TUGAS AKHIR – TE145561

RANCANG BANGUN JOULEMETER UNTUK MENGUKUR KONSUMSI ENERGI KENDARAAN LISTRIK

Putra Arif Wardhana
NRP 2212 030 029
Muhammad Ainul Yaqin Febrianto
NRP 2212 030 091

Dosen Pembimbing
Suwito, ST., MT.

PROGRAM STUDI D3 TEKNIK ELEKTRO
Fakultas Teknologi Industri
Institut Teknologi Sepuluh Nopember
Surabaya 2015

TUGAS AKHIR – TE145561

RANCANG BANGUN JOULEMETER UNTUK MENGUKUR KONSUMSI ENERGI KENDARAAN LISTRIK

Putra Arif Wardhana
NRP 2212 030 029
Muhammad Ainul Yaqin Febrianto
NRP 2212 030 091

Dosen Pembimbing
Suwito, ST., MT.

PROGRAM STUDI D3 TEKNIK ELEKTRO
Fakultas Teknologi Industri
Institut Teknologi Sepuluh Nopember
Surabaya 2015

FINAL PROJECT – **TE145561**

JOULEMETER DESIGN FOR MEASURING ENERGY CONSUMPTION OF ELECTRICAL VEHICLES

Putra Arif Wardhana
NRP 2212 030 029
Muhammad Ainul Yaqin Febrianto
NRP 2212 030 091

Counselor
Suwito, ST., MT.

DEPARTMENT OF ELECTRICAL ENGINEERING DIPLOME
Industrial Technology of Faculty
Sepuluh Nopember Institute of Technology
Surabaya 2015

FINAL PROJECT – **TE145561**

JOULEMETER DESIGN FOR MEASURING ENERGY CONSUMPTION OF ELECTRICAL VEHICLES

Putra Arif Wardhana
NRP 2212 030 029
Muhammad Ainul Yaqin Febrianto
NRP 2212 030 091

Counselor
Suwito, ST., MT.

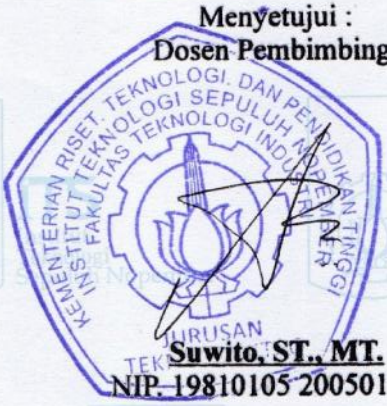
DEPARTMENT OF ELECTRICAL ENGINEERING DIPLOME
Industrial Technology of Faculty
Sepuluh Nopember Institute of Technology
Surabaya 2015

RANCANG BANGUN JOULEMETER UNTUK MENGUKUR KONSUMSI ENERGI KENDRAAN LISTRIK

TUGAS AKHIR

Diajukan Guna Memenuhi Sebagian Persyaratan
Untuk Memperoleh Gelar Ahli Madya
Pada
Bidang Studi Elektro Industri
Program Studi D3 Teknik Elektro
Fakultas Teknologi Industri
Institut Teknologi Sepuluh Nopember

Menyetujui :
Dosen Pembimbing,



Suwito, ST., MT.
NIP. 19810105 200501 004

SURABAYA
JUNI, 2015

RANCANG BANGUN JOULEMETER UNTUK MENGUKUR KONSUMSI ENERGI KENDARAAN LISTRIK

Nama Mahasiswa	**:**	**Putra Arif Wardhana**
NRP	**:**	**2212030029**
Nama Mahasiswa	**:**	**Muhammad Ainul Yaqin Febrianto**
NRP	**:**	**2211030105**
Dosen Pembimbing	**:**	**Suwito, ST., MT.**
NIP	**:**	**19810105 200501 004**

ABSTRAK

Kendaraan berbahan bakar listrik menjadi salah satu jenis kendaraan alternatif yang banyak diminati oleh masyarakat karena pemakaian bahan bakarnya yang lebih hemat dan lebih ramah dengan lingkungan. Namun dalam pemakaiannya, kendaraan listrik sering terjadi berhenti secara tiba-tiba seperti kondisi energi(bahan bakar) yang kurang terawasi. Sehingga kendaraan listrik mengalami kerusakan.

Oleh sebab itu, dalam Tugas Akhir ini kami telah membuat rancang bangun *joulemeter* untuk mengukur konsumsi energi kendaraan listrik. Arus dikonversi oleh ACS712, tegangan dikonversi oleh sensor tegangan, dan kecepatan yang dikonversi oleh sensor *reed switch*. Kemudian tegangan *output* dibaca oleh bagian ADC Arduino dan disimpan dalam bentuk variabel data sensor. Setelah pembacaan sensor selesai. Arduino mengambil data dari RTC yang berupa data waktu. Setelah mengubah data waktu tersebut kedalam teks, ditambahkan juga data sensor untuk kemudian disimpan pada *SD card* dalam bentuk *teks Excel*. Selain itu, data hasil pengolahan ditampilkan di ponsel berbasis android menggunakan *bloetooth*. Dari hasil pengujian alat yang telah dilakukan, didapat bahwa pengujian alat memiliki presentasi *error* rata-rata 20% terhadap perhitungan secara teori.

Kata kunci: *Android*, *Bloetooth*, *Data Logger*, *ACS712*

Halaman ini sengaja dikosongkan

JOULEMETER DESIGN FOR MEASURING ENERGY CONSUMTION OF ELECTRIC VEHICLES

Name of Student	**:**	**Putra Arif Wardhana**
NRP	**:**	**2212030029**
Name of Student	**:**	**Muhammad Ainul Yaqin Febrianto**
NRP	**:**	**2212030091**
Cunsultative Lucturer 1	**:**	**Suwito, ST., MT.**
NIP	**:**	**19810105 200501 004**

ABSTRACT

Vehicles fueled electricity to be one type of alternative vehicles are much in demand by the public for the use of more fuel-efficient and more environmentally friendly. But in use, electric vehicles often stop abruptly as the condition of energy (fuel) that are less supervised. So that electric vehicles were damaged.

Therefore, in this final project we have made design joulemeter to measure the energy consumption of electric vehicles. Flow converted by the ACS712, the voltage is converted by a voltage sensor, and the velocity is converted by the sensor reed switch. Then the output voltage read by Arduino ADC section and stored in the form of a variable sensor data. After the reading of the sensor is completed. Arduino took the form of data from the RTC time data. After changing the time data into the text, added also sensor data to be stored on the SD card in the form of text Excel. In addition, the data processing results displayed on android based phone using bloetooth. From the results of testing tools that have been conducted, found that testing error presentation tool has an average of 20% of the theoretical calculations.

Keywords: Android, Bloetooth, The Data Logger, ACS712

Halaman ini sengaja dikosongkan

KATA PENGANTAR

Dengan mengucapkan puji syukur kehadirat Allah SWT yang telah melimpahkan rahmat, hidayah serta karunia-Nya sehingga penulis dapat menyelesaikan Tugas Akhir dengan judul :

"RANCANG BANGUN JOULEMETER UNTUK MENGUKUR KONSUMSI ENERGI KENDARAAN LISTRIK"

Tugas Akhir ini merupakan sebagian syarat untuk menyelesaikan studi dan memperoleh gelar Ahli Madya Teknik di Jurusan D3 Teknik Elektro, Fakultas Teknologi Industri, Institut Teknologi Sepuluh Nopember.

Dengan selesainya Tugas Akhir ini penulis menyampaikan terima kasih sebesar-besarnya kepada :

1. Kedua orang tua atas limpahan doa, kasih sayang dan teladan hidup bagi penulis.
2. Bapak Suwito, ST., MT. selaku dosen pembimbing.
3. Seluruh staf pengajar dan administrasi Jurusan D3 Teknik Elektro FTI-ITS.
4. Semua pihak yang telah banyak membantu untuk menyelesaikan Tugas Akhir ini yang tidak dapat penulis sebutkan satu persatu.

Harapan kami sebagai penulis adalah semoga terselesaikannya Tugas Akhir ini dapat bermanfaat bagi kami serta pembaca. Sadar atas keterbatasan yang dimiliki oleh penulis karena hasil dari Tugas Akhir ini jauh dari kesempurnaan. Demikian penulis sudah berusaha semaksimal mungkin dan pintu maaf serta saran kritik yang membangun penulis harapkan.

Surabaya, Juni 2015

Penulis

Halaman ini sengaja dikosongkan

DAFTAR ISI

JUDUL i
PENGESAHAN iii
ABSTRAK v
ABSTRACT vii
KATA PENGANTAR ix
DAFTAR ISI xi
DAFTAR GAMBAR xv
DAFTAR TABEL xvii

BAB I. PENDAHULUAN 1
1.1. Latar Belakang 1
1.2. Permasalahan 1
1.3. Batasan Masalah 1
1.4. Maksud dan Tujuan 2
1.5. Metodologi 2
1.6. Sistematika Pembahasan 3
1.7. Relevansi 4

BAB II. TEORI PENUNJANG 5
2.1. Sistem Minimum 5
2.1.1. Mikrokontroler ATmega 328 5
2.2. Sensor Arus *ACS*7712 7
2.3. Sensor Tegangan 8
2.4. IC *Regulator* 7805 dan IC *Regulator* 7809 9
2.5. *Logging Recorder Shield Data Logger* Modul 9
2.5.1. *Real-Time Clock* (RTC) 10
2.5.2. *MMC Shield* 11
2.5.3. *SD Card* 12
2.6. *Bloetooth* 12
2.7. *Android* 13
2.7.1. Sistem Operasi *Android* 13
2.8. LCD 15
2.9. *Reed Switch* 16
2.10. *APP Investor* 17
2.10. *Arduino IDE* 19

BAB III. PERENCANAAN ALAT 21
3.1. Perancangan *Hardware* (Perangkat Keras) 21

3.1.1. Ardunino Uno 22
3.1.2. Perancangan Rangkaian LCD 23
3.1.3. Perancangan Rangkaian *Push Button* 23
3.1.4. Perancangan Rangkaian Sensor Kecepatan 23
3.1.5. Perancangan Rangkaian Sensor Tegangan 24
3.1.6. Perancangan Rangkaian Sensor Arus *ACS*712 *with OpAmp* 25
3.1.8. Perancangan *Bloetooth* 25
3.1.6. Perancangan *Data Logge*r dan RTC (*Real Time Clock*) 26
3.1.7. Perancangan Rangkaian Catu Daya 27
3.2. Perancangan *Software* (Perangkat Lunak) 27
3.2.1. Perancangan Sistem Secara Keseleuruhan 28
3.2.2. Perancangan Program *APP Inventor* 29
3.2.3. Perancangan *Software Client (Android)* 31
3.3. Perancangan Mekanik 35

BAB IV. PENGUKURAN DAN ANALISA 37
4.1. Pengujian Sumber Tegangan pada Arduino 37
4.2. Pengujian *Input/Output* Arduino Uno 38
4.3. Pengujian Pembacaan RTC 39
4.4. Pengujian Sensor *ACS*712 40
4.5. Pengujian Sensor Tegangan 42
4.6. Pengujian *Data Logger* 43
4.7. Pengujian LCD 44
4.8. Pengujian Sensor Kecepatan 44
4.9. Pengujian *Bloetooth* 45
4.10. Pengujian *Oddometer* 46
4.11. Pengujian Aplikasi *Android* 46
4.12. Pengujian Keseluruhan Sistem 47

BAB V. PENUTUP 51
5.1. Kesimpulan 51
5.2. Saran 51

DAFTAR TABEL

Tabel 3.1 *Tools* pada *Screen* Aplikasi *Android* 32
Tabel 4.1 Hasil Pengujian *Power Supply* 38
Tabel 4.2 Hasil Pengukuran Tegangan *Output* pada Pin Arduino 39
Tabel 4.3 Hasil Pengujian RTC 40
Tabel 4.4 Hasil Pengukuran Sensor Arus 41
Tabel 4.5 Hasil Pengukuran Sensor Tegangan 42
Tabel 4.6 Hasil Pengujian Sensor Kecepatan 45
Tabel 4.7 Hasil Pengujian *Bloetooth* 45
Tabel 4.8 Hasil Pengujian *Oddometer* 46
Tabel 4.9 Hasil Pengujian Waktu Pengiriman Data 47
Tabel 4.10 Hasil Pengujian Keseluruhan Sistem 49

Halaman ini sengaja dikosongkan

DAFTAR GAMBAR

Gambar 2.1 Konfigurasi Pin ATmega 328 6
Gambar 2.2 Pin Konfigurasi *ACS* 712 8
Gambar 2.3 Sensor Tegangan 9
Gambar 2.4 IC *Regulator* 7805 dan IC *Regulator* 7809 9
Gambar 2.5 Modul *Logging Recorder Shield Data Logger* 10
Gambar 2.6 Pin Konfigurasi DS1307 11
Gambar 2.7 Modul *MMC Card* 11
Gambar 2.8 *SD Card* 12
Gambar 2.9 *Bloetooth* 13
Gambar 2.10 LCD 16
Gambar 2.11 Sensor *Reed Switch* 16
Gambar 2.12 *APP Inventor Designer* 18
Gambar 2.13 Blok Editor 18
Gambar 2.14 Jendela *Arduino IDE* 19
Gambar 3.1 Diagram Fungsional Sistem 22
Gambar 3.2 Skematik Arduino Uno 22
Gambar 3.3 Skematik *Shield* LCD 23
Gambar 3.4 Skematik Rangkaian *Pull Up* Resistor 23
Gambar 3.5 Skematik Rangkaian *Reed Switch* 24
Gambar 3.6 Skematik Rangkaian Sensor Tegangan 24
Gambar 3.7 Skematik DT *Sense Current Sensor* 25
Gambar 3.8 *Wirring* Modul *Bloetooth* HC 05 26
Gambar 3.9 Skematik Data *Logger* dan *RTC* 27
Gambar 3.10 Skematik Catu Daya 27
Gambar 3.11 *Flowchart Software* Arduino 28
Gambar 3.12 *Flowchart Software APP Inventor1* 29
Gambar 3.13 *Flowchart Software APP Inventor 1* 30
Gambar 3.14 *Flowchart Software APP Inventor 1* 31
Gambar 3.15 *Screen* Awal di Apllikasi *Android* 32
Gambar 3.16 *Screen* Intruksi 33
Gambar 3.17 *Screen* Data 33
Gambar 3.18 *Screen Google Map* 34
Gambar 3.19 *Screen Read Me* 34
Gambar 3.20 Sketsa Desain di *Sketchup* 35
Gambar 3.21 Realisasi Desain Box 35
Gambar 4.1 Pengujian *Power Supply* Keluaran 5 Volt 38
Gambar 4.2 Tampilan LCD dan Jam Komputer pada Pengujian RTC 39
Gambar 4.3 Pengujian Sensor Arus 40

Gambar 4.4 Diagram Linearisasi Sensor Arus 41
Gambar 4.5 Pengujian Sensor Tegangan 42
Gambar 4.6 Diagram Linearisasi Sensor Teganagn 43
Gambar 4.7 Pengujian Hasil Pengujian *Data Logger* 43
Gambar 4.8 *Listing* Program Pengujian LCD 44
Gambar 4.9 Hasil Pengujian LCD 44
Gambar 4.10 Tampilan LCD dan Ponsel Berbas *Android* pada *Joulemeter* Ketika Kondisi Sepeda Listrik *Statis* 47
Gambar 4.11 Kondisi Sepeda Listrik *Statis* 48
Gambar 4.12 Tampilan LCD dan Ponsel Berbas *Android* pada *Joulemeter* Ketika Kondisi Sepeda Listrik *Dinamis* .. 48
Gambar 4.13 Kondisi Sepeda Listrik *Dinamis* 49

BAB I
PENDAHULUAN

1.1 Latar Belakang

Pada saat ini, kompetisi kendaraan listrik di indonesia mulai berkembang dengan pesat. salah satunya kompetisi yang digagas oleh Dirjen Pendidikan Tinggi (DIKTI) yaitu Indonesia *Energy Marathon Challenge* (IEMC). konsep dari kompetisi IEMC menitikberatkan penghematan konsumsi energi pada kendaraaan listrik, namun sebagian besar peserta masih belum memiliki alat untuk menegetahui besar konsumsi energi pada kendaraan listrik yang telah dibuatnya.

Dengan bertambah banyaknya kompetisis mobil listrik, maka perlu adanya sebuah sistem alat yang dapat me*monitoring* pemakaian energi pada kendaraan listrik tersebut. Alat ini dirancang dengan arus dikonversi oleh *ACS*712, tegangan dikonversi oleh sensor tegangan, dan kecepatan yang dikonversi oleh sensor *reed switch*. Kemudian tegangan output dibaca oleh bagian ADC Arduino dan disimpan dalam bentuk variabel data sensor. Setelah pembacaan sensor selesai. Arduino mengambil data dari RTC yang berupa data waktu. Setelah mengubah data waktu tersebut kedalam teks, ditambahkan juga data sensor untuk kemudian disimpan pada *SD card* dalam bentuk *teks Excel*. Selain itu, data hasil pengolahan ditampilkan di ponsel berbasis *Android* menggunakan *bloetooth*. Dengan adanya alat ini, diharapkan dapat membatu masyarakat dalam memonitoring pemakaian energi pada kendaraan listrik.

1.2 Perumusan Masalah

Sistem *monitoring* yang digunakan untuk mengukur konsumsi energi kendaraan listrik masih kurang sehingga perlu dipantau setiap saat agar energi pada kendaraan listrik sesuai dengan kebutuhan. Selain itu, proses *monitoring* masih dipanatau secara manual.

1.3 Batasan Masalah

Dalam pembuatan Tugas Akhir memiliki batasan masalah sebagai berikut:

1. *Joulemeter* masih diaplikasikan pada sepeda listrik.
2. Instrument pengatur berupa Arduino Uno.
3. Komunikasi alat ini menggunakan *bloetoooth*.
4. Tampilan alat ini dilakukan oleh LCD 16x2 dan Ponsel berbasis *Android*.
5. Kemampuan sensor arus yang digunakan sampai 5 Ampere.

1.4 Maksud dan Tujuan

Pembuatan rancang bangun *Joulemeter* ini dapat diuraikan sebagai berikut :

1. Membuat dan merancang sistem *monitoring* konsumsi energi listrik.
2. Merancang dan membuat sistem penyimpanan data setiap detiknya dari sensor ke Arduino dengan meggunakan *SD Card* sebagai media penyimpanan kemudian diakses pada laptop/PC dalam bentuk *file.tx*.

