



**SISTEM MONITORING ARUS DAN TEGANGAN PADA
BATERAI MOBIL LISTRIK BERBASIS
MIKROKONTROLER ATMEGA16**

SKRIPSI

Oleh:

**Ridik Yudha Susila
NIM 101910201061**

JURUSAN TEKNIK ELEKTRO STRATA 1

FAKULTAS TEKNIK

UNIVERSITAS JEMBER

2015



**SISTEM MONITORING ARUS DAN TEGANGAN PADA
BATERAI MOBIL LISTRIK BERBASIS
MIKROKONTROLER ATMEGA16**

SKRIPSI

**Diajukan guna untuk melengkapi skripsi dan memenuhi syarat-syarat untuk
menyelesaikan Program Studi Teknik Elektro (S1)
dan guna mencapai gelar Sarjana Teknik**

Oleh:

**Ridik Yudha Susila
NIM 101910201061**

**JURUSAN TEKNIK ELEKTRO STRATA 1
FAKULTAS TEKNIK
UNIVERSITAS JEMBER**

2015

PERSEMBAHAN

Alhamdulillah, segala puji bagi Allah SWT, yang telah memberikan limpahan nikmat yang sangat luar biasa kepada penulis, dan tidak lupa juga sholawat dan salam selalu tercurahkan kepada Nabi besar Muhammad SAW yang telah membawa kita semua menuju peradaban manusia yang lebih baik. Dengan kerendahan hati, penulis mempersembahkan tugas akhir ini untuk:

1. Ibu Sri Sunarmi dan Bapak Pamuji yang selalu mendoakan dan selalu mendukung baik secara moral dan materi.
2. Keluarga besar penulis yang telah memberikan doa serta menjadi motivasi penulis untuk sukses.
3. Semua Dosen Jurusan Teknik Elektro Fakultas Teknik Universitas Jember yang telah membimbing dan memberikan ilmu. Terutama Bapak Dr. Bambang Sri Kaloko, S.T., M.T. selaku DPU dan Bapak Satryo Budi Utomo, S.T., M.T. selaku DPA yang telah meluangkan waktu dan pikirannya seta perhatiannya guna memberikan bimbingan dan pengarahan demi terselesainya skripsi ini.
4. Teman-teman Teknik Elektro 2010, yang dengan bangga mengusung slogan “Sing Penting Wani Dhisek!” dan budaya kumpul angkatan yang sangat membantu dalam pembentukan semangat bagi penulis. #SPWD.
5. Teman-teman seperjuangan dan seperantauan angkatan 2000 X (Bala Kurawa) dengan budaya cangkruk berbagi opini dalam curahan kopi, yang selama ini telah menjadi keluarga saat menempuh perkuliahan di Jember. Bangga menjadi bagian penting dari kalian. #MMX
6. Pihak-pihak yang membaca serta menjadikan skripsi ini sebagai referensi penelitiannya.
7. Serta pihak-pihak lain yang telah membantu penulis untuk menyelesaikan skripsi ini yang tidak bisa penulis sebutkan satu persatu.

MOTTO

“Jika kamu dalam masa sulit, berserah dirilah kepada Gusti Allah SWT. Karena Beliau Yang Maha Tahu, maka perbaikilah ibadahmu!”

(Ayahanda)

“Jangan kejar perempuanmu, tapi kejarlah ilmu dan selesaikan study mu!
Ketika kamu berilmu, perempuanmu yang akan datang sendiri kepadamu”

(Ibunda Ratu)

“Jangan terlalu serius menghabiskan waktu, barang kali kita lupa mengucapkan Amin pada doa seseorang yang diam-diam dipanjatkannya untuk kita”

(Ridik Yudha Susila)

“Perjuanganku lebih mudah karena mengusir penjajah, tapi perjuanganmu akan lebih sulit karena melawan bangsamu sendiri”

(Ir. Soekarno)

“Kemenangan yang seindah-indahnya dan sesukar-sukarnya yang boleh direbut manusia ialah menundukkan diri sendiri”

(R.A. Kartini)

“Lebih baik mengerti sedikit dari pada salah mengerti”

(A. France)

“Agama tanpa ilmu adalah buta. Ilmu tanpa agama adalah lumpuh.”

(Albert Eisntein)

“Menunggu kesuksesan adalah tindakan sia-sia yang sangat bodoh.”

(Anonim)

PERNYATAAN

Saya yang bertanda tangan dibawah ini :

Nama : Ridik Yudha Susila

NIM : 101910201061

menyatakan dengan sesungguhnya bahwa skripsi yang berjudul “*Sistem Monitoring Arus Dan Tegangan Pada Baterai Mobil Listrik Berbasis Mikrokontroler ATmega16*” adalah benar – benar hasil karya sendiri, kecuali jika dalam pengutipan subansi disebutkan sumbernya dan belum pernah diajukan pada institusi manapun, serta bukan karya jiplakan. Saya bertanggung jawab atas keabsahan dan kebenaran isinya sesuai dengan sikap ilmiah yang harus dijunjung tinggi.

Demikian pernyataan ini saya buat dengan sebenarnya, tanpa adanya tekanan dan paksaan dari pihak manapun serta bersedia mendapat sanksi akademik jika ternyata dikemudian hari pernyataan ini tidak benar.

Jember, Februari 2015

Yang menyatakan,

Ridik Yudha Susila

NIM 101910201061

SKRIPSI

**SISTEM MONITORING ARUS DAN TEGANGAN PADA
BATERAI MOBIL LISTRIK BERBASIS MIKROKONTROLER
ATMEGA16**

Oleh

Ridik Yudha Susila

NIM 101910201061

Pembimbing :

Dosen Pembimbing Utama : Dr. Bambang Sri Kaloko, S.T., M.T.

Dosen Pembimbing Anggota : Satryo Budi Utomo, S.T.,M.T.

PENGESAHAN

Skripsi dengan judul : "*Sistem Monitoring Arus Dan Tegangan Pada Baterai Mobil Listrik Berbasis Mikrokontroler ATmega16*" telah diuji dan disahkan oleh Jurusan Teknik Elektro, Fakultas Teknik, Universitas Jember pada :
hari, tanggal : Jumat, 20 Februari 2015
tempat : Fakultas Teknik, Universitas Jember.

Tim Penguji

Ketua,



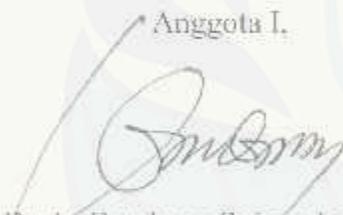
Dr. Bambang Sri Kaloko, S.T., M.T.
NIP 19710402 200312 1 001

Sekretaris,



Satryo Budi Utomo, S.T., M.T.
NIP 19850126 200801 1 002

Anggota I,



Dr. Ir. Bambang Sujanarko, M.M.
NIP 19631201 199402 1 002

Anggota II,



H.R.B. Moch. Gozali, S.T., M.T.
NIP 19710402 200312 1 001

Mengesahkan,
Dekan,



Ir. Widyono Hadi, M.T.
NIP 19610414 198902 1 001

**SISTEM MONITORING ARUS DAN TEGANGAN
PADA BATERAI MOBIL LISTRIK BERBASIS
MIKROKONTROLER ATMEGA16**

Ridik Yudha Susila

Jurusan Teknik Elektro, Fakultas Teknik, Universitas Jember

ABSTRAK

Mobil listrik pertama kali dibuat pada tahun 1834 oleh Thomas Davenport. Akan tetapi baru dikembangkan lagi pada tahun 1970an dikarenakan krisis energi dan kelangkaan minyak. Berbeda dengan mobil mesin pembakaran dalam yang menggunakan bahan bakar cair sebagai sumber energi, mobil listrik menggunakan motor listrik untuk penggerak dengan baterai sebagai sumber energinya. Keunggulan mobil listrik adalah tidak ada emisi gas buang, efisiensi tinggi dan tidak bergantung pada bahan bakar minyak, kemudian operasinya tenang dan halus. Dalam penelitian kali ini, kita akan membahas tentang baterai timbal asam sebagai sumber energi dari mobil listrik. Kenapa menggunakan baterai timbal asam, karena baterai timbal asam adalah jenis baterai yang memiliki kehandalan dan mampu menjadi sumber yang besar. Disamping itu baterai timbal asam memiliki umur yang relatif lama dan bisa di isi ulang (*charging*). Dalam penelitian kali ini akan dilakukan pengujian arus baterai menggunakan beban yang dimodelkan *starting*, *konstan*, *breaking*. Perumpamaan yang digunakan adalah 3buah lampu DC yang berkapasitas 35 Watt, 25 Watt, 18 Watt. Untuk mengukur data arus yang mengalir kita menggunakan sensor arus tipe ACS 712, kemudian data yang dihasilkan akan diolah oleh Mikrokontroler Atmega16 dan terakhir akan ditampilkan dalam bentuk *Graphic User Interface* oleh perangkat lunak *Visual Studio 2010 Ultimate*.

Kata kunci : *mobil listrik, baterai timbale asam, starting, konstan, breaking, sensor arus acs 712, mikrokontroler atmega16, graphic user interface, visual studio 2010 ultimate.*

**MONITORING SYSTEM CURRENT AND VOLTAGE
ON ELETRIC CAR BATTERY BASED
MICROCONTROLLER ATMEGA16**

Ridik Yudha Susila

Department of Electrical Engineering, Engineering Faculty, University of Jember

ABSTRACT

Electric car was first made in 1834 by Thomas Davenport. However, it developed again in the 1970s because of the energy crisis and dearth of oil. Different from the internal combustion engine car that uses liquid fuel as an energy source, an electric car uses an electric motor for driving with battery as energy source. The advantages of the electric car is no exhaust emissions, high efficiency and does not rely on oil fuels, the operation is quite and smooth. In this research, we will discuss about lead acid battery as the energy source of the electric car. Why do we use a lead acid battery, because the lead acid battery is a type of battery that has reliability and capable of being a great source. Besides that, lead acid is relatively long life and can refill (charging). In this present research will tested battery current using burden that modeled starting, constant, breaking. Visual display used is 3 lamps DC 35watt capacity, 25watt, 18watt. To measure the flowing current data we use current sense tipe ACS 712, then the resulting data will be processed by the Microcontroller ATmega16 and the latter will be displayed in Graphic User Interface form by Visual Studio 2010 Ultimate software..

Keyword : *electric car, lead acid battery, starting, konstan, breaking, current sense acs712, microcontroller atmega16, graphic user interface, visual studio 2010 ultimate.*

RINGKASAN

Sistem Monitoring Arus dan Tegangan Pada Baterai Mobil Listrik Berbasis Mikrokontroler ATmega16; Ridik Yudha Susila; 101910201061; 2015; 63 halaman; Program Studi Strata Satu Teknik, Jurusan Teknik Elektro, Fakultas Teknik, Universitas Jember.

Seiring dengan perkembangan jaman saat ini, mobil merupakan kebutuhan sekunder yang sudah sangat banyak dimiliki oleh masyarakat kalangan menengah. Karena dilihat dari fungsinya mobil sangatlah meringankan dan sangat membantu dalam kehidupan masyarakat. Dan dari situlah mobil dianggap menjadi kebutuhan pokok bagi seorang pengusaha ataupun pejabat sekalipun. Akan tetapi dari banyaknya mobil yg diproduksi tiap tahun, masyarakat luas tidak pernah memikirkan dampak negatif yang ditimbulkan dari penggunaan alat transportasi kendaraan bermotor tersebut. Dampak yang ditimbulkan antara lain : polusi udara atau pencemaran udara yang disebabkan oleh gas buang dari knalpot kendaraan bermotor. Kemudian bahan bakar minyak yang pada saat ini sudah dianggap langka oleh berbagai Negara, bagaimana apabila BBM benar-benar habis? Mungkin dunia akan kacau.

Oleh sebab itu, pada penelitian ini adalah solusi terbaik dari sebuah kendaraan yang ramah lingkungan. Dan pada penelitian ini yang akan dibahas tentang baterai atau sumber energi sebagai penggerak dari mobil listrik.

Baterai yang digunakan pada mobil listrik adalah jenis baterai timbal asam. Baterai timbal asam merupakan baterai yang menggunakan sel-sel larutan kimia model basah, oleh sebab itu baterai bisa di isi ulang. Dalam penelitian ini baterai timbal asam yang digunakan berkapasitas 45 AH dengan merk GS Astra Premium tipe 46 B 24 R (NS60). Pada baterai timbal asam saat di isi ulang, dan berstatus penuh , kapasitas yang sebenarnya tidak samapai 100 % atau 45 AH, akan tetapi hanya 80 % sampai 85 %, atau sekitar 36 sampai 38,25 %. Dari baterai tersebut kemudian akan dilakukan pengujian dengan *monitoring* arus dan tegangannya apabila baterai diberi beban. Kemudian untuk beban sendiri divariasikan menjadi 3 varian, yaitu : *starting*, *konstan*, *breaking*. Dari ketiga varian beban tersebut digunakan lampu DC sebesar 35 Watt, 25 Watt, 18 Watt.

Untuk tahap selanjutnya adalah pembuatan perangkat keras rangkaian *system minimum*. Dari rangkaian tersebut terdapat otak dari program sensor arus dan tegangan, yaitu Mikrokontroler ATmega16. Kemudian deprogram dan dicari rumus kalibrasi untuk sensor dan mikrokontroler. Setelah kalibrasi sukses, dibuat aplikasi GUI (*Graphic User Interface*) untuk monitoring dari pengujian menggunakan perangkat lunak Visual Studio 2010 Ultimate.

Setelah semua alat selesai, dilakukan pengujian alat. Apabila dari keseluruhan didapat *error %* kurang dari 10 %, maka alat dikatakan sukses dan dilanjutkan pengujian secara keseluruhan. Dari data pengujian *starting*, *konstan*, *breaking* didapat nilai *error %* terbesar adalah 9,18 % yaitu pada pengujian beban *breaking*. Sedangkan untuk pengujian dengan *error %* terkecil adalah 3,4 % yaitu pada pengujian beban *breaking*. Kemudian didapat *error %* rata-rata dari pengujian keseluruhan adalah 6,61 %. Dari penelitian kali ini faktor terbesar yang mempengaruhi *error %* adalah pada kondisi baterai yang tidak stabil.

PRAKATA

Bismillahirrohmanirrohim

Puji syukur kehadiran Allah SWT atas segala rahmat dan karunia-Nya sehingga penulis dapat menyelesaikan skripsi yang berjudul “*sistem monitoring arus dan tegangan pada baterai mobil listrik berbasis mikrokontroler atmega16*”. Skripsi ini disusun untuk memenuhi salah satu syarat menyelesaikan pendidikan strata satu (S1) pada Jurusan Teknik Elektro Fakultas Teknik Universitas Jember.

Penyusunan skripsi ini tidak lepas dari bantuan beberapa pihak. Oleh karena itu, penulis menyampaikan terima kasih kepada :

1. Allah SWT yang telah melimpahkan rahmat dan rizki-Nya serta memberi kelancaran dan kemudahan sehingga terselesaikannya skripsi ini.
2. Bapak Dr. Bambang Sri Kaloko, S.T., M.T., selaku Dosen Pembimbing Utama dan Bapak Satryo Budi Utomo, S.T., M.T., selaku Dosen Pembimbing Anggota yang telah meluangkan waktu, pikiran guna memberikan bimbingan dan pengarahan dalam penyusunan skripsi ini.
3. Bapak Dr. Ir. Bambang Sujanarko, M.T., dan Bapak H.R.B. Moch. Gozali, S.T., M.T., selaku Tim Penguji yang telah meluangkan banyak waktu, pikiran dan perhatiannya guna memberikan pengarahan demi terselesaikannya penulisan skripsi ini.
4. Seluruh Dosen yang ada di Fakultas Teknik khususnya Teknik Elektro beserta karyawan.
5. Ibunda Ratu Sri Sunarmi, Ayahanda Pamuji, Kakek Hadi Subari, Nenek Rupini dan adek Radhitta Diena Amellia. Terima kasih atas semua dukungan baik secara materi maupun moral dan kasih sayang serta doa restunya yang tulus kepada saya. Kalian semua tetap sabar dan selalu menyelipkan do'a untuk saya disetiap shalat dan perkataan kalian, hingga terselesaikannya skripsi ini.
6. Teman-teman Fakultas Teknik atau bahkan bisa dibilang lebih dari sekedar teman biasa khususnya Teknik Elektro Angkatan 2010 yang tidak dapat saya sebutkan satu persatu. Dengan kalian saya tahu makna kebersamaan,

kekompakan dan kesederhanaan dalam hidup. Terima kasih atas dukungan dan bantuannya selama proses perkuliahan hingga penyusunan skripsi ini. Tanpa kalian, saya bukan apa-apa di Jember ini, SING PENTING WANI DHISEK.!

7. Teman-teman seperantauan, seperkuliahan, seduluruan, seangkatan 18 sks, alias bala kurawa angkatan 2000 X. Dwipa-Gerbong, Mujib-Mbah, Vian-Mboet, Husin-Oyek, Marsandi-Parto, Udin-Penyok, Bryan-Mi'ing, Resan-Sur, kita berdiri bersama dalam langkah dewasa saudara-sodara. *Thanks a lot!*
8. Teman-teman komunitas fans club UIOJ (United Indonesia Official Jember) yang tidak dapat saya sebutkan satu persatu. Terima kasih telah menjadi bagian dari sejarah yang menyenangkan selama hidup saya di bumi Jember. *Glory Glory Man.United. #GGMU. UIOJ Solid.!*
9. Riki Setian S.T., dan saudara Budi Setiawan S.T., tanpa kalian saya tidak kan selancar ini. *Big Thank Brothers!*
10. *The special one in my life* yang selalu menjadi motivasi dan semangat hidup saya selama ini, terima kasih Rosy Mardiana.
11. Pihak-pihak yang tidak dapat saya sebutkan satu persatu, terima kasih atas dukungan dan motivasi kalian dalam penyusunan skripsi ini.

Semoga skripsi ini dapat bermanfaat dalam mengembangkan ilmu pengetahuan khususnya untuk disiplin ilmu teknik elektro. Kritik dan saran yang membangun diharapkan terus mengalir untuk lebih menyempurnakan skripsi ini dan dapat dikembangkan untuk penelitian selanjutnya.

