2 KB memori SRAM
- GSM Shield, SIM900 GSM/GPRS SHIELD ICOM SAT Quad-Band 850/900/1800/1900MHz
- Smartphone Xiaomi MI4 LTE
- Catu daya 12V, 2A
- Terminal listrik 4 pole
- Beban Fan 60W

4.1.2 Perangkat Lunak

Spesifikasi perangkat lunak yang digunakan dalam implementasi sistem adalah sebagai berikut:

- Microsoft Windows 7 32-bit sebagai sistem operasi PC/ laptop
- Android OS, v6.0 (Marshmallow) sebagai sistem operasi Android
- Android Studio sebagai IDE untuk mengimplementasikan aplikasi Android
- Java JDK 1.7
- Microsoft visio 2013 untuk merancang diagram alir
- Arduino IDE 1.6.2 sebagai ide untuk mengimplementasikan sistem Arduino.

4.2 Implementasi Perangkat Keras

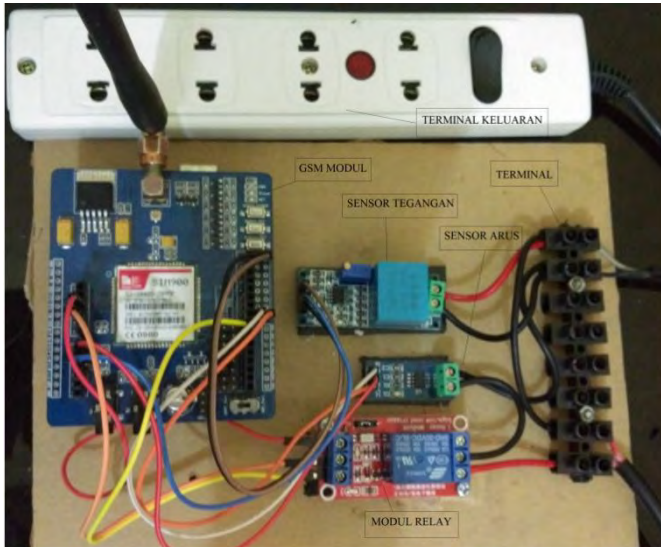
Implementasi perangkat keras ini diawali dengan sebuah perakitan perangkat Arduino, modem SIM900, modul *relay*, sensor arus dan sensor tegangan pada sebuah papan akrilik. Semua sesnor dihubungkan ke masukan analog Arduino dan masukan *relay* dihubungkan ke keluaran digital Arduino. Modul sim 900 dihubungkan ke arduino *plugin* pada terminal arduino. Power supply 12v dihubungkan pada terminal catu daya Arduino.

Perangkat keras yang digunakan oleh sistem ini sebagai berikut:

- 1 buah mikrokontroler Arduino Uno Atmega 328
- 1 buah modul GSM
- 1 buah Power supply 12/2A
- 1 buah sensor arus
- 1 buah sensor tegangan
- 1 buah *relay*
- 1 *Sim Card* untuk perangkat *modem*
- 1 buah *smartphone* Android
- 1 buah terminal listrik

- 1 buah Fan 60 W att sebagai beban

Semua perangkat dirangkai seperti pada Gambar 4.1 berikut.



Gambar 4. 1. Rangkaian Perangkat Keras

Ada pun biaya yang dihabiskan untuk membeli perangkat keras yang digunakan dalam sistem ini adalah sebagai berikut:

Mikrokontroler Arduino Uno Atmega 328	Rp 160.000
Modul GSM	Rp 550.000
Power supply 12/2A	Rp 20.000
Sensor arus	Rp 70.000
Sensor tegangan	Rp 120.000
Relay	Rp 35.000
Sim Card untuk perangkat modem	Rp 20.000
Smartphone Android	Rp 1.500.000
Terminal listrik	Rp 10.000
Total	RP 2.485.000

4.3 Implementasi Proses

Pada sub bab ini terdiri dari penjelasan tentang proses yang terjadi pada sistem monitoring pemakaian daya listrik ini. Sistem proses terdiri dari 2 bagian sistem proses pada perangkat Arduino dan sistem proses pada perangkat Android. Proses pada Arduino terdiri dari beberapa langkah, yaitu :

1. Reset kuota daya listrik
2. Limit peringatan pemakaian daya listrik
3. Cek kuota listrik
4. Mematikan sambungan keluaran listrik
5. Menyalakan sambungan keluaran listrik
6. Mematikan sambungan keluaran jika kuota listrik = 0

Sementara itu untuk proses pada perangkat Android terdiri dari beberapa proses, yaitu :

1. Mengirim pesan untuk menyalakan sambungan keluaran listrik
2. Mengirim pesan untuk mematikan sambungan keluaran listrik
3. Mengirim pesan untuk mengatur limit peringatan pemakaian daya listrik
4. Mengirim pesan untuk mereset kuota listrik
5. Mengirim pesan untuk mengetahui besar kuota listrik sekarang

4.3.1 Implementasi kondisi awal sistem

Untuk implementasi awal sistem perlu dilakukan inisialisasi variabel yang mewakili kondisi-kondisi sistem awal mula dijalankan. Berikut inisialisasi awal pada program Arduino (Gambar 4.2).

```
boolean SMS_REPLY_KUOTA,SMS_REPLY_LIMIT;  
unsigned long KUOTA_LISTRIK;  
unsigned long timer_rtc,wait_send;  
double temp_voltage,temp_current;  
float voltage_average,current_average;  
int mVperAmp = 185; 30A Module  
double Voltage = 0;  
double VRMS = 0;  
double AmpsRMS = 0;  
double WATT_SECOND;  
float KILO_WATT_HOUR;  
double COS_PHI = 1;  
float SISA_KWH;  
int i,k;  
boolean WARNING_LOW_DEPOSITE_NULL;  
boolean RUN_OUPUT=true,CHECK_KUOTA,SEND_SMS;
```