1.5 Metodologi

Dalam pelaksanaan Tugas Akhir yang berupa "Rancang Bangun *Joulemeter* untuk Mengukur Konsumsi Energi Kendaraan Listrik", ada beberapa kegiatan yang dapat diuraikan sebagai berikut:

1. Studi Pustaka dan Survei Data Awal

   Kegiatan ini dilakukan dengan mengidentifikasi masalah yang akan dibahas dan mempelajari buku pedoman dari e-book atau jurnal yang menunjang teori atau prinsip dasar rangkaian, baik *software* maupun *hardware* dalam pembuatan alat tersebut. Selain itu, dalam kegiatan ini juga dilakukan beberapa survei lapangan atau analisa kondisi lingkungan, guna mengetahui seberapa besar alat ini mempunyai nilai kebermanfaatan jika direalisasikan dalam kehidupan sehari-hari.
2. Perencanaan dan Pembuatan Alat

   Pada tahapan ini dilakukan perancangan dan pemodelan sistem Joulmeter sendiri,mulai dari bagian elektriknya sampai bagian mekaniknya. Dan alat akan didesain sedemikian rupa agar nantinya dapat direalisasikan dalam kehidupan sehari-hari. Setelah melakukan perancangan bentuk, desain dan struktur dari sistem, barulah dilakukan pembuatan *hardware* dari sistem tersebut. Yang meliputi pembuatan sistem mekanik dan sistem elektroniknya.
3. Perencanaan dan Pembuatan *Software*

   Perancangan dan pembuatan software dilakukan pada sistem minimum mikrokontroler ATMega328 menggunakan bahasa pemrograman *basic* yaitu dengan aplikasi *Arduino* dan ponsel berbasi *Android* sebagai tampilan data serial.
4. Uji Coba dan Analisis Data

   Pengujian ini dimaksudkan untuk memastikan bahwa kinerja masing-masing komponen dari hasil pembuatan alat dapat

berfungsi sesuai yang diharapkan. Untuk proses pengujian sistem dilakukan melalui percobaan dengan menggunakan sepeda listrik dan meng*interfacekan* data ke ponsel berbasi *Android* dengan koneksi *bluetooth*.Dari pengujian ini juga dilakukan penganalisaan sistem untuk memperoleh data-data sistem,seberapa besar nilai *error* pada sistem.

5. Penyusunan Laporan

   Penyusunan laporan dilakukan setelah semua tahap terselesaikan sehingga hasil yang diperoleh dari pembuatan alat dapat dijelaskan secara rinci dan spesifik sesuai dengan data-data yang diperoleh.

1.6 Sistematika Laporan

Untuk pembahasan lebih lanjut, laporan Tugas Akhir ini disusun dengan sistematika sebagai berikut:

Bab I : PENDAHULUAN

Membahas tentang latar belakang, perumusan masalah, batasan masalah, maksud dan tujuan, sistematika laporan, metodologi, serta relevansi Tugas Akhir yang dibuat.

Bab II : TEORI PENUNJANG

Menjelaskan teori yang berisi teori-teori penunjang yang dijadikan landasan dan mendukung dalam perencanaan dan pembuatan alat yang dibuat.

Bab III : PERANCANGAN ALAT

Membahas perencanaan dan pembuatan tentang perencanaan dan pembuatan perangkat keras (*Hardware*) yang meliputi desain mekanik dan perangkat lunak (*Software*) yang meliputi program yang akan digunakan untuk menjalankan alat tersebut.

Bab IV : PENGUKURAN DAN ANALISA

Membahas pengujian alat dan menganalisa data yang didapat dari pengujian tersebut serta membahas tentang pengukuran, pengujian, dan penganalisaan terhadap alat.

Bab V : PENUTUP

Berisi penutup yang menjelaskan tentang kesimpulan yang didapat dari Tugas Akhir ini dan saran-saran untuk pengembangan alat ini lebih lanjut.

1.7 Relevansi

Dengan adanya perancangan prototipe sistem pengontrol dan *monitoring* ruang kelas berbasis *Android* diharapkan nantinya akan :

1. Memudahkan para *montir* dibengkel, karena para *montir* tidak susah-susah membuka kendaraan listrik untuk mengecek sebuah kerusakan,dengan mengambil *SD Card* yang tertancap di *Joulemeter* dan kemudian memasukan *SD Card* di laptop/PC data dari P,V,I pada kendaraan listrik dapat dilihat.
2. Dapat me*monitoring* konsumsi energi pada kendaraan listrik dari jarak yang jauh dengan menggunakan ponsel berbasis *Android* sebagai *display*-nya
3. Sebagai inovasi teknologi untuk *monitoring* konsumsi energi kendaraan listrik.

BAB II
TEORI PENUNJANG

2.1. Sistem Minimum [1]

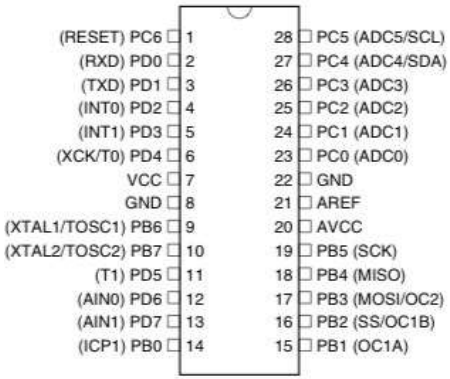
Sistem minimum merupakan suatu rangkaian elektronika yang terdiri komponen-komponen dasar yang diperlukan untuk menjalankan fungsi dari IC mikrokontroler. Sistem minimum ini nantinya bisa dihubungkan dengan rangkaian elektronika lain agar bisa menjalankan fungsi tertentu.

2.1.1. Mikrokontroler ATmega 328 [1]

Mikrokontroler adalah sebuah *chip* yang berfungsi sebagai pengontrol rangkaian elektronik dan umunya dapat menyimpan program didalamnya. Mikrokontroler umumnya terdiri dari CPU (*Central Processing Unit*), memori, I/O tertentu dan unit pendukung seperti *Analog-to-Digital Converter* (ADC) yang sudah terintegrasi di dalamnya. Kelebihan utama dari mikrokontroler ialah tersedianya RAM dan peralatan I/O pendukung sehingga ukuran *board* mikrokontroler menjadi sangat ringkas. ATMega328 memiliki 3 buah *PORT* utama yaitu *PORT*B, *PORT*C, dan *PORT*D dengan *to*tal pin *input/output* sebanyak 23 pin. *PORT* tersebut dapat difungsikan sebagai *input/output digital*.

Mikrokontroler ini memiliki beberapa fitur antara lain :

1. 130 macam instruksi yang hampir semuanya dieksekusi dalam satu siklus *clock*.
2. 32 x 8-bit *register* serba guna.
3. Kecepatan mencapai 16 MIPS dengan *clock* 16 MHz.
4. 32 KB *Flash memory* dan pada Arduino memiliki *bootloader* yang menggunakan 2 KB dari *flash* memori sebagai *bootloader*.
5. Memiliki *EEPROM* (*Electrically Erasable Programmable Read Only Memory*) sebesar 1KB sebagai tempat penyimpanan data semi *permanent* karena *EEPROM* tetap dapat menyimpan data meskipun catu daya dimatikan.
6. Memiliki *SRAM* (*Static Random Access Memory*) sebesar 2KB.
7. Memiliki pin I/O *digital* sebanyak 14 pin 6 diantaranya *PWM*(*Pulse Width Modulation*) output.
8. *Master / Slave* SPI Serial *interface*.



Gambar 2.1 Konfigurasi Pin ATMega328

ATmega328 mempunyai 28 Pin yang setiap pin nya memiliki fungsi berbeda-beda baik sebagai *PORT* maupun fungsi yang lainnya. Fungsi dari masing-masing kaki ATmega328 seperti yang terlihat pada gambar 2.1. yaitu:

1. *PORT B*

PORT B merupakan jalur data 8 bit yang dapat difungsikan sebagai *Input/output*. Selain itu *PORT*B juga dapat memiliki fungsi alternatif seperti di bawah ini.

1. ICP1 (PB0), berfungsi sebagai *Timer Counter* 1 *Input capture* pin.
2. OC1A (PB1), OC1B (PB2) dan OC2 (PB3) dapat difungsikan sebagai keluaran *PWM* (*Pulse Width Modulation*).
3. MOSI (PB3), MISO (PB4), SCK (PB5), SS (PB2) merupakan jalur komunikasi SPI.
4. Selain itu pin ini juga berfungsi sebagai jalur pemograman serial (ISP).
5. TOSC1 (PB6) dan TOSC2 (PB7) dapat difungsikan sebagai sumber *clock* external untuk *timer*.
6. XTAL1 (PB6) dan XTAL2 (PB7) merupakan sumber *clock* utama mikrokontroler.

2. *PORT* C

PORT C merupakan jalur data 7 bit yang dapat difungsikan sebagai *Input/output digital*. Fungsi alternatif *PORT*C antara lain sebagai berikut.

1. ADC 6 *channel* (PC0,PC1,PC2,PC3,PC4,PC5) dengan resolusi sebesar 10 bit. ADC dapat kita gunakan untuk mengubah *Input* yang berupa tegangan *analo*g menjadi data *digital*
2. I2C (SDA dan SDL) merupakan salah satu fitur yang terdapat pada *PORT*C. I2C digunakan untuk komunikasi dengan sensor atau *devICe* lain yang memiliki komunikasi data tipe I2C seperti sensor kompas, *accelerometer nunchuck*.

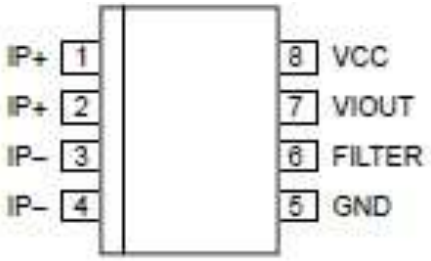
3. *PORT* D

PORT D merupakan jalur data 8 bit yang masing-masing pin-nya juga dapat difungsikan sebagai *Input*/output. Sama seperti *PORT B* dan *PORT C*, *PORT D* juga memiliki fungsi alternatif dibawah ini.

1. USART (TXD dan RXD) merupakan jalur data komunikasi serial dengan level sinyal TTL. Pin TXD berfungsi untuk mengirimkan data serial, sedangkan RXD kebalikannya yaitu sebagai pin yang berfungsi untuk menerima data serial.
2. *Interrupt* (INT0 dan INT1) merupakan pin dengan fungsi khusus sebagai interupsi *hardware*. Interupsi biasanya digunakan sebagai selaan dari program, misalkan pada saat program berjalan kemudian terjadi interupsi *hardware/software* maka program utama akan berhenti dan akan menjalankan program interupsi.
3. XCK dapat difungsikan sebagai sumber *clock external* untuk USART, namun kita juga dapat memanfaatkan *clock* dari CPU, sehingga tidak perlu membutuhkan *external clock.*
4. T0 dan T1 berfungsi sebagai masukan *counter external* untuk *timer* 1 dan *timer* 0.
5. AIN0 dan AIN1 keduanya merupakan masukan *Input* untuk *analog comparator*.

2.2. Sensor Arus *ACS*712 [2]

*ACS*712 merupakan suatu IC terpaket yang mana berguna untuk sensor arus menggantikan trafo arus yang relatif besar dalam hal ukuran. Pada prinsipnya *ACS*712 sama dengan sensor efek *hall* lainnya yaitu dengan memanfaatkan medan magnetik di sekitar arus kemudian dikonversi menjadi tegangan yang linier dengan perubahan arus. Nilai variabel dari sensor ini merupakan *Input* untuk mikrokontroler yang kemudian diolah. Keluaran dari *ACS*712 masih berupa sinyal tegangan AC, agar dapat diolah oleh mikrokontroller maka sinyal tegangan AC ini di searahkan oleh rangkaian penyearah.



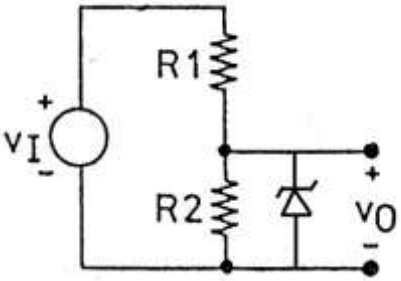
Gambar 2.2 Pin Konfigurasi *ACS*712

*ACS*712 mempunyai 8 Pin yang setiap pin nya memiliki fungsi berbeda-beda. Fungsi dari masing-masing kaki *ACS*712 seperti yang terlihat pada gambar 2.2. yaitu:

IP +	:Terminal yang mendeteksi arus, terdapat *sekring* di dalamnya
IP -	:Terminal yang mendeteksi arus, terdapat *sekring* di dalamnya
GND	:Terminal sinyal *ground*
FILTER	:Terminal untuk kapasitor eksternal yang berfungsi sebagai pembatas *bandwith*
VIout	:Terminal keluaran sinyal *analog*
Vcc	:Terminal masukan catu daya

2.3. Sensor Tegangan [3]

Sensor tegangan adalah komponen yang digunakan untuk mengkonversi besaran tegangan pada listrik menjadi besaran *analog* dengan memperkecil nilainya menjadi tegangan referensi sehingga dapat dibaca pada rangkaian elektronik. Penggunaan sensor tegangan kali ini menggunakan rangkaian pembagi tegangan. Rangkaian pembagi tegangan pada dasarnya dapat dibuat dengan 2 buah resistor, dan satu buah diode zener seperti dapat dilihat pada gambar 2.3.



Gambar 2.3 Sensor Tegangan

2.4. IC *Regulator* 7805 dan *IC Regulator* 7809 [4]

Regulator merupakan suatu rangkaian elektronika yang memiliki fungsi untuk mengatur agar tegangan keluaran yang dihasilkan tetap berada pada posisi yang ditentukan walau tegangan masukkannya berubah-ubah. Rangkaian *regulator* tegangan ini kemudian dibuat di dalam bentuk IC (*integrated circuit).* IC *regulator* 7805 pada gambar 2.4 adalah *regulator* tegangan yang memiliki output +5 Volt. Sedangkan IC *regulator* 7809 pada gambar 2.4 adalah *regulator* tegangan yang memiliki output +9 Volt. Untuk memudahkan dalam mengingat tegangan *output* dari *regulator* tegangan seri 78xx adalah dua *digit* terakhir dari nomer tersebut.

Gambar 2.4 IC *Regulator* 7805 dan IC *Regulator* 7809

2.5. *Logging Recorder Shield Data Logger* Modul [5]

Logging Recorder Shield Data Logger Modul merupakan kombinasi antara modul *RTC* dengan Modul *MMC*, jadi prinsip kerja dari modul ini adalah mengambil data dari sensor tiap detiknya dan kemudian disimpan di *SD Card*.Dengan adanya modul ini kita dapat mengambil data dari sensor dan diketahui waktunya juga,penggunaan modul ini

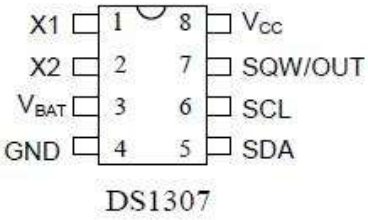
sangat tepat digunakan dalam pembuatan sistem akusisi data atau monitoring. Bentuk fisik Modul *Logging Recorder Shield Data Logger* seperti pada gambar 2.5

Gambar 2.5 Modul *Logging Recorder Shield Data Logger*

2.5.1. *Real-Time Clock* (RTC) [5].

Real-time clock disingkat RTC adalah jam di komputer yang umumnya berupa sirkuit terpadu yang berfungsi sebagai pemelihara waktu. RTC umumnya memiliki catu daya terpisah dari catu daya komputer (umumnya berupa baterai *litium*) sehingga dapat tetap berfungsi ketika catu daya komputer terputus. Kebanyakan RTC menggunakan *osilator* kristal. Jenis RTC yang kami gunakan adalah *DS1307* yang mempunyai spesifikasi sebagai berikut :

- *Real-time clock* (RTC) meyimpan data- data detik, menit, jam, tanggal, bulan, hari dalam seminggu, dan tahun valid hingga 2100.
- 56-*byte*, *battery-backed*, RAM *nonvolatile* (NV) RAM untuk penyimpanan.
- Antarmuka serial *Two-wire* (I2C).
- Sinyal luaran gelombang-kotak terprogram (*Programmable squarewave*).
- Deteksi otomatis kegagalan-daya (*power- fail*) dan rangkaian *switch.*
- Konsumsi daya kurang dari 500nA menggunakn mode baterei cadangan dengan operasional *osilator.*



Gambar 2.6 Pin konfigurasi DS1307

DS1307 mempunyai 8 Pin yang setiap pin nya memiliki fungsi berbeda-beda. Fungsi dari masing-masing kaki *DS1307* seperti yang terlihat pada gambar 2.6. yaitu:

1. X1 dan X2 adalah pin yang dihubungkan dengan kristal 32.768 KHz.
2. VBAT adalah pin yang dihubungkan masukan baterai +3V.
3. GND adalah pin yang dihubungkan *Ground*.
4. SDA adalah pin yang difungsikan sebagai jalur data.
5. SCL adalah pin yang fungsikan sebagai jalur *clock*.
6. SQW/OUT adalah pin yang digunakan sebagai keluaran sinyal kotak.
7. VCC adalah pin untuk mencatu tegangan 5V.

2.5.2 *MMC Shield* [5]

MMC Shield adalah sebuah rangkaian yang digunakan untuk menyimpan data kedalam sebuah media. Pada alat ini media yang digunakan adalah *SD Card*. Data yang akan disimpan adalah hasil pembacaan sensor yang telah diproses terlebih dahulu oleh Arduino. Modul *MMC Card* dapat dilihat pada gambar 2.7.

Gambar 2.7 Modul *MMC Card*

2.5.3 *SD Card [5]*

SD Card adalah salah satu media penyimpanan yang berupa kartu memori flash. Yang dimaksud dengan kartu memori *flash* adalah kartu memori yang data yang ditulis didalamnya dapat tersimpan tanpa membutuhkan suplai listrik.*SD Card* dapat dilihat pada gambar 2.8.