Jember, Februari 2015

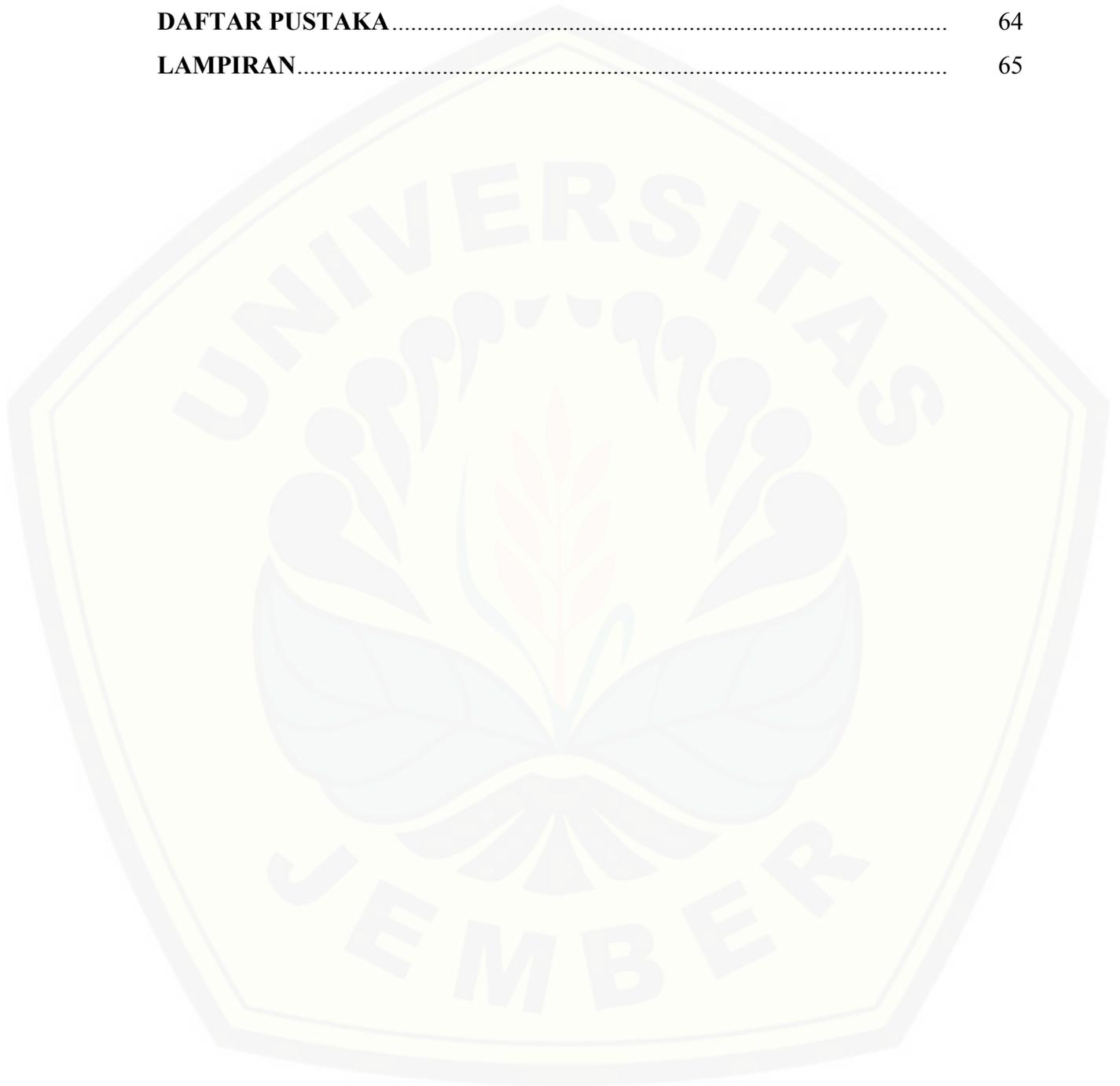
Penulis

DAFTAR ISI

	Halaman
HALAMAN JUDUL	i
HALAMAN PERSEMBAHAN	ii
HALAMAN MOTTO	iii
HALAMAN PERNYATAAN.....	iv
HALAMAN PEMBIMBINGAN.....	v
HALAMAN PENGESAHAN.....	vi
ABSTRAK	vii
RINGKASAN	ix
PRAKATA.....	xi
DAFTAR ISI.....	xiii
DAFTAR TABEL	xvi
DAFTAR GAMBAR.....	xvii
BAB 1 PENDAHULUAN	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah	3
1.3 Batasan Masalah	3
1.4 Tujuan.....	3
1.5 Manfaat.....	4
1.6 Sistematika Penulisan.....	4
BAB 2 TINJAUAN PUSTAKA.....	5
2.1 Sistem <i>Monitoring</i>	5
2.2 Mobil Listrik.....	5
2.3 Baterai Timbal Asam.....	7
2.4 Mikrokontroler ATmega 16	9
2.5 Komunikasi <i>Serial</i>	13
2.6 Sensor Arus	14
2.7 <i>Analog To Digital Converter (ADC0804)</i>	15
2.8 Visual Studio 10 Ultimate	17

BAB 3 METODOLOGI PENELITIAN.....	19
3.1 Jadwal Penelitian	19
3.2 Tahap Perencanaan	19
3.3 Perancangan Perangkat Keras	20
3.3.1 Blok Perangkat Keras <i>System Minimum</i>	21
3.3.2 Blok Perangkat Keras Sensor Arus dan Tegangan.....	22
3.3.3 Konsep Sistem dan Penataan Alat.....	23
3.4 Perancangan Perangkat Lunak.....	25
3.4.1 Perancangan Leader Diagram Proteus.....	25
3.4.2 Pemrograman Code Vision AVR.....	25
3.4.3 Aplikasi GUI (<i>Graphic User Interface</i>)	26
3.5 <i>Flowchart</i> Sistem Alat Secara Keseluruhan.....	27
3.6 Rumus.....	29
BAB 4 HASIL DAN PEMBAHASAN.....	30
4.1 Pengujian Sensor	30
4.1.1 Pengujian Sensor Arus	30
4.1.2 Pengujian Sensor Tegangan	33
4.2 Pengujian <i>Serial</i>	36
4.3 Pengujian GUI (<i>graphic User Interface</i>)	
4.4 Pemodelan Simulasi	38
4.5 Pengujian Sistem Keseluruhan.....	41
4.5.1 Pengujian Beban 35 Watt (<i>Starting</i>).....	42
4.5.2 Pengujian Beban 25 Watt (<i>Konstan</i>)	45
4.5.3 Pengujian Beban 18 Watt(<i>Breaking</i>).....	47
4.5.4 Pengujian Beban Paralel (<i>Couple</i>).....	49
4.5.4.1 Pengujian Beban <i>Couple</i> 43 Watt.....	49
4.5.4.2 Pengujian Beban <i>Couple</i> 53 Watt.....	52
4.5.4.3 Pengujian Beban <i>Couple</i> 60 Watt.....	51
4.5.4.4 Pengujian Beban <i>Couple</i> 78 Watt.....	56
4.5.5 Perbandingan Pengujian Beban Keseluruhan.....	49

BAB 5 PENUTUP	62
5.1 Kesimpulan.....	62
5.2 Saran.....	62
DAFTAR PUSTAKA.....	64
LAMPIRAN.....	65



DAFTAR TABEL

	Halaman
Tabel 2.1 Konfigurasi <i>pin port</i> Mikrokontroler ATmega 16	12
Tabel 2.2 <i>Pin port</i> pada sensor arus ACS 712	16
Tabel 4.1 Data hasil pengujian sensor arus.....	33
Tabel 4.2 Data hasil pengujian sensor tegangan	36
Tabel 4.3 Pengujian <i>serial</i> dari mikrokontroler ke laptop	36
Tabel 4.4 Data pengujian <i>starting</i> beban 35 Watt	44
Tabel 4.5 Data pengujian <i>konstan</i> beban 25 Watt	46
Tabel 4.6 Data pengujian <i>breaking</i> beban 18 Watt.....	48
Tabel 4.7 Data pengujian beban <i>couple</i> 43 Watt	51
Tabel 4.8 Data pengujian beban <i>couple</i> 53 Watt	53
Tabel 4.9 Data pengujian beban <i>couple</i> 60 Watt	55
Tabel 4.10 Data pengujian beban <i>couple</i> 78 Watt	58
Tabel 4.11 Hasil data pengujian secara keseluruhan	59

DAFTAR GAMBAR

	Halaman
Gambar 2.1 Bentuk alur sistem <i>monitoring</i>	5
Gambar 2.2 Skema mobil listrik.....	6
Gambar 2.3 Baterai timbal asam.....	9
Gambar 2.4 <i>Pin-pin</i> ATmega 16 model 40 <i>pin</i>	11
Gambar 2.5 Konfigurasi <i>usb to serial</i> menggunakan IC FT232.....	13
Gambar 2.6 <i>Usb to serial</i>	14
Gambar 2.7 Sensor arus ACS 712.....	14
Gambar 2.8 Konfigurasi <i>pin</i> ADC 0804.....	16
Gambar 2.9 Microsoft Visual Studio 2010 Express.....	18
Gambar 3.1 Diagram blok sistem monitoring arus dan tegangan.....	21
Gambar 3.2 Rangkaian <i>System Minimum</i> ATmega 16.....	22
Gambar 3.3 Sensor Arus.....	22
Gambar 3.4 Sensor Tegangan.....	23
Gambar 3.5 Skema konsep penataan alat.....	23
Gambar 3.6 Sistem letak alat.....	24
Gambar 3.7 Gambar rangkaian <i>System Minimum</i>	25
Gambar 3.8 Tampilan Visual Basic sebagai <i>monitoring</i> alat.....	26
Gambar 3.9 <i>Flowchat</i> sistem secara keseluruhan.....	27
Gambar 4.1 Pengujian sensor arus dengan beban 18 Watt.....	30
Gambar 4.2 Pengujian sensor arus dengan beban 25 Watt.....	30
Gambar 4.3 Pengujian sensor arus dengan beban 35 Watt.....	31
Gambar 4.4 Pengujian sensor arus dengan beban 43 Watt.....	31
Gambar 4.5 Pengujian sensor arus dengan beban 53 Watt.....	32
Gambar 4.6 Pengujian sensor arus dengan beban 60 Watt.....	32
Gambar 4.7 Pengujian sensor arus dengan beban 78 Watt.....	32
Gambar 4.8 Pengujian tegangan 3 Volt.....	34
Gambar 4.9 Pengujian tegangan 6 Volt.....	34
Gambar 4.10 Pengujian tegangan 9 Volt.....	34

Gambar 4.11 Pengujian tegangan 12 Volt	35
Gambar 4.12 Pengujian tegangan 15 Volt	35
Gambar 4.13 Tampilan GUI sebagai <i>monitoring</i> alat.....	37
Gambar 4.14 Simulasi model karakteristik baterai timbal asam dengan Simulink Matlab	38
Gambar 4.15 Model simulasi tegangan dengan beban 35 Watt.....	39
Gambar 4.16 Model simulasi arus dengan beban 35Watt	39
Gambar 4.17 Grafik perbandingan tegangan simulasi dan pengujian	40
Gambar 4.18 Grafik perbandingan arus simulasi dan pengujian	40
Gambar 4.19 Baterai GS Astra 45 AH.....	41
Gambar 4.20 Kondisi baterai penuh pada alat Test Car Battery.....	42
Gambar 4.21 Kondisi baterai habis pada alat Test Car Battery.....	42
Gambar 4.22 Tampilan GUI <i>starting</i> pada saat pengujian dimulai	43
Gambar 4.23 Tampilan GUI <i>starting</i> saat pengujian selesai	43
Gambar 4.24 Grafik arus dan tegangan pada kondisi <i>starting</i>	44
Gambar 4.25 Tampilan GUI <i>konstan</i> saat pengujian dimulai.....	45
Gambar 4.26 Tampilan GUI <i>konstan</i> saat pengujian selesai	46
Gambar 4.27 Grafik arus dan tegangan pada kondisi <i>konstan</i>	47
Gambar 4.28 Tampilan GUI <i>breakibg</i> saat pengujian dimulai.....	47
Gambar 4.29 Tampilan GUI <i>breaking</i> saat pengujian selesai.....	48
Gambar 4.30 Grafik arus dan tegangan pada kondisi <i>breaking</i>	49
Gambar 4.31 Tampilan GUI <i>couple</i> 43 Watt pada saat mulai	50
Gambar 4.32 Tampilan GUI <i>couple</i> 43 Watt pada saat selesai	50
Gambar 4.33 Grafik arus dan tegangan pada kondisi <i>couple</i> 43 Watt.....	51
Gambar 4.34 Tampilan GUI <i>couple</i> 53 Watt pada saat mulai	52
Gambar 4.35 Tampilan GUI <i>couple</i> 53 Watt pada saat selesai	52
Gambar 4.36 Grafik arus dan tegangan pada kondisi <i>couple</i> 53 Watt.....	53
Gambar 4.37 Tampilan GUI <i>couple</i> 60 Watt pada saat mulai	54
Gambar 4.38 Tampilan GUI <i>couple</i> 60 Watt pada saat selesai	54
Gambar 4.39 Grafik arus dan tegangan pada kondisi <i>couple</i> 60 Watt.....	55
Gambar 4.40 Tampilan GUI <i>couple</i> 78 Watt pada saat mulai	56

Gambar 4.41 Tampilan GUI <i>couple</i> 78 Watt pada saat selesai	57
Gambar 4.42 Grafik arus dan tegangan pada kondisi <i>couple</i> 78 Watt.....	58
Gambar 4.43 Grafik <i>monitoring</i> sistem pengujian secara keseluruhan	60



BAB 1. PENDAHULUAN

1.1 Latar Belakang

Kendaraan berbahan bakar minyak (Mobil) pada masa sekarang seakan sudah menjadi kebutuhan pokok bagi masyarakat luas. Peranan mobil sendiri sangat signifikan untuk semua kalangan, dari dipakai untuk mata pencaharian hingga hanya untuk materi kekayaan. Tetapi tanpa disadari, semakin banyaknya permintaan produksi mobil justru semakin membahayakan kelangsungan hidup manusia. Poin pokok dari ancaman tersebut adalah semakin tercemarnya polusi udara di planet bumi. Oleh sebab itu, banyak dilakukan riset tentang mobil listrik untuk mengembangkan modern di negara-negara maju.

Mobil listrik baterai pertama kali dibuat oleh Thomas Davenport pada tahun 1834. Jadi, mobil listrik telah dibuat beberapa dekade lebih dulu daripada mobil mesin pembakaran dalam yang pertama. Mobil pertama yang melampaui kecepatan 100 km/jam juga mobil baterai, yaitu 'Jamais Contente' yang dikendarai oleh Camille Jenatzy pada tahun 1899. Dibandingkan dengan mobil mesin pembakaran dalam, mobil listrik lebih nyaman, tenang dan bersih. Akan tetapi, karena keterbatasan kapasitas penyimpan energi baterai, jangkauannya sangat terbatas. Pada saat yang sama, teknologi mobil mesin pembakaran dalam berkembang dengan pesat. Dan akibatnya, mobil listrik menghilang pada tahun 1930an. (<http://dx.doi.org/>)

Baru pada tahun 1970an ketika terjadi krisis energi dan kelangkaan minyak pada saat itu, pabrikan mobil dan pembuat kebijakan mulai berpikir ulang tentang mobil listrik. Alasan yang dikemukakan adalah, pertama, mobil listrik menawarkan efisiensi energi tinggi dan mengizinkan diversifikasi sumber energi. Dan yang kedua, memiliki emisi lokal nol dan membantu memperbaiki kualitas udara perkotaan.

Berbeda dengan mobil mesin pembakaran dalam yang menggunakan bahan bakar cair sebagai sumber energi, mobil listrik menggunakan motor listrik untuk penggerak dengan baterai sebagai sumber energinya. Keunggulan mobil listrik adalah tidak ada emisi gas buang, efisiensi tinggi dan tidak bergantung pada

bahan bakar fosil, dan operasinya tenang serta halus. Mobil listrik terdiri dari tiga sub-sistem utama:

1. Sistem penggerak motor listrik meliputi bisa pengendali mobil, konverter elektronika daya, motor listrik dan transmisi.
2. Sistem baterai, meliputi sistem manajemen baterai dan unit pengisian
3. Sistem pembantu, meliputi pemanas/pendingin, pompa elektronika dan pembantu elektronika lainnya. (<http://elkimkor.com/>)

Kemudian dari beberapa uraian mobil listrik diatas, disini penelitian saya akan lebih khusus membahas tentang baterai mobil listrik. Permasalahan dalam mendisain mobil listrik dengan sumber energi listrik dari baterai adalah sistem pengaturan energi, penentuan daya dan metode pengaturannya sehingga dapat memenuhi kebutuhan sistem penggerak mobil listrik (Kaloko et all, 2009). Untuk itu perlu dikembangkan suatu model baterai cerdas yang dapat memenuhi kebutuhan mobil listrik (Kaloko et all, 2011).

Baterai sebagai sumber energi pada mobil listrik merupakan suatu sel elektrokimia yang terdiri dari empat komponen dasar yaitu plat positif, plat negatif, larutan elektrolit, dan separator atau pemisah yang berfungsi sebagai isolasi antara elektroda *positif* dan *negatif*. Reaksi kimia diantara plat-plat dengan larutan elektrolit akan menghasilkan potensial listrik. Kapasitas baterai ditentukan oleh banyaknya muatan listrik yang dapat diperoleh dari suatu baterai dan besarnya tergantung pada bahan aktif yang ada pada plat (Kiehne, H. A, 2003).

Mobil listrik umumnya menggunakan baterai sebagai sumber energi untuk menjalankan motor penggerak. Namun baterai pada mobil listrik memiliki kapasitas yang terbatas. Pemakaian baterai dalam waktu tertentu akan menyebabkan kapasitas baterai berkurang. Untuk mengembalikan kapasitas baterai seperti semula diperlukan proses pengisian ulang (*recharging*) (Park et all, 2001). Untuk menjaga keandalan ketersediaan sumber energi dari baterai maka diperlukan pemilihan jenis baterai yang sesuai dan pengaturan penggunaan energi listrik pada mobil listrik (Thomas B.G., 2000). Pada pengoperasian mobil listrik kita harus dapat memantau kinerja baterai selama mobil beroperasi. Untuk itu perlu dikembangkan suatu model dari baterai. Dari pelacakan jejak hasil

penelitian tentang baterai ini maka perlu dikembangkan model baterai cerdas untuk meningkatkan kinerja mobil listrik.

Hal tersebut yang mendasari penulis melakukan penelitian tentang pengembangan baterai timbal asam, untuk bisa mengontrol baterai mobil listrik kapan dan dimana saat kapasitas dayanya habis. Dengan cara memonitoring baterai dengan melakukan beberapa tes ketika *starting*, *konstan* dan pengereman.

1.2 Perumusan Masalah

Permasalahan yang dibahas dalam penelitian ini adalah :

1. Bagaimana menggunakan sensor arus untuk membaca *output* dari baterai timbal asam
2. Bagaimana menampilkan data arus, tegangan, daya beban menjadi grafik di komputer untuk *monitoring* baterai pada mobil listrik
3. Bagaimana perbandingan arus, tegangan, daya pada saat *starting*, *konstan*, *breaking*.

1.3 Batasan Masalah

Untuk memperjelas dan menghindari meluasnya masalah, maka batasan masalahnya yaitu :

1. Baterai yang digunakan adalah baterai timbal asam 12 Volt atau 45 AH.
2. Beban yang digunakan lampu DC 18 Watt, 25 Watt, 35 Watt, kemudian untuk beban tambahan dilakukan *couple* pada masing-masing beban.
3. Pengujian dilakukan pada kapasitas nominal baterai maksimal 80 % sampai 85 %.
4. Pada beban mobil listrik dimodelkan *starting*, *konstant*, *breaking*.
5. Software pada komputer menggunakan *Visual Studio 10 Ultimate*.

1.4 Tujuan

Tujuan penelitian ini adalah :

1. Merancang, membangun dan menganalisa sistem monitoring baterai timbal asam dengan beban model mobil listrik.
2. Mempelajari hubungan arus, tegangan, dan kapasitas daya baterai timbal asam dengan model beban mobil listrik.

3. Mengembangkan baterai cerdas buatan untuk mobil listrik.

1.5 Manfaat

Penelitian ini diharapkan dapat bermanfaat sebagai berikut :

1. Mengetahui kapan baterai harus diisi atau diganti
2. Dapat mengetahui seberapa lama waktu kapasitas baterai akan habis ketika beban berubah-ubah.

1.6 Sistematika Penulisan

Secara garis besar penyusunan proposal skripsi ini adalah sebagai berikut:

BAB 1. PENDAHULUAN

Berisi tentang latar belakang, perumusan masalah, batasan masalah, tujuan dan manfaat, dan sistematika pembahasan

BAB 2. TINJAUAN PUSTAKA

Berisi penjelasan tentang teori yang berhubungan dengan penelitian.

BAB 3. METODOLOGI PENELITIAN

Menjelaskan tentang metode yang digunakan untuk menyelesaikan skripsi.

BAB 4. HASIL DAN ANALISA DATA

Berisi tentang pengujian alat dan pembahasan dari penelitian yang dilakukan.