Gambar 4.2. Pseudocode Inisialisasi Awal Arduino

Inisialisasi variabel dapat berubah jika kondisi yang tersimpan dalam eeprom Arduino memenuhi suatu kondisi tertentu. Misal variabel kuota listrik saat ini sama dengan nilai limit peringatan, oleh sebab itu nilai SMS_REPLY_KUOTA akan bernilai *true*. Setelah pesan dikirimkan maka variabel SMS_REPLY_KUOTA akan kembali bernilai *false*.

4.3.2 Implementasi kondisi sistem telah berjalan

Untuk implementasi sistem yang telah berjalan ada dua proses yang secara berulang dilakukan, mengukur penggunaan daya listrik sekaligus meng-*update* kuota pada memori dan membaca pesan yang masuk. Proses pengukuran penggunaan daya listrik dilakukan dengan interval setiap 1 detik. Untuk mendapatkan pembacaan daya listrik setiap 1 detik maka proses yang digunakan adalah mengubah pembacaan daya listrik dalam kilowatt hour ke dalam watt second. Akumulasi data pemakaian daya listrik akan dihitung dalam watt second. Berikut kode untuk proses pembacaan daya listrik (Gambar 4.3).

```

sensorValue1=sensorValue;
sensorValue = analogRead(PIN_VOLTAGE_SENSOR);
if (sensorValue>sensorValue1 && sensorValue>511){
    climbhill=1;
    VmaxD=sensorValue;
}
if (sensorValue<sensorValue1 && climbhill==1){
    climbhill=0;
    VmaxD=sensorValue1;
    VeffD=VmaxD/sqrt(2);
    Veff=((VeffD-420.76)/-90.24)*-210.2+210.2;
if(k < 1000){
    readValue = analogRead(PIN_CURRENT_SENSOR);
    if (readValue > maxValue) { maxValue = readValue; }
    if (readValue < minValue) { minValue = readValue; }
    k++;
} else {
    k = 0;
    Voltage = ((float)maxValue - (float)minValue)/1024.0;
    VRMS = (Voltage/2.0) *0.707;
    AmpsRMS = (VRMS * 5000)/mVperAmp;
    if(AmpsRMS<0.07)AmpsRMS = 0;
}

```

```

        KILO_WATT_HOUR    =    (AmpsRMS    *
voltage_average);
    if(KILO_WATT_HOUR > 0){
        if(millis()-timer_delay > 5000){
            EEPROMWritelong(0,KUOTA_PULSA-
            KILO_WATT_HOUR); } }
    Serial.print("Voltage:");
    Serial.print(voltage_average);
    Serial.print(" Current :");
    Serial.print(AmpsRMS,3);
    Serial.print(" Power use :");
    Serial.print(KILO_WATT_HOUR,3);
    Serial.print(" Watt Second");
    Serial.print(" Sisa Pulsa:");
    Serial.print(KUOTA_PULSA/3600000,6);
    Serial.print(" KWH");
    Serial.print(" Set limit:");
    Serial.print(DATA_SET/3600000,6);
    Serial.println(" KWH");
    maxValue = 0;
    minValue = 1024;
}
timer 1sec = micros();

```

Gambar 4.3. Pseudocode Proses Pembacaan Daya Listrik

VRMS adalah nilai *root mean square* dari V. Lalu 0.707 adalah konstanta yang digunakan untuk menghitung VRMS. AmpsRMS adalah nilai *root mean square* dari I. Lalu nilai konstanta mVperAmp untuk sensor tegangan 20 A adalah 100.

Perlu dilakukan inialisasi variabel yang mewakili kondisi-kondisi sistem awal mula dijalankan. Berikut inialisasi awal pada program Arduino.

Implementasi sistem saat menerima pesan dapat dijelaskan dalam beberapa proses. Penerimaan pesan yang dikirimkan dari perangkat Android dapat dijelaskan dalam beberapa proses, yaitu :

1. Perintah untuk mematikan sambungan keluaran listrik (Gambar 4.4).

Pesan sms yang diterima Arduino diperoleh dari serial komunikasi data dengan modem. Data hasil buffer dari serial komunikasi di parse atau dipecah-pecah untuk mendapatkan data pesan.

```

if(GSM_SHIELD.available(>0){
  String SERIAL_INCOMING =
  GSM_SHIELD.readStringUntil('\n');
  Serial.println(SERIAL_INCOMING);
  if(SERIAL_INCOMING.charAt(0)!='+'||
  SERIAL_INCOMING.charAt(0)!='*'){
    if(SERIAL_INCOMING.charAt(0)!='A' &&
    SERIAL_INCOMING.charAt(1)!='T'){
      if(SERIAL_INCOMING.charAt(0)!='\t'){
        char BUFFER[50];
        for(int c=0;c<SERIAL_INCOMING.length()-
        1;c++){ BUFFER[c]=SERIAL_INCOMING.charAt(
        c);
          }
        if(BUFFER[0]=='#' && BUFFER[1]=='*' &&
        BUFFER[2]=='0' && BUFFER[3]=='#') {
          RUN_OUPUT = false;
          SEND_SMS = true;
        }
      }
    }
  }
}