Gambar 2.8 *SD Card*

Secara fisik, *SD Card* dilengkapi dengan sebuah kunci. Kunci ini digunakan sebagai perlindungan akses isi dari *SD Card*. Apabila dalam kondisi tidak terkunci, *SD Card* dapat dilihat isinya maupun ditambahi isi baru. Sedangkan dalam posisi terkunci, *SD Card* hanya bisa diakses untuk dilihat saja isinya. Sebuah *SD Card* didalamnya terdiri atas sebuat chip pengontrol dan modul penyimpanan. Modul penyimpanan berfungsi sebagai tempat dimana data disimpan. Ketika kita memformat sebuah *SD Card* sebuah *file* sistem akan ditulis pada bagian ini. Sedangkan chip pengontrol berfungsi sebagai penghubung antara modul penyimpanan dengan alat lain misalnya mikrokontroler dengan menggunakan *SD command*.

2.6. *Bluetooth* [6]

Bluetooth adalah standar komunikasi untuk bertukar data tanpa perantara kabel (wireless) pada jarak yang tidak terlalu jauh. Dalam Tugas Akhir ini untuk komunikasi *Bluetooth* kami mengggunakan *Bluetooth* HC05 dan resistor 2kΩ dengan resistor 1 kΩ. .Modul *Bluetooth* HC-05 adalah sebuah modul *Bluetooth* SPP (*Serial Port Protocol*) yang mudah digunakan untuk komunikasi serial *wireless* (*nirkabel*) yang mengkonversi *Port* serial ke *Bluetooth*. HC-05 menggunakan modulasi *bluetooth* V2.0 + EDR (*Enchanced Data Rate*) 3 Mbps dengan memanfaatkan gelombang radio berfrekuensi 2,4 GHz. Modul ini dapat digunakan sebagai *slave* maupun master. HC-05 memiliki 2 mode konfigurasi, yaitu AT *mode* dan *CommunICation mode*. AT *mode* berfungsi untuk melakukan pengaturan konfigurasi dari HC-05. Sedangkan *CommunICation mode* berfungsi untuk melakukan komunikasi *bluetooth* dengan piranti lain. Dalam penggunaannya, HC-05

dapat beroperasi tanpa menggunakan *driver* khusus. *bloetooth* dapat dilihat pada gambar 2.9. Untuk berkomunikasi antar *Bluetooth*, minimal harus memenuhi dua kondisi berikut :

1. Komunikasi harus antara master dan *slave*.
2. *Password* harus benar (saat melakukan *pairing*).

Gambar 2.9 *Bloetooth*

Dengan deskripi sebagai berikut Sensitivitas -80dBm (*TypICal*). Daya *transmit* RF sampai dengan +4dBm., Operasi daya rendah 1,8V – 3,6V I/O. Kontrol PIO, Antarmuka UART dengan *baudrate* yang dapat diprogram, Dengan antena terintegrasi.

2.7. *Android* [7]

Android merupakan sistem operasi berbasis *linux* yang digunakan untuk *mobile devICe* seperti *smartphone* atau PC *tablet*. Seperti *MICrosoft* dengan *Windows* yang sangat dikenal oleh para pengguna komputer maupun laptop, *Android* merupakan sistem operasi seperti *windows* sedangkan *smartphone* atau PC tablet merupakan perangkat kerasnya seperti komputer maupun laptop. *Android* dibuat dengan sistem distribusi *open sources* yang memungkinkan para pengembang untuk dapat menciptakan berbagai macam aplikasi menarik agar dapat dinikmati oleh para penggunanya.

2.7.1. Sistem Operasi *Android* [7]

Sistem Operasi *Android* merupakan sebuah sistem operasi yang berbasis *Linux* untuk telepon seluler seperti telepon pintar dan komputer tablet. *Android* menyediakan *platform* terbuka bagi para pengembang untuk menciptakan aplikasi mereka sendiri untuk digunakan oleh bermacam peranti bergerak. Pada awalnya yang mengembangkan Sistem Operasi *Android* ini adalah *Android Inc.,* kemudian dibeli oleh *Google Inc*. yang setelah itu terus menerus melakukan pengembangan ke teknologi yang lebih baik. Berikut ini merupakan macam-macam sistem operasi *Android* :

1. *Android* 1.1

   Android versi 1.1 dikeluarkan pada 9 Maret 2009 oleh Google. *Android* versi dilengkapi aplikasi yang *disupport* oleh *Google Mail ServICe* dengan pembaruan estetis pada aplikasi, jam alarm, *voICe search* (pencarian suara), pengiriman pesan dengan Gmail dan pemberitahuan email.
2. *Android* 1.5 Cup Cake

   Android Cup Cake dikeluarkan pada pertengahan Mei 2009 oleh Google Inc. *Android* ini dilengkapi *software development kit* dengan berbagai pembaharuan termasuk penambahan beberapa fitur antara lain yakni kemampuan merekam dan menonton video dengan modus kamera, mengunggah video ke Youtube, *upload* gambar ke PICasa langsung dari telepon, serta mendapat dukungan *Bluetooth* A2DP.
3. *Android* 1.6 Donut

   Android Donut dikeluarkan pada September 2009. *Android* versi ini dapat menampilkan proses pencarian yang lebih baik dibandingkan versi-versi sebelumnya. *Android* Donut juga dilengkapi baterai indikator dan kontrol applet VPN.
1. *Android* 2.0/2.1 Eclair

   Android Eclair dikeluarkan pada 3 Desember 2009. *Android* Eclair merupakan *Android* pertama yang mulai dipakai oleh banyak smartphone, fitur utama Eclair yaitu perubahan total struktur dan tampilan *user interface*. Perubahan yang ada pada *Android* versi ini dari versi sebelumnya antara lain adalah pengoptimalan *hardware*, peningkatan *Google Maps* 3.1.2, perubahan UI dengan *browser* baru dan dukungan HTML5, daftar kontak yang baru, dukungan *flash* untuk kamera 3,2 MP, digital *Zoom*, dan *Bluetooth* 2.1.
2. *Android* 2.2 Froyo (Frozen Yogurt)

   Android Froyo dikeluarkan pada 20 mei 2012. *Android* versi ini memiliki kecepatan kinerja dan aplikasi 2 sampai 5 kali dari versi-versi sebelumnya. Selain itu ada penambahan fitur-fitur baru seperti dukungan *Adobe Flash* 10.1, intergrasi V8 *JavaScript engine* yang dipakai *Google Chrome* yang mempercepat kemampuan rendering pada browser, pemasangan aplikasi dalam SD Card, kemampuan *WiFi Hotspot Portabel*, dan kemampuan *auto update* dalam aplikasi *Android Market*.
3. *Android* 2.3 Gingerbread

   Android Gingerbread dikeluarkan pada 6 Desember 2010. Perubahan-perubahan yang didapat dari *Android* versi ini antara

lain peningkatan kemampuan permainan *(gaming),* peningkatan fungsi *copy paste*, layar antar muka *(User Interface)* didesain ulang, dukungan format video VP8 dan WebM, efek audio baru (*reverb*, *equalization*, *headphone virtualization*, dan *bass boost*), dukungan kemampuan *Near Field CommunICation (NFC)* dan dukungan jumlah kamera yang lebih dari satu.

4. *Android* 3.0/3.1 *Honeycomb*

   Android Honeycomb dikeluarkan pada awal 2012. *Android* Honeycomb merupakan versi *Android* yang dirancang khusus untuk *devICe* dengan layar besar seperti Tablet PC. Fitur baru yang ada pada *Android* Honeycomb antara lain yaitu dukungan terhadap prosessor *multICore* dan grafis dengan *hardware acceleration*. *User Interface* pada Honeycomb juga berbeda karena sudah didesain untuk tablet.

5. *Android* 4.0 ICS (*ICe Cream SandwICh*)

   Android ICe Cream SandwICh dikeluarkan pada 10 Mei 2011. *Android ICe Cream SandwICh* membawa fitur *Honeycomb* untuk *smartphone* serta ada penambahan fitur baru seperti membuka kunci dengan pengenalan tampilan, jaringan data pemantauan penggunaan dan kontrol, terpadu kontak jaringan sosial, perangkat tambahan fotografi, mencari email secara offline, dan berbagi informasi dengan menggunakan NFC.

6. *Android* 4.1 Jelly Bean

   Android Jelly Bean dikeluarkan pada 10 Mei 2011. Versi ini membawa sejumlah keunggulan dan fitur baru, diantaranya peningkatkan *Input keyboard*, desain baru fitur pencarian, UI yang baru dan pencarian melalui *VoICe Search* yang lebih cepat. Versi ini juga dilengkapi *Google Now* yang dapat memberikan informasi yang tepat pada waktu yang tepat pula. Salah satu kemampuannya adalah dapat mengetahui informasi cuaca, lalu-lintas, ataupun hasil pertandingan olahraga.

7. *Android* 4.4 KitKat

   Android KitKat merupakan versi terbaru dari OS *Android*. Sistem terbaru ini dihadirkan oleh Google dengan spesifikasi minimum yang dapat dijalankan dengan menggunakan *smartphone kelas low end*.

2.8. LCD [8]

LCD (*Liquid Crystal Display*) merupakan modul untuk menampilkan karakter data dari sebuah alat masukan seperti mikrokontroler yang menggunakan kristal cair sebagai penampil utama.

Modul tersebut dilengkapi dengan mikrokontroler yang dirancang khusus untuk mengendalikan LCD. Pemrograman yang digunakan untuk menampilkan karakter atau string ke LCD ini adalah *CodeVision* AVR. Berikut ini merupakan gambar dari LCD dengan jumlah karakter 2x16 pada gambar 2.10.

Gambar 2.10 LCD 2x16

Modul LCD memiliki karakteristik:

1. Terdapat 16 x 2 karakter huruf yang bisa ditampilkan
2. Terdapat 192 macam karakter
3. Terdapat 80 x 8 bit display RAM (maksimal 80 karakter)
4. Memiliki kemampuan penulisan dengan 8 bit maupun dengan 4 bit
5. Dibangun dengan *osilator local*
6. Satu sumber tegangan 5 Volt
7. Otomatis reset saat tegangan dihidupkan
8. Bekerja pada suhu 0 ^{0}C sampai 55 ^{0}C

2.9. *REED SWITCH* [9]

Red switch adalah saklar listrik dioperasikan oleh medan magnet diterapkan. Ini di ciptakan di Bell Telephone Laboratories pada tahun 1936 oleh WB Ellwood. Ini terdiri dari sepasang kontak *plate* tipis pada kaca tertutup rapat. Apabila bagian permukaan dari sensor terkena medan magnet maka dua buah kontak *plate* tipis yang terdapat dibagian dalam sensor akan tertarik oleh medan magnet, sehingga kontak akan terhubung. Jika magnet dijauhkan daribagian permukaan sensor (*reed switch*), maka kontak *plate reed switch* akan membuka. Bentuk sensor *reed swit*ch dapat dilihat pada gambar 2.11.

Gambar 2.11 Sensor *Reed Switch*

2.10. *APP INVENTOR* [10]

G*oogle APP Inventor adalah* sebuah aplikasi yang disediakan oleh *google* yang memungkinkan orang untuk membuat aplikasi perangkat lunak untuk *OS Android*. *APP Inventor* menggunakan *graphICal interface* yang memungkinkan *user* melakukan *drag and drop* objek visual untuk membuat aplikasi yang berjalan pada sistem *Android*. Pengembangan aplikasi ini adalah bagaimana kode aplikasi disusun. Berbeda dengan tool pengembangan aplikasi yg lain seperti Eclipse, netbeans, Visual C, Delphi, dll. *Tool* ini tidak menggunakan koding tekstual, melainkan dengan menyusun blok-blok kode. *APP Inventor* memiliki dua bagian utama yaitu *APP Inventor Designer* dan *Block Editor*. *APP Inventor Designer* adalah Komponen desainer berjalan pada browser yang digunakan untuk memilih komponen yang dibutuhkan dan mengatur *property* nya.

Pada lingkungan kerja *APP Inventor* ini terdapat beberapa komponen yang terdiri dari :

1. Komponen Desainer, Komponen desainer berjalan pada *browse*r yang digunakan untuk memilih komponen yang dibutuhkan dan mengatur propertinya. Pada komponen desainer sendiri terdapat 5 bagian, yaitu *palette, viewer, component*, media dan *properties*.

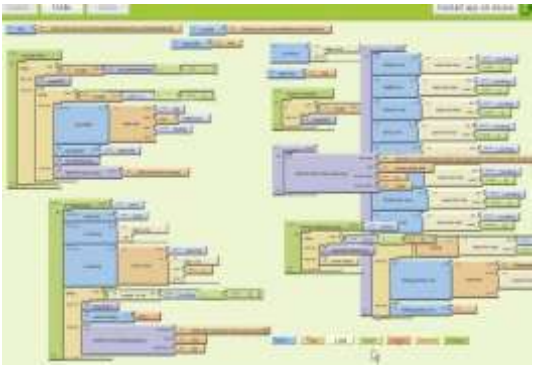
1.*Palette*	:*list* komponen yang bisa digunakan
2. *Viewer*	:untuk menempatkan komponen dan mengaturnya sesuai tampilan yang diinginkan
3.*Component*	:tempat *list* komponen yang dipakai pada *project*
4.Media	:mengambil media *audio* atau gambar untuk *project*
5.*Properties*	:mengatur *properties* komponen yang digunakan, seperti *width, height, name*

2. *Block Editor*, *Block Editor* berjalan di luar *browser* dan digunakan untuk membuat dan mengatur *behaviour* dari komponen-komponen yang kita pilih dari komponen desainer. Nah, berhubung *block editor* ini basisnya java. Tampilan *Block Editor* terlihat seperti gambar 2.13.
3. Emulator, Emulator digunakan untuk menjalankan dan *mengetest project* yang telah kita buat. Jadi yang belum punya *Android* pun tetep bisa belajar karna *APP Inventor*

menyediakan emulatornya juga. Tampilan *APP Inventor Designer* terlihat seperti gambar 2.12

Gambar 2.12 *APP Inventor Designer*

Blok Editor Adalah bagian dari *APP Inventor* untuk merancang logika pemrograman komonen-komponen yang telah dibuat sebelumnya di *APP Inventor designer*. Pada blok *editor*, *user* dapat menggabungkan blok-blok kode yang terkait. Blok-blok tersebut terdapat pada *drawers* bagian sebelah kiri dari *userinterface* blok *editor*. *User* hanya perlu *drag* blok yang berada pada *drawers* ke dalam *workspace* dalam rangka untuk menambahkanya ke proyek.



Gambar 2.13 Blok *Editor*

2.11. *Arduino IDE* [11]

Board Arduino dapat di program menggunakan *software open source* bawaan *Arduino IDE*. *Arduino IDE* adalah sebuah aplikasi *cross-platform* yang berbasis Bahasa pemrograman *Processing* dan *Wiring. Arduino IDE* di desain untuk mempermudah pemrograman, dengan adanya kode *editor* yang dilengkapi dengan *syntax highlighting*, *brace matching*, dan indentasi otomatis untuk kemudahan pembacaan program, serta dapat meng-*compile* dan meng-*upload* program ke *board* dalam satu klik. Jendela *Arduino IDE* dapat dilihat pada gambar 2.14.

.

Gambar 2.14 Jendela *Arduino IDE*

Selain membuat pemrograman Arduino lebih mudah. *Software* ini juga dilengkapi jendela untuk memonitor koneksi serial. Dengan contoh-contoh pemrograman yang dapat diakses secara baik *online* maupun *offline* sehingga memudahkan pengguna untuk mengakses referensi tentang pemrograman Arduino

Halaman ini sengaja dikosongkan

BAB III
PERANCANGAN ALAT

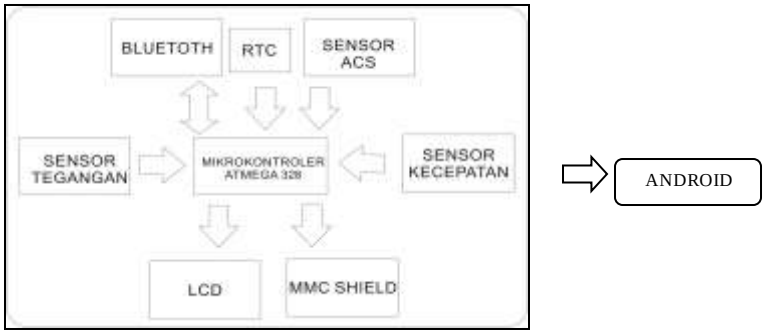
Dalam bab ini akan membahas mengenai perancangan alat yang meliputi perencanaan perangkat keras (*hardware*) dan perangkat lunak (*software*). Hal tersebut guna mewujudkan Tugas Akhir yang berjudul " Rancang bangun *Joulemeter* untuk mengukur konsumsi energi kendaraan listrik" . Perancangan alat akan dibahas perbagian disertai dengan gambar skematik. Sedangkan penjelasan *software* akan dijelaskan mengenai pembuatan program di Arduino, serta program komunikasi serialnya menggunakan *software APP Inventor*.

Untuk memudahkan dalam pembahasan bab ini akan dibagi menjadi tiga yaitu:

1. Perancangan *hardware* (perangkat keras) yang terdiri dari konfigurasi pin Arduino, pembuatan rangkain *shield* LCD, pembuatan rangkain sensor kecepatan menggunakan *reed switch*, konfigurasi sensor *ACS*712, konfigurasi *bluetooth*, pembuatan rangkaian sensor tegangan, pembuatan rangkain *push button*, konfigurasi pin RTC dan *MMC Shield*, rangkain catu daya.
2. Perancangan *software* (perangkat lunak) meliputi perancangan pembuatan program di Arduino serta program yang digunakan untuk mengirim data secara serial ke ponsel berbasis *Android* menggunakan *software APP Inventor*.
3. Perancangan Mekanik meliputi desain perancangan *box* yang digunakan untuk memasukan semua perangkat elektronik supaya terlihat lebih menarik dan ringkas. Bahan yang digunakan untuk mebuat b*ox* ini menggunakan bahan *acrylic*.