BAB 5. KESIMPULAN

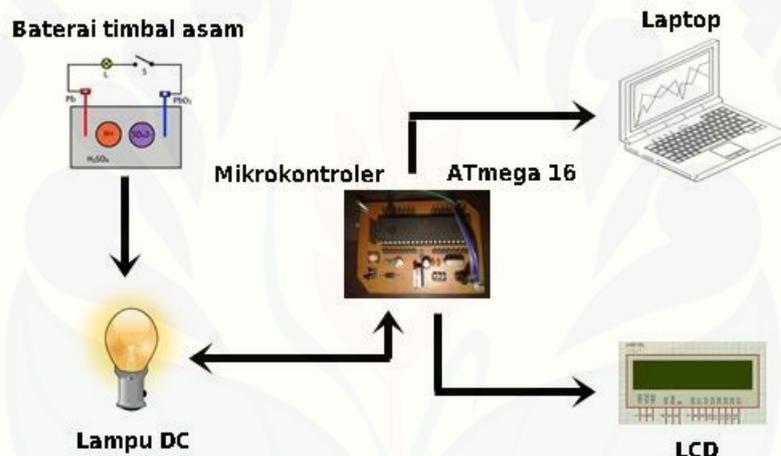
Berisi tentang kesimpulan dari hasil akhir penelitian

BAB 2. TINJAUAN PUSTAKA

2.1 Sistem *Monitoring*

Pengertian sistem secara umum adalah sekelompok elemen-elemen yang berkaitan untuk membentuk suatu kesatuan. Sistem adalah sekelompok elemen yang bekerja sama untuk suatu tujuan tertentu dengan memasukkan dan menghasilkan *output* dengan melalui sebuah proses. Sistem terdiri dari 3 komponen utama yaitu *input*, proses, dan *output*. (O'Brien, 2003)

Sistem *monitoring* merupakan sistem yang didesain untuk bisa memberikan *feedback* ketika program sedang menjalankan fungsinya. *Feedback* dimaksudkan untuk memberikan informasi atau keadaan sistem pada saat itu.



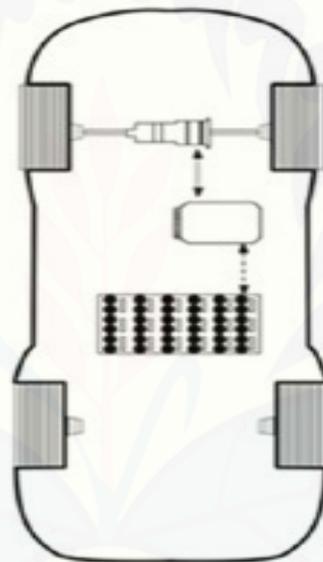
Gambar 2.1 Bentuk alur sistem *monitoring*
(Sumber : www.riddickazhu.tumblr.com)

Pada gambar 2.1 adalah bentuk dari sistem *monitoring* arus dan tegangan dengan menggunakan 2 bentuk *monitor*, yaitu LCD dan komputer atau laptop. (Sumber : www.nonprofitbasic.org/TopicAreaGlossary.aspx)

2.2 Mobil Listrik

Mobil listrik baterai pertama kali dibuat oleh Thomas Davenport pada tahun 1834. Jadi, mobil listrik ini telah dibuat beberapa dekade lebih dulu daripada mobil mesin pembakaran dalam pertama. Mobil pertama yang melampaui kecepatan 100 km/jam juga mobil baterai, yaitu 'Jamais Contente' yang dikendarai oleh Camille Jenatzy pada tahun 1899. Dibandingkan dengan mobil

mesin pembakaran dalam, mobil listrik lebih nyaman, tenang dan bersih. Akan tetapi, karena keterbatasan kapasitas penyimpan energi baterai, jangkauannya sangat terbatas. Pada saat yang sama, teknologi mobil mesin pembakaran dalam berkembang dengan pesat. Sebagai akibatnya, mobil listrik menghilang pada tahun 1930an. Baru pada tahun 1970an ketika terjadi krisis energi dan kelangkaan minyak pada saat itu, pabrikan mobil dan pembuat kebijakan mulai berpikir ulang tentang mobil listrik. Alasan yang dikemukakan adalah, pertama, mobil listrik menawarkan efisiensi energi tinggi dan mengijinkan diversifikasi sumber energi, dan kedua memiliki emisi lokal nol dan membantu memperbaiki kualitas udara perkotaan.



Gambar 2.2 Skema mobil listrik.

(Sumber: <http://dx.doi.org/10.1016/j.electacta.2012.03.172>)

Berbeda dengan mobil mesin pembakaran dalam yang menggunakan bahan bakar cair sebagai sumber energi, mobil listrik menggunakan motor listrik untuk penggerak dengan baterai sebagai sumber energinya (Gambar 2.1). Keunggulan mobil listrik adalah tidak ada emisi gas buang, efisiensi tinggi dan tidak bergantung pada bahan bakar fosil, dan operasinya tenang dan halus. Mobil listrik terdiri dari tiga sub-sistem utama:

- Sistem penggerak motor listrik meliputi pengendali mobil, konverter elektronika daya, motor listrik dan transmisi;

- Sistem baterai meliputi baterai, Sistem Manajemen Baterai dan unit pengisian;
- Sistem pembantu meliputi pemanas/pendingin, pompa elektronika dan pembantu elektronika lainnya.

Prinsip kerja mobil listrik adalah sebagai berikut. Input pengendali diperoleh dari pedal akselerator dan rem. Pengendali ini menyediakan sinyal yang sesuai ke konverter daya elektronika yang mengatur aliran daya antara motor listrik dan baterai. Motor juga memainkan peran generator, yang mengkonversi energi pengereman menjadi elektron dan mengisi baterai. Unit manajemen energi, bekerja sama dengan pengendali mobil, mengendalikan pengereman regenerasi dan rekovert energi. Motor listrik menghasilkan torsi yang besar dari keadaan mobil berhenti. Hal ini menghasilkan kinerja yang sangat bagus. Percepatan dan daya mobil listrik jauh melampaui mobil mesin pembakaran dalam.

(Sumber: B.G. Pollet, I. Staffell, J.L. Shang, 2012)

2.3 Baterai Timbal Asam

Baterai adalah alat listrik kimiawi yang menyimpan energi, dan mengeluarkan tenaganya dalam bentuk listrik. Sebuah baterai biasanya terdiri dari tiga komponen penting, yaitu:

1. Batang karbon sebagai anode (kutub positif baterai)
2. Seng (Zn) sebagai katode (kutub negatif baterai)
3. Pasta sebagai elektrolit (penghantar)

Baterai yang biasa dijual (*disposable* / sekali pakai) mempunyai tegangan listrik 1,5 volt. Baterai ada yang berbentuk tabung atau kotak. Ada juga yang dinamakan *rechargeable battery*, yaitu baterai yang dapat diisi ulang, seperti yang biasa terdapat pada telepon genggam. Baterai sekali pakai disebut juga dengan baterai primer, sedangkan baterai isi ulang disebut dengan baterai sekunder.

Baik baterai primer maupun baterai sekunder, kedua-duanya bersifat mengubah energi kimia menjadi energi listrik. Baterai primer hanya bisa dipakai sekali, karena menggunakan reaksi kimia yang bersifat tidak bisa dibalik (*irreversible reaction*). Sedangkan baterai sekunder dapat diisi ulang karena reaksi kimianya bersifat bisa dibalik (*reversible reaction*). (<http://id.wikipedia.org/wiki/Baterai>)

Dan yang akan dibahas adalah baterai sekunder yaitu baterai timbal asam atau yang biasa disebut accu / aki, adalah salah satu komponen utama dalam kendaraan bermotor, baik mobil atau motor, semua memerlukan aki untuk dapat menghidupkan mesin mobil (mencatu arus pada dinamo stater kendaraan). Aki mampu mengubah tenaga kimia menjadi tenaga listrik.

Accu untuk mobil biasanya mempunyai tegangan sebesar 12 Volt, sedangkan untuk motor ada tiga jenis yaitu, dengan tegangan 12 Volt, 9 volt dan ada juga yang bertegangan 6 Volt. Selain itu juga dapat ditemukan pula accu yang khusus untuk menyalakan tape atau radio dengan tegangan juga yang dapat diatur dengan rentang 3, 6, 9, dan 12 Volt. Tentu saja aki jenis ini dapat dimuati kembali (*recharge*) apabila muatannya telah berkurang atau habis. Namun untuk pembahasan pada skripsi saya nanti, saya menggunakan accu / baterai timbal asam untuk sumber energi dari mobil listrik sebagai pengganti bahan bakar.

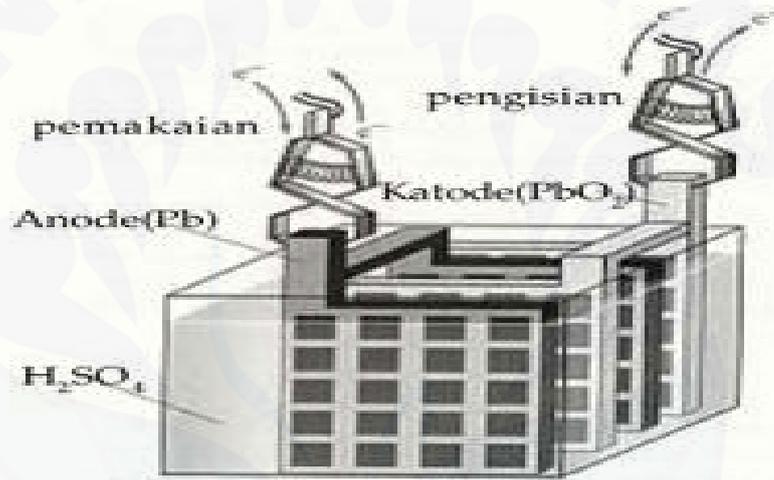
Reaksi kimia pada baterai timbal asam. Pada saat accu digunakan, tiap molekul asam sulfat (H_2SO_4) pecah menjadi dua ion hidrogen yang bermuatan positif (2H^+) dan ion sulfat yang bermuatan negatif (SO_4^-). Tiap ion SO_4^- yang berada dekat lempeng Pb akan bersatu dengan satu atom timbal murni (Pb) menjadi timbal sulfat (PbSO_4) sambil melepaskan dua elektron. Sedang sepasang ion hidrogen tadi akan ditarik lempeng timbal dioksida (PbO_2), mengambil dua elektron dan bersatu dengan satu atom oksigen membentuk molekul air (H_2O).

Dari proses ini terjadi pengambilan elektron dari timbal dioksida (sehingga menjadi positif) dan memberikan elektron itu pada timbal murni (sehingga menjadi negatif), yang mengakibatkan adanya beda potensial listrik di antara dua kutub tersebut. Proses tersebut terjadi secara simultan, reaksi secara kimia dinyatakan sebagai berikut :



Di atas ditunjukkan terbentuknya timbal sulfat selama penggunaan (*discharging*). Keadaan ini akan mengurangi reaktivitas dari cairan elektrolit karena asamnya menjadi lemah (*encer*), sehingga tahanan antara kutub sangat lemah untuk pemakaian praktis.

Sementara proses kimia selama pengisian aki (*charging*) terjadi setelah accu melemah (tidak dapat memasok arus listrik pada saat kendaraan hendak dihidupkan). Kondisi accu dapat dikembalikan pada keadaan semula dengan memberikan arus listrik yang arahnya berlawanan dengan arus yang terjadi saat discharging. Pada proses ini, tiap molekul air terurai dan tiap pasang ion hidrogen yang dekat dengan lempeng negatif bersatu dengan ion SO_4 pada lempeng negatif membentuk molekul asam sulfat. Sedangkan ion oksigen yang bebas bersatu dengan tiap atom Pb pada lempeng positif membentuk PbO_2 . Reaksi kimia yang terjadi adalah :



Gambar 2.3 Baterai timbal asam.

(Sumber: <http://www.fisikanet.lipi.go.id/>)

(Sumber: Pikiran Rakyat, 2014)

2.4 Mikrokontroler ATmega 16

Mikrokontroler adalah sebuah sistem komputer lengkap dalam satu serpih (*chip*). Mikrokontroler lebih dari sekedar sebuah mikroprosesor karena sudah terdapat atau berisikan ROM (*Read-Only Memory*), RAM (*Read-Write Memory*), beberapa bandar masukan maupun keluaran, dan beberapa peripheral seperti pencacah / pewaktu, ADC (*Analog to Digital converter*), DAC (*Digital to Analog converter*) dan serial komunikasi.

Beberapa keunggulan dari AVR ATmega16 antara lain:

2.1 *Advanced RISC Architecture*

- *130 Powerful Instructions – Most Single Clock Cycle Execution*
- *32 x 8 General Purpose Fully Static Operation*
- *Up to 16 MIPS Throughput at 16 MHz*
- *On-chip 2-cycle Multiplier*

2.2 Nonvolatile Program and Data Memories

- *8K Bytes of In-System Self-Programmable Flash*
- *Optional Boot Code Section with Independent Lock Bits*
- *512 Bytes EEPROM*
- *512 Bytes Internal SRAM*
- *Programming Lock for Software Security*

2.3 Peripheral Features

- *Two 8-bit Timer/Counters with Separate Prescalers and Compare Mode*
- *Two 8-bit Timer/Counters with Separate Prescalers and Compare Modes*
- *One 16-bit Timer/Counter with Separate Prescaler, Compare Mode, and Capture Mode*
- *Real Time Counter with Separate Oscillator*
- *Four PWM Channels*
- *8-channel, 10-bit ADC*
- *Byte-oriented Two-wire Serial Interface*
- *Programmable Serial USART*

2.4 Special Microcontroller Features

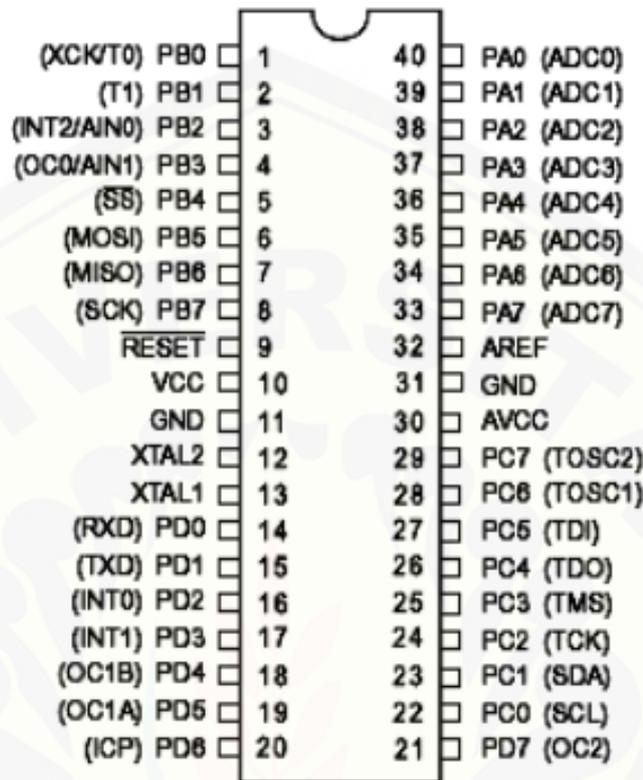
- *Power-on Reset and Programmable Brown-out Detection*
- *Internal Calibrated RC Oscillator*
- *External and Internal Interrupt Sources*
- *Six Sleep Modes: Idle, ADC Noise Reduction, Power-save, Powerdown, Standby and Extended Standby*

2.5 I/O and Package

- *32 Programmable I/O Lines*
- *40-pin PDIP, 44-lead TQFP, 44-lead PLCC, and 44-pad MLF*

2.6 Operating Voltages

- 2.7 - 5.5V for Atmega16L
- 4.5 - 5.5V for Atmega16



Gambar 2.4 Pin pin ATmega16 model 40-pin.

(Sumber: www.atmel.com)

Pin-pin pada ATmega16 dengan kemasan 40-pin DIP (*dual inline package*) ditunjukkan oleh gambar 2.3. Guna memaksimalkan performa, AVR menggunakan arsitektur Harvard (dengan memori dan bus terpisah untuk program dan data).

Port sebagai *input/output digital*. ATmega16 mempunyai empat buah port yang bernama PortA, PortB, PortC, dan PortD. Keempat *port* tersebut merupakan jalur *bidirectional* dengan pilihan *internal pull-up*. Tiap *port* mempunyai tiga buah *register bit*, yaitu DDxn, PORTxn, dan PINxn. Huruf 'x' mewakili nama huruf dari *port* sedangkan huruf 'n' mewakili nomor bit. Bit DDxn terdapat pada I/O address DDRx, bit PORTxn terdapat pada I/O address PORTx, dan bit PINxn terdapat pada I/O address PINx. Bit DDxn dalam *register DDRx (Data Direction Register)* menentukan arah *pin*. Bila DDxn diset 1 maka Px berfungsi sebagai *pin*

output. Bila DD_{xn} diset 0 maka P_x berfungsi sebagai pin input. Bila $PORT_{xn}$ diset 1 pada saat *pin* terkonfigurasi sebagai *pin input*, maka *resistor pull-up* akan diaktifkan. Untuk mematikan *resistor pull-up*, $PORT_{xn}$ harus diset 0 atau *pin* dikonfigurasi sebagai *pin output*. *Pin port* adalah *tri-state* setelah kondisi *reset*. Bila $PORT_{xn}$ diset 1 pada saat *pin* terkonfigurasi sebagai *pin output* maka *pin port* akan berlogika 1. Dan bila $PORT_{xn}$ diset 0 pada saat *pin* terkonfigurasi sebagai *pin output* maka *pin port* akan berlogika 0. Saat mengubah kondisi *port* dari kondisi *tri-state* ($DD_{xn}=0$, $PORT_{xn}=0$) ke kondisi *output high* ($DD_{xn}=1$, $PORT_{xn}=1$) maka harus ada kondisi peralihan apakah itu kondisi *pull-up enabled* ($DD_{xn}=0$, $PORT_{xn}=1$) atau kondisi *output low* ($DD_{xn}=1$, $PORT_{xn}=0$).

Biasanya, kondisi *pull-up enabled* dapat diterima sepenuhnya, selama lingkungan impedansi tinggi tidak memperhatikan perbedaan antara sebuah *strong high driver* dengan sebuah *pull-up*. Jika ini bukan suatu masalah, maka bit PUD pada *register SFIOR* dapat diset 1 untuk mematikan semua *pull-up* dalam semua *port*. Peralihan dari kondisi input dengan *pull-up* ke kondisi *output low* juga menimbulkan masalah yang sama. Kita harus menggunakan kondisi *tri-state* ($DD_{xn}=0$, $PORT_{xn}=0$) atau kondisi *output high* ($DD_{xn}=1$, $PORT_{xn}=0$) sebagai kondisi transisi.

Tabel 2.1. Konfigurasi *pin port* Mikrokontroler ATmega 16

DD_{xn}	$PORT_{xn}$	PUD (In SFIOR)	I/O	Pull – up	Comment
0	0	X	Input	No	Tri-state (Hi-Z)
0	1	0	Input	Yes	P_{xn} will source current if ext. pulled low
0	1	1	Input	No	Tri-State (Hi-Z)
1	0	X	Output	No	Output Low (Sink)
1	1	X	Output	No	Output High (Source)

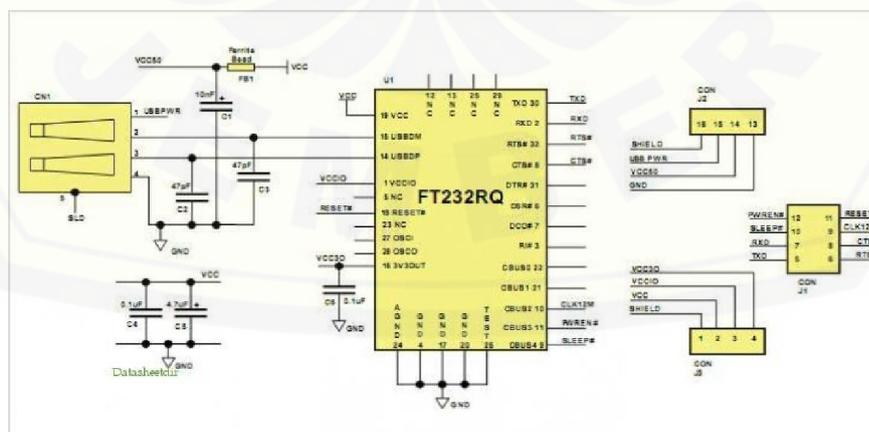
Bit 2 – PUD : *Pull-up Disable*. Bila bit diset bernilai 1 maka *pull-up* pada *port I/O* akan dimatikan walaupun *register* DD_{xn} dan $PORT_{xn}$ dikonfigurasi untuk menyalakan *pull-up* ($DD_{xn}=0$, $PORT_{xn}=1$).