```

Gambar 4.4. Pseudocode Memutuskan Sambungan Listrik

2. Perintah untuk menyalakan sambungan keluaran listrik ditunjukkan pada kode berikut (Gambar 4.5).

```
if(BUFFER[0]=='#' && BUFFER[1]=='*' &&
  BUFFER[2]=='1' && BUFFER[3]=='#') {
    RUN_OUPUT = true;
    SEND_SMS = true;
}
```

Gambar 4.5. Pseudocode Menyalakan Sambungan Listrik

3. Perintah untuk cek kuota listrik ditunjukkan pada kode berikut (Gambar 4.6).

```
if(BUFFER[0]=='#' && BUFFER[1]=='*' &&
  BUFFER[2]=='2' && BUFFER[3]=='#') {
    CHECK_KUOTA = true;
}
```

Gambar 4.6. Pseudocode Cek Kuota

4. Perintah untuk reset kuota listrik ditunjukkan pada kode berikut (Gambar 4.7).

```

if(BUFFER[0]=='#' && BUFFER[1]=='*' &&
  BUFFER[2]=='9' && BUFFER[3]=='#') {
  for(int      c=4;c<SERIAL_INCOMING.length()-
  2;c++){
  KUOTA+=
  String(SERIAL_INCOMING.charAt(c)); }
  KUOTA_LISTRIK= KUOTA.toInt()*3600000;
  SMS_REPLY_KUOTA = true;

  EEPROMWritelong(0,KUOTA_LISTRIK);
  }

```

Gambar 4.7. Pseudocode Reset Kuota

5. Perintah untuk set limit ditunjukkan pada kode berikut (Gambar 4.8).

```

String DATA_SETTING;
  if(BUFFER[0]=='#' && BUFFER[1]=='*'
  && BUFFER[2]=='8' && BUFFER[3]=='#') {
  for(int
  c=4;c<SERIAL_INCOMING.length()-2;c++){
  DATA_SETTING+=
  String(SERIAL_INCOMING.charAt(c)); }
  DATA_SET = DATA_SETTING.toInt()*3600000;
  SMS_REPLY_LIMIT = true;
  EEPROMWritelong(10,DATA_SET);
  WARNING_LOW_DEPOSITE_0 = false;
  }

```

Gambar 4.8. Pseudocode Set Limit

Untuk implementasi pada perangkat Android dapat dijelaskan dalam beberapa proses yaitu :

1. Pengiriman pesan menyalakan perangkat.
Pada proses pengiriman pesan dapat dijelaskan pada kode Android berikut (Gambar 4.9).

```
nyalakan.setOnClickListener(new
View.OnClickListener() {
@Override
public void onClick(View v) {
if(phoneNumber.equals("")) {
Toast.makeText(MainActivity.this, "Please set device
number in setting", Toast.LENGTH_LONG).show();
} else {
sendSMS(phoneNumber, "#*1#");
}
}
});
```

Gambar 4.9 Pseudocode Pengiriman Pesan Menyalakan Perangkat

2. Pengiriman pesan mematikan perangkat.
Pada proses pengiriman pesan dapat dijelaskan pada kode Android berikut (Gambar 4.10).

```
matikan.setOnClickListener(new
View.OnClickListener() {
@Override
public void onClick(View v) {
if(phoneNumber.equals("")) {
Toast.makeText(MainActivity.this, "Please set device
number in setting", Toast.LENGTH_LONG).show();
} else {
sendSMS(phoneNumber, "#*0#");
}
}
});
```

Gambar 4.10 *Pseudocode* Pengiriman Pesan Mematikan Perangkat

3. Pengiriman pesan cek kuota perangkat.
Pada proses pengiriman pesan dapat dijelaskan pada kode Android berikut (Gambar 4.11).

```
cekKuota.setOnClickListener(new
View.OnClickListener() {
@Override
public void onClick(View v) {
if(phoneNumber.equals("")) {
Toast.makeText(MainActivity.this, "Please set device
number in setting", Toast.LENGTH_LONG).show();
} else {
sendSMS(phoneNumber, "#*2#");
}
}
});
```

Gambar 4.11 *Pseudocode* Pengiriman Pesan Cek Kuota

4. Pengiriman pesan reset kuota perangkat.
Pada proses pengiriman pesan dapat dijelaskan pada kode Android berikut (Gambar 4.12).

```
btnOk.setOnClickListener(new
View.OnClickListener(){
@Override
public void onClick(View v)
{
if(inputUser.getText().toString().equals("")){
    Toast.makeText(getApplicationContext(),
    "Please enter new quota",
    Toast.LENGTH_LONG).show();
} else {
    kuota = inputUser.getText().toString();
    sendSMS(phoneNumber,
    "#*9#+kuota+#");
    dialog.dismiss();
}
}
});
```

Gambar 4.12 Pseudocode Pengiriman Pesan Reset Kuota

5. Pengiriman pesan set limit peringatan perangkat.
Pada proses pengiriman pesan dapat dijelaskan pada kode Android berikut (Gambar 4.13).

```
btnOk.setOnClickListener(new  
View.OnClickListener()  
{  
    @Override  
    public void onClick(View v){  
        String limit =  
String.valueOf(spinner.getSelectedItem());  
        sendSMS(phoneNumber,"#*8#"+limit+"#");  
        dialog.dismiss();  
    }  
});
```