3.1. Perancangan *Hardware* (Perangkat Keras)

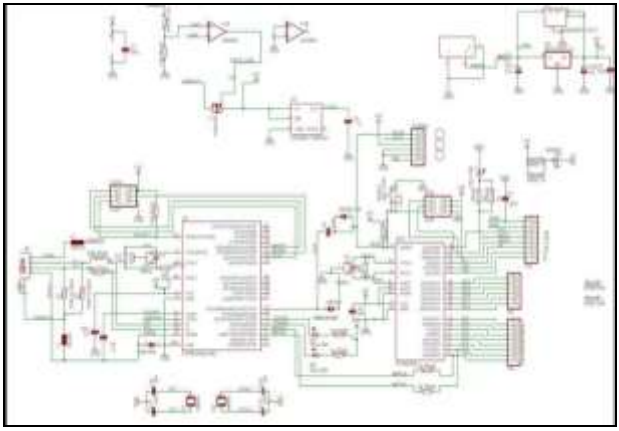
Secara garis besar sistem kerja alat dapat dilihat dari diagram fungsional pada gambar 3.1. Sistem kerja alat secara keseluruhan yaitu pertama arus dikonversi oleh *ACS*712, tegangan dikonversi oleh sensor tegangan, dan kecepatan yang dikonversi oleh sensor *reed switch*. Kemudian tegangan output dibaca oleh bagian ADC Arduino dan disimpan dalam bentuk variabel data sensor. Setelah pembacaan sensor selesai. Arduino mengambil data dari RTC yang berupa data waktu. Setelah mengubah data waktu tersebut kedalam teks, ditambahkan juga data sensor untuk kemudian disimpan pada *SD card* dalam bentuk *teks Excel*. Selain itu, data hasil pengolahan ditampilkan di ponsel berbasis *Android* menggunakan *bloetooth*.



Gambar 3.1 Diagram Fungsional Sistem

3.1.1 Arduino Uno

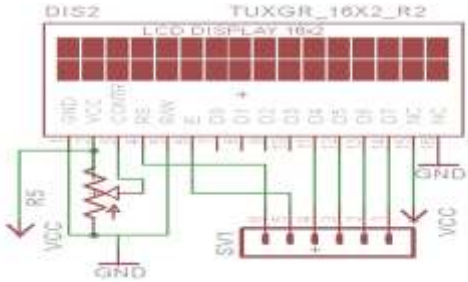
Perencanaan Tugas Akhir ini menggunakan Arduino Uno yang ada dipasarand dengan *board* berbasis mikrokontroler Atmega328. Board ini memiliki 14 digital *Input* output pin (dimana 11 pin dapat digunakan sebagai output PWM), 6 *Input* analog, 16 MHz osilator kristal, koneksi USB, *jack* listrik tombol *reset*. Pin-pin ini berisi semua yang diperlukan untuk mendukung mikrokontroler, hanya terhubung ke komputer dengan kabel USB atau sumber tegangan bisa didapat dari adaptor AC-DC atau baterai untuk menggunakannya. Program Arduino Uno dapat menggunakan STK500, AVRASP, Arduino Pro mini. Skematik Arduino Uno dapat dilihat pada gambar 3.2.



Gambar 3.2 Skematik Arduino Uno

3.1.2 Perancangan Rangkaian LCD

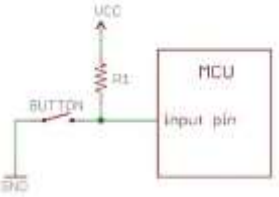
Perencanaan Tugas Akhir ini menggunakan *LCD* yang digunakan pada alat ini adalah 16x2. Maksudnya adalah 16 kolom dan 2 baris. Tegangan yang diberikan pada *LCD* tersebut adalah sebesar 5 Vdc. *LCD* ini dapat menampilkan gambar atau tulisan karena didalam *LCD* terdapat titik-titik cahaya. Skematik *LCD* dapat dilihat pada gambar 3.3.



Gambar 3.3 Skematik *Shield* LCD

3.1.3 Perancangan Rangkaian *Push Button*

Perencanaan Tugas Akhir ini menggunakan Rangkaian *Push Button* ini mengunakan jenis rangkaian *Push Button* Tipe *Pull Up* jadi ketika *push button* ditekan akan memberikan logika 1 karena, *Input* pin akan ditarik ke 5 volts. Ketika *button* ditekan, jalur dengan resistor yang lebih rendah akan menghubungkan pin dengan *ground*. Bila tanpa resistor antara 5 volt dan *ground* maka akan terjadi *short circuit* (konslet) dan dapat merusak rangkaian atau *power suplly* seperti gambar 3.4.

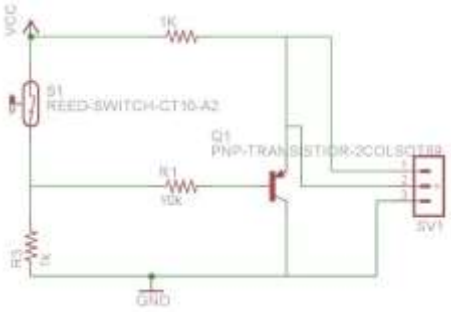


Gambar 3.4 Skematik Rangkaian *Pull Up* Resistor

3.1.4 Perancangan Rangkaian Sensor Kecepatan

Perencanaan Tugas Akhir ini menggunakan Rangkaian sensor kecepatan ini menggunakan kombinasi antara dua sensor yaitu sensor

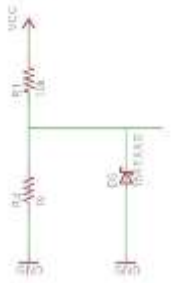
magnet dan sensor tegangan.Jadi sensor magnet dipasang di dekat *velg* dan di *velg* sendiri terpasang satu magnet permanen jadi apabila sensor magnet dan magnet permanen berdekatan maka akan muncul ADC dari pembagian tegangan kedua resistor.Dan setiap ADC tertentu keluar fungsi *counter* akan aktif di di Arduino,dan data diolah menjadi bentuk RPM. Skematik Rangkaian *Reed Switch* dapat dilihat pada gambar 3.5.



Gambar 3.5 Skematik Rangkaian *Reed Switch*

3.1.5 Perancangan Rangkaian Sensor Tegangan

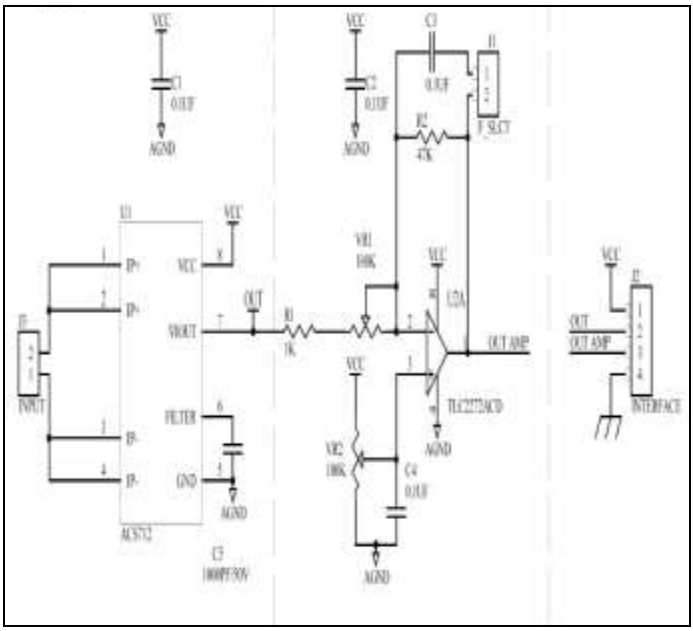
Perencanaan Tugas Akhir ini menggunakan Sensor yang digunakan menggunakan prinsip *voltage divider* yang dilengkapi dengan dioda *zene*r 5v yang berfungsi untuk membatasi tegangan yang yang dibaca oleh *analog read* dari mikrokontroler.Karena di mikrokontroler tegangan yang bisa diolah oleh fungsi ADC hanyalah sampai tegangan 5v.Karena 0-5V dari tegangan mempunyai nilai ADC sebesar 0 -1023. Skematik Rangkaian Sensor Tegangan dapat dilihat pada gambar 3.6.



Gambar 3.6 Skematik Rangkaian Sensor Tegangan

3.1.6 Perancangan Rangkaian Sensor Arus *ACS*712 *with OpAmp*

Perencanaan Tugas Akhir ini menggunakan Sensor *ACS*712 sebagai sensor arus yang memiliki prinsip *Hall Effect* sehingga arus yang mampu dibaca bernilai dari + 5 A hingga - 5 A. Sensor ini mampu bekerja dengan catu daya 5 VDC. Pin vout dihubungkan dengan pin I/O *analog* pada sistem minimum Arduino. Sumber tegangan dan *ground* menggunakan pin pada sistem minimum Arduino untuk menghindari kesalahan pembacaan akibat beda tegangan referensi. Berikut skematik dari sensor arus *ACS* 712 DT *Sense Current Sensor* yang ditunjukkan pada gambar 3.7 berikut ini.



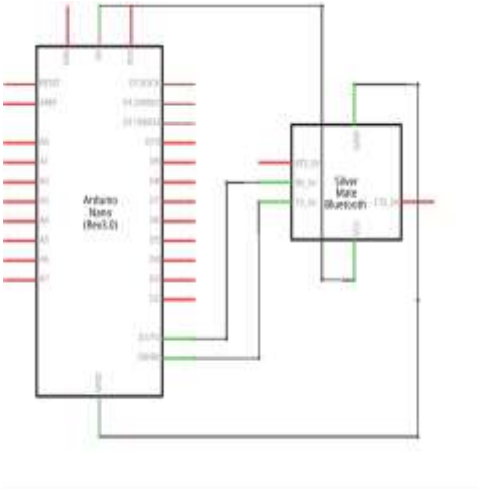
Gambar 3.7 Skematik DT *Sense Current Sensor*

3.1.8 Perancangan *Bluetooth*

Untuk komunikasi serial dalam Tugas Akhir ini menggunakan *Bluetooth* HC-05 yang sudah ada *Bluetooth serial module transceiver*. *Bluetooth* HC-05 dapat dikoneksikan sebagai *slave*, ataupun sebagai *master*. Dalam penggunaannya, *Bluetooth* HC-05 ini diatur terlebih dahulu sebagai *slave* dengan menggunakan *AT Commands*. Pengaturan *slave mode* pada *AT Commands* dengan mengketik "AT+ROLE = 0" di

serial monitor Arduino Uno maka akan tampil "OK" sebagai respon *Bluetooth* HC-05 sehingga *Bluetooth* HC-05 sudah menjadi *slave mode*.

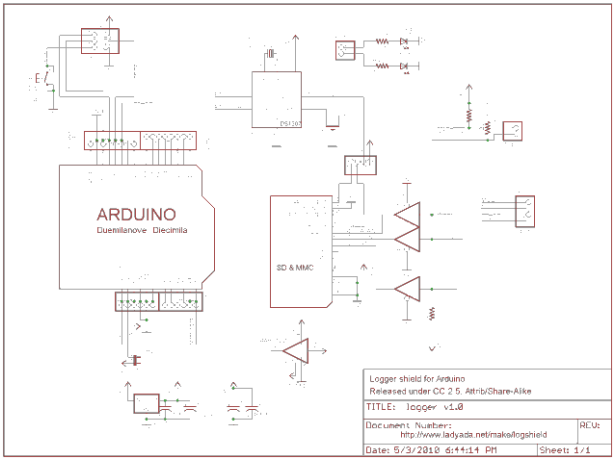
Dengan begitu proses pengaturan mulai pengiriman dan penerimaan data berupa angka untuk lampu lalu lintas dari aplikasi *APP Inventor 2* di *handphone Android* dapat dilakukan secara nirkabel. Dalam pemasangannya pin RX Arduino Uno dengan pin TX *Bluetooth* HC-05 dan pin TX *Bluetooth* HC-05 dengan pin RX Arduino Uno. skematik rangkaian komunikasi serial *Bluetooth* HC-05 dapat dilihat pada gambar 3.8.



Gambar 3.8 *Wirring* Modul *Bluetoth* HC 05

3.1.7 Perancangan *Data Logger* dan RTC*(Real Time Clock)*

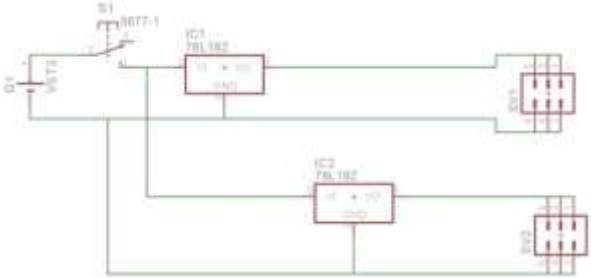
Perencanaan Tugas Akhir ini menggunakan Komunikasi *Data Logger* dengan Arduino adalah menggunakan *MMC Shield* berisi *SD card* dan digabungkan dengan RTC. Oleh karena itu, untuk membuat rangkaian penyimpanannya kita hanya perlu mengkoneksikan antara MISO, MOSI, SCK, dan CS(*pin select*) saja. Selain itu perlu dilengkapi AMS1117 untuk menyesuaikan tegangan agar selevel dengan tegangan TTL.RTC disini menggunakan IC DS1307 Pada penggunaan IC DS1307 dilakukan 2 proses yakni menulis data ke IC DS1307 (dalam hal ini adalah *setting* waktu) dan yang kedua adalah membaca data waktu dari IC DS1307. Prosedur kedua proses ini mengacu pada protokol I2C agar komunikasi berjalan sebagai mana mestinya. skematik Datalogger dan RTC dapat dilihat pada gambar 3.9.



Gambar 3.9 Skematik *Data Logger* dan RTC

3.1.7 Perancangan Rangkaian Catu Daya

Perencanaan Tugas Akhir ini menggunakan rangkaian catu daya digunakan untuk menurunkan tegangan dari sumber baterai lippo sebesar 11.1 v kemudian tegangan tersebut dipecah menjadi dua sumber yaitu sebesar 5v dan 9v.Tegangan 9v digunakan untuk mengaktifkan Arduino sementara tegangan 5v digunakan untuk mengaktifkan sensor dan LCD. SchematIC Rangkaian catu daya dapat dilihat pada gambar 3.10.



Gambar 3.10 Skematik Catu Daya

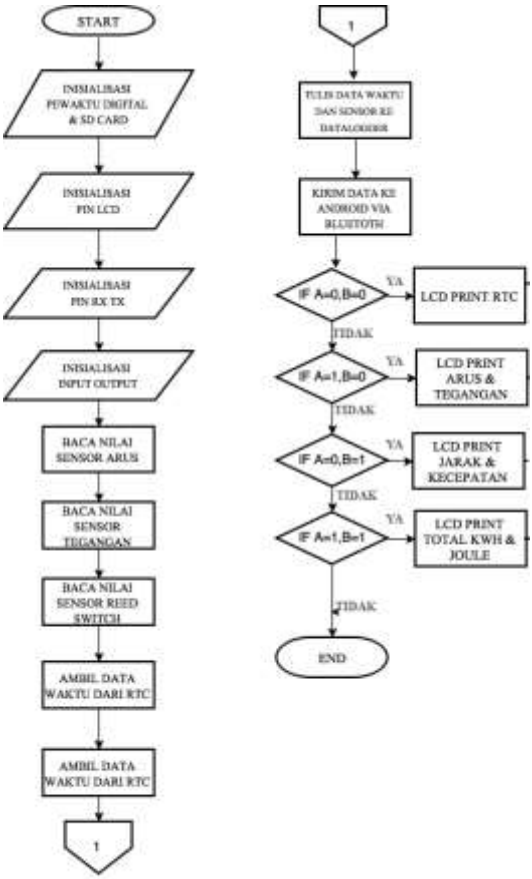
3.2 Perancangan Perangkat Lunak (Software)

Dalam pembuatan *software* pada hardware dalam pembuatan alat ini difungsikan untuk memrogram Arduino menggunakan *Arduino*

IDE dan *software* pada *Android* dalam pembuatan aplikasi ini menggunakan aplikasi *APP Inventor*.

3.2.1 Perancangan Sistem Secara Keseluruhan

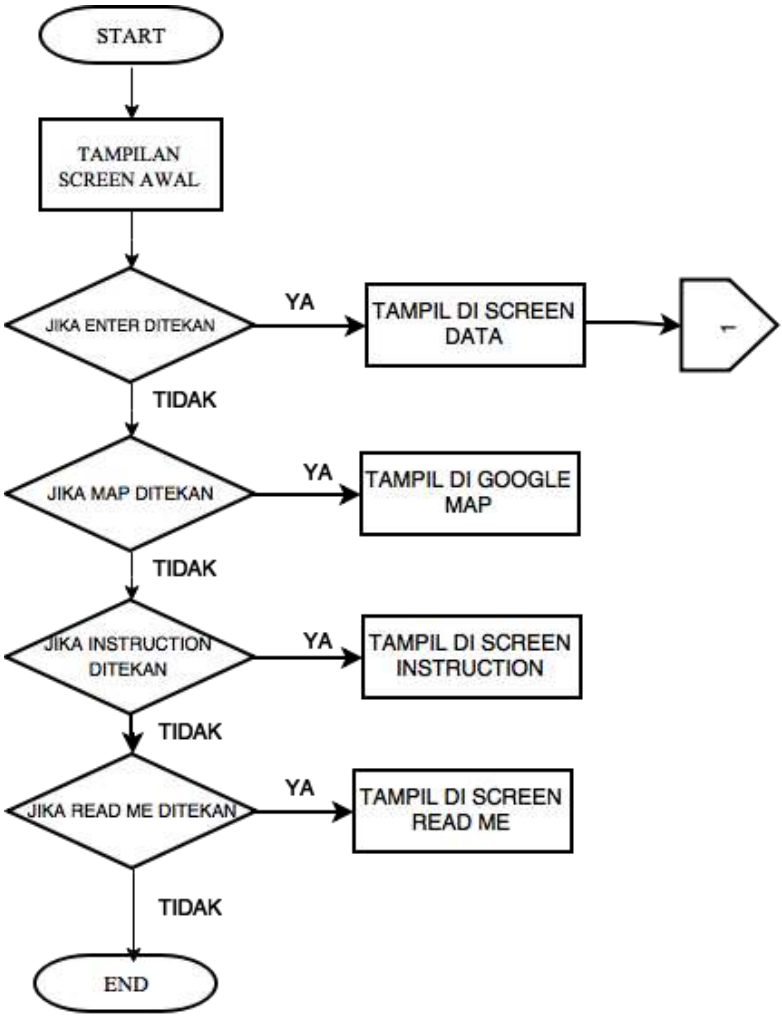
Dalam pembuatan *software* pada hardware dalam pembuatan alat ini difungsikan untuk memrogram Arduino menggunakan *Arduino IDE*. Pemrograman dibuat dengan menyesuaikan alur *flowchart* pada gambar 3.11. .