(Sumber: www.atmel.com. *Datasheet ATmega16*)

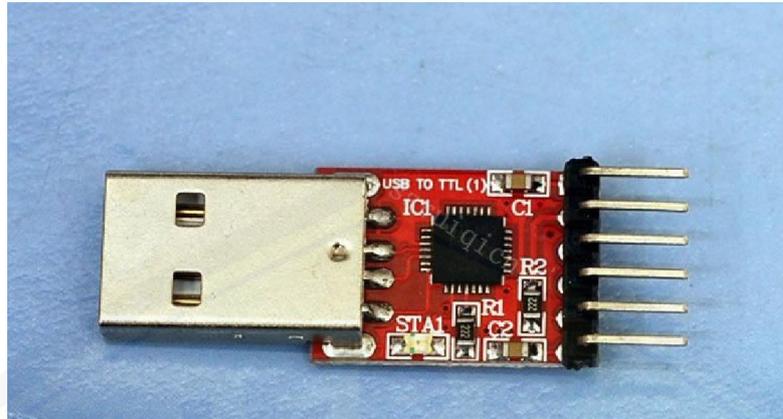
2.5 Komunikasi Serial

Komunikasi *serial* adalah komunikasi yang mengantarkan data *digital* secara bit per bit secara bergantian melalui media *interface serial*, contoh: modem, *mouse* dll. Pengiriman data melalui *interface serial* dapat dilakukan secara bit per bit (setiap satu step waktu 1 bit) atau juga dalam satuan *baud* dimana 1 baud tidak mesti senilai dengan 1 *bit per second*, tergantung besaran data untuk setiap kali *clock transfer*.

Pada PC atau Laptop-laptop keluaran terbaru, jumlah *port serial RS232* semakin lama semakin berkurang, jika pada PC lama biasanya terdapat dua buah konektor RS232 maka sekarang berkurang menjadi satu buah konektor RS232 saja. Keberadaan *port serial RS232* sekarang telah digantikan oleh *port USB* yang memang mempunyai banyak kelebihan dibandingkan *port serial RS232*. Cara untuk merubah *port USB* agar dapat berkomunikasi layaknya melalui *port RS232* misalnya adalah, dengan menggunakan IC FT232. IC ini adalah IC USB UART. Secara garis besar IC ini berfungsi untuk merubah data USB yang berasal dari *port USB* menjadi data serial dengan level tegangan TTL. Sehingga dengan menggunakan IC FT232 para pengguna dapat melakukan komunikasi data serial (UART) melalui *port USB*. Semua proses *handshaking, enumerasi* dan lain-lain yang diperlukan agar dapat menggunakan *port USB* telah ditangani oleh IC FT232 tersebut, sehingga memudahkan pengguna untuk menggunakannya. Sistem komunikasi yang digunakan sebagai komunikasi dari usb komputer ke masukan ATmega16 adalah *USB to Serial* dengan memanfaatkan IC FT232.

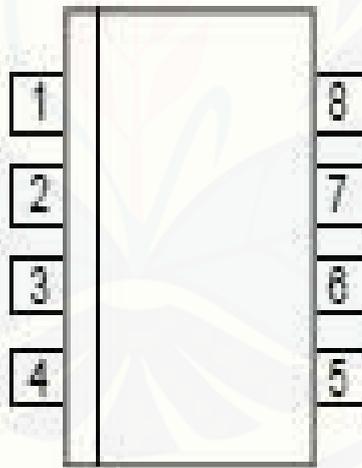


Gambar 2.5 Konfigurasi *usb to serial* menggunakan IC FT232

Gambar 2.6 *Usb to serial*(Sumber : www.yourduino.com)

2.6 Sensor Arus

Sensor arus adalah alat yang digunakan untuk mengukur kuat arus listrik. Sensor arus ini menggunakan metode *Hall Effect Sensor*. *Hall Effect Sensor* merupakan sensor yang digunakan untuk mendeteksi medan magnet. Disini saya menampilkan sensor arus tipe ACS712 sebagai contoh.



Gambar 2.7 Sensor arus ACS712

(Sumber: www.joeldra.blogspot.com)

Hall Effect Sensor akan menghasilkan sebuah tegangan yang proporsional dengan kekuatan medan magnet yang diterima oleh sensor tersebut. Pendeteksian perubahan kekuatan medan magnet cukup mudah dan tidak memerlukan apapun selain sebuah induktor yang berfungsi sebagai sensornya. Kelemahan dari

detektor dengan menggunakan induktor adalah kekuatan medan magnet yang statis (kekuatan medan magnetnya tidak berubah) tidak dapat dideteksi. Oleh sebab itu diperlukan cara yang lain untuk mendeteksinya yaitu dengan sensor yang dinamakan dengan *'hall effect'sensor*. Sensor ini terdiri dari sebuah lapisan silikon yang berfungsi untuk mengalirkan arus listrik. Dengan metode ini arus yang dilewatkan akan terbaca pada fungsi besaran tegangan berbentuk gelombang sinusoidal dengan menggunakan aplikasi Visual Studio 2010 Ultimate yang akan ditampilkan dikomputer.

Tabel 2.2 *Pin port* pada sensor arus ACS 712

No.	Nama	Keterangan
1 dan 2	IP+	Masukan arus
3 dan 4	IP-	Keluaran arus
5	GND	<i>Ground</i>
6	N>C>	Terminal untuk kapasitor eksternal, untuk menentukan <i>bandwitch</i>
7	VOUT	Keluaran tegangan analog
8	VCC	<i>Power Supply</i>

(Sumber: Dwi Sutadi, 2002)

Disini sensor arus tersebut akan digunakan sebagai pengukur arus, yang nantinya akan berhubungan dengan tegangan, daya beban, dan kapasitas baterai. Dengan menggunakan beberapa komponen lain yang nantinya nilai arus, tegangan dan kapasitas daya akan ditampilkan dalam bentuk grafik dengan aplikasi *Visual Studio 10 Ultimate* pada komputer

2.7 *Analog To Digital Converter (ADC0804)*

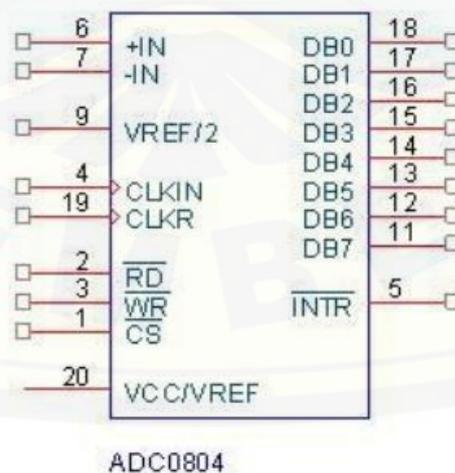
Piranti-piranti dan sistem logika hanya mengenal besaran *analog*, sehingga sebelum diumpankan ke sistem *digital*, besaran ini harus diubah ke dalam bentuk digital. ADC adalah kepanjangan dari *Analog to Digital Converter* yang artinya pengubah dari *analog* ke *digital*.

Suatu pengubah *analog* ke *digital* (ADC) adalah suatu alat yang mampu mengubah sinyal masukan *analog* menjadi keluaran *digital*. Data-data *digital* yang dihasilkan ADC hanyalah merupakan pendekatan proporsional terhadap masukan *analognya*. Hal ini karena tidak mungkin melakukan konversi secara

sempurna berkaitan dengan kenyataan bahwa informasi *digital* berubah dalam step-step, sedangkan *analog* berubahnya secara kontinyu (Agfianto, 2002).

Fungsi dari ADC adalah untuk mengubah data *analog* menjadi data *digital* yang nantinya akan masuk ke suatu komponen *digital*. Inputan dari ADC ini ada 2 yaitu *input* positif (+) dan *input* negatif (-). ADC 0804 ini terdiri dari 8 bit *microprocessor Analog to Digital Converter*. $V (+)$ dan $V (-)$ adalah inputan tegangan *analog differensial* sehingga data tegangan yang akan diproses oleh ADC adalah selisih antara $V_i (+)$ dan $V_i (-)$. V_{ref} adalah tegangan referensi ADC yang digunakan untuk mengatur tegangan input pada V_{i+} dan V_{i-} . Besarnya tegangan referensi ini adalah setengah dari tegangan *input* maksimal. Hal ini bertujuan agar pada saat inputan maksimal data *digital* juga akan maksimal.

Chip select fungsinya untuk mengaktifkan ADC yang diaktifkan dengan logika *low*. *Read* adalah inputan yang digunakan untuk membaca data *digital* hasil konversi yang aktif pada kondisi logika *low*. *Write* berfungsi untuk melakukan *start* konversi ADC diaktifkan pada kondisi logika *low*. Instruksi berfungsi untuk mendeteksi apakah konversi telah selesai atau tidak, jika sudah selesai maka pin instruksi akan mengeluarkan logika *low*. Data *output-an digital* sebanyak 8 byte (DB0-DB7) biner 0000 0000 sampai dengan 1111 1111, sehingga kemungkinan angka desimal yang akan muncul adalah 0 sampai 255 dapat diambil pada *pin* D0 sampai D7. DB0-DB7 mempunyai sifat *latching*.



Gambar 2.8 Konfigurasi pin ADC0804
(Sumber: Heri Kuswanto, 2012)

Deskripsi Fungsi *Pin* ADC 0804

- a. WR, pulsa transisi *high to low* pada *input write* maka ADC akan melakukan konversi data, tegangan *analog* menjadi data *digital*.
- b. INT, bila konversi data *analog* menjadi *digital* telah selesai maka *pin* INT akan mengeluarkan pulsa transisi *high to low*. Perangkat ADC dapat dioperasikan dalam mode *free running* dengan menghubungkan *pin* INT ke *input* WR.
- c. CS, agar ADC dapat aktif melakukan konversi data maka *input chip select* harus diberi logika *low*.
- d. RD, agar data ADC data dapat dibaca oleh sistem mikroprosesor maka *pin* RD harus diberi logika *low*.
- e. Tegangan *analog input* deferensial, *input* Vin (+) dan Vin (-) merupakan *input* tegangan deferensial yang akan mengambil nilai selisih dari kedua *input*. Dengan memanfaatkan *input* Vin maka dapat dilakukan *offset* tegangan nol pada ADC.
- f. Vref, tegangan referensi dapat diatur sesuai dengan *input* tegangan pada Vin (+) dan Vin (-).
- g. CLOCK, *clock* untuk ADC dapat diturunkan pada *clock* CPU atau RC eksternal dapat ditambahkan untuk memberikan generator *clock* dari dalam CLK In menggunakan *schmitt trigger*

(Sumber: Heri Kuswanto, 2010)

2.8 Visual Studio 10 Ultimate

Microsoft Visual Studio merupakan sebuah perangkat lunak lengkap (*suite*) yang dapat digunakan untuk melakukan pembuatan aplikasi, pengembangan aplikasi, baik itu aplikasi bisnis, aplikasi personal, ataupun komponen aplikasinya, dalam bentuk aplikasi console, aplikasi Windows, ataupun aplikasi Web.

(Sumber : id.wikipedia)

Visual Basic merupakan salah satu bahasa pemrograman komputer yang mendukung (*Object Oriented Programming* = OOP). Visual basic yaitu bahasa pemrograman keluaran Microsoft yang merupakan pengembangan dari pendahulunya yaitu bahasa pemrograman BASIC yang dikembangkan pada era

1950-an. Visual Basic merupakan salah satu Development Tool yaitu alat bantu untuk membuat berbagai macam program komputer, khususnya yang menggunakan sistem operasi Windows.



Gambar 2.9 Microsoft Visual Studio 2010 Express

(Sumber : <http://www.ilmukomputer.com>)

BAB 3. METODOLOGI PENELITIAN

3.1 Jadwal Penelitian

Pembuatan perangkat keras dan perangkat lunak dari sistem monitoring arus dan tegangan pada baterai mobil listrik dilakukan setelah pelaksanaan seminar proposal yang pelaksanaan kegiatan penelitian dilakukan di Laboratorium Jaringan Komputer Fakultas Teknik Universitas Jember selama 5 bulan.

3.2 Tahap Perencanaan

Dalam perancangan system monitoring arus dan tegangan pada baterai mobil listrik berbasis mikrokontroler ATmega 16 dibutuhkan langkah – langkah sebagai berikut :

1. Studi literatur

Tahap awal dari penelitian ini adalah mencari literatur tentang penelitian yang sedang diteliti. Diharapkan dengan literatur yang didapat bisa memberikan keyakinan bahwa penelitian ini dapat dilaksanakan dan memberikan arahan untuk mengurangi kesalahan dalam penelitian.

2. Pembelian bahan pembuatan

Tahap kedua adalah pembelian material yang dibutuhkan untuk membuat system monitoring arus dan tegangan pada baterai mobil listrik yang meliputi baterai timbal asam, sensor arus, sensor tegangan dan mikrokontroler ATmega 16 sebagai komponen utama, komponen-komponen elektronik aktif dan pasif yang dibutuhkan seperti kabel serial dan perangkat pendukung lainnya.

3. Pengerjaan perangkat keras

Tahap ketiga adalah proses pengerjaan pembuatan *system minimum* dengan mikrokontroler ATmega 16 serta kalibrasi sensor arus dan tegangan yang meliputi pembuatan perangkat mekanik dan elektronik dari bahan pembuatan yang sudah dibeli.

4. Pembuatan perangkat lunak

Tahap keempat adalah pembuatan perangkat lunak, yang pertama menggunakan *software Proteus* untuk mendesain pemodelan pcb pada *system minimum*. Yang kedua menggunakan *software Code Vision AVR* sebagai media kalibrasi sensor arus dan tegangan pada rangkaian *system minimum* mikrokontroler ATmega16. Yang ketiga *software Visual Studio 2010 C#* sebagai media *monitoring* grafik arus dan tegangan yang diharapkan akan menjadi akhir dari analisa data dan penelitian.

5. Pengujian alat

Tahap kelima adalah pengujian alat diharapkan dengan adanya tahap ini fungsi kerja baik perangkat keras dan perangkat lunak yang telah dibuat dan menguji sistem yang telah terintegrasi secara menyeluruh, sehingga nantinya alat yang telah dibuat dapat bekerja secara maksimal.

6. Analisa sistem

Tahap keenam adalah menganalisa kinerja alat baik yang berupa perangkat keras dan perangkat lunak sistem yang telah diuji. Sehingga diharapkan dari penganalisaan sistem keseluruhan perangkat dapat ditemukan suatu pembahasan atas analisa yang telah dibuat terhadap sistem yang telah dibuat dan dapat ditemukan kesimpulan terhadap apa yang sedang diamati.

3.3 Perancangan Perangkat Keras

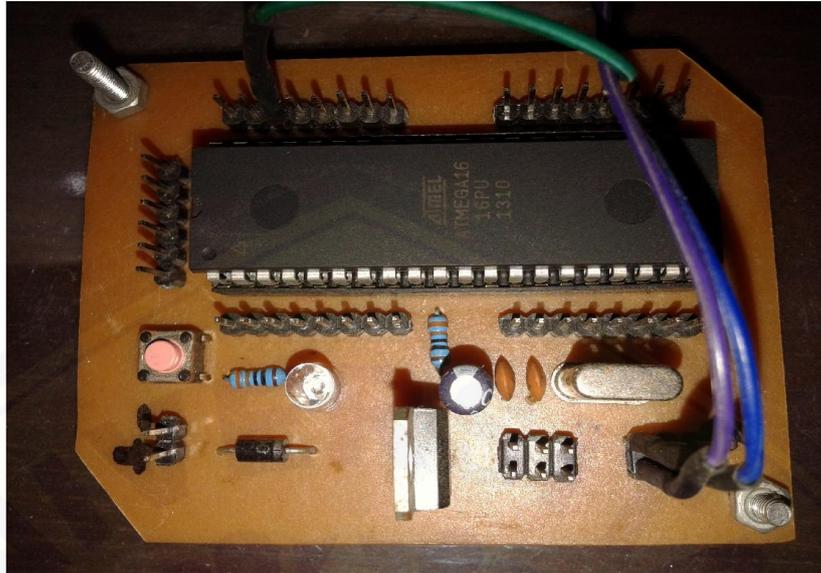
Pembuatan perangkat keras ini berdasarkan blok diagram sistem pada gambar 3.1 yaitu tersusun dari baterai, beban lampu DC, 2buah sensor, rangkaian *system minimum*, dan laptop sebagai tampilan monitoring untuk GUI. Kemudian dari diagram blok perangkat keras tersebut dapat kita pilah lagi pada spesifikasi dari rangkaian *system minimum*, sensor tegangan serta sensor arus. Dan untuk skema atau konsep dari penataan sendiri akan diperlihatkan pada gambar 3.5 sesuai bentuk rancangan perangkat keras.



Gambar 3.1 Diagram Blok Sistem Monitoring Arus dan Tegangan Pada Baterai Mobil Listrik Berbasis ATmega16

3.3.1 Blok Perangkat Keras *System Minimum* Mikrokontroler ATmega16

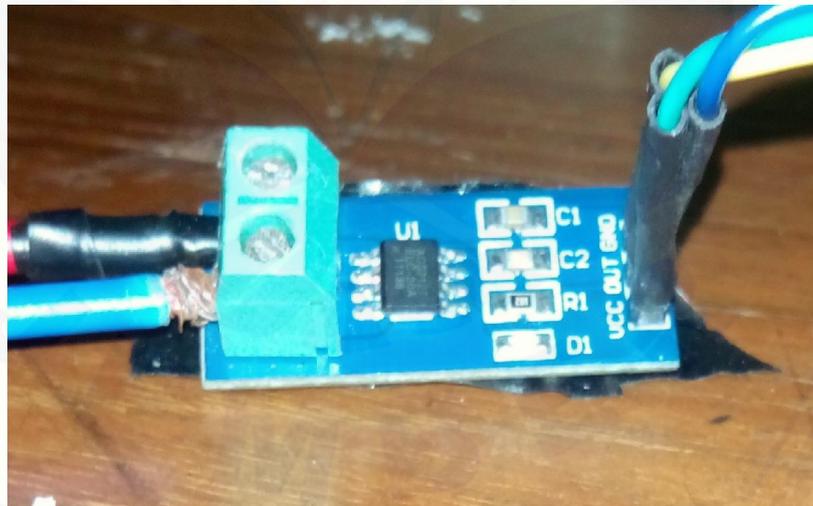
Pada perangkat keras rangkaian *system minimum* tersebut terdiri dari Mikrokontroler ATmega16, IC805, elco, resistor, kapasitor, crystal, LED. Mikrokontroler ini akan berfungsi untuk menerima data sensor yang kemudian akan diolah dan berproses pada ADC dilanjutkan mengirim data ke komputer. Mikrokontroler ini adalah otak yang akan mengolah dari *input* sensor arus maupun sensor tegangan. Karena pengkalibrasian dari sensor arus sendiri didapat dari pengolahan pada mikrokontroler ATmega 16.