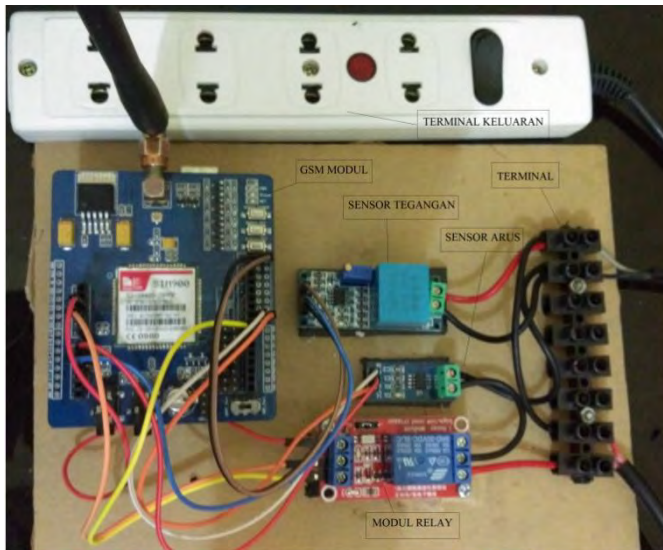
Gambar 4.13 Pseudocode Pengiriman Pesan Set Limit

4.4 Implementasi Sistem

Setelah dilakukan analisis dan perancangan, proses berikutnya adalah implementasi sistem sesuai dengan perancangan yang telah dibuat pada tahap sebelumnya.

4.4.1 Implementasi Sistem *Smart Home*

Implementasi sistem yang telah dibuat dilakukan dengan cara melakukan pengujian perangkat keras dan perangkat lunak secara terpadu. Pengujian dilakukan untuk mendapatkan hasil dari sistem keseluruhan apakah telah berjalan sesuai dengan tujuan pembuatan sistem *smart home*. Pengujian dilakukan dengan menyusun semua perangkat dan menghubungkan tiap perangkat sesuai dengan perancangan (Gambar 4.14).



Gambar 4.14. Rangkaian *Smart Home* Secara Keseluruhan

[Halaman ini sengaja dikosongkan]

BAB V

UJI COBA DAN ANALISA

Pada bab ini akan membahas uji coba dan evaluasi dari sistem yang telah dibuat. Pengujian dilakukan untuk mengetahui bahwa sistem telah bekerja dengan baik atau tidak. Pengujian fungsi dari sistem dilakukan dengan skenario pengujian yang sesuai dengan rumusan masalah pembuatan sistem ini.

a. Lingkungan Uji Coba

Pada sub bab ini dijelaskan mengenai proses pengujian sistem antara perangkat Arduino dengan perangkat Android. Uji coba aplikasi ini dilakukan dengan menggunakan sebuah *smartphone* Android dan 1 buah mikrokontroler Arduino beserta modul pendukung dan dirangkai sesuai dengan perancangan. Spesifikasinya adalah sebagai berikut:

- Laptop Asus A42JC
 - Spesifikasi perangkat keras
 - Intel® Core™ i5 CPU 430M @ 2.3GHz
 - RAM 2048 MB
 - Spesifikasi perangkat lunak
 - Windows 7 Home Premium 32-bit sebagai sistem operasi yang digunakan
 - Android Studio sebagai IDE pengembangan Android
 - IDE Arduino 1.6.8 sebagai perangkat untuk mengimplementasikan program mikrokontroler Arduino.

- Perangkat *smartphone* Xiaomi Mi4 LTE
 - Sistem Operasi: Android OS, v6 (Marshmallow)
 - Chipset: Qualcomm MSM8974AC Snapdragon 801
 - CPU: Quad-core 2.5 GHz Krait 400
 - GPU: Adreno 330
 - Memori internal: 16 GB
 - RAM: 2 GB
 - Speed: HSPA 7.2/5.76 Mbps, LTE Cat4 150/50 Mbps
 - WLAN: Wi-Fi 802.11 a/b/g/n/ac, hotspot

- Mikrokontroler Arduino Uno Rev3
 - Mikrokontroler: Atmega328
 - Tegangan operasi: 5V
 - Pin digital I/O : 14 (6 digunakan untuk keluaran PWM)
 - Pin masukan analog: 6
 - Flash Memory: 32 KB (Atmega328) 0.5 KB digunakan untuk *bootloader*
 - SRAM: 2 KB
 - EEPROM: 1 KB
 - Clock Speed: 16 MHz
 - *GSM Shield* (modul tambahan)

b. Skenario Uji Coba

Uji coba ini dilakukan untuk menguji apakah fungsi dari sistem yang dibuat telah sesuai dengan tujuan. Pengujian dilakukan dengan beberapa skenario uji agar dihasilkan perangkat yang benar-benar fungsional.

i. Uji Coba Fungsionalitas Perangkat Android

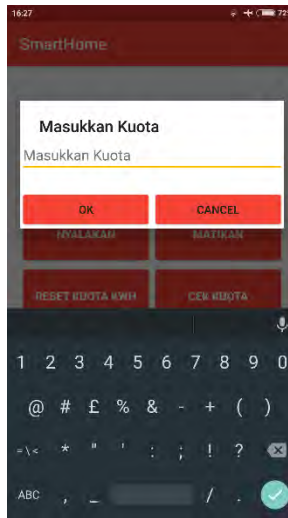
Pada uji coba fungsionalitas akan diuji semua fungsi-fungsi yang telah dibuat pada program. Uji fungsi yang dilakukan pengguna terhadap sistem pada perangkat Android.

Gambar 5.1 berikut adalah tampilan aplikasi android saat dijalankan. Pada tampilan utama berisi menu-menu yang memiliki fungsi berbeda-beda.