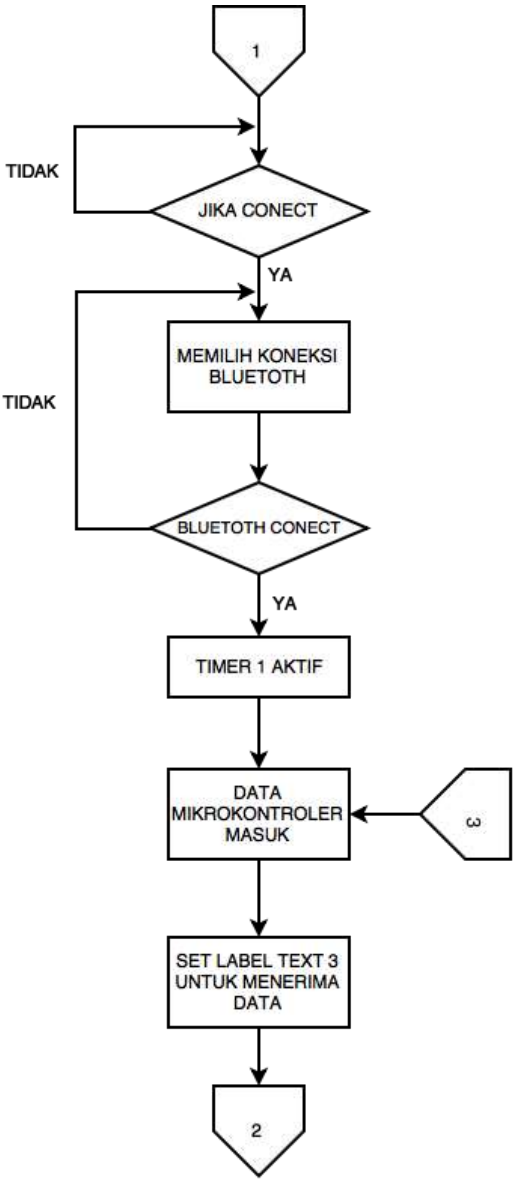
Gambar 3.11 *Flowchart Software* Arduino

3.2.2 Perancangan Program *APP Inventor*

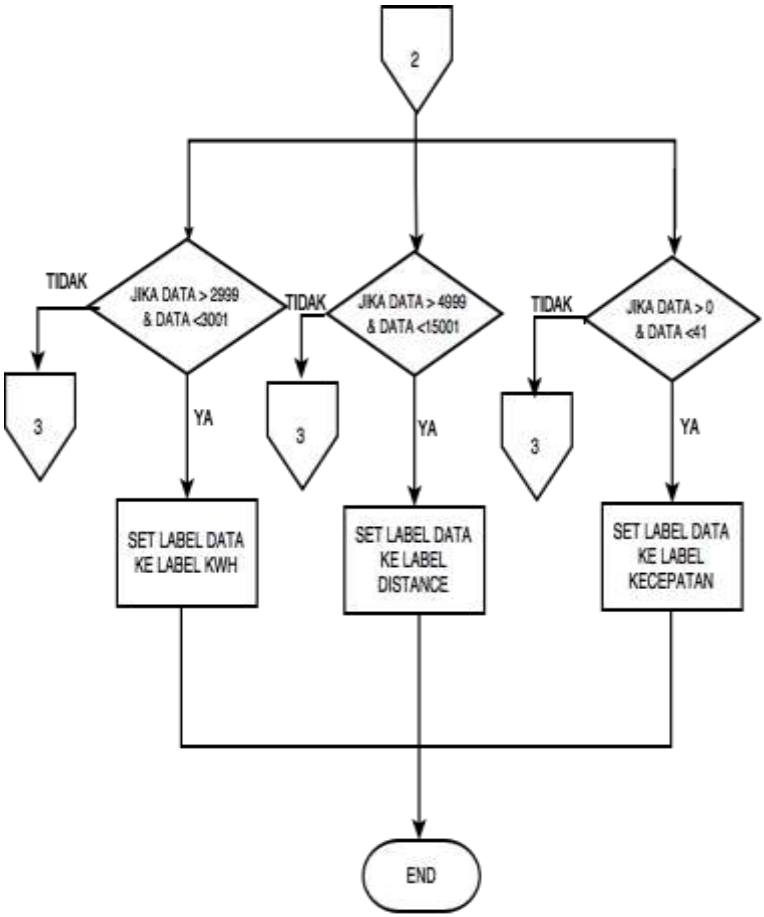
Dalam pembuatan *software* pada *Android* supaya data dari mikrokontroler bisa diinterfacekan ke ponsel berbasis *Android* menggunakan software *APP Inventor*. Pemrograman dibuat dengan menyesuaikan alur *flowchart* pada gambar 3.12, 3.13 dan 3.14.



Gambar 3.12 *Flowchart Software APP Inventor 1*



Gambar 3.13 *Flowchart Software APP Inventor 1*



Gambar 3.14 *Flowchart Software APP Inventor 1*

3.2.3 Perancangan *Software Client* (*Android*)

Perancangan *software* pembuatan *APP Inventor* sistem *Android* yang digunakan untuk menerima data dari mikrokontroler yang akan dikirim lewat *bluetooth*. Namun di tampilan awal *applIcation* ada beberapa tombol pilihan seperti *enter, instruksion, map, read me* dan *exit*. Masing-masing tombol memiliki fungsi sendiri-sendiri. Tampilan *screen* aplikasi di *Android* dapat dilihat pada gambar 3.15 serta penjelasan *tools screen* aplikasi dapat dilihat pada table 3.1.

Gambar 3.15 *Screen* Awal di Aplikasi *Android*

Tabel 3.1 Tools pada Screen Aplikasi *Android*

Simbol	Penjelasan
ENTER	Untuk masuk ke screen data yang digunakan untuk mengetahui data masuk yang dikirik lewat bluetoth.
MAP	Untuk masuk ke screen google map yang digunakan untuk mengetahui dimana kita berada
INSTRUCTIO N	Untuk masuk ke screen instruction yang menjelaskan tentang bagaimana menggunakan applikasi ini.
READ ME	Untuk masuk ke screen read me yang menjelaskan tentang profil dari pembuat aplikasi.
EXIT	Untuk keluar dari aplikasi

Dalam menggunakan aplikasi *inventor* kita bisa menggunakan dua tipe yaitu *offline* dan *online* untuk penggunaan *offline* kita bisa mengaktifkan *cmd* yang terdapat pada *APP Inventor* setelah itu kita bisa membuka *browser* dan menuliskan *localhost* setelah itu secara otomatis kita dapat langsung menggunakan aplikasi ini sementara untuk *online* kita harus ada jaringan internet dengan menggunakan *email* dari *gmail* masuk ke alamat *browse* ini http://ai2.appinventor.mit.edu/. Tampilan intruksi pada program *Android* dapat dilihat pada gambar 3.16

Gambar 3.16 *Screen* Instruksi

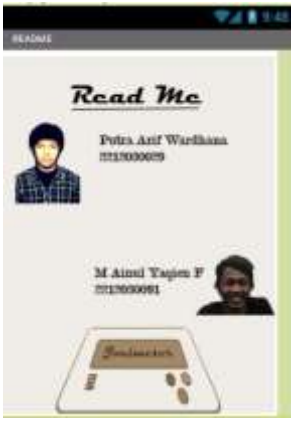
Pada gambar 3.17 adalah tampilan data dari sistem kerja dari program *monitoring* konsumsi energi yang menggunakan *google APP Inventor*

Gambar 3.17 *Screen* Data

Pada gambar 3.18 adalah tampilan *map* dari sistem kerja dari program *monitoring* konsumsi energi yang menggunakan *google APP Inventor*

Gambar 3.18 *Screen Google Map*

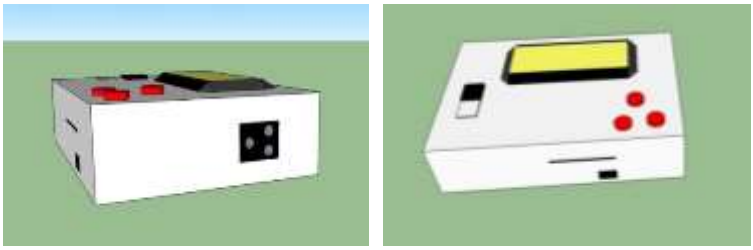
Pada gambar 3.19 adalah *screen read me* dari sistem kerja dari program *monitoring* konsumsi energi yang menggunakan *google APP Inventor*



Gambar 3.19 *Screen Read Me*

3.3 Perancangan Mekanik

Box ini dirancang seperti ini sebagai wadah beberapa perangkat elektronik. Kotak ini berisi sensor arus, sensor tegangan, sensor kecepatan, Arduino Uno dan *MMC Shield*, Baterai Lippo, b*luetooth*. Kotak ini terbuat dari bahan acrylic dengan tebal 2mm dan ukuran panjang 24cm,lebar 15cm dan tinggi 9cm.Sketsa desain dapat dilihat pada gambar 3.20 dan realisasinya dapat dilihat pada gambar 3.21



Gambar 3.20 Sketsa Desain di *Sketchup*

Gambar 3.21 Realisai Desain *Box*

Halaman ini sengaja dikosongkan

BAB IV
PENGUKURAN DAN ANALISA

Pada bab ini akan membahas mengenai hasil pengujian dan analisa atas penyusunan alat untuk ”Rancang Bangun *Joulemeter* untuk Mengukur Konsumsi Energi Kendaraan Listrik ”. Data pengujian sangat diperlukan untuk implementasi dalam dunia nyata. Kinerja suatu sistem sangat dipengaruhi oleh kinerja per bagian dari sistem tersebut.

Pengujian merupakan salah satu langkah yang harus dilakukan untuk mengetahui apakah sistem yang telah dibuat sesuai dengan yang direncanakan. Kesesuaian sistem dengan perencanaan dapat dilihat dari hasil-hasil yang dicapai pada pengujian sistem. Pengujian juga bertujuan untuk mengetahui kelebihan dan kekurangan dari sistem yang telah dibuat. Hasil pengujian tersebut akan dianalisa untuk mengetahui penyebab terjadinya kekurangan atau kesalahan dalam sistem.

Pengujian tersebut meliputi:

1. Pengujian Tegangan *Output Power Supply*
2. Pengujian *Input/Output* Arduino
3. Pembacaan RTC
4. Pembacaan *ACS*712
5. Pembacaan Sensor Tegangan
6. Pengujian *Data Logger*
7. Pengujian LCD
8. Pengujian Sensor Kecepatan
9. Pengujian *Bloetooth*
10. Pengujian *Oddometer*
11. Pengujian *APP Inventor*
12. Uji Alat secara Keseluruhan

4.1 Pengujian Sumber Tegangan pada Arduino

Sumber tegangan dari Arduino, sensor dan *bloetooth* dari *Power Supply*. Pada rangkaian Arduino dibutuhkan tegangan 9 volt DC serta sensor, LCD dan *bloetooth* dibutuhkan tegangan 5 V DC. Pengujian *Power Supply* 5 dan 9 volt DC bertujuan untuk menguji apakah tegangan sumber yang akan digunakan sudah sesuai dengan keperluan. Pengujian *Power Supply* dapat dilihat pada gambar 4.1.

Cara pengujian *Power Supply* dilakukan dengan mengukur tegangan *output Power Supply* saat tanpa beban dan mengukur tegangan *output Power Supply* setelah dibebani Arduino, sensor *ACS*712, Sensor tegangan , sensor *reed switch*, LCD, dan *bloetooth*. Data yang di dapat adalah seperti pada table 4.1.

Gambar 4.1 Pengujian *Power Supply* Keluaran 5 Volt

Dengan hasil pengujian yang didapat diketahui *error* keluaran dari rangkaian *Power Supply* saat tanpa beban dan ada beban. Nilai *error* di dapat dari persamaan 4.1

Persentase *error* diperoleh dengan rumus 4.1:

$$\%Error=\frac{[Tegangan\ Keluaran-Teganagn]}{Tegangan} \text{ x } 100\% \quad \text{.....} \quad (4.1)$$

Tabel 4.1 Hasil Pengujian *Power Supply*

Tegangan	Tegangan Keluaran		Kesalahan (%)	
	Tanpa beban	**Berbeban**	**Tanpa beban**	**Berbeban**
+5 Volt	5.01 Volt	4,94 Volt	0,2 %	0,2%
+9 Volt	8.96 Volt	8.94 Volt	0,7 %	0,4 %

Hasil dari pengujian yang dilakukan menunjukkan bahwa *Power Supply* dapat digunakan sebagai sumber untuk mensuplai peralatan tersebut yang digunakan.

4.2 Pengujian *Input/Output* Arduino Uno

Pengujian Mikrokontroler Arduino Uno bertujuan untuk mengetahui tegangan output pada saat diberi *Input High Voltage* dan *Input Low Voltage* pada setiap 14 pin digital Arduino Uno.

Cara pengujian ini dilakukan pada waktu yang bersamaan ketika Vcc dari mikrokontroler Arduino Uno mendapat *Input* tegangan sebesar 9 volt dari *Power Supply*. Hasil dari pengukuran tegangan tersebut terdapat pada tabel 4.2.

Tabel 4.2 Hasil Pengukuran Tegangan *Output* pada Setiap Pin Arduino

Nomor Pin	High (volt)	Low (volt)
0	4,98	0.00
1	4,98	0.00
2	4,98	0.00
3	4,98	0.00
4	4,98	0.00
5	4,98	0.00
6	4,98	0.00
7	4,98	0.00
8	4,98	0.00
9	4,98	0.00
10	4,98	0.00
11	4,98	0.00
12	4,98	0.00
13	4,98	0.00

Dari hasil data tegangan pada setiap pin Arduino Uno diatas dapat dikatakan bahwa *output* tegangan tersebut mencapai kestabilan pada nilai 4.98 *volt* pada saat pin di*setting* berlogika 1 dan bernilai 0 *volt* saat pin berlogika 0 . hal ini menunjukkan bahwa tiap-tiap pin pada rangkaian Arduino Uno masih dapat bekerja dengan baik.

4.3 Pengujian Pembacaan RTC

Pengujian terhadap RTC (*Real Time Clock*) bertujuan untuk mengetahui data waktu pada RTC yang ditampilkan ke LCD. Pengujian RTC dapat dilihat pada gambar 4.2.

Cara pengujian dilakukan dengan membandingkan jam komputer dengan jam yang ditampilkan pada LCD . Hasil pengujian RTC tertera pada tabel 4.3.

Gambar 4.2 Tampilan LCD dan Jam Komputer pada Pengujian *RTC*

Tabel 4.3 Tabel Pengujian RTC

No	Waktu Sebenarnya	Waktu yang ditampilkan di LCD	Selisih
1	23:47:00	23:46:50	10 detik
2	13:30:00	13:29:50	10 detik
3	10:10:00	10:09:50	10 detik
4	20:30:00	20:29:50	10 detik
5	22:40:00	22:30:50	10 detik

Dengan mengacu data pada table 4.3 didapatkan bahwa selisih antara jam RTC dan jam sesungguhnya (komputer) adalah tetap yaitu 10 detik. Sehingga dapat disimpulkan bahwa RTC dapat digunakan sebagai acuan karena selisih waktunya selalu tetap dan tidak berubah ubah.

4.4. Pengujian Sensor *ACS* 712

Pengujian sensor *ACS* 712 bertujuan untuk mengukur arus pada beban. Pengujian sensor arus *ACS* 712 ini dilakukan dengan cara mengukur arus *output* pada sensor dengan multimeter pada beban. Pengujian sensor arus dapat dilihat pada gambar 4.3 serta data yang didapat adalah seperti pada tabel 4.4.

Persentase *error* diperoleh dengan rumus 4.2:

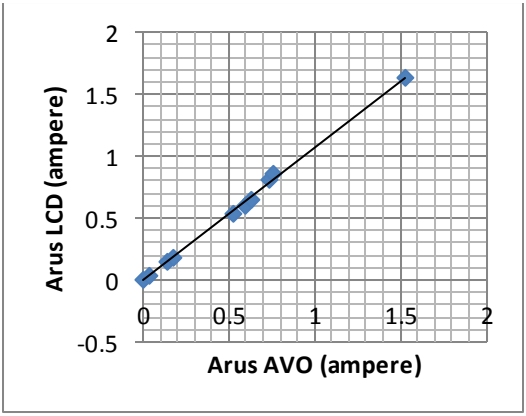
$$\%Error=\frac{[Arus\ pada\ LCD - Arus\ pada\ AVO]}{Arus\ pada\ AVO} \text{ x } 100\% \quad \ldots\ldots\ldots\ldots\ldots\ldots \quad (4.2)$$

Gambar 4.3 Pengujian Sensor Arus

Tabel 4.4 Hasil Pengukuran Sensor Arus

No	Arus pada kalibrator (AVO)	Arus pada mikrokontoler (LCD)	Error(%)
1	0 A	0.04 A	0.4%
2	0.090 A	0.098 A	0.8%
3	0.14 A	0.15 A	1%
4	0.17 A	0.18 A	1%
5	0.53 A	0.54 A	1%
6	0.59 A	0.60 A	1%
7	0.63 A	0.65 A	2%
8	0.73 A	0.80 A	7%
9	0.76 A	0.85 A	9%
10	1.59 A	1.63 A	4%

Grafik liniearisasi sensor arus dengan menampilkan nilai tegangan yang terukur oleh sensor yaitu arus pada kalibrator dan arus pada mikro kontroler. Diagram linieresasi dapat dilihat pada gambar 4.4



Gambar 4.4 Diagram Linearisasi Sensor Arus

Nilai pada pengujian menunjukkan bahwa sensor arus telah berfungsi dengan baik. Dalam keadaan tidak berbeban arus *output* sensor bernilai 0 Ampere. Dan nilai tersebut naik apabila beban bertambah semakin besar.

4.5. Pengujian Sensor Tegangan

Pengujian sensor tegangan bertujuan untuk mengatahui nila ADC yang dikonversi menjadi tegangan pada Arduino dan nilai tegangan ketika diberi beban motor, pengujian sensor tegangan dapat dilihat pada gambar 4.5.