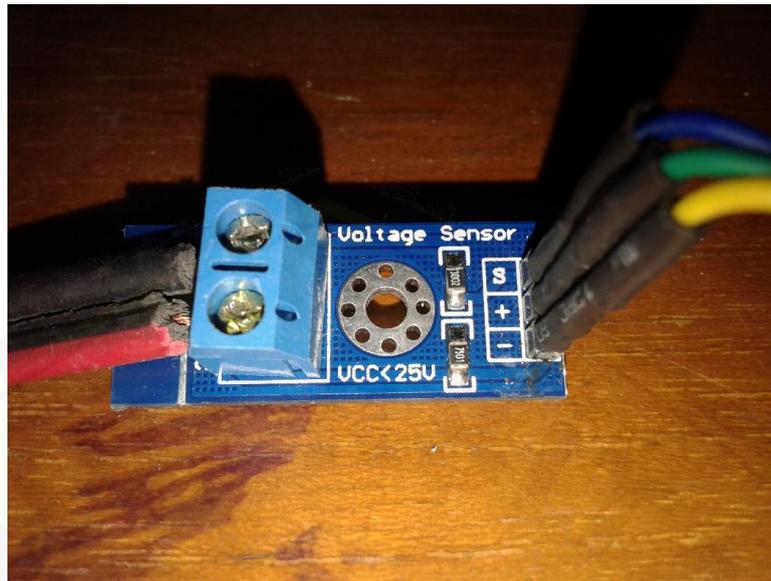
Gambar 3.2 Rangkaian *System Minimum* ATmega16

3.3.2 Blok Perangkat Keras Sensor Arus dan Sensor Tegangan

Pada perangkat keras sensor arus menggunakan tipe ACS712 (Gambar 3.3). Pada sensor arus dan sensor tegangan (Gambar 3.4) terdapat 3 *Port*, yaitu VCC sebagai kutub positif (+). GROUND sebagai kutub negative (-), dan OUT atau S sebagai *output* dari masing-masing sensor.

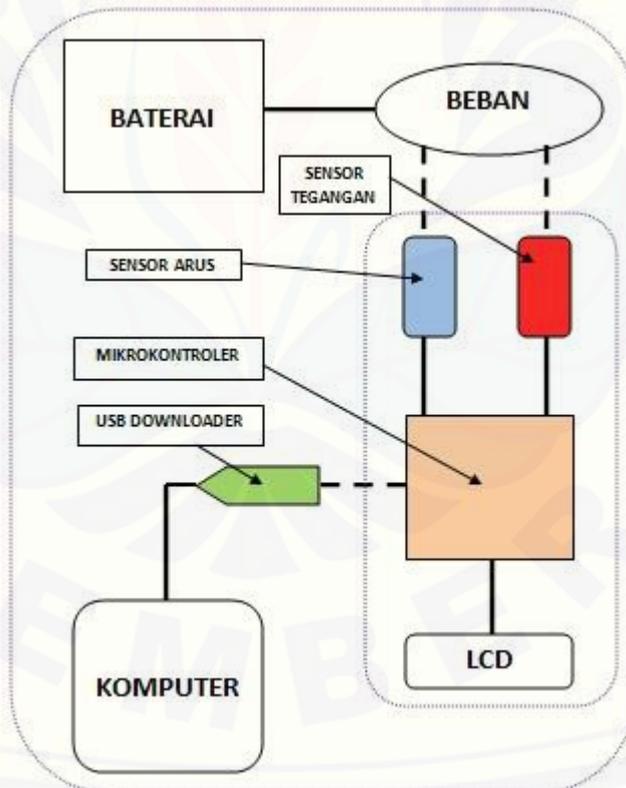


Gambar 3.3 Sensor Arus



Gambar 3.4 Sensor Tegangan

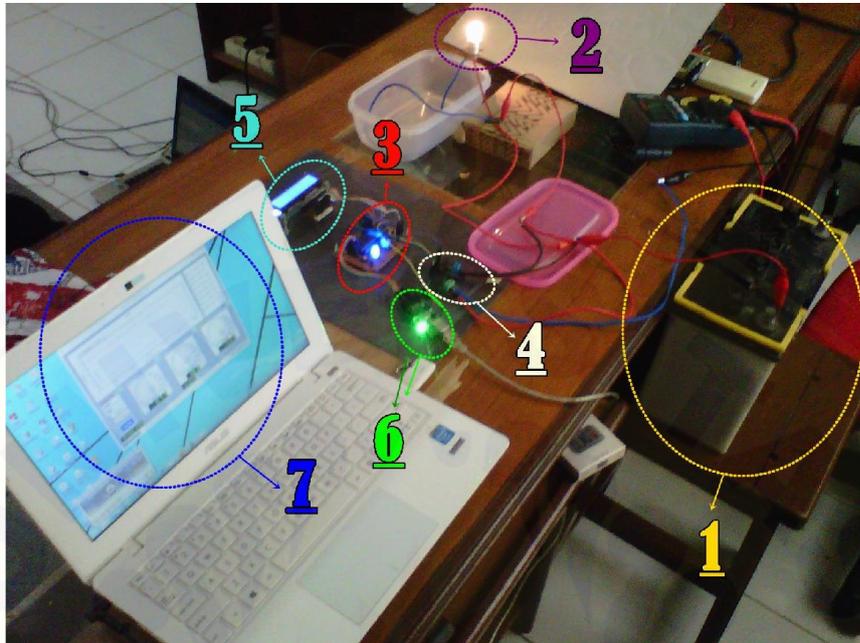
3.3.3 Konsep Sistem Dan Penataan Alat



Gambar 3.5 Skema konsep penataan alat

Konsep sistem terlihat seperti gambar 3.5. Pada gambar tersebut merupakan bentuk nyata dari pemodelan alat, tahap pertama yaitu dari baterai,

kemudian sensor, masuk ke mikrokontroler, dan dikirim ke komputer dalam bentuk *serial*.



Gambar 3.6 Sistem letak alat

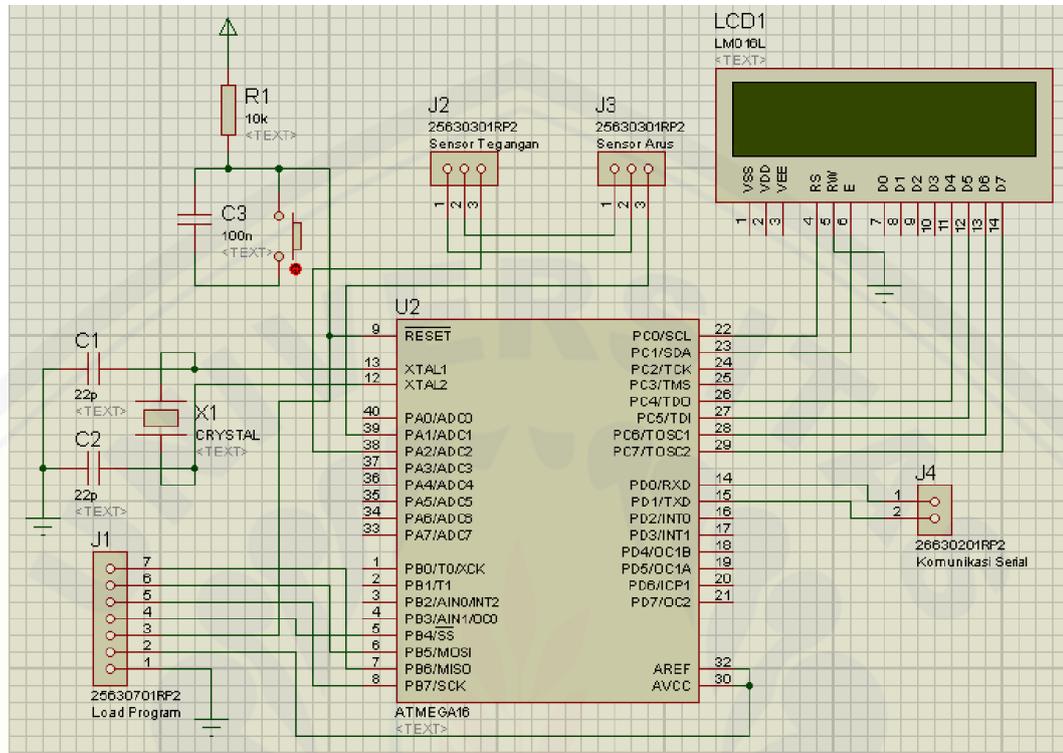
Keterangan :

1. Baterai timbal asam 45 AH
2. Beban lampu DC
3. Rangkaian *system minimum* mikrokontroler ATmega 16
4. Sensor arus dan sensor tegangan
5. LCD
6. USB downloader K125 R
7. Tampilan GUI

Pada penataan alat sebenarnya tidak terlalu banyak perbedaan, tetapi di sini sangat perlu diperhatikan. Karena pada rangkaian sangatlah rawan terjadi hubung singkat pada kabel-kabel jumper. Apabila terjadi hubung singkat pastinya akan membuat kinerja alat menjadi tidak maksimal. Untuk kabel jumper sendiri juga dapat dicek dengan menggunakan avometer, apakah kabel aman ataupun layak digunakan.

3.4 Perancangan Perangkat Lunak

3.4.1 Perancangan Leader Diagram Proteus



Gambar 3.7 Gambar rangkaian *System Minimum*

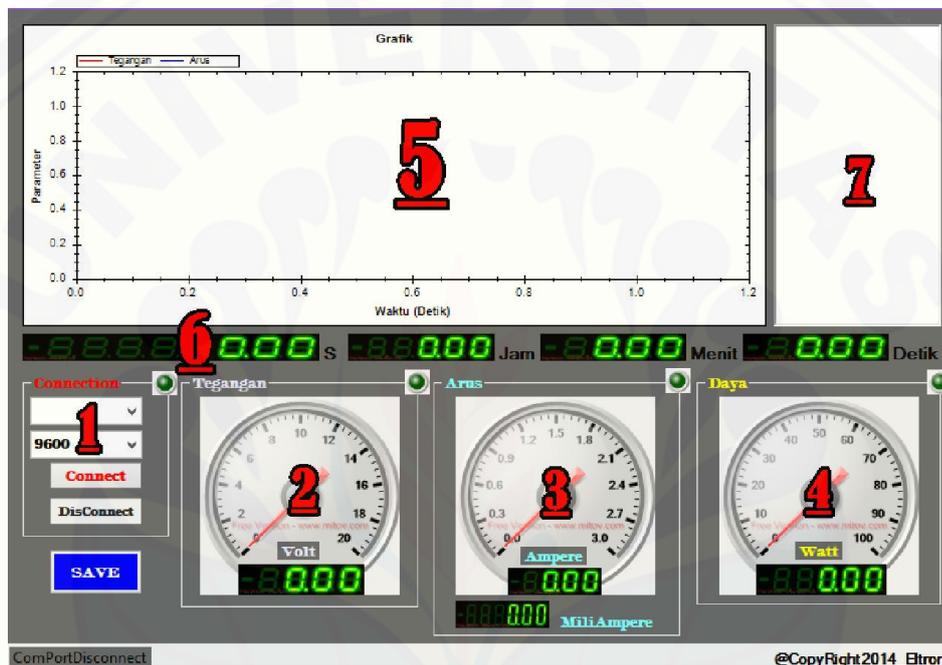
Pada gambar 3.6 adalah keseluruhan dari skema rangkaian pada *system minimum* Mikrokontroler ATmega16 disertai sensor arus dan tegangan. Sensor arus adalah tipe ACS0712. Pada rangkaian ini sensor arus masuk ke mikrokontroler menggunakan alamat Port PA1/ADC1, sedangkan untuk sensor tegangan menggunakan alamat Port PA2/ADC2. Pada tampilan LCD sendiri melalui Port C, dan untuk serial komunikasinya menggunakan Port D.

3.4.2 Pemrograman CodeVision AVR

Pada perangkat lunak CV AVR, disini kita akan melakukan proses kalibrasi antara mikrokontroler dengan sensor arus dan tegangan. Pada bentuk pemrograman CV AVR kita akan mencari perhitungan rumus secara manual untuk mendapatkan sinkronisasi antara mikrokontroler dengan sensor arus dan tegangan.

3.4.3 Aplikasi GUI (*Graphic User Interface*)

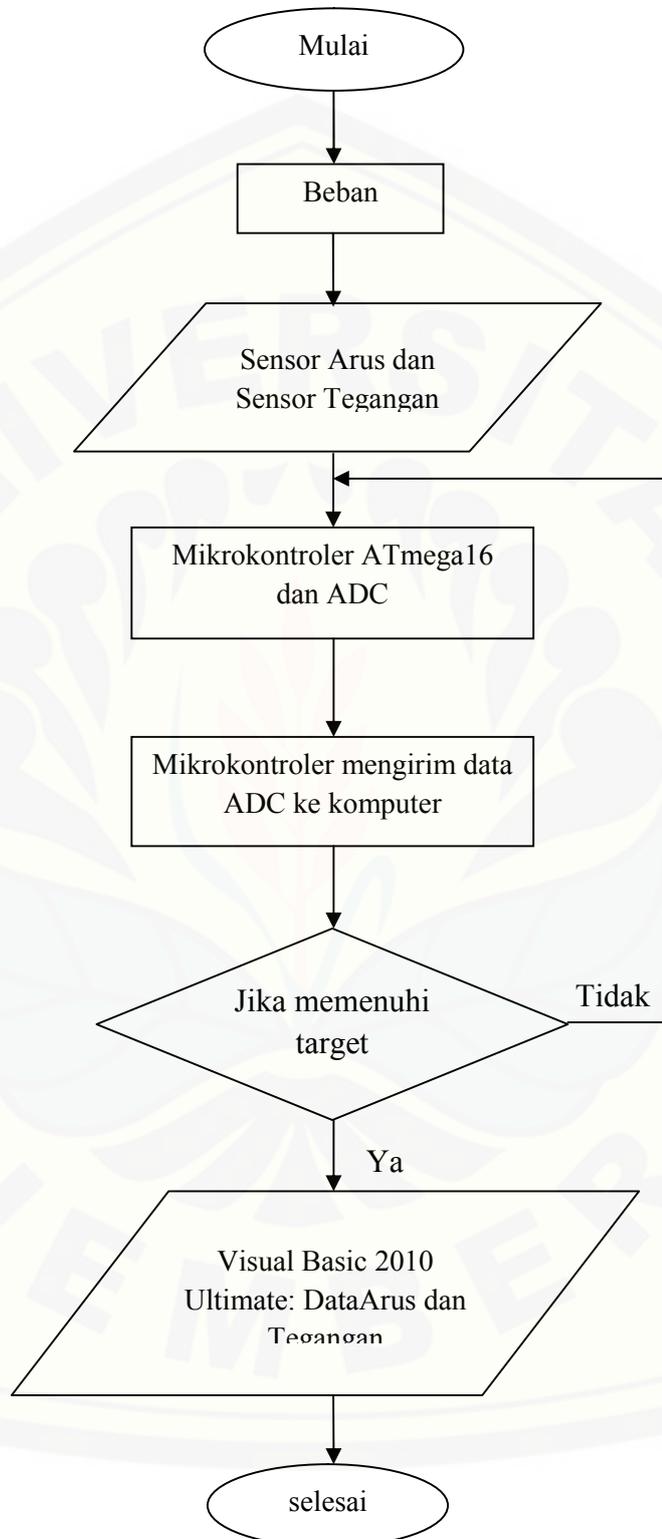
Aplikasi GUI yang telah dibuat menggunakan Microsoft visual studio 2010 ultimate untuk memantau nilai-nilai setiap sensor. Aplikasi ini menggunakan komunikasi serial yang digunakan sebagai *receiver*. Nilai-nilai tersebut dikirim melalui telemetri pada COM *Receiver* menggunakan USB *Downloader* K125R. GUI akan menampilkan data yang dikirimkan Mikrokontroler dan ditampilkan dalam bentuk grafik. Bentuk tampilan *monitoring* adalah sebagai berikut :



.Gambar 3.8 Tampilan Visual Basic sebagai *monitoring* alat

1. Tampilan pengaturan *connecting* dengan rangkaian sensor arus dan tegangan dalam bentuk serial.
2. Tampilan bentuk *analog* dan *digital* nilai tegangan
3. Tampilan bentuk *analog* dan *digital* nilai arus
4. Tampilan bentuk *analog* dan *digital* nilai kapasitas beban
5. Tampilan grafik, untuk grafik biru adalah arus dan grafik merah adalah tegangan
6. Tampilan digital waktu, meliputi sekon atau detik, menit, dan jam
7. Tampilan data nilai arus dan tegangan tiap jam, mulai detik jam 1, 2, 3.

3.5 Flowchart Sistem Alat Secara Keseluruhan



Gambar 3.9 Flowchart sistem secara keseluruhan

Pada *flowchart* diatas pertama kita memastikan bahwa baterai pada keadaan terisi penuh dengan kapasitas daya 12Volt 45AH. Kemudian baterai dialiri beban dengan 3 varian, yaitu 18Watt, 25Watt, 35Watt. Pengujian dilakukan bertahap, di mulai dari nilai beban yang paling besar. Karena bisa dibuat perbandingan bahwa nilai beban yang paling besar adalah posisi ketika mesin pada mobil listrik dalam keadaan *starting*. Sebagai contohnya kita menggunakan nilai beban 35 Watt. Dan pada beban 2 5Watt adalah keadaan *konstan*, sedangkan beban 18 Watt adalah keadaan *breaking* atau pengereman. Kemudian ditambah beban *couple* dengan kapasitas 43 Watt, 53 Watt, 60 Watt, dan 78 Watt.

Setelah tegangan dan arus melalui beban, disini sensor arus akan membaca berapa nilai arus yang terdapat pada rangkaian tersebut. Nilai yang didapat oleh sensor arus akan dibaca mikrokontroler. Disini sebelumnya mikrokontroler akan kita program sesuai dengan karakteristik sensor arus yang akan kita pakai.

Setelah mikrokontroler berhasil menampilkan data sesuai program, disini akan diteruskan ke ADC. Fungsi dari ADC sendiri adalah mengubah data atau sinyal yang berupa analog (dari mikrokontroler) menjadi bentuk digital. Dan disinilah kita menggunakan *serial port* untuk bisa merubah data menjadi bentuk angka atau bit yang selanjutnya akan dikirim ke aplikasi GUI atau komputer.

Pada tahap ini bisa saja terjadi kesalahan sehingga data yang dikirim ke komputer tidak terbaca. Untuk menghindari *error* pada rangkaian, disini kita menggunakan *If* pada *flowchart*. Fungsi dari *If* sendiri adalah untuk mengulangi terjadinya kesalahan program pada saat *transfer* ke komputer. Jika data yang dikirimkan tidak sesuai dan aplikasi GUI tidak bisa membaca masukan, otomatis proses program akan kembali ke tahap *input* pada mikrokontroler.

Tetapi apabila aplikasi GUI dapat membaca *serial* yang dikirimkan melalui USB *downloader* maka program dapat menghasilkan *output* yang kita inginkan. Yaitu menampilkan data grafik arus, tegangan dan nilai kapasitas daya beban. Sehingga dapat diramalkan kapan kapasitas baterai akan habis.

3.6 Rumus

Ada beberapa rumus yang digunakan pada penelitian ini, salah satunya adalah hubungan antara arus, tegangan, dan daya. Yaitu

$$P = V \cdot I \quad (3.1)$$

Keterangan :

1. P = Daya (Watt)
2. V = Tegangan (Volt)
3. I = Arus (Ampere)

Rumus 3.1 digunakan untuk menghitung berapa besar kapasitas beban dalam beberapa waktu disaat pengujian berlangsung. Rumus tersebut juga dimasukkan dalam GUI untuk kapasitas beban. Selain itu, rumus hubungan arus dan tegangan tersebut juga digunakan untuk menghitung nilai kapasitas baterai pada saat pengujian sistem keseluruhan akan dimulai.