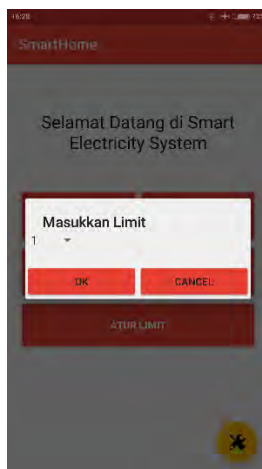


Gambar 5.1. Tampilan Menu Utama Aplikasi Android

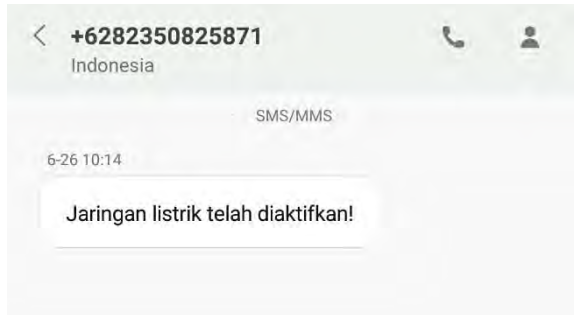
Pada program utama terdiri dari beberapa menu yang masing-masing menu dilakukan pengujian. Berikut ini adalah gambar-gambar pengujian masing-masing menu yang terdapat pada aplikasi Android (Gambar 5.2-5.5).



Gambar 5.2. Tampilan Menu Reset Kuota



Gambar 5.3. Tampilan Menu Pengaturan Limit

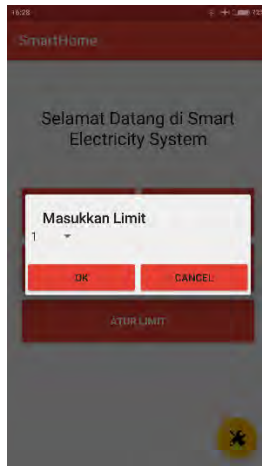


Gambar 5.4. Tampilan Pesan Balasan Dari Perangkat



Gambar 5.5. Tampilan Pesan Balasan Dari Perangkat

Pada pengujian peringatan limit kuota listrik pada perangkat Android, pertama kali sistem Android akan mengirimkan pesan pengaturan limit seperti pada Gambar 5.6.



Gambar 5.6. Tampilan Menu Pengaturan Limit

Setelah pengguna memilih variabel limit yang diinginkan sesuai dengan pilihan limit pada spinner, maka sistem akan mengirimkan pesan ke perangkat mikrokontroller Arduino dan menyimpannya ke eeprom memori internal Arduino.

Saat sistem pada Arduino dijalankan, maka proses cek daya listrik yang tersedia apakah dibawah pengaturan limit. Jika daya listrik yang tersedia telah berada dibawah batas atau limit maka sistem pada Arduino akan mengirimkan pesan ke Android berupa peringatan bahwa daya listrik mencapai limit. Berikut gambar tampilan peringatan pada Android.

Pengujian peringatan bahwa kuota listrik telah berada dibawah limit yang telah ditentukan dapat dilihat pada 5 pengujian yang dilakukan pada gambar berikut. Pengujian pengaturan limit diatur pada 1,2,3,4 dan 5 kWh (Gambar 5.7-5.11).