Pengujian sensor arus *ACS* 712 ini dilakukan dengan cara mengukur teganga *output* pada sensor dengan multimeter pada beban. Data yang didapat adalah seperti pada tabel 4.5.

Persentase *error* diperoleh dengan rumus 4.3:

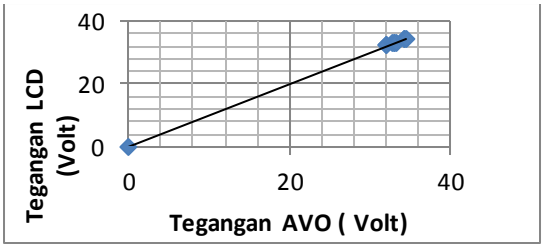
$$\%Error=\frac{[Tegangan\ LCD - Tegangan\ AVO]}{Teganagn\ AVO} \text{ x } 100\% \quad \ldots\ldots\ldots\ldots\ldots\ldots \quad (4.3)$$

Gambar 4.5 Pengujian Sensor Tegangan

Tabel 4.5 Hasil Pengukuran Tegangan

No	Tegangan pada kalibrator (AVO)	Tegangan pada mikrokontoler (LCD)	Error(%)
1	0	0.005	0.5%
2	32.13	32.10	3%
3	32.19	32.24	5%
4	32.84	32.80	4%
5	33	32.97	3%
6	33.08	33.08	0%
7	33.33	33.25	8%
8	34.28	34.32	4%
9	34.43	34.45	2%
10	34.57	34.59	2%

Grafik liniearisasi sensor arus dengan menampilkan nilai tegangan yang terukur oleh sensor yaitu tegangan pada kalibrator dan tegangan pada mikrokontroler. Diagram linieresasi dapat dilihat pada gambar 4.6.



Gambar 4.6 Diagram Linearisasi Sensor Tegangan

Nilai pada pengujian menunjukkan bahwa sensor tegangan telah berfungsi dengan baik. Dalam keadaan tidak berbeban tegangan *output* sensor bernilai 0 Volt. Dan nilai tersebut naik apabila beban bertambah semakin kecil.

4.6. Pengujian *Data Logger*

Pengujian *Data Logger* bertujuan untuk mengetahui besar interval waktu pengambilan data antara data pertama dan data kedua. Pengujian *Data Logger* ini dilakukan dengan cara melihat pembacaan hasil data kartu *SDCard* pada komputer . Data yang didapat adalah seperti pada gamba 4.7.

	A	B	C	D	E	F	G	H	I	J
1	Tanggal	Waktu	Arus(A)	Tegangan(V)	Daya(wh)	Joule	Total(Daya)	Total(Joule)	Jarak(m)	Velocity(km/h)
2	02-07-2015	0:23:14	0.09A	0.27V	0.00wh	0.02J	0.00wh	0.02J	0.00m	0kph
3	02-07-2015	0:23:16	0.09A	0.22V	0.00wh	0.02J	0.00wh	0.04J	0.00m	0kph
4	02-07-2015	0:23:18	0.09A	0.22V	0.00wh	0.02J	0.00wh	0.06J	0.00m	0kph
5	02-07-2015	0:23:20	0.09A	0.22V	0.00wh	0.02J	0.00wh	0.08J	0.00m	0kph
6	02-07-2015	0:23:22	0.09A	0.22V	0.00wh	0.02J	0.00wh	0.10J	0.00m	0kph
7	02-07-2015	0:23:24	0.09A	0.22V	0.00wh	0.02J	0.00wh	0.12J	0.00m	0kph
8	02-07-2015	0:23:26	0.09A	0.22V	0.00wh	0.02J	0.00wh	0.14J	0.00m	0kph
9	02-07-2015	0:23:28	0.09A	0.22V	0.00wh	0.02J	0.00wh	0.16J	0.00m	0kph
10	02-07-2015	0:23:30	0.09A	0.22V	0.00wh	0.02J	0.00wh	0.18J	0.00m	0kph
11	02-07-2015	0:23:32	0.09A	0.22V	0.00wh	0.02J	0.00wh	0.20J	0.00m	0kph
12	02-07-2015	0:23:34	0.09A	0.22V	0.00wh	0.02J	0.00wh	0.21J	0.00m	0kph
13	02-07-2015	0:23:36	0.09A	0.22V	0.00wh	0.02J	0.00wh	0.23J	0.00m	0kph
14	02-07-2015	0:23:38	0.09A	0.22V	0.00wh	0.02J	0.00wh	0.25J	0.00m	0kph
15	02-07-2015	0:23:40	0.09A	0.22V	0.00wh	0.02J	0.00wh	0.27J	0.00m	0kph
16	02-07-2015	0:23:42	0.09A	0.22V	0.00wh	0.02J	0.00wh	0.29J	0.00m	0kph
17	02-07-2015	0:23:44	0.09A	0.22V	0.00wh	0.02J	0.00wh	0.31J	0.00m	0kph
18	02-07-2015	0:23:46	0.09A	0.27V	0.00wh	0.02J	0.00wh	0.33J	0.00m	0kph
19	02-07-2015	0:23:48	0.09A	0.22V	0.00wh	0.02J	0.00wh	0.35J	0.00m	0kph

Gambar 4.7 Hasil Pengujian *Data Logger*

Nilai pada pengujian menunjukkan bahwa *Data Logger* telah berfungsi dengan baik terlihat pada gambar 4.7, bahwa selisih interval waktu sebesar 2 detik.Interval waktu bisa diatur sesuai dengan keinginan kita dengan mengatur *delay* pada program.

4.7. Pengujian LCD

Pengujian LCD dilakukan dengan cara memberikan program dengan perintah menampilkan karakter pada LCD seperti pada gambar 4.8 dan hasil dari *running* program menampilkan karakter seperti program di atas ditampilkan pada gambar 4.9

```
buttonState = digitalRead(buttonPin);
buttonState1 = digitalRead(buttonPin1);
if (buttonState == HIGH && buttonState1 == LOW ){
  lcd.clear();
  lcd.setCursor(0,0);
  lcd.print("Arus=");
  lcd.print(arus);
  lcd.print("A");
  lcd.setCursor(0,1);
  lcd.print("Tegangan=");
  lcd.print(tegangan);
  lcd.print("V");
  delay(200);
 }
```

Gambar 4.8 *Listing* Program Pengujian LCD

Gambar 4.9 Hasil Pengujian LCD

4.8 Pengujian Sensor Kecepatan

Pengujian kecepatan digunakan untuk menentukan seberapa besar kecepatan kendaraan melaju. Pengujian kecepataan dilakukan

dengan dengan cara mengukur waktu yang ditempuh kendaraan ketika melintasi lintasan lurus sepanjang 50 meter. Hasil pengujian dapat dilihat pada pada tabel.4.6.

Persentase *error* diperoleh dengan rumus 4.4:

$$\%Error=\frac{[kecepatan\ pada\ LCD-kecepatan\ pada\ sebenarnya]}{kecepatan\ pada\ sebenarnya} \text{ x } 100\% \quad \text{..... (4.4)}$$

Tabel 4.6 Pengujian Sensor Kecepatan

No	Kecepatan Sebenarnya	Kecepatan pada mikrokontroler (LCD)	Error(%)
1	9 km/s	8 km/s	11 %
2	6,4 km/s	6 km/s	6 %
3	4,5 km/s	4 km/s	5 %
4	3,9 km/s	4 km/s	1 %
5	7,9 km/s	7 km/s	9 %

Nilai pada pengujian menunjukkan bahwa sensor kecepatan memiliki *error* rata-rata 6,4 % , sehingga sensor tersebut dapat digunakan sebagai indikator kecepatan.

4.9 Pengujian *Bloetooth*

Pengujian *Bloetooth* bertujuan untuk mengetahui seberapa jauh *Bloetooth* ini bias mengirim data dan menerima data dengan baik. Pengujian *Bloetooth* ini dengan cara melihat koneksi antara *bloetooth* dengan aplikasi *Android* pada jarak tertentu pembacaan. Hasil pengujian dapat dilihat pada tabel.4.7.

Tabel 4.7 Pengujian Jarak Koneksi *Bluetooth*

No	Jarak *Bluetooth* (meter)	Kondisi *Bluetooh*
1	3	*Connect*
2	6	*Connect*
3	9	*Connect*
4	10	*Connect*
5	15	*Disconnect*
6	18	*Disconnect*

Dari data pengujian pada tabel 4.7 dapat didisimpulkan bahwa jarak maksimal *Bluetooth* yang kami gunakan dapat mengirim data dan

menerima data adalah antara 0 – 10 meter, setelah jarak itersebut *Bluetooth* tidak dapat bekerja maksimal.

4.10 Pengujian *Oddometer*

Oddometer adalah sebuah alat yang digunakan sebagai penunjuk jarak tempuh kendaraan.Pengujian *oddometer* betujuan untuk mengetahui seberapa jarak yang di tempuh oleh kendaraan. Pengujian dilakukan dengan cara membandingkan jarak sebenarnya dengan jarak yang ditampilkan pada LCD. Hasil pengujian dapat dilihat pada pada tabel.4.8.

Persentase *error* diperoleh dengan rumus 4.5:

$$\%Error=\frac{[jarak\ pada\ LCD - jaral\ pada\ sebenarnya]}{jarak\ pada\ sebenarnya} \text{ x } 100\% \quad \ldots\ldots\ldots\ldots \quad (4.5)$$

Tabel 4.8 Pengujian *Oddometer*

No	Jarak Sebenarnya	Jarak Mikrokontroler (LCD)	Error (%)
1	5 m	4.87 m	2 %
2	10 m	9.74 m	2 %
3	15 m	14,61 m	2 %
4	20 m	19,48 m	2 %
5	25 m	24,35 m	2 %
6	30 m	29,22 m	2 %
7	35 m	34,09 m	2 %
8	40 m	38,96 m	2 %
9	45 m	43,83 m	2 %
10	50 m	45.77 m	2 %

Nilai pada pengujian menunjukkan bahwa sensor kecepatan memiliki *error* rata-rata 2 % , sehingga sensor tersebut dapat digunakan sebagai indikator jarak.

4.11 Pengujian Aplikasi *Android*

Pengujian ini dimaksudkan untuk mengetahui kecepatan pengiriman dan penerimaan data yang dikirim oleh mikrokontoler ke ponsel berbasis *Android*. Pengujian dilakukan dengan cara menghitung waktu mulai pengiriman oleh mikrokontroler yang ditampilkan di LCD sampai dengan penerimaan data pada aplikasi ponsel berbasis *Android*..

. Untuk menghitung waktu tersebut digunakan alat bantu berupa *stopwatch*. Hasil pengujian dapat dilihat pada tabel 4.9 berikut.

Tabel 4.9 Pengujian Waktu Pengiriman Data

Percobaan ke	Waktu (detik)
1	1
2	1
3	1
4	1
5	1

Waktu pada pengujian tersebut sudah termasuk waktu pembacaan ADC dan delay yang ada pada mikrokontroler. Sehingga pengiriman membutuhkan waktu rata-rata 1 detik. Waktu tersebut juga tergantung pada kekuatan sinyal pada *bloetooth* yang digunakan. Apabila kekuatan sinyal lemah jika didalam suatu ruangan, maka waktu pengiriman bisa lebih dari 1 detik.

4.12 Pengujian Keseluruhan Sistem

Saat dilakukan pengujian alat secara keseluruhan, dilakukan seluruh koordinasi semua sensor dan tampilan pada ponsel berbasis *Android*.

Langkah pertama menghubungkan setiap komponen *joulemeter* yaitu sepeda listrik beserta *joulemeter* panel yang didalamnya terdapat mikrokontroler, sensor dan *bloetooth*. Pengujian dilakukan dengan dua kondisi, yaitu :

1. Kondisi Sepeda Listrik *Statis*

Kondisi Sepeda Listrik Statis dilakukan dengan menyalakan sistem kelistrikan pada sepeda listrik dan *joulemeter* panel serta mengatur kecepatan sepeda listrik yang diinginkan dengan jarak tempuh 0 Km. Kondisi sepeda listrik *statis* ini dapat dilihat pada gambar 4.10 dan 4.11.

Gambar 4.10 Tampilan LCD dan Ponsel Berbas *Android* pada *Joulemeter* Ketika Kondisi Sepeda Listrik *Statis*

Gambar 4.11 Kondisi Sepeda Listrik *Statis*

2. Kondisi Sepeda Listrik *Dinamis*

Kondisi Sepeda Listrik *Dinamis* dilakukan dengan menyalakan sistem kelistrikan pada sepeda listrik beserta *joulemeter* panel, mengatur kecepatan, jarak dan sepeda listrik di beri beban sepeda listrik yang diinginkan. Beban pada sepeda listrik yaitu berat badan manusia yang berbeda-beda. Kondisi sepeda listrik *dinamis* ini dapat dilihat pada gambar 4.12 dan 4.13. Data pengujian keseluruan sistem dapat dilihat pada table 4.10.

Gambar 4.12 Tampilan LCD dan Ponsel Berbas *Android* pada *Joulemeter* Ketika Kondisi Sepeda Listrik *Dinamis.*

Gambar 4.13 Kondisi Sepeda Listrik *Dinamis*

Persentase *error* diperoleh dengan rumus 4.6:

$$\%Error = \frac{[Joule\ listrik - joule\ mekanik]}{joule\ mekanik} \text{ x } 100\% \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad (4.6)$$

Nilai joule mekanik didapatkan denga rumus 4.7:

$$\text{Joule mekanik} = \frac{1}{2} \text{ x m x } v^2 \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad (4.7)$$

Keterangan:
m = massa (Kg)
v = kecepatan (m/s)

Nilai joule Listrik didapatkan denga rumus 4.8:

$$\text{Joule Listrik} = \text{Wh x } 3600 \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad (4.8)$$

Keterangan:
Wh = *Watt hour* (Wh)

Tabel 4.10 Pengujian Keseluruhan Sistem

Berat	Kecepatan (Km/jam)	Joule Mekanik (Wh)	Joule Listrik (Wh)	Error %
90	17	994,05	841,05	10
90	18	1080,45	714,64	30
105	16	10063	880,71	17
100	19	1352	1166,13	13
95	17	1049	829,62	20

Berdasarkan pada pengujian alat keseluruhan diatas telah didapatkan presentasi *error* rata-rata 20 %.

Halaman ini sengaja dikosongkan

BAB V
PENUTUP

Dari hasil yang telah didapatkan selama proses pembuatan alat untuk Tugas Akhir ini, maka dapat ditarik beberapa kesimpulan dan saran agar nantinya bisa bermanfaat bagi siapa saja yang membacanya.

5.1 Kesimpulan

Hasil dari pengujian serta analisa data dari Joulmeter (Alat yang digunakan untuk mengetahui besar usaha kendaraan listrik) adalah sebagai berikut :

1. Sensor Arus yang digunakankan memiliki persentasi *error* rata-rata 2,7 % terhadap pengukuran kalibrator (AVO).
2. Sensor Tegangan yang digunakankan memiliki persentasi *error* rata-rata 3,15 % terhadap pengukuran kalibrator (AVO).
3. Sensor Kecepatan yang digunakankan memiliki persentasi *error* rata-rata 6,4% terhadap perhitungan teori.
4. Alat ini memiliki persentasi *error* rata-rata 20% terhadap perhitungan secara teori.

5.2 Saran

Saran untuk membuat alat ini memiliki kinerja lebih baik maka perlu diperhatikan:

1. Diperlukan sistem *interface* data yang bisa digunakan dalam jarak jangkau yang jauh supaya apabila ingin mengetahui data masuk kita tidak usah repot-repot mencabut *SDCard*.
2. Sistem *Android* dibuat lebih banyak lagi pilihannya seperti sistem navigasi kondisi baterai dan lainnya.

Halaman ini sengaja dikosongkan

DAFTAR PUSTAKA

[1] ….., 2012. ***Pengertian Dan Kelebihan Mikrokontroler,*** http://elektronika-dasar.web.id/artikel-elektronika/pengertian-dan-kelebihan-mikrokontroler.html, 23 April 2015

[2] …….., 2013. **Liquid Crystal Display (LCD) 16 x 2,** http://baskarapunya.blogspot.com/2013/01/liquid-crystal-display-lcd-16-x-2.html,.29 April 2015

[3] Siswoyo,Bambang, 2012. ***Belajar Arduino: pengantar,*** http://bsiswoyo.lecture.ub.ac.id/2012/06/belajar-Arduino-pengantar/, 02 Mei 2015

[4] Irwan Arifin, ***Elektronika 1,*** Gunadarma, 2004.

[5] Jazi Eko Istiyanto, ***Pengantar Elektronika dan Instrumentasi***, Andi Yogyakarta, Yogyakarta, 2013.

[6] . . . ., *DS1307 Serial Real Time Clock, datasheet* tentang RTC, *URL: https://www.sparkfun.com/datasheets/components/DS1307*

[7] . . . ., *User Manual Fargo Maestro 100 Fargo Maestro 20*, Fargo *Telecom* O/B Fargo *ServICes* (H.K) Ltd.

[8] . . . ., *ATCommands Interface Guide*, Wavecom, 2002.