Selanjutnya adalah rumus *error* % dibagi dalam 2 rumus. Yang pertama adalah rumus *error* % untuk target dari alat, sensor arus, sensor tegangan.

$$\text{error \%} = \frac{\text{data pembacaan} - \text{data alat ukur}}{\text{data alat ukur}} \quad (3.2)$$

Dengan perhitungan rumus 3.2 dapat ditentukan apakah alat sensor arus maupun tegangan sudah layak digunakan atau masih perlu dibenahi. Untuk ketentuan *error* % alat adalah 10 % .

Kemudian adalah *error* waktu, rumus ini digunakan untuk membandingkan waktu target dari seharusnya kapasitas baterai habis dengan waktu pada saat pengujian selesai. Untuk rumus waktu target sendiri adalah sebagai berikut:

$$\text{waktu awal} = \frac{\text{kapasitas baterai (45 AH)}}{\text{arus awal}} \quad (3.3)$$

Dari rumus 3.3 kemudian dapat digunakan untuk menghitung *error* waktu, dengan rumus sebagai berikut

$$\text{error waktu} = \text{waktu target} - \text{waktu pengujian selesai} \quad (3.4)$$

Dari keseluruhan rumus yang dipakai akan didapat berapa kelayakan alat untuk dipakai dalam penelitian dithahap selanjutnya.

BAB 4. HASIL DAN ANALISA DATA

4.4.1. Pengujian Sensor

4.1.1 Pengujian Sensor Arus

Pada pengujian sensor arus akan didapatkan arus dengan satuan *ampere*. Tipe sensor yang digunakan adalah ACS712, tahap pertama yang dilakukan dilakukan adalah dengan menemukan rumus kalibrasi dengan mikrokontroler ATmega 16. Pada pengkalibrasian digunakan perangkat lunak CV AVR, dan di temukan rumus kalibrasinya adalah $arus = arus + (0.049*a-25.45)$. Fungsi dari kalibrasi ini sendiri adalah untuk menentukan nilai pembacaan dari sensor arus ketika tidak ada sumber dan tidak ada beban bernilai 0.

Untuk pengujian alat sensor arus ini digunakan lampu DC sebagai beban, kemudian kita bandingkan antara nilai pembacaan avometer dan nilai pembacaan pada sensor arus. Adapun variabel bebannya menggunakan 7 variasi beban.



Gambar 4.1 Pengujian sensor arus dengan beban 18 Wat

Pada gambar 4.1 terlihat bahwa pembacaan pada avometer adalah 1 A, sedangkan pada alat sensor aru adalah 0,97 A, didapat selisihnya adalah 0,03 A.



Gambar 4.2 Pengujian sensor arus dengan beban 25 Watt

Terlihat pada gambar 4.2 nilai arus yang keluar pada avometer adalah 1,31 A, sedangkan pada alat sensor arus adalah 1,29 A, sehingga didapat selisihnya 0,02 A. Pada gambar 4.3 nilai apa avometer 1,77 A, sedangkan pada sensor arus 1,78 A, didapat selisihnya sangat lah kecil hanya 0,01 A.



Gambar 4.3 Pengujian sensor arus dengan beban 35 Watt



Gambar 4.4 Pengujian sensor arus dengan beban 43 Watt

Pada pengujian sensor arus dengan yang ke 4 adalah dengan menghubungkan parallel atau di *couple* dari beban 18 Watt dan 25 Watt. Disini dilakukan hubung *parallel* dikarenakan untuk menambah kapasitas beban supaya lebih tinggi. Untuk pengujian ke 5 dan seterusnya juga digunakan beban dengan hubung paralel. Pada gambar 4.4 didapat nilai arus pada avometer adalah 2,3 A, sedangkan pada alat sensor arus 2,3 A. Pada pengujian alat sensor arus beban ke 4 ini bisa dikatakan sempurna, karena selisihnya adalah 0, jadi *error %* adalah 0.



Gambar 4.5 Pengujian sensor arus dengan beban 53 Watt



Gambar 4.6 Pengujian sensor arus dengan beban 60 Watt



Gambar 4.7 Pengujian sensor arus dengan beban 78 Watt

Untuk gambar 4.5 didapat nilai dari avometer 2,76 A dan pada sensor arus 2,83 A sehingga mempunyai selisih 0,07 A. Dan untuk gambar 4.6 dengan beban 60 Watt didapat nilai dari avometer 3,09 A dan pada sensor arus 3,13 A, sehingga selisihnya 0,04 A. Untuk pengujian sensor yang terakhir dengan beban 78 Watt dapat

dilihat pada gambar 4.7. Didapat nilai pada avometer 4,07 dan pada sensor arus 4,15 A, sehingga didapat selisihnya 0,08 A.

Tabel 4.1 Dat hasil pengujian sensor arus

Beban (Watt)	Avometer (Ampere)	Sensor Arus (Ampere)	<i>Error %</i>
0	0	0,8	0
18	1	0,97	3
25	1,31	1,29	1,52
35	1,77	1,78	0,56
18+25	2,3	2,3	0
18+35	2,76	2,83	2,53
25+35	3,09	3,13	1,29
18+25+35	4,03	4,15	2,97

Dan disini didapat data bahwa *error* persen tertinggi adalah 3 % pada beban 18 Watt dengan nilai arus pada avometer 1 A dan pada sensor arus 0,97 A. Kemudian nilai *error* persen terendah adalah 0 %, disini bisa dikatakan sempurna. Yaitu pada beban *parallel* 18 +25 Watt dengan nilai arus 2,3 A pada avometer maupun sensor arus. Karena *error* persen dari data pengujian sensor arus tidak ada yang melebihi 10 % maka alat sensor arus siap digunakan pada tahap penelitian selanjutnya.

4.1.2 Pengujian Sensor Tegangan

Pada pengujian sensor tegangan ini didapatkan nilai tegangan. Satuan pengukuran tegangan adalah *volt*. Pengujian ini dilakukan di gedung C Fakultas teknik Universitas jember. Pengujian ini dilakukan dengan tujuan mengetahui kinerja dari tegangan. Pengujian ini dilakukan dengan cara mengaliri sensor tegangan dengan sumber (*power supply*). Kemudian dihubungkan *parallel* dengan alat ukur yaitu Avometer. Pada kondisi tersebut dapat dilakukan *monitoring* pada tegangan. Pengujian dilakukan sebanyak 6 kali dengan tegangan terendah 0 V dan maksimal 15 V.



Gambar 4.8 Pengujian tegangan 3 Volt

Pada gambar 4.8 adalah data dari alat ukur avometer dan data pembacaan dari sensor tegangan, tegangan yang dialirkan sebesar 3 V. Dan terlihat selisih dari data pembacaan dan alat ukur adalah 0,07 V. Dari selisih tersebut nantinya dapat diperoleh *error* %, begitu juga dengan pengujian nilai tegangan selanjutnya.



Gambar 4.9 Pengujian tegangan 6 Volt



Gambar 4.10 Pengujian tegangan 9 Volt

Dapat dilihat pada gambar 4.9 adalah pengujian sensor tegangan dengan tegangan 6 V sesuai pada pembacaan avometer, sedangkan pada sensor tegangan sendiri terbaca 6,16 V, selisihnya 0,16 V. Untuk gambar 4.10 pada tampilan avometer

terbaca 9 V, sedangkan pada alat sensor tegangan 9,27 V, sehingga didapat selisihnya 0,27 V.



Gambar 4.11 Pengujian tegangan 12 Volt

Pada gambar 4.11 nilai yang terbaca pada avometer adalah 12 V, sedangkan pada alat sensor tegangan 12,39 V, sehingga diperoleh selisihnya 0,39 V. Untuk gambar 4.12 sendiri didapat data pada avometer adalah 15 V, sedangkan pembacaan pada alat adalah 15,48 V, dan didapat selisih 0,48 V.



Gambar 4.12 Pengujian tegangan 15 Volt

Pada pengujian sensor tegangan didapat nilai selisih tertinggi adalah pada pengujian dengan tegangan 15 V yang mempunyai selisih 0,48 V. Akan tetapi untuk menghitung nilai *error* % tidak hanya dengan mengetahui selisih tegangannya saja.

Untuk ketentuan nilai *error* % pada pengujian sensor tegangan digunakan rumus yang sama seperti pada pengujian sensor arus. Begitu juga dengan keberhasilan perancangan alat yang *error* % harus dibawah 10 %. Data dari pengujian sensor tegangan adalah sebagai berikut :

Tabel 4.2 Data hasil pengujian sensor tegangan

No.	Avometer (Volt)	Sensor Tegangan (Volt)	<i>Error %</i>
1.	0	0	0
2.	3	3,07	2,33
3.	6	6,16	2,67
4.	9	9,27	3
5.	12	12,39	3,25
6.	15	15,48	3

Dari data diatas diperoleh hasil dari pengujian alat sensor tegangan dengan nilai *error* persen tertinggi adalah 3,25 % pada tegangan 12 V di avometer, dan 12,39 V di alat sensor tegangan. Kemudian untuk *error %* terendah adalah 2,33 % pada tegangan 3 V di avometer dan 3,07 V di alat sensor tegangan. Kesimpulannya pada pengujian sensor tegangan ini adalah sudah siap digunakan untuk tahap penelitian selanjutnya.

4.4.2. Pengujian *Serial*

Tabel 4.3 Pengujian serial dari Mikrokontroler ke Laptop

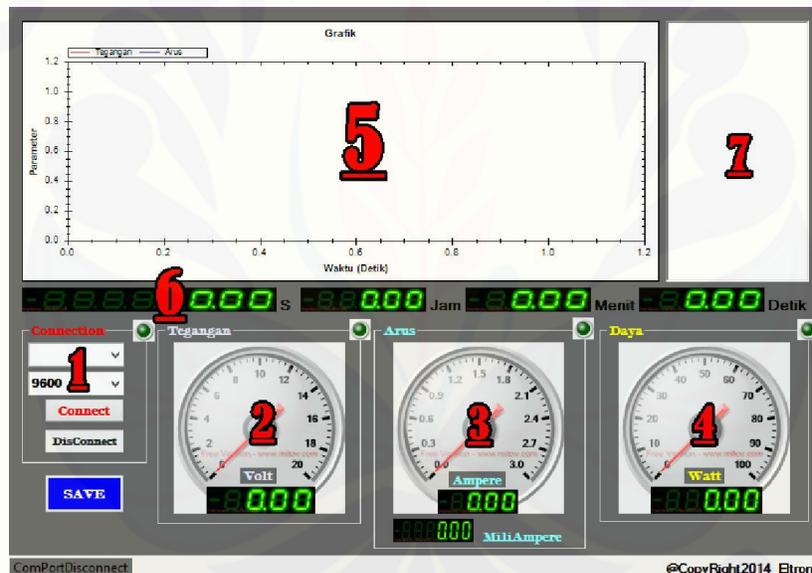
No.	Nilai Arus atau Tegangan		Keterangan
	Data pada LCD	Data pada Laptop	
1.	3	3	Benar
2.	5	5	Benar
3.	17	17	Benar
4.	48	48	Benar
5.	72	82	Benar
6.	114	114	Benar
7.	213	213	Benar

Pengujian serial adalah pengujian data yang dikirim dari mikrokontroler ke laptop dengan USB *downloader* K125R. Data yang dikirim adalah data dari sensor arus dan sensor tegangan. Kemudian dilakukan perbandingan antara data yang tampil pada LCD dan data pada laptop.

Dari data tabel 4.3 Dapat dilihat bahwa pengujian serial bisa dinyatakan sudah benar, karena pada pembacaan pada LCD dan Laptop sudah sama. Oleh karena itu pada pengujian serial sudah dapat dinyatakan sukses.

4.4.3. Pengujian GUI (*Graphic User Interface*)

Aplikasi GUI yang digunakan adalah Visual Basic 2010 Ultimate.



Gambar 4.13 Tampilan GUI sebagai *monitoring* alat

Graphic User Interface (GUI) adalah aplikasi yang digunakan untuk *monitoring* arus dan tegangan secara *real time*. Berikut adalah keterangan dan fungsi pada tampilan GUI tersebut :

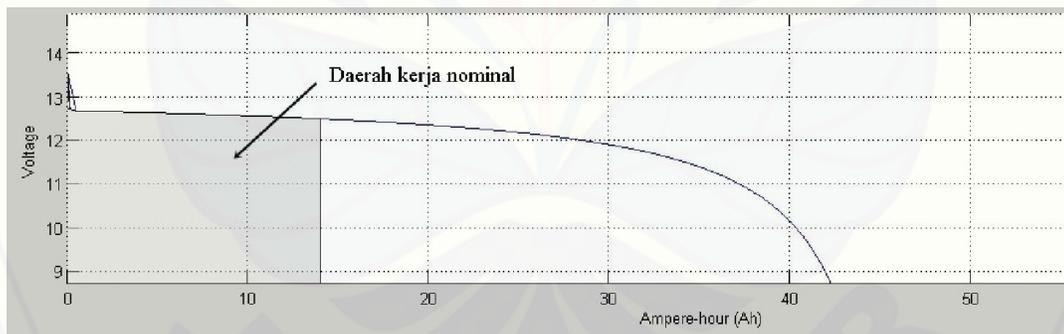
1. Tampilan pengaturan *connecting* dengan rangkaian sensor arus dan tegangan dalam bentuk serial.
2. Tampilan bentuk *analog* dan *digital* nilai tegangan

3. Tampilan bentuk *analog* dan *digital* nilai arus
4. Tampilan bentuk *analog* dan *digital* nilai kapasitas beban
5. Tampilan grafik, untuk grafik biru adalah arus dan grafik merah adalah tegangan
6. Tampilan digital waktu, meliputi sekon atau detik, menit, dan jam
7. Tampilan data nilai arus dan tegangan tiap jam, mulai detik jam 1, 2, 3.

Aplikasi GUI tersebut juga dilengkapi dengan pembacaan kapasitas daya beban, terlihat pada tampilan nomor 4. Yaitu dengan memasukkan rumus $P = V \cdot I$ pada program.

4.4.4. Pemodelan Simulasi

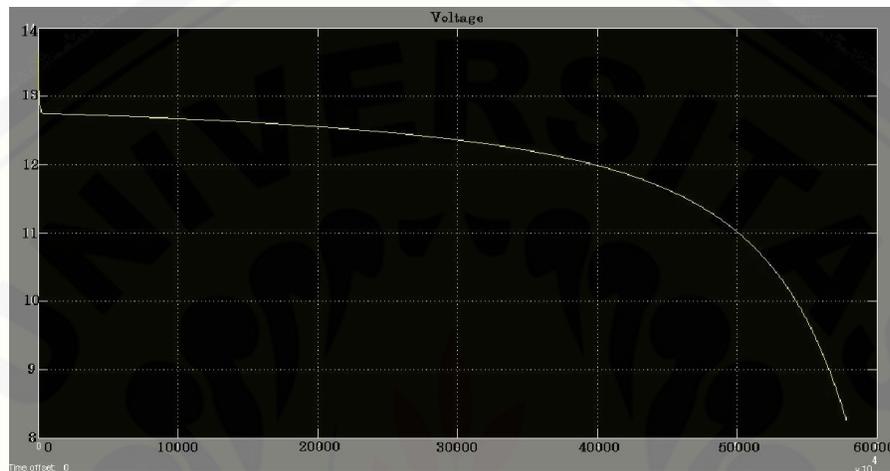
Pada pengujian matematis, digunakan perangkat lunak matlab sebagai aplikasi untuk membandingkan hasil pengujian praktek dengan bentuk simulasi. Dari data tersebut kemudian akan terlihat selisih antara bentuk *real time* dan bentuk simulasinya.



Gambar 4.14 Simulasi model karakteristik baterai timbal asam dengan Simulink/Matlab

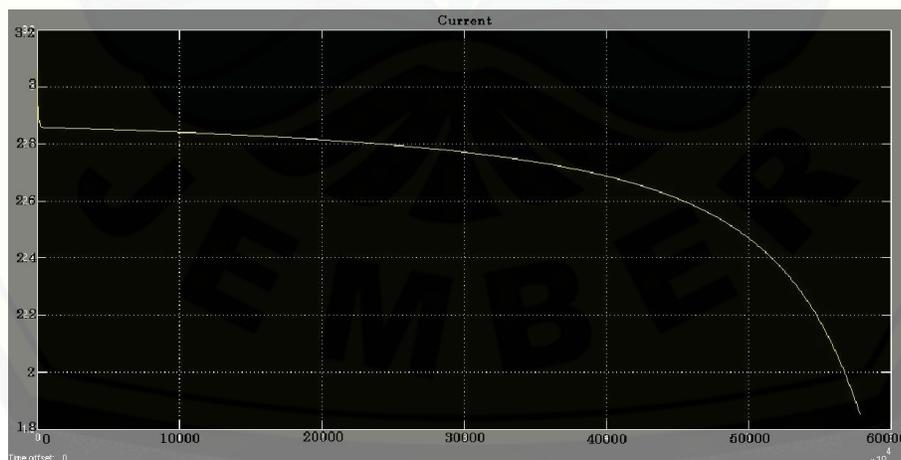
Pada gambar 4.14 adalah bentuk simulasi dari baterai timbal asam dengan aplikasi matlab. Pada gambar tersebut simulasi berbentuk grafik yang menjelaskan tentang daerah kerja nominal baterai ketika diberi beban. Dapat dilihat bahwa daerah kerja nominal baterai tersebut berkisar antara 14 AH sampai 15 AH.

Pada bentuk pengujian matematis beban yang akan digunakan adalah lampu DC 35 Watt, dalam simulasi tersebut akan diambil 2 bentuk simulasi yaitu simulasi tegangan dan arus.



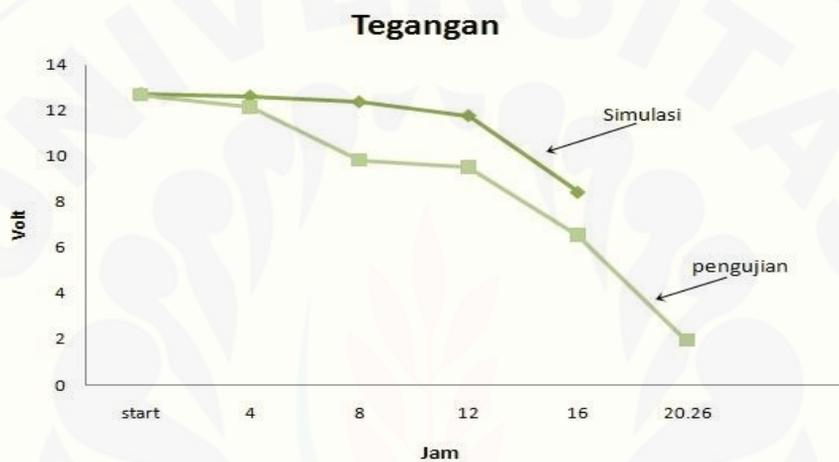
Gambar 4.15 Model simulasi tegangan dengan beban 35 Watt

Pada gambar 4.15 adalah bentuk simulasi tegangan sampai selesai, dari simulasi tersebut dapat diambil data bahwa dengan tegangan awal sekitar 12,5 V dan tegangan akhir sekitar 8,3 V. Dan pada simulasi pengujian selesai dengan waktu 57852 detik, atau sekitar 16 jam 4 menit 15,6 detik.