Gambar 5.7 Tampilan Peringatan Limit 1 Kwh



Gambar 5.8 Tampilan Peringatan Limit 2 Kwh



Gambar 5.9 Tampilan Peringatan Limit 3 Kwh



Gambar 5.10 Tampilan Peringatan Limit 4 Kwh

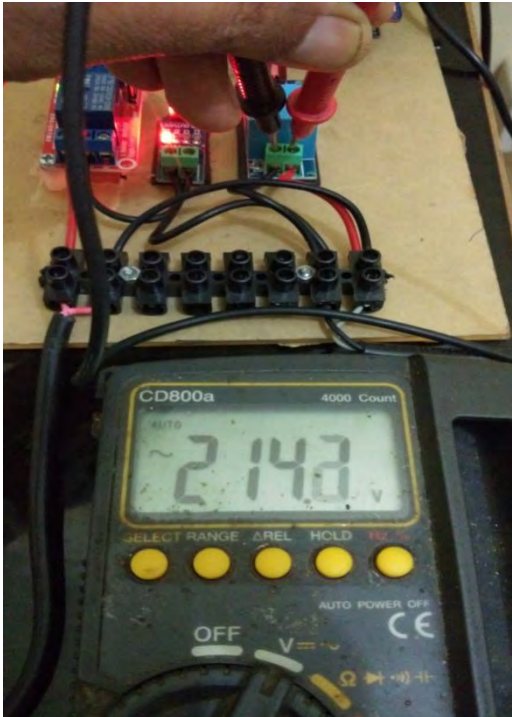


Gambar 5.11 Tampilan Peringatan Limit 5 Kwh

5.2.2 Uji Coba Fungsionalitas Perangkat Arduino

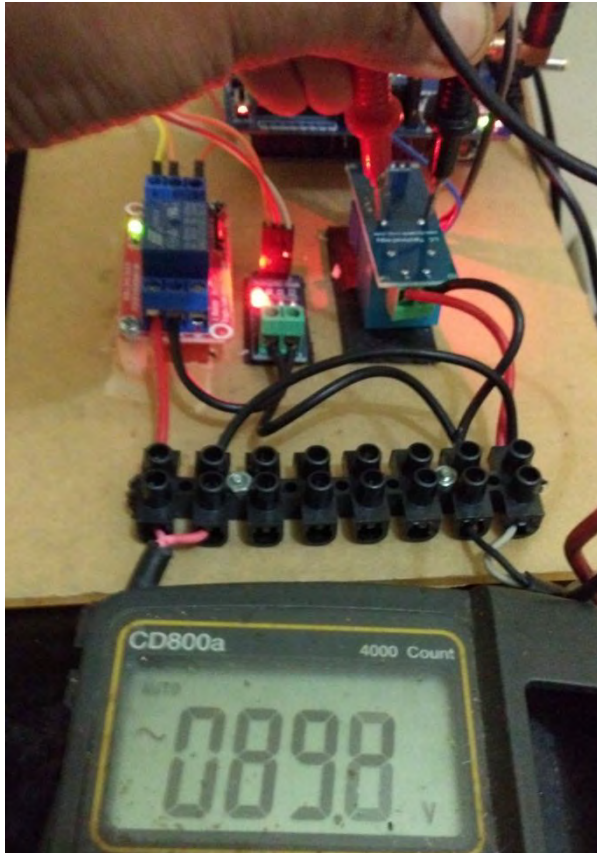
5.2.2.1 Pengujian Sensor tegangan

Pengujian sensor tegangan dilakukan untuk mengetahui apakah sensor tegangan dapat bekerja dengan baik. Sensor tegangan menerapkan trafo penurun tegangan dari tegangan sumber 220Vac ke tegangan ac 90 Vac. Pengujian dilakukan dengan menghubungkan catu daya dan sumber tegangan listrik 220V ke rangkaian. Berikut gambar pengujian dan pengukuran menggunakan voltmeter merk SANWA CD800a (Gambar 5.12).



Gambar 5.12. Pengujian Dan Pengukuran Tegangan Masukan AC Pada Sensor Tegangan

Pengukuran dan pengujian keluaran penurun tegangan dilakukan untuk membuktikan bahwa sensor pengukur tegangan dapat bekerja dengan baik. Pada keluaran trafo penurun tegangan hasil pengujian dan pengukuran menunjukkan angka pengukuran sekitar 89 VAC. Berikut gambar pengukuran pada keluaran trafo *step down* (Gambar 5.13).



Gambar 5.13. Pengujian dan Pengukuran Tegangan Keluaran Trafo *Step Down*

Berikut potongan kode program arduino yang melakukan proses perhitungan dan pengukuran tegangan AC untuk mendapatkan angka pembacaan yang mendekati angka sebenarnya (Gambar 5.14).

```

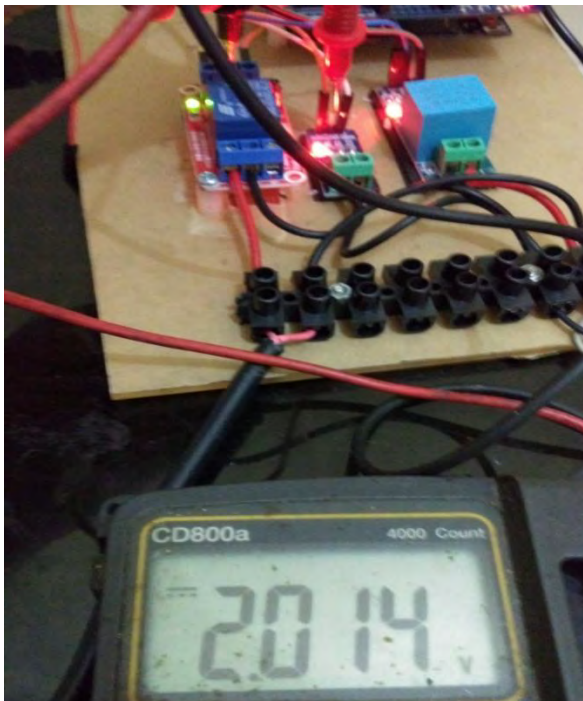
sensorValue = analogRead(PIN_VOLTAGE_SENSOR);
if (sensorValue>sensorValue1 && sensorValue>511){
  climbhill=1;
  VmaxD=sensorValue;
}
if (sensorValue<sensorValue1 && climbhill==1){
  climbhill=0;
  VmaxD=sensorValue1;
  VeffD=VmaxD/sqrt(2);
  Veff=((VeffD-420.76)/-90.24)*-210.2+210.2;
  if(i<100){
    temp_voltage+=Veff;
    i++;
  }else{
    voltage_average=(temp_voltage/118)*2;
    temp_voltage = 0;
    i = 0;
  }
  VmaxD=0;
}

```

Gambar 5.14. Pseudocode Perhitungan dan Pengukuran Tegangan Arus AC

5.2.2.2 Pengujian Sensor arus

Pengujian sensor arus dilakukan untuk mengetahui apakah sensor arus dapat bekerja dengan baik. Sensor arus menerapkan prinsip pengubah arus ke tegangan. Pengujian dilakukan dengan mengukur tegangan keluaran sensor arus dengan voltmeter DC. Pengujian dan pengukuran menggunakan voltmeter merk SANWA CD800a disajikan pada Gambar 5.15.