Halaman ini sengaja dikosongkan

LAMPIRAN

LAMPIRAN A
Listing Program

Program Pada *Arduino IDE*

```
#include <Wire.h>
#include "RTClib.h"
#include <SPI.h>
#include <SD.h>
#include <LiquidCrystal.h>
LiquidCrystal lcd (8,7,6,5,4,3);
RTC_DS1307 RTC;
const int chipSelect = 10;
//
String tanggal;
String waktu;
String aruscsv;
String tegangancsv;
String dayacsv;
String totaldayacsv;
String joulecsv;
String totaljoulecsv;
String jarakcsv;
String velocitycsv;
//variabel
int dataADC1;
int dataADC2;
float arus;
float tegangan;
float daya;
float sumdaya;
float joule;
float sumjoule;
float x;
float y;
float dayadroid;
float jouledroid;
```

```
float distancedroid;
int velocitydroid;
#define reed A0
int reedVal;
int maxDebounceTicks = 200;
int currentDebounceTicks;
int time = 0;
float radius = 0.00017;
float circumference;
float velocity = 0.00;
float distance = 0.00;
int velocity1;
float distance1;
//pushbutton
const int buttonPin = 2;
const int buttonPin1 = 9;
int buttonState = 0;
int buttonState1 = 0;
void setup()
{
Serial.begin(9600);
currentDebounceTicks = maxDebounceTicks;
circumference = 2 * 3.14 * radius;
pinMode(reed, INPUT);
cli();
TCCR1A = 0;
 // set entire TCCR1B register to 0
TCCR1B = 0;

 // initialize counter value to 0;
TCNT1  = 0;
// set timer count for 1khz increments
// = (16*10^6) / (1000*8) - 1
OCR1A = 1999;
// turn on CTC mode
TCCR1B |= (1 << WGM12);
// Set CS11 bit for 8 prescaler
TCCR1B |= (1 << CS11);
// enable timer compare interrupt
TIMSK1 |= (1 << OCIE1A);
```

```
sei();
Wire.begin();
RTC.begin();
lcd.begin(16,2);
//pushbutton
pinMode(buttonPin, INPUT_PULLUP);
pinMode(buttonPin1, INPUT_PULLUP);
//datalog
pinMode(10, OUTPUT);
SD.begin(chipSelect);
File dataFile = SD.open("joule1.txt", FILE_WRITE);
if(dataFile)
{
  dataFile.print("Tanggal");
  dataFile.print(" ");
  dataFile.print("Waktu");
  dataFile.print(" ");
  dataFile.print("Arus(A)");
  dataFile.print(" ");
  dataFile.print("Tegangan(V)");
  dataFile.print(" ");
  dataFile.print("Daya(wh)");
  dataFile.print(" ");
  dataFile.print("Joule");
  dataFile.print(" ");
  dataFile.print("Total(Daya)");
  dataFile.print(" ");
  dataFile.print("Total(Joule)");
  dataFile.print(" ");
  dataFile.print("Jarak(m)");
  dataFile.print(" ");
  dataFile.println("Velocity(km/h)");
  dataFile.close();
}
}
void loop()
{
 String dataString="";
 arus1();
 tegangan1();
```

```
daya1();
daya2();
joule1();
joule2();
datadroid();
kecepatan();

  dataString=String(arus);
  dataString=String(tegangan);
  dataString=String(daya);
  dataString=String(joule);
  dataString=String(x);
  dataString=String(y);
  dataString=String(distance1);
  dataString=String(velocity1);
DateTime now = RTC.now();
File dataFile = SD.open("joule1.txt", FILE_WRITE);
  //
  if(dataFile){
  dataFile.print(now.year(), DEC);
  dataFile.print('/');
  dataFile.print(now.month(), DEC);
  dataFile.print('/');
  dataFile.print(now.day(), DEC);
  dataFile.print(' ');
  dataFile.print(now.hour(), DEC);
  dataFile.print(':');
  dataFile.print(now.minute(), DEC);
  dataFile.print(':');
  dataFile.print(now.second(), DEC);
  dataFile.print(' ');
  dataFile.print(arus);
  dataFile.print("A");
  dataFile.print(' ');
  dataFile.print(tegangan);
  dataFile.print("V");
  dataFile.print(' ');
  dataFile.print(daya);
  dataFile.print("wh");
  dataFile.print(' ');
```

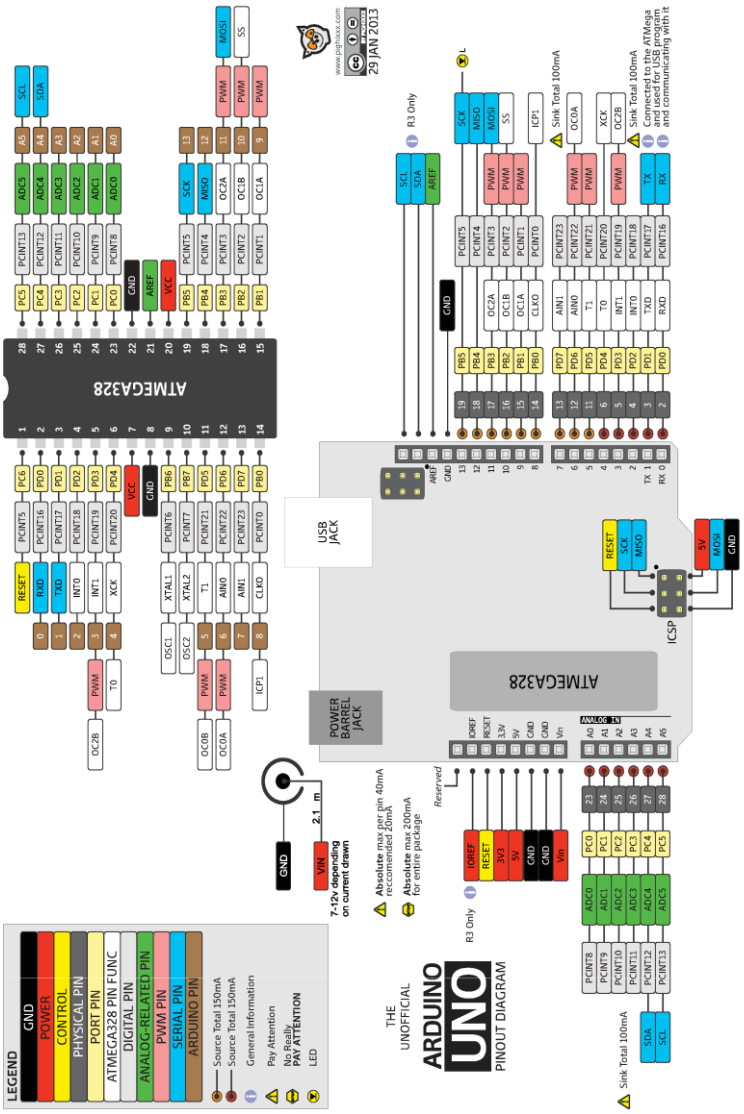
```
    dataFile.print(joule);
    dataFile.print("j");
    dataFile.print(' ');
    dataFile.print(x);
    dataFile.print("wh");
    dataFile.print(' ');
    dataFile.print(y);
    dataFile.print("j");
    dataFile.print(' ');
    dataFile.print(distance1);
    dataFile.print("m");
    dataFile.print(' ');
    dataFile.print(velocity1);
    dataFile.println("kph");
    dataFile.close();
    delay(1000);
    }
  //
buttonState = digitalRead(buttonPin);
buttonState1 = digitalRead(buttonPin1);
if (buttonState == HIGH && buttonState1 == LOW ){
  lcd.clear();
  lcd.setCursor(0,0);
  lcd.print("Arus=");
  lcd.print(arus);
  lcd.print("A");
  lcd.setCursor(0,1);
  lcd.print("Tegangan=");
  lcd.print(tegangan);
  lcd.print("V");
  delay(200);
 }
else if (buttonState == LOW && buttonState1 == HIGH){

     lcd.clear();
     lcd.setCursor(0,0);
     lcd.print("Kecepatan=");
     lcd.print(velocity1);
     lcd.setCursor(0,1);
     lcd.print("Jarak=");
```

```
    lcd.print(distance1);
    delay(200);
  }
else if (buttonState == HIGH && buttonState1 == HIGH){
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("SumKwh=");
    lcd.print(x);
    lcd.setCursor(0,1);
    lcd.print("SumJoule=");
    lcd.print(y);
    delay(200);
}
else{
   lcd.clear();
   lcd.setCursor(3,0);
  lcd.print("JOULEMETER");
  lcd.setCursor(0, 1);
  lcd.print(now.hour(), DEC);
  lcd.print(":");
  lcd.print(now.minute(), DEC);
  lcd.print(":");
  lcd.print(now.second(), DEC);
  lcd.print(" ");

  lcd.setCursor(8, 1);
  lcd.print(now.day(), DEC);
  lcd.print("/");
  lcd.print(now.month(), DEC);
  lcd.print("/");
  lcd.print(now.year(), DEC);
  lcd.print("");

  delay(200);
}
}
void arus1()
{
 dataADC1 = analogRead(A1);
 arus=abs((dataADC1*0.047627)-23.0938);
```

```
}
void tegangan1()
{
  dataADC2 = analogRead(A2);
  tegangan =abs(( dataADC2*0.054052)-0.18582);

}
void daya1()
{
  daya =abs ((tegangan*arus)*0.00027);
}
void joule1()
{
 joule=daya*3600;
}
void daya2()
{
  sumdaya = ((tegangan*arus)*0.00027);
  x+= sumdaya;
}
void joule2()
{
  sumjoule=daya*3600;
  y+=sumjoule;
}
void datadroid()
{
    dayadroid = x;
    distancedroid =(int (distance*1000));
    distancedroid = map(distancedroid,0,10000,5000,15000);
    velocitydroid = velocity1;
    velocitydroid = map(velocitydroid,0,30,100,130);
    Serial.print(' ');
    Serial.println(dayadroid);
    delay(200);
    Serial.print(' ');
    Serial.println(distancedroid);
    delay(200);
    Serial.print(' ');
    Serial.println(velocitydroid);
```

```
    delay(200);
}

ISR(TIMER1_COMPA_vect) {
  reedVal = digitalRead(reed);
  if (reedVal) {
    // wait the given number of ticks, before calculating the velocity
    if (currentDebounceTicks == 0) {
      // circumference in kilometers, time in hours
      velocity = circumference/(float(time)/1000/3600);
      time = 0;
      currentDebounceTicks = maxDebounceTicks;
      distance+=circumference;
    } else {
      if (currentDebounceTicks > 0) {
        currentDebounceTicks -= 1;
      }
    }
  } else {
    if (currentDebounceTicks > 0) {
      currentDebounceTicks -= 1;
    }
  }

if (time > 800) {
    // set velocity to 0 when tire is still for 2 seconds
    velocity = 0;
  } else {
    time += 1;
  }
}
void kecepatan()
{
 velocity1=(int(velocity));
 distance1=(float(distance*1000)/3.1);
}
```

LAMPIRAN B
Datasheet Arduino Uno



ACS712

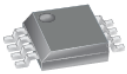
Fully Integrated, Hall Effect-Based Linear Current Sensor with 2.1 kVRMS Voltage Isolation and a Low-Resistance Current Conductor

Features and Benefits

- Low-noise analog signal path
- Device bandwidth is set via the new FILTER pin
- 5 µs output rise time in response to step input current
- 80 kHz bandwidth
- Total output error 1.5% at T_A = 25°C
- Small footprint, low-profile SOIC8 package
- 1.2 mΩ internal conductor resistance
- 2.1 kV_{RMS} minimum isolation voltage from pins 1-4 to pins 5-8
- 5.0 V, single supply operation
- 66 to 185 mV/A output sensitivity
- Output voltage proportional to AC or DC currents
- Factory-trimmed for accuracy
- Extremely stable output offset voltage
- Nearly zero magnetic hysteresis
- Ratiometric output from supply voltage

TÜV America Certificate Number: U8V 06 05 54214 010

Package: 8 Lead SOIC (suffix LC)



Approximate Scale 1:1

Description

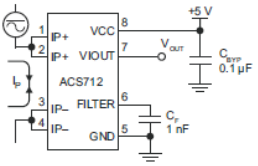
The Allegro® ACS712 provides economical and precise solutions for AC or DC current sensing in industrial, commercial, and communications systems. The device package allows for easy implementation by the customer. Typical applications include motor control, load detection and management, switched-mode power supplies, and overcurrent fault protection.

The device consists of a precise, low-offset, linear Hall sensor circuit with a copper conduction path located near the surface of the die. Applied current flowing through this copper conduction path generates a magnetic field which is sensed by the integrated Hall IC and converted into a proportional voltage. Device accuracy is optimized through the close proximity of the magnetic signal to the Hall transducer. A precise, proportional voltage is provided by the low-offset, chopper-stabilized BiCMOS Hall IC, which is programmed for accuracy after packaging.

The output of the device has a positive slope (>$V_{IOUT(Q)}$) when an increasing current flows through the primary copper conduction path (from pins 1 and 2, to pins 3 and 4), which is the path used for current sensing. The internal resistance of this conductive path is 1.2 mΩ typical, providing low power

Continued on the next page…

Typical Application



Application 1. The ACS712 outputs an analog signal, V_{OUT}, that varies linearly with the uni- or bi-directional AC or DC primary sensed current, I_P, within the range specified. C_F is recommended for noise management, with values that depend on the application.

Description (continued)

loss. The thickness of the copper conductor allows survival of the device at up to 5× overcurrent conditions. The terminals of the conductive path are electrically isolated from the sensor leads (pins 5 through 8). This allows the ACS712 current sensor to be used in applications requiring electrical isolation without the use of opto-isolators or other costly isolation techniques.

The ACS712 is provided in a small, surface mount SOIC8 package. The leadframe is plated with 100% matte tin, which is compatible with standard lead (Pb) free printed circuit board assembly processes. Internally, the device is Pb-free, except for flip-chip high-temperature Pb-based solder balls, currently exempt from RoHS. The device is fully calibrated prior to shipment from the factory.

Selection Guide

Part Number	Packing*	T_A (°C)	Optimized Range, I_P (A)	Sensitivity, Sens (Typ) (mV/A)
ACS712ELCTR-05B-T	Tape and reel, 3000 pieces/reel	–40 to 85	±5	185
ACS712ELCTR-20A-T	Tape and reel, 3000 pieces/reel	–40 to 85	±20	100
ACS712ELCTR-30A-T	Tape and reel, 3000 pieces/reel	–40 to 85	±30	66

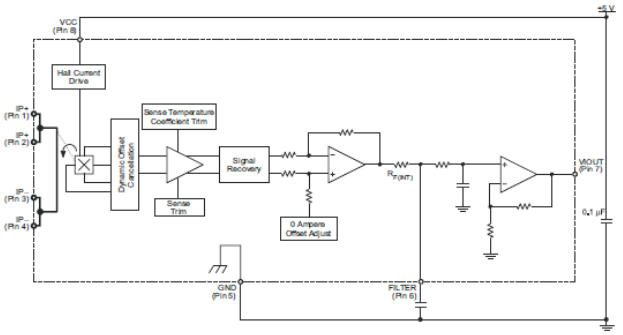
*Contact Allegro for additional packing options.

Absolute Maximum Ratings

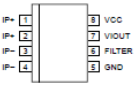
Characteristic	Symbol	Notes	Rating	Units
Supply Voltage	V_{CC}		8	V
Reverse Supply Voltage	V_{RCC}		–0.1	V
Output Voltage	V_{IOUT}		8	V
Reverse Output Voltage	V_{RIOUT}		–0.1	V
Reinforced Isolation Voltage	V_{ISO}	Pins 1-4 and 5-8; 60 Hz, 1 minute, T_A=25°C	2100	V
		Voltage applied to leadframe (Ip+ pins), based on IEC 60950	184	V_{peak}
Basic Isolation Voltage	$V_{ISO(bsc)}$	Pins 1-4 and 5-8; 60 Hz, 1 minute, T_A=25°C	1500	V
		Voltage applied to leadframe (Ip+ pins), based on IEC 60950	354	V_{peak}
Output Current Source	$I_{IOUT(Source)}$		3	mA
Output Current Sink	$I_{IOUT(Sink)}$		10	mA
Overcurrent Transient Tolerance	I_P	1 pulse, 100 ms	100	A
Nominal Operating Ambient Temperature	T_A	Range E	–40 to 85	°C
Maximum Junction Temperature	T_J(max)		165	°C
Storage Temperature	T_{stg}		–65 to 170	°C

Parameter	Specification
Fire and Electric Shock	CAN/CSA-C22.2 No. 60950-1-03 UL 60950-1:2003 EN 60950-1:2001

Functional Block Diagram



Pin-out Diagram



Terminal List Table

Number	Name	Description
1 and 2	IP+	Terminals for current being sensed; fused internally
3 and 4	IP–	Terminals for current being sensed; fused internally
5	GND	Signal ground terminal
6	FILTER	Terminal for external capacitor that sets bandwidth
7	VIOUT	Analog output signal
8	VCC	Device power supply terminal

COMMON OPERATING CHARACTERISTICS[1] over full range of T_A, C_F = 1 nF, and V_{CC} = 5 V, unless otherwise specified

Characteristic	Symbol	Test Conditions	Min.	Typ.	Max.	Units
ELECTRICAL CHARACTERISTICS						
Supply Voltage	V_{CC}		4.5	5.0	5.5	V
Supply Current	I_{CC}	V_{CC} = 5.0 V, output open	–	10	13	mA
Output Capacitance Load	C_{LOAD}	VIOUT to GND	–	–	10	nF
Output Resistive Load	R_{LOAD}	VIOUT to GND	4.7	–	–	kΩ
Primary Conductor Resistance	$R_{PRIMARY}$	T_A = 25°C	–	1.2	–	mΩ
Rise Time	t_r	I_P = I_P(max), T_A = 25°C, C_{OUT} = open	–	5	–	µs
Frequency Bandwidth	f	–3 dB, T_A = 25°C; I_P is 10 A peak-to-peak	–	80	–	kHz
Nonlinearity	E_{LIN}	Over full range of I_P	–	1.5	–	%
Symmetry	E_{SYM}	Over full range of I_P	98	100	102	%
Zero Current Output Voltage	$V_{IOUT(Q)}$	Bidirectional; I_P = 0 A, T_A = 25°C	–	V_{CC} x 0.5	–	V
Power-On Time	t_{PO}	Output reaches 90% of steady-state level, T_J = 25°C, 20 A present on leadframe	–	35	–	µs
Magnetic Coupling[2]			–	12	–	G/A
Internal Filter Resistance[3]	$R_{F(INT)}$			1.7		kΩ

[1]Device may be operated at higher primary current levels, I_P, and ambient, T_A, and internal leadframe temperatures, T_A, provided that the Maximum Junction Temperature, T_J(max), is not exceeded.

[2]1G = 0.1 mT.

[3]$R_{F(INT)}$ forms an RC circuit via the FILTER pin.

COMMON THERMAL CHARACTERISTICS[1]

			Min.	Typ.	Max.	Units
Operating Internal Leadframe Temperature	T_A	E range	–40	–	85	°C
					Value	Units
Junction-to-Lead Thermal Resistance[2]	$R_{\theta JL}$	Mounted on the Allegro ASEK 712 evaluation board			5	°C/W
Junction-to-Ambient Thermal Resistance	$R_{\theta JA}$	Mounted on the Allegro 85-0322 evaluation board, includes the power consumed by the board			23	°C/W

[1]Additional thermal information is available on the Allegro website.