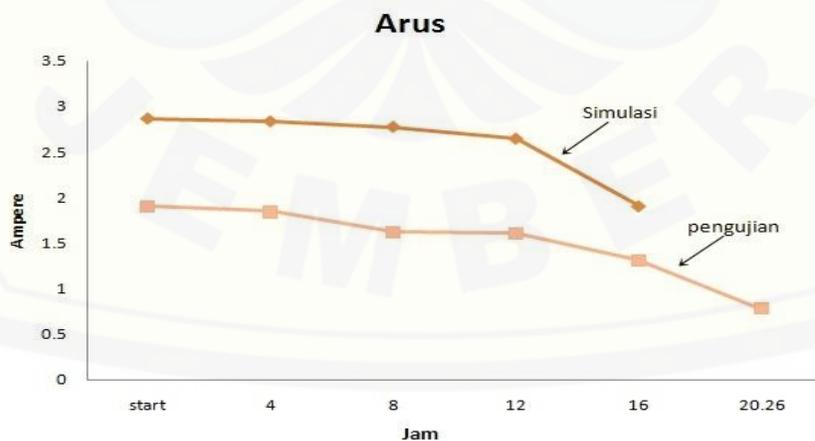
Gambar 4.16 Model simulasi arus dengan beban 35Watt

Pada gambar 4.16 adalah bentuk dari model simulasi arus, dari simulasi tersebut dapat diambil data bahwa arus awal pada beban 35 Watt adalah 2,9 A. sedangkan ketika baterai dikatakan habis yaitu arus akhirnya sekitar 1,9 A. Dan sama seperti tegangan, baterai dinyatakan habis dengan waktu 57852 detik atau 16 jam 4 menit 15,6 detik.



Gambar 4.17 Grafik perbandingan tegangan simulasi dan pengujian

Dapat dilihat pada gambar grafik 4.17 adalah perbandingan antara simulasi dan pengujian. Pada simulasi terlihat stabil, sedangkan pada pengujian masih ada drop tegangan per 4 jam.



Gambar 4.18 Grafik perbandingan arus simulasi dan pengujian

Pada gambar 4.18 dapat dilihat bahwa perbedaan nilai arus cukup signifikan, pada teori dan simulasi arus awal yang didapat pada beban adalah sekitar 2,9 A. Sedangkan pada pengujian arus awal yang didapat hanya 1,9 A. Sehingga sangat mempengaruhi pada lama waktu pengujian, ketika pada simulasi dikatakan selesai pada jam ke 16, akan tetapi pada pengujian selesainya sampai jam ke 20. Jadi arus merupakan penentu untuk lama waktu pada pengujian, sedangkan arus sendiri ditentukan oleh besarnya beban dan kondisi baterai yang bagus.

4.4.5. Pengujian Sistem Keseluruhan

Pengujian sistem keseluruhan dilakukan dengan menggunakan baterai timbal asam sebagai sumber dan lampu DC sebagai beban. Baterai yang digunakan adalah baterai dengan merk GS Astra Premium tipe 46 B 24 R (NS 60) dengan kapasitas 45 AH. Sedangkan pada beban menggunakan 3 varian, yaitu 35 Watt sebagai kategori mobil listrik *starting*, 25 Watt ketika mobil listrik keadaan *konstan*, dan 18 Watt pada saat mobil listrik pada keadaan *breaking*.



Gambar 4.19 Baterai GS Astra 45 AH

Untuk mengetahui kapasitas baterai penuh dan habis digunakan alat Test Car Battery. Untuk keadaan habis, semua tergantung berapa arus yang dikeluarkan dan berapa arus yang dibutuhkan oleh beban. Terlihat pada gambar 4.20 Adalah keadaan pada saat baterai penuh, sedangkan pada gambar 4.21 Merupakan kondisi pada saat

baterai habis yang ditentukan oleh alat Test Car Battery. Akan tetapi perlu diperhatikan pada saat baterai penuh, sebuah baterai timbal asam setiap kali di *charging* kapasitas hanya sampai 85 %.



Gambar 4.20 Kondisi baterai penuh pada alat Test Car Battery



Gambar4.21 Kondisi baterai habis pad alat Test Car Battery

4.5.1. Pengujian Beban 35 Watt (*Starting*)

Pengujian yang pertama adalah pengujian dengan beban 35 Watt, yang dinyatakan sebagai kondisi pada saat mobil listrik *starting*. Disini diperumpamakan seperti *starting* karena pada saat mobil listrik melakukan penyalaan pada mesin,

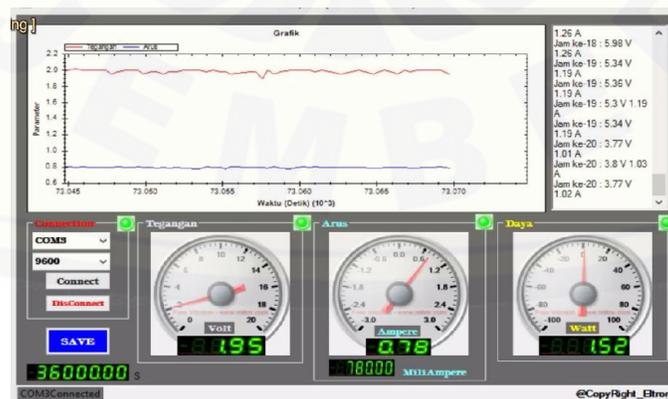
otomatis membutuhkan arus dan tegangan yang relatif besar. Oleh karena itu sebagai penggantinya digunakan lampu DC sebesar 35 Watt sebagai kondisi *starting*.

Selanjutnya pada penelitian ini dilakukan *monitoring* antara hubungan arus, tegangan, daya beban dan kapasitas baterai. Digunakan aplikasi GUI sebagai aplikasi *monitoring* yang diperjelas dalam bentuk grafik.



Gambar 4.22 Tampilan GUI *starting* saat pengujian dimulai

Dapat dilihat pada saat mulai atau *start* dengan kapasitas baterai penuh, didapat nilai pada tegangan 12,68 V, arus 1,9 A, dan daya beban 24,9 Watt. Disini daya beban sangat berbeda jauh dengan daya beban yang sebenarnya yaitu 35 Watt. Mungkin disini dikarenakan dari faktor baterai yang kurang bagus, oleh karena itu berdampak pada nilai yang didapat pada arus dan kemudian mempengaruhi pada nilai kapasitas beban..



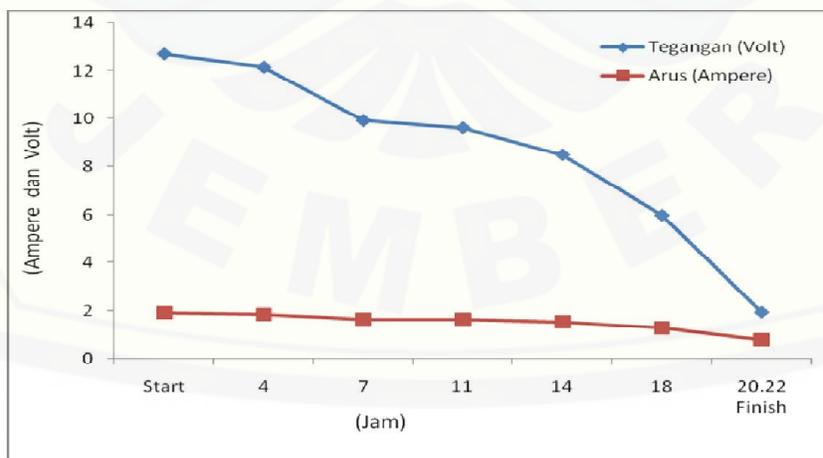
Gambar 4.23 Tampilan GUI *starting* saat pengujian selesai

Pada gambar 4.23 adalah pada saat kapasitas baterai bisa dikatakan habis, karena pada pengujian lampu sebagai beban sudah padam. Didapat hasil pengujian selesai pada saat tegangan tersisa 1,95 V dan arusnya 0,78 A. Sehingga didapat daya lampu yang menyala hanya 1,52 Watt.

Tabel 4.4 Data pengujian *starting* beban 35 Watt

No.	Beban (Watt)	Waktu (jam)	Tegangan (Volt)	Arus (Ampere)	Daya Beban (Watt)
1.	35	Start	12,68	1,9	24,09
2.		4	12,14	1,84	22,34
3.		7	9,91	1,62	16,04
4.		11	9,61	1,62	15,57
5.		14	8,5	1,51	12,84
6.		18	5,98	1,26	7,34
7.		Finish	1,95	0,78	1,52

Data dari tabel 4.4 hanya sebagian dari data pengujian. Didapat hasil ketika pengujian dimulai tegangan baterai ketika penuh adalah 12,68 V dan ketika tidak mampu menyalakan beban 1,95 V. Untuk arus sendiri ketika pengujian dimulai 1,9 A dan ketika habis 0,78 A. Pengujian selesai pada jam ke 20 lebih 13 menit.

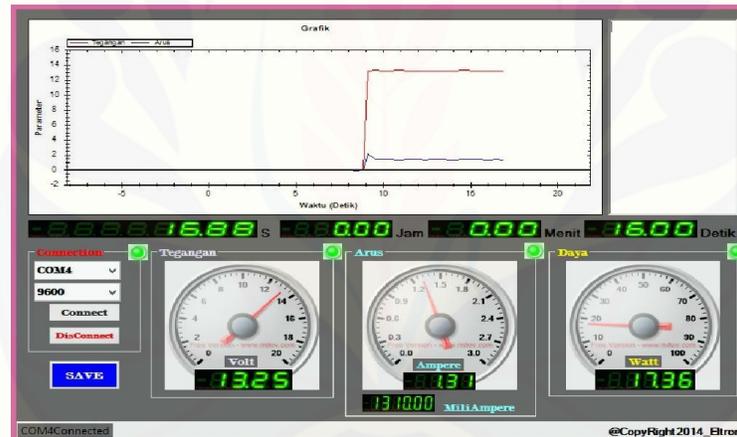


Gambar 4.24 Grafik arus dan tegangan pada kondisi *starting*

Dari gambar grafik 4.24 adalah bentuk jalannya pengujian *starting* dengan pengambilan dari beberapa data. Dari keseluruhan data pengujian *starting* dapat diambil *error* waktu dari pengujian dengan waktu target adalah 23 jam 40 menit. Kemudian hasil pengambilan data selesai pada waktu berjalan 20 jam 16 menit, sehingga didapat *error* waktunya 3 jam 24 menit atau 3,4 %.

4.5.2. Pengujian Beban 25 Watt (*Konstan*)

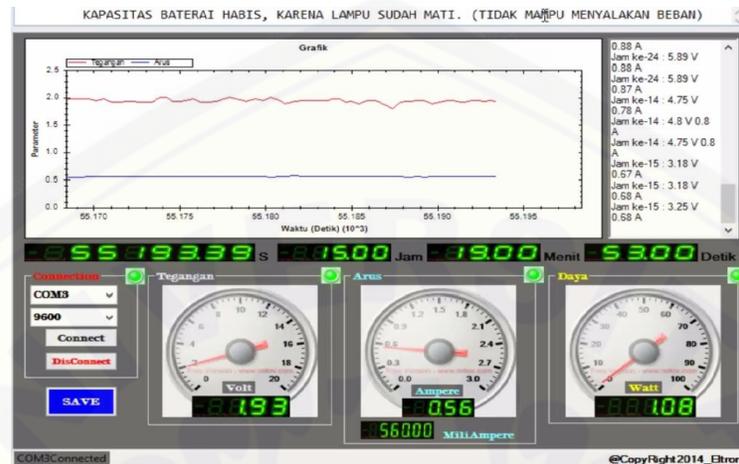
Pengujian kedua adalah pengujian dengan beban 25 Watt. Pengujian kali ini adalah beban dengan variasi sedang. Beban tersebut diperumpamakan sebagai *konstan*, maksudnya adalah ketika saat mobil listrik berjalan dengan kecepatan stabil dengan beberapa lama.



Gambar 4.25 Tampilan GUI *konstan* saat pengujian dimulai

Dapat dilihat pada gambar 4.25 ketika kapasitas baterai penuh tegangan awalnya adalah 13,25 V, berbeda sekitar 1 V dengan pengujian pada beban *starting*. Untuk arusnya sendiri didapat nilai 1,31 A. Dan kapasitas sebesar 17,36 Watt, selisih 7,64 Watt dari kapasitas tertulisnya.

Untuk gambar 4.26 menjelaskan pengujian *konstan* ketika selesai, data yang didapat ketika baterai sudah habis. Untuk arus yang tersisa adalah 0,56 A dan tegangannya 1,93 V, dan didapat kapasitas beban terakhir 1,08 Watt. Lama waktu habis baterai sendiri adalah sekitar 26 jam 20 menit

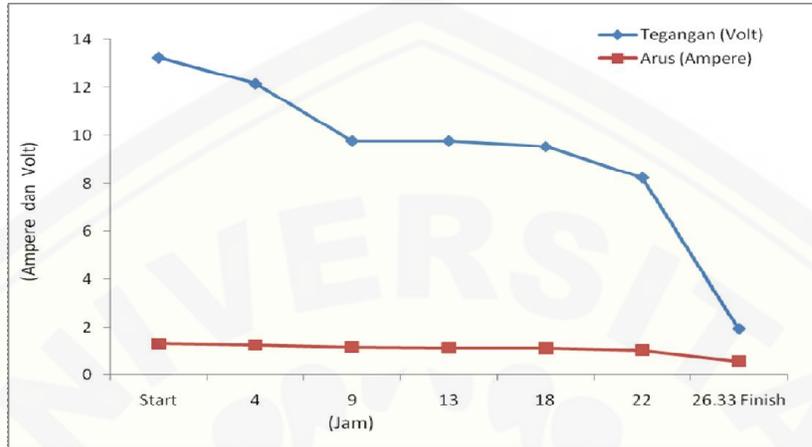


Gambar 4.26 Tampilan GUI *konstan* saat pengujian selesai

Tabel 4.5 Data pengujian *konstan* beban 25 Watt

No.	Beban (Watt)	Waktu (jam)	Tegangan (Volt)	Arus (Ampere)	Daya Beban (Watt)
1.	25	Start	13,25	1,31	17,36
2.		4	12,18	1,26	15,34
3.		9	9,77	1,16	11,33
4.		13	9,77	1,13	11,04
5.		18	9,52	1,11	10,57
6.		22	8,25	1,03	8,5
7.		Finish	1,93	0,56	1,08

Dari tabel data 4.5 bahwa pada pengujian *konstan* memiliki tegangan awal yang cukup besar, yaitu 13,25 V. Dari data tabel diatas kemudian dapat dibuat bentuk perbandingan grafik arus dan grafik tegangan terhadap waktu, untuk lebih mengetahui bentuk jelas dari menurunnya arus dan tegangan itu sendiri sampai baterai dinyatakan habis. Grafik tersebut akan di tampilkan pada gambar 4.22 di halaman berikutnya.

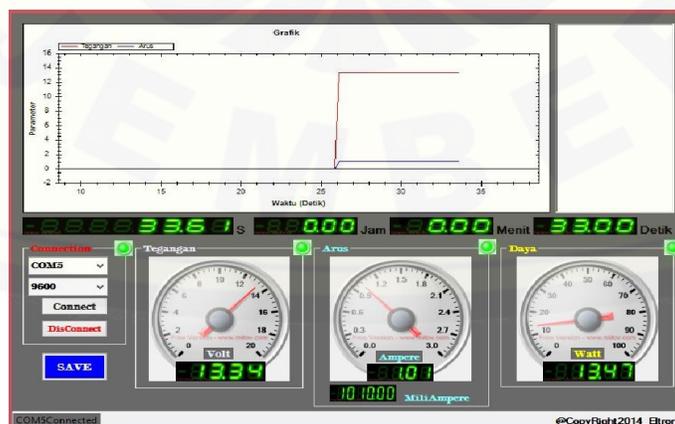


Gambar 4.27 Grafik arus dan tegangan pada kondisi konstan

Pada pengujian konstan didapat waktu targetnya adalah 34 jam 21 menit., kemudian untuk waktu pengujian selesai adalah 26 jam 19 menit. Dari data tersebut didapat *error* waktunya adalah 8 jam 2 menit atau 8,03 %.

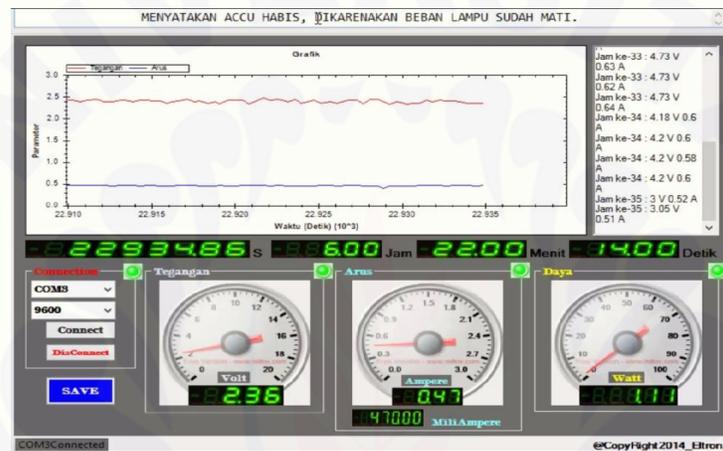
4.5.3. Pengujian Beban 18 Watt (*Breaking*)

Pengambilan beban ketiga adalah beban variasi kecil dengan menggunakan kapaitas 18 Watt. Kemudian dalam penelitian ini dinamakan sebagai beban *breaking* atau pengereman pada mobil listrik. Karena pada umumnya ketika terjadi pengereman pada suatu kendaraan bermotor, arus dan energi maupun sumber tenaga yang dibutuhkan penggerak adalah relatif sangat kecil dibandingkan dengan kondisi apapun saat motor bekerja.



Gambar 4.28 Tampilan GUI *breaking* saat pengujian dimulai

Dapat diambil data ketika pengujian *breaking* dimulai dengan kapasitas baterai penuh menghasilkan tegangan awal 13,34 V dan arus yang dihasilkan 1,01 A dan didapat kapasitas daya beban 13,47 Watt. Kemudian selisih daya bebannya adalah 4,53 Watt.



Gambar 4.29 Tampilan GUI *breaking* pada saat pengujian selesai

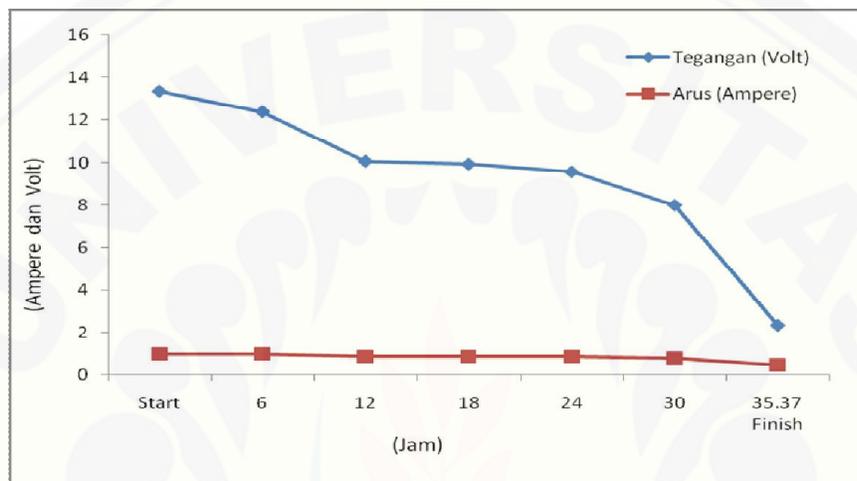
Pada tampilan GUI diatas adalah ketika kondisi baterai habis, karena beban lampu sudah tidak dapat menyala. Pada kondisi tersebut didapat data tegangan tersisa 2,36 V dan arusnya 0,47 A sehingga didapat besar kapasitas beban tersisa adalah 1,11 Watt.

Tabel 4.6 Data pengujian *breaking* beban 18Watt

No.	Beban (Watt)	Waktu (Jam)	Tegangan (Volt)	Arus (Ampere)	Daya Beban (Watt)
1.	18	Start	13,34	1,01	13,47
2.		6	12,39	0,99	12,27
3.		12	10,09	0,89	8,98
4.		18	9,95	0,88	8,76
5.		24	9,59	0,87	8,34
6.		30	8	0,79	6,32

7.		Finish	2,36	0,47	1,11
----	--	--------	------	------	------

Dari tabel 4.6 kemudian dapat dibuat bentuk grafik antara arus dan tegangan terhadap waktu ketika pengujian, dapat dilihat pada gambar 4.30.