Gambar 5.15. Pengujian dan Pengukuran Tegangan Keluaran Sensor Arus

Berikut potongan kode program arduino yang melakukan proses pembacaan arus AC (Gambar 5.16).

```
readValue = analogRead(PIN_CURRENT_SENSOR);
if (readValue > maxValue)
{
    maxValue = readValue;
}
if (readValue < minValue)
{
    minValue = readValue;
}
Voltage = ((float)maxValue - (float)minValue)/1024.0;
VRMS = (Voltage/2.0) *0.707;
AmpsRMS = (VRMS * 5000)/mVperAmp;
KILO_WATT_HOUR = (AmpsRMS * voltage_average);
```

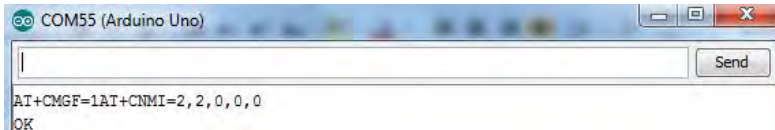
Gambar 5.16. Pseudocode Proses Pembacaan Arus AC

5.2.2.3 Pengujian Modem GSM

Pengujian modem GSM dilakukan agar diperoleh kinerja yang baik dan sesuai dengan tujuan perancangan sebagai media komunikasi melalui pesan SMS.

Pengujian modem dengan menghubungkan rangkaian melalui USB ke PC. Pada pengujian dapat diamati perubahan data melalui *hyper terminal* Arduino IDE. Jika modem merespon dengan baik maka pada terminal nampak data respon dari modem.

Respon modem saat mikrokontroler Arduino memberikan perintah untuk pengaturan modem pada modem Teks dapat dilihat pada Gambar 5.17.



Gambar 5.17. Respon Modem pada Mode Teks

Respon modem saat mendapat pesan masuk untuk mematikan sistem yang dikirim melalui Android dapat dilihat pada Gambar 5.18.

```
+CMT: "+6282350825754", "", "16/05/31,17:29:40+28"  
#*0#  
  
AT+CMGS="+6282350825754"  
> Jaringan listrik telah dimatika
```

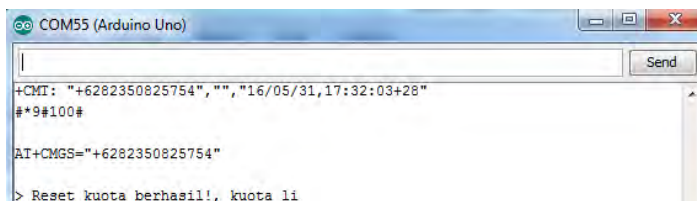
Gambar 5.18. Respon Modem Saat Mendapat Pesan Masuk

Respon modem saat mendapat pesan masuk untuk menyalakan sistem yang dikirim melalui Android dapat dilihat pada Gambar 5.19.

```
+CMT: "+6282350825754","", "16/05/31,17:31:03+28"  
#*1#  
  
AT+CMGS="+6282350825754"  
  
> Jaringan listrik telah diaktifk
```

Gambar 5.19. Respon Modem Saat Mendapat Pesan Masuk

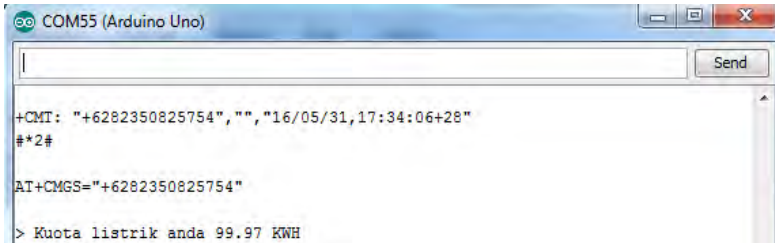
Respon modem saat mendapat pesan masuk untuk mereset kuota pulsa dapat dilihat pada Gambar 5.20.



```
COM55 (Arduino Uno)  
|  
| Send  
+CMT: "+6282350825754","", "16/05/31,17:32:03+28"  
#*9#100#  
  
AT+CMGS="+6282350825754"  
  
> Reset kuota berhasil!, kuota li
```

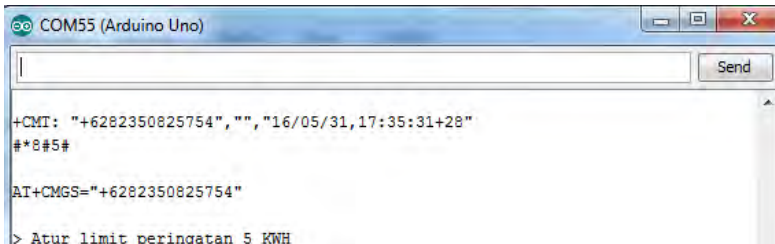
Gambar 5.20. Respon Modem Saat Mendapat Pesan Masuk

Respon modem saat mendapat pesan masuk untuk mengecek kuota pulsa dapat dilihat pada Gambar 5.21.



Gambar 5.21. Respon Modem Saat Mendapat Pesan Masuk

Respon modem saat mendapat pesan masuk untuk mengatur limit peringatan dapat dilihat pada gambar 5.22.



Gambar 5.22. Respon Modem Saat Mendapat Pesan Masuk


```

DATA_SETTING+=
String(SERIAL_INCOMING.charAt(c));
    }
    DATA_SET =
DATA_SETTING.toInt()*3600000;
    SMS_REPLY_LIMIT = true;
    EEPROMWritelong(10,DATA_SET);
    WARNING_LOW_DEPOSITE_0 = false;
    }
    /*#0#
    if(BUFFER[0]=='#' && BUFFER[1]=='*' &&
BUFFER[2]=='0' && BUFFER[3]=='#') {
        RUN_OUPUT = false;
        SEND_SMS = true;
    }
    if(BUFFER[0]=='#' && BUFFER[1]=='*' &&
BUFFER[2]=='1' && BUFFER[3]=='#') {
        RUN_OUPUT = true;
        SEND_SMS = true;
    }
    if(BUFFER[0]=='#' && BUFFER[1]=='*' &&
BUFFER[2]=='2' && BUFFER[3]=='#') {
        CHECK_KUOTA = true;
    }
    }
    }
    }
    }
}

```

Gambar 5.23. Pseudocode Memproses Pesan Yang Masuk

5.2.3 Uji Coba Kalibrasi Sistem

Uji coba kalibrasi sistem ini dilakukan dengan cara manual, yaitu dengan melakukan cek kuota dengan bantuan perangkat *smartphone* Android dan membandingkannya dengan angka yang ditampilkan oleh meteran listrik yang terpasang di tempat aliran listrik terinstalasi (Tabel 5.1-5.3).