[2]The Allegro evaluation board has 1500 mm² of 2 oz. copper on each side, connected to pins 1 and 2, and to pins 3 and 4, with thermal vias connecting the layers. Performance values include the power consumed by the PCB. Further details on the board are available from the Frequently Asked Questions document on our website. Further information about board design and thermal performance also can be found in the Applications Information section of this datasheet.

x05B PERFORMANCE CHARACTERISTICS T_A = –40°C to 85°C[1], C_F = 1 nF, and V_{CC} = 5 V, unless otherwise specified

Characteristic	Symbol	Test Conditions	Min.	Typ.	Max.	Units
Optimized Accuracy Range	I_P		–5	–	5	A
Sensitivity	Sens	Over full range of I_P, T_A = 25°C	180	185	190	mV/A
Noise	$V_{NOISE(PP)}$	Peak-to-peak, T_A = 25°C, 185 mV/A programmed Sensitivity, C_F = 47 nF, C_{OUT} = open, 2 kHz bandwidth	–	21	–	mV
Zero Current Output Slope	$\Delta I_{OUT(Q)}$	T_A = –40°C to 25°C	–	–0.26	–	mV/°C
		T_A = 25°C to 150°C	–	–0.08	–	mV/°C
Sensitivity Slope	ΔSens	T_A = –40°C to 25°C	–	0.054	–	mV/A/°C
		T_A = 25°C to 150°C	–	–0.008	–	mV/A/°C
Total Output Error[2]	E_{TOT}	I_P =±5 A, T_A = 25°C	–	±1.5	–	%

[1]Device may be operated at higher primary current levels, I_P, and ambient temperatures, T_A, provided that the Maximum Junction Temperature, $T_{J(max)}$, is not exceeded.

[2]Percentage of I_P, with I_P = 5 A. Output filtered.

x20A PERFORMANCE CHARACTERISTICS T_A = –40°C to 85°C[1], C_F = 1 nF, and V_{CC} = 5 V, unless otherwise specified

Characteristic	Symbol	Test Conditions	Min.	Typ.	Max.	Units
Optimized Accuracy Range	I_P		–20	–	20	A
Sensitivity	Sens	Over full range of I_P, T_A = 25°C	96	100	104	mV/A
Noise	$V_{NOISE(PP)}$	Peak-to-peak, T_A = 25°C, 100 mV/A programmed Sensitivity, C_F = 47 nF, C_{OUT} = open, 2 kHz bandwidth	–	11	–	mV
Zero Current Output Slope	$\Delta I_{OUT(Q)}$	T_A = –40°C to 25°C	–	–0.34	–	mV/°C
		T_A = 25°C to 150°C	–	–0.07	–	mV/°C
Sensitivity Slope	ΔSens	T_A = –40°C to 25°C	–	0.017	–	mV/A/°C
		T_A = 25°C to 150°C	–	–0.004	–	mV/A/°C
Total Output Error[2]	E_{TOT}	I_P =±20 A, T_A = 25°C	–	±1.5	–	%

[1]Device may be operated at higher primary current levels, I_P, and ambient temperatures, T_A, provided that the Maximum Junction Temperature, T_J(max), is not exceeded.

[2]Percentage of I_P, with I_P = 20 A. Output filtered.

x30A PERFORMANCE CHARACTERISTICS T_A = –40°C to 85°C[1], C_F = 1 nF, and V_{CC} = 5 V, unless otherwise specified

Characteristic	Symbol	Test Conditions	Min.	Typ.	Max.	Units
Optimized Accuracy Range	I_P		–30	–	30	A
Sensitivity	Sens	Over full range of I_P , T_A = 25°C	64	66	68	mV/A
Noise	$V_{NOISE(PP)}$	Peak-to-peak, T_A = 25°C, 66 mV/A programmed Sensitivity, C_F = 47 nF, C_{OUT} = open, 2 kHz bandwidth	–	7	–	mV
Zero Current Output Slope	$\Delta I_{OUT(Q)}$	T_A = –40°C to 25°C	–	–0.35	–	mV/°C
		T_A = 25°C to 150°C	–	–0.08	–	mV/°C
Sensitivity Slope	ΔSens	T_A = –40°C to 25°C	–	0.007	–	mV/A/°C
		T_A = 25°C to 150°C	–	–0.002	–	mV/A/°C
Total Output Error[2]	E_{TOT}	I_P = ±30 A, T_A = 25°C	–	±1.5	–	%

[1]Device may be operated at higher primary current levels, I_P, and ambient temperatures, T_A, provided that the Maximum Junction Temperature, T_J(max), is not exceeded.

[2]Percentage of I_P, with I_P = 30 A. Output filtered.

ABSOLUTE MAXIMUM RATINGS[1][2]

Supply Voltage			28V
Supply Current (Zener Options)			25 mA
Collector Voltage			28V
Differential Input Voltage	Tachometer		28V
	Op Amp/Comparator		28V
Input Voltage Range	Tachometer	LM2907-8, LM2917-8	±28V
		LM2907, LM2917	0.0V to +28V
	Op Amp/Comparator		0.0V to +28V
Power Dissipation	LM2907-8, LM2917-8		1200 mW
	LM2907-14, LM2917-14[1]		1580 mW
Operating Temperature Range			−40°C to +85°C
Storage Temperature Range			−65°C to +150°C
Soldering Information	PDIP Package	Soldering (10 seconds)	260°C
	SOIC Package	Vapor Phase (60 seconds)	215°C
		Infrared (15 seconds)	220°C

ELECTRICAL CHARACTERISTICS

V_{CC} = 12 V_{DC}, T_A = 25°C, see test circuit

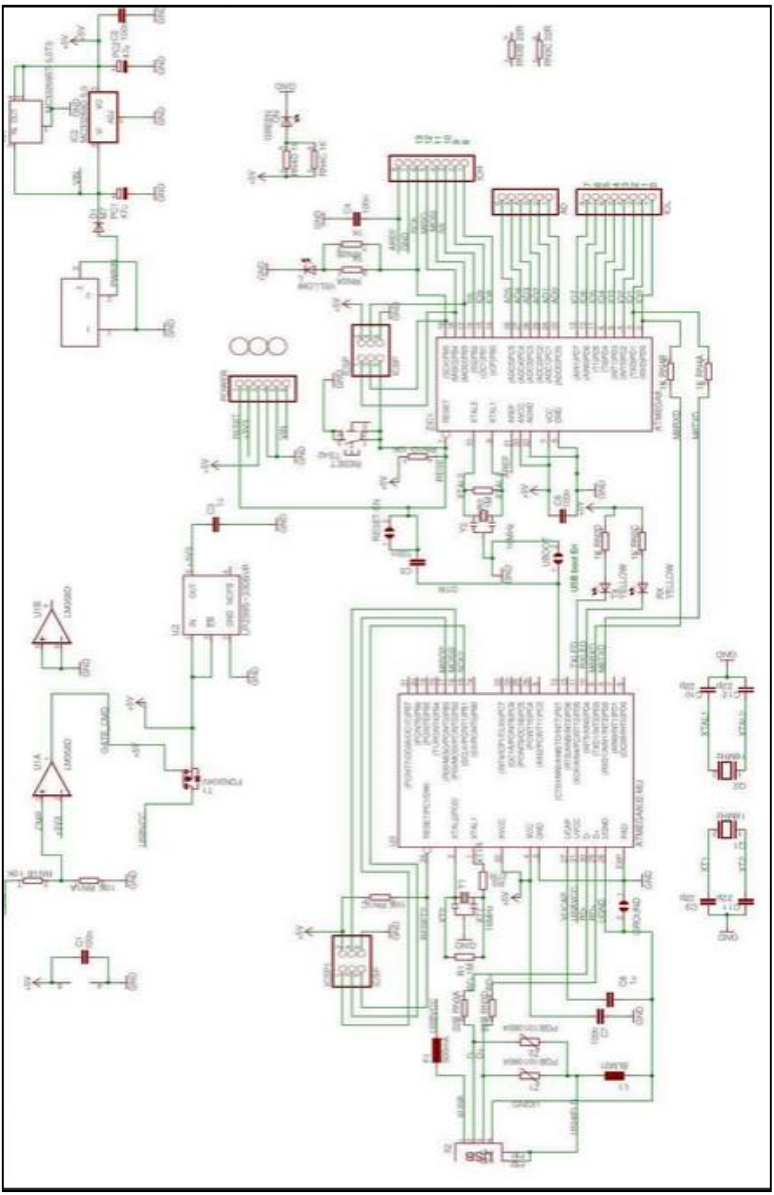
Symbol	Parameter	Conditions	Min	Typ	Max	Units
TACHOMETER						
	Input Thresholds	V_{IN} = 250 mVp-p @ 1 kHz[1]	±10	±25	±40	mV
	Hysteresis	V_{IN} = 250 mVp-p @ 1 kHz[1]		30		mV
	Offset Voltage	V_{IN} = 250 mVp-p @ 1 kHz[1]				
	LM2907/LM2917			3.5	10	mV
	LM2907-8/LM2917-8			5	15	mV
	Input Bias Current	V_{IN} = ±50 mV_{DC}		0.1	1	µA
V_{OH}	Pin 2	V_{IN} = +125 mV_{DC}[2]		8.3		V
V_{OL}	Pin 2	V_{IN} = −125 mV_{DC}[2]		2.3		V
I_2, I_3	Output Current	V2 = V3 = 6.0V[3]	140	180	240	µA
I_3	Leakage Current	I2 = 0, V3 = 0			0.1	µA
K	Gain Constant	See[2]	0.9	1.0	1.1	
	Linearity	f_{IN} = 1 kHz, 5 kHz, 10 kHz[4]	−1.0	0.3	+1.0	%
OP/AMP COMPARATOR						
V_{OS}		V_{IN} = 6.0V		3	10	mV
I_{BIAS}		V_{IN} = 6.0V		50	500	nA
	Input Common-Mode Voltage		0		V_{CC}−1.5V	V

ELECTRICAL CHARACTERISTICS (continued)

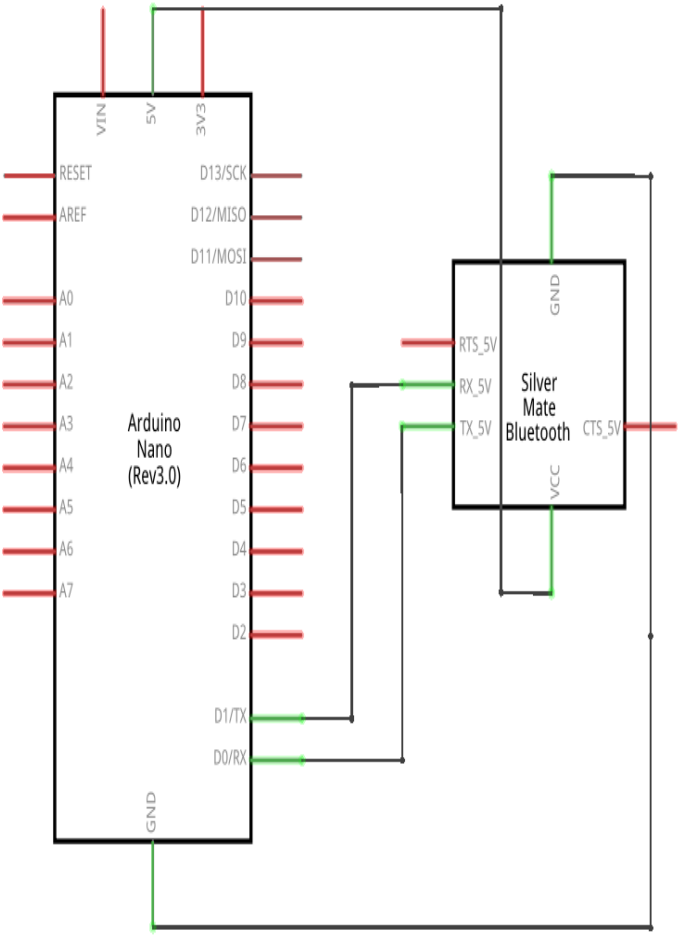
V_{CC} = 12 V_{DC}, T_A = 25°C, see test circuit

Symbol	Parameter	Conditions	Min	Typ	Max	Units
	Voltage Gain			200		V/mV
	Output Sink Current	V_C = 1.0	40	50		mA
	Output Source Current	V_E = V_{CC} −2.0		10		mA
	Saturation Voltage	I_{SINK} = 5 mA		0.1	0.5	V
		I_{SINK} = 20 mA			1.0	V
		I_{SINK} = 50 mA		1.0	1.5	V
ZENER REGULATOR						
	Regulator Voltage	R_{DROP} = 470Ω		7.56		V
	Series Resistance			10.5	15	Ω
	Temperature Stability			+1		mV/°C
	Total Supply Current			3.8	6	mA

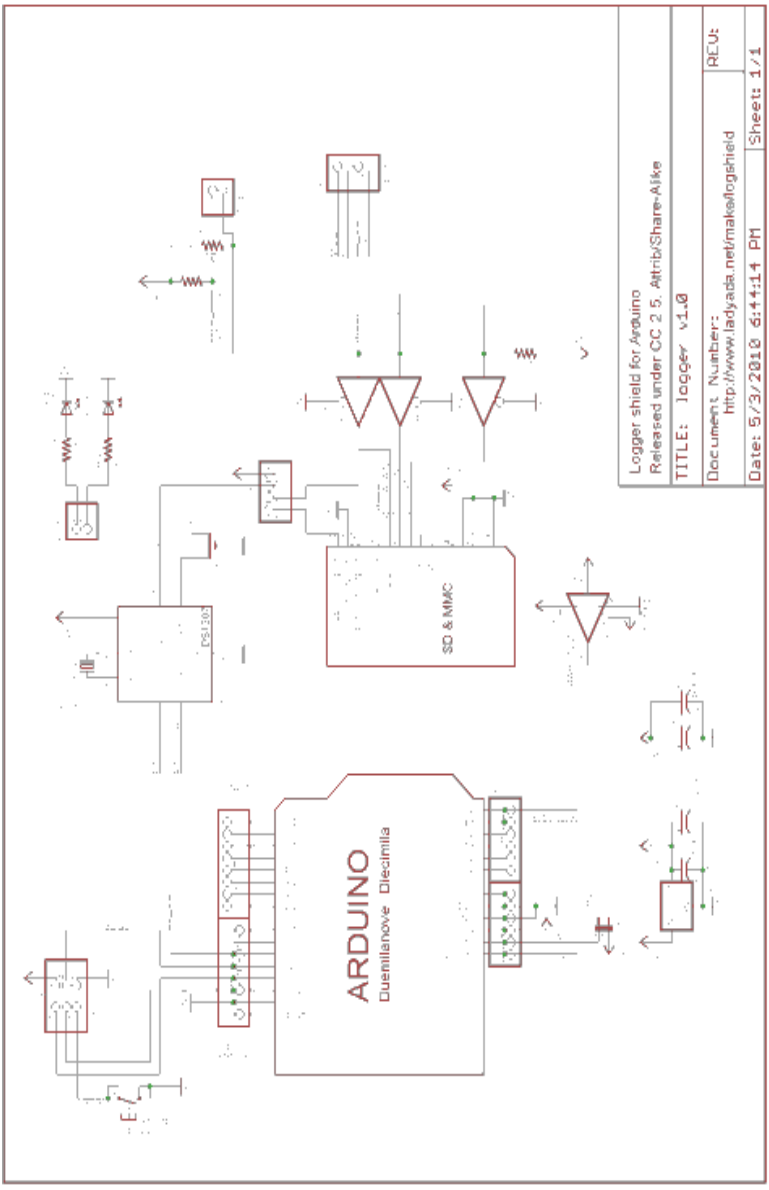
Skematik Arduino Uno



***Wirring* Modul *Bluetoth* HC 05**



Skematik *Data Logger* dan RTC



Halaman ini sengaja dikosongkan

RIWAYAT HIDUP PENULIS

Nama	:	Putra Arif Wardhana
TTL	:	Pati , 31 Agustus 1994
Jenis Kelamin	:	Laki-laki
Agama	:	Islam
Alamat Rumah	:	Jalan Jambean raya 100 Sidokerti Pati
Telp/HP	:	08564151867
E-mail	:	putraarifw@gmail.com
Hobi	:	Olah Raga

RIWAYAT PENDIDIKAN

- 2000 – 2006 :SDNSidokerto 03 PATI
- 2006 – 2009 :SMP Negeri 3 PATI
- 2009 – 2011 :SMA Negeri 1 PATI
- 2012 – sekarang : Bidang Studi Komputer Kontrol, Program D3 Teknik Elektro, ITS

PENGALAMAN KERJA

- Kerja Praktek di PLN Waru (2013-2014)

PENGALAMAN ORGANISASI

- Staff Departemen KOMINFO BEM Fakultas Teknologi Industri ITS (2013-2014)
- Asisten Laboratorium Elektronika Terapan D3 Teknik Elektro ITS (2014-2015)

Halaman ini sengaja dikosongkan

RIWAYAT HIDUP PENULIS

Nama	:	Muhammad Ainul Yaqin Febrianto
TTL	:	Pasuruan, 11 Februari 1994
Jenis Kelamin	:	Laki-Laki
Agama	:	Islam
Alamat Rumah	:	Jl. Nener 408 Kalianyar Bangil,Kabupaten Pasuruan
Telp/HP	:	085755889591
E-mail	:	doditfebrianto@gmail.com
Hobi	:	Hiking

RIWAYAT PENDIDIKAN

- 2000 – 2006 : SDN plus 1 Kaliarejo
- 2006 – 2009 : SMP Negeri 1 Bangil
- 2009 – 2012 : SMA 1 Bangil
- 2012 – sekarang : Bidang Studi Komputer Kontrol, Program D3 Teknik Elektro, ITS

PENGALAMAN KERJA

- Kerja Praktek di PT. Yamah MusIC Product Indonesia (Juni-Juli 2014)

PENGALAMAN ORGANISASI

- Staff Dewan Perwakilan Mahasiswa Jurusan D3 Teknik Elektro ITS (2012 – 2013)
- Asisten Laboratorium Elektronika Dasar D3 Teknik Elektro ITS (2013-2014)

Halaman ini sengaja dikosongkan