Gambar 4.30 Grafik arus dan tegangan pada kondisi *breaking*

Untuk pengujian *breaking* didapat waktu target sebesar 44 jam 33 menit, dan pada saat pengujian hanya 35 jam 22 menit. Sehingga didapat *error* waktunya adalah 9 jam 11 menit atau 9,18 %.

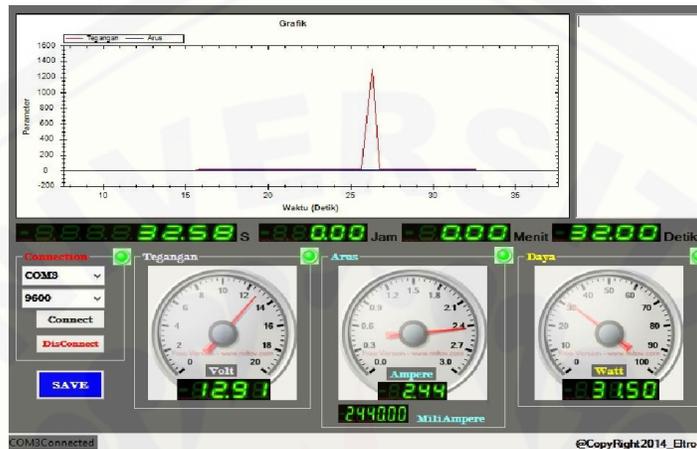
4.5.4. Pengujian Beban Paralel (*Couple*)

Pada pengujian *couple* ini digunakan untuk perbandingan *error* waktu pada pengujian sistem secara keseluruhan. Beban *couple* tersebut juga menggunakan lampu DC, paralel antara kapasitas 18 Watt, 25 Watt, dan 35 Watt. Sehingga didapat 4 variasi beban lagi yaitu 43 Watt, 53 Watt, 60 Watt dan 78 Watt.

Data yang akan diambil pada pengujian *couple* meliputi tegangan, arus, dan kapasitas beban pada saat mulai maupun selesai. Serta akan dihitung *error* waktu dengan mengurangi waktu target dan waktu pengujian seperti pada pengujian sebelumnya dan bisa dikatakan ini adalah pengujian tambahan untuk menentukan rata-rata *error* dari keseluruhan pengujian.

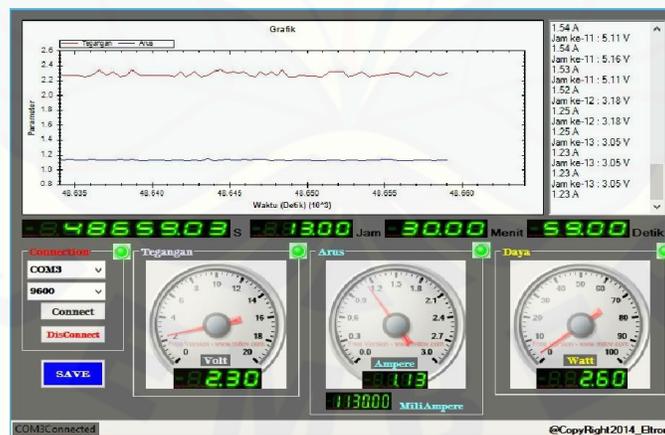
4.4.4.1 Pengujian Beban *Couple* 43 Watt

Pengujian ini menggunakan beban lampu DC 18 Watt diparalel dengan lampu 25 Watt.



Tabel 4.31 Tampilan GUI *couple* 43Watt pada saat mulai

Pada saat pengujian dimulai dengan kondisi baterai penuh didapat besar tegangan 12,91 V dan arus 2,44 A sehingga nilai kapasitas bebannya 31,5 Watt. Disini dapat diambil selisih antara kapasitas tertulis dengan pengujian sebesar 12,5 Watt.



Tabel 4.32 Tampilan GUI *couple* 43 Watt pada saat selesai

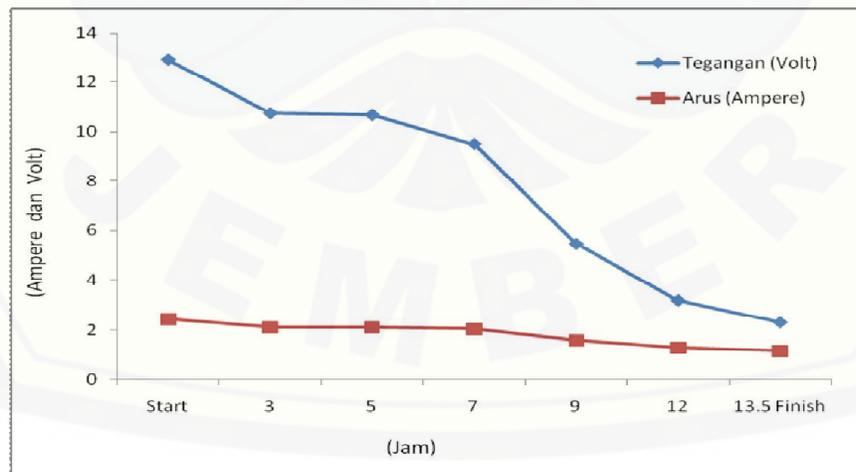
Pada saat pengujian beban *couple* 43 Watt selesai, didapat data tegangan akhir sebesar 2,3 V dan untuk arus sendiri 1,13 A dengan kapasitas 2,6 Watt. Untuk

pengujian kali ini dianggap selesai dengan membutuhkan waktu selama 13 jam 30 menit.

Tabel 4.7 Data pengujian beban *couple* 43Watt

No.	Beban (Watt)	Waktu (jam)	Tegangan (Volt)	Arus (Ampere)	Daya Beban (Watt)
1.	43	Start	12,91	2,44	31,5
2.		3	10,75	2,11	22,68
3.		5	10,68	2,1	22,42
4.		7	9,52	2,03	19,32
5.		9	5,45	1,56	8,5
6.		12	3,18	1,25	3,98
7.	43	Finish	2,3	1,13	2,6

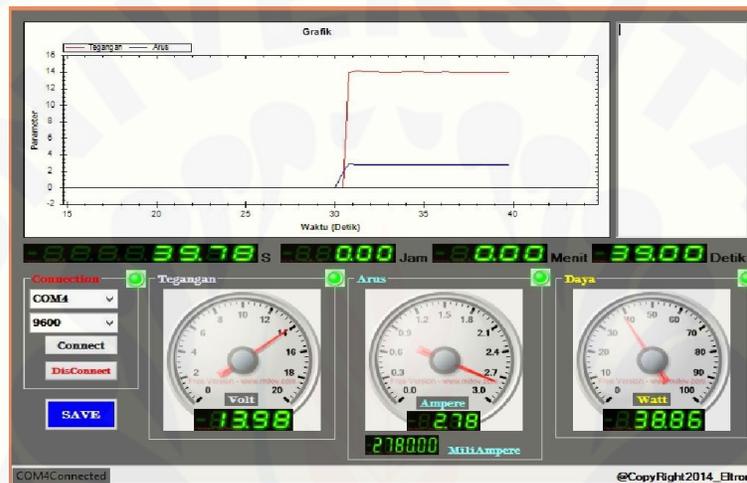
Dari data tabel 4.7 dapat dibuat grafik hubungan antara arus dan tegangan terhadap waktu. Seperti pada gambar 4.28 adalah bentuk alur grafik saat keadaan baru dimulai dan sampai selesai.



Gambar 4.33 Grafik arus dan tegangan pada kondisi *couple* 43 Watt

Dari seluruh data pada pengujian *couple* 43 Watt dengan data arus awal 2,44 A didapat target waktu habis baterai adalah 18,44 atau 18 jam 26 menit. Kemudian pada saat pengujian selesai membutuhkan waktu selama 13 jam 30 menit. Dari data waktu tersebut dapat diambil *error* waktunya sebesar 4 jam 56 menit atau 4,93 %.

4.4.4.2 Pengujian Beban *Couple* 53 Watt



Tabel 4.34 Tampilan GUI *couple* 53 Watt pada saat mulai

Untuk pengujian *couple* 53Watt didapat data tegangan awal sebesar 13,98 V dan untuk arusnya 2,78 A sehingga didapat nilai kapasitas beban 38,86 Watt. Mempunyai selisih 14,14 Watt dari beban tertulis.



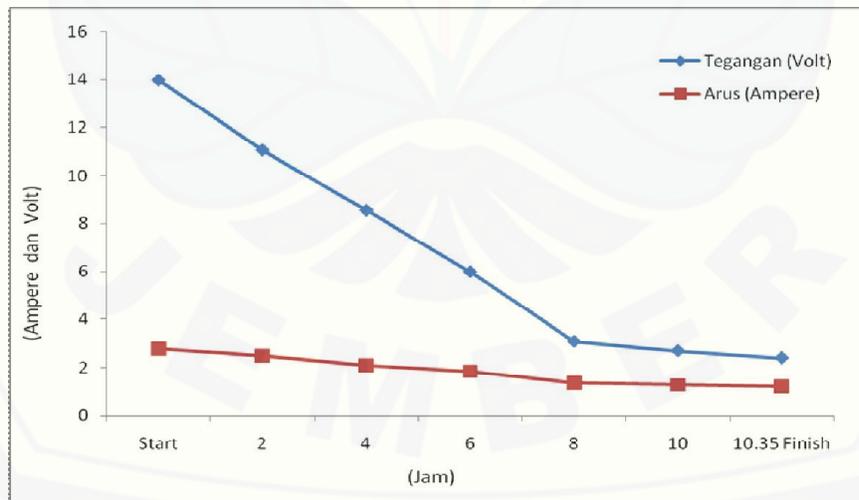
Tabel 4.35 Tampilan GUI *couple* 53 Watt pada saat selesai

Pada saat pengujian selesai dapat diambil data tegangan akhir sebesar 2,39 V dan arusnya 1,22 A dengan kapasitas beban 2,9 2Watt. Dari pengujian kali ini membutuhkan waktu 10 jam 21 menit untuk menghabiskan energi baterai.

Tabel 4.8 Data pengujian beban *couple* 53Watt

No.	Beban (Watt)	Waktu (jam)	Tegangan (Volt)	Arus (Ampere)	Daya Beban (Watt)
1.	53	Start	13,98	2,78	38,86
2.		2	11,07	2,48	27,45
3.		4	8,55	2,1	17,96
4.		6	5,98	1,85	11,06
5.		8	3,07	1,38	4,24
6.		10	2,68	1,28	3,43
7.	53	Finish	2,39	1,22	2,92

Dari data tabel 4.8 kemudian dapat dibuat grafik hubungan antara arus dan tegangan terhadap waktu. Seperti pada gambar 4.31.

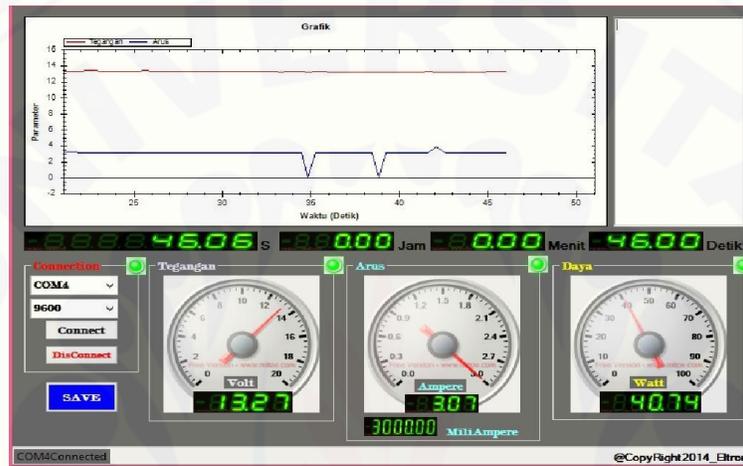


Gambar 4.36 Grafik arus dan tegangan pada kondisi *couple* 53 Watt

Dari data keseluruhan pengujian *couple* 53 Watt diambil besarnya arus awal sebesar 2,78 A akan didapat waktu target habisnya kapasitas baterai selama 16 jam 10

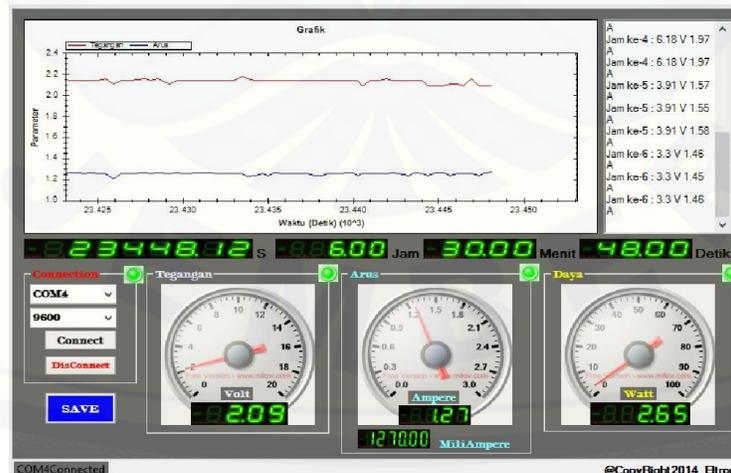
menit. Pada pengujian kali ini dibutuhkan waktu selama 10 jam 21 menit, sehingga didapat *error* waktunya sebesar 5 jam 49 menit atau 5,82 %.

4.4.4.3 Pengujian Beban *Couple* 60 Watt



Tabel 4.37 Tampilan GUI *couple* 60 Watt pada saat mulai

Dari pengujian *couple* 60 Watt mulai didapat tegangan awal sebesar 13,27 V dan arusnya 3,07 A, sehingga nilai kapasitas bebannya adalah 40,74 Watt, dengan data tersebut kapasitas beban memiliki selisih 19,26 Watt dari nilai aslinya 60 Watt.



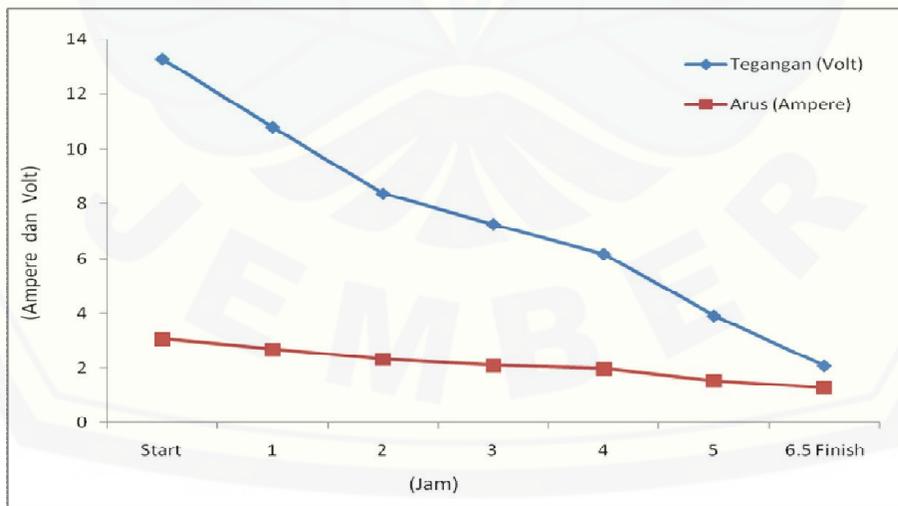
Tabel 4.38 Tampilan GUI *couple* 60 Watt pada saat selesai

Ketika kondisi baterai dinyatakan habis, dari tampilan GUI tersebut didapat data tegangan akhir sebesar 2,09 V dan arusnya 1,27 A serta kapasitas beban yang tersisa sebesar 2,65 Watt. Untuk lama waktu pengujiannya sendiri adalah 6 jam 30 menit.

Tabel 4.9 Data pengujian beban *couple* 60 Watt

No.	Beban (Watt)	Waktu (jam)	Tegangan (Volt)	Arus (Ampere)	Daya Beban (Watt)
1.	60	Start	13,27	3,07	40,74
2.		1	10,8	2,69	29,05
3.		2	8,36	2,33	19,48
4.		3	7,25	2,1	15,23
5.		4	6,18	1,97	12,17
6.		5	3,91	1,52	5,94
7.	60	Finish	2,09	1,27	2,65

Dari data tabel 4.9 kemudian didapat grafik hubungan arus dan tegangan terhadap waktu pada gambar 4.34.



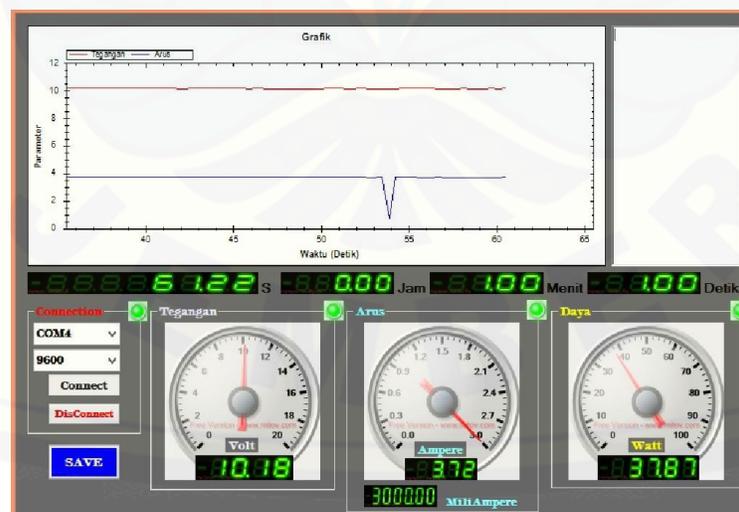
Gambar 4.39 Grafik arus dan tegangan pada kondisi *couple* 60 Watt

Dari keseluruhan data pengujian, dapat dicari waktu target baterai habis yaitu dengan membagi kapasitas baterai penuh dengan nilai arus awal. Dan didapat nilai waktu target sebesar 14,65 atau 14 jam 39 menit. Untuk waktu pengujian sendiri membutuhkan waktu selama 6 jam 30 menit, sehingga dapat diambil *error* waktunya adalah 8 jam 9 menit atau 8,15 %.

4.4.4.4 Pengujian Beban *Couple* 78 Watt

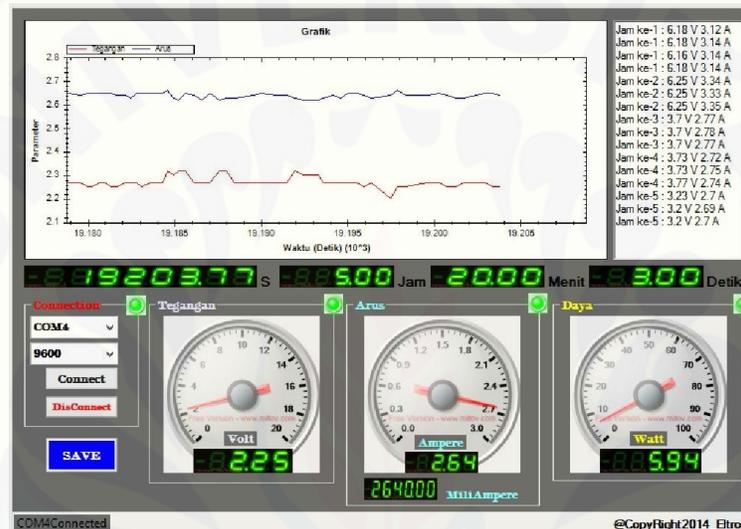
Untuk pengujian beban *couple* 78 Watt kali ini ada masalah yang terjadi, yaitu pada baterai timbal asam. Masalah yang terjadi adalah pada tegangan yang keluar pada baterai. Apabila dilihat dari pengujian-pengujian sebelumnya, nilai rata-rata tegangan yang terbaca oleh sensor sekitar