Tabel.5.1. Uji Coba Pertama

Kuota Sistem (KWh)	Meteran Listrik (KWh)	Penggunaan daya terhitung oleh sistem (KWh)	Penggunaan daya terhitung oleh meteran (KWh)	Selisih (KWh)
10.00	7041.00	-	-	-
9.98	7041.02	0.02	0.02	-
9.95	7041.04	0.03	0.02	0.01
9.93	7041.06	0.02	0.02	-
9.90	7041.09	0.03	0.03	-
9.87	7041.11	0.03	0.02	0.01

Tabel 5.2. Uji Coba Kedua

Kuota Sistem (KWh)	Meteran Listrik (KWh)	Penggunaan daya terhitung oleh sistem (KWh)	Penggunaan daya terhitung oleh meteran (KWh)	Selisih (KWh)
10.00	7041.11	-	-	-
9.97	7041.13	0.03	0.02	0.01
9.94	7041.16	0.03	0.03	-
9.92	7041.18	0.02	0.02	-
9.89	7041.20	0.03	0.02	0.01
9.86	7041.22	0.03	0.02	0.01

Tabel 5.3. Uji Coba Ketiga

Kuota Sistem (KWh)	Meteran Listrik (KWh)	Penggunaan daya terhitung oleh sistem (KWh)	Penggunaan daya terhitung oleh meteran (KWh)	Selisih (KWh)
9.68	7041.22	-	-	-
9.67	7041.23	0.01	0.01	-
9.63	7041.25	0.04	0.02	0.02
9.59	7041.27	0.04	0.02	0.02
9.57	7041.28	0.02	0.01	0.01
9.54	7041.30	0.03	0.02	0.01

Dari ujicoba yang dilakukan, dapat dilihat bahwa perhitungan daya yang dilakukan oleh sistem dan meteran listrik memiliki perbedaan sekitar 0.01 KWh – 0.02 KWh selama uji coba dilakukan.

[Halaman ini sengaja dikosongkan]

BAB VI

KESIMPULAN DAN SARAN

6.1 Kesimpulan

Penelitian dalam tugas akhir ini memberikan beberapa kesimpulan sebagai berikut:

1. Implementasi mikorokontroler dan modem GSM sim900 dapat dimanfaatkan untuk sistem informasi pemakaian pulsa listrik
2. Pengiriman pesan pada modem tidak dapat dilakukan dengan *interval* waktu yang singkat, hal ini mengakibatkan salah satu pesan tidak dapat terkirim. Dalam pengujian penulis menggunakan *delay* 5 detik sebelum mengirim pesan berikutnya.
3. Penggunaan sistem monitoring dan pengisian daya listrik via sms dapat mempermudah untuk mengetahui pemakaian daya listrik dan pengisian pulsa daya listrik.

6.2 Saran

Berikut ini adalah beberapa saran perbaikan dan pengembangan aplikasi ini

1. Penggunaan sistem monitoring dan pengisian daya listrik secara *online* akan memperbaiki dari sisi monitoring dan pengisian pulsa daya listrik yang lebih dinamis dan *real time*.
2. Jika ingin dikembangkan untuk penggunaan perangkat listrik berdaya besar, harus dilakukan penyesuaian perangkat pada sistem.

[Halaman ini sengaja dikosongkan]

DAFTAR PUSTAKA

- [1] “Arduino,” [Online]. Available: <http://arduino.cc/en/Main/arduinoBoardUno> . [Diakses 6 February 2016].
- [2] “GSM shield,” [Online]. Available: <https://www.arduino.cc/en/Main/ArduinoGSMShield>. [Diakses 5 Maret 2016].
- [3] “Tegangan listrik naik turun | PT PLN (Persero),” [Online]. Available: <http://www.pln.co.id/riau/?p=3736>. [Diakses 5 Maret 2016].
- [4] “Android,” [Online]. Available: http://www.academia.edu/6194825/class_Research_Method_Pemograman_Android. [Diakses 5 Maret 2016].
- [5] “Arduino IDE,” [Online]. Available: <http://arduino.cc/en/main/software>. [Diakses 6 Maret 2016].

[Halaman ini sengaja dikosongkan]

BIODATA PENULIS



Muhammad Taufiq Hidayat, lahir di Palembang, pada tanggal 20 Februari 1992. Penulis menempuh pendidikan mulai dari SD II Yayasan Sosial Pendidikan PUSRI Palembang (1997-2003), SMP Yayasan Sosial Pendidikan PUSRI Palembang (2003-2006), SMA Plus Negeri 17 Palembang (2006-2009) dan selanjutnya pada tahun 2009 penulis melanjutkan pendidikan sarjana di jurusan Teknik Informatika Fakultas

Teknologi Informasi Institut Teknologi Sepuluh Nopember Surabaya (2009-2016).

Berkaitan dengan tugas akhir ini, komunikasi dengan penulis dapat melalui email: taufiq_dx@yahoo.co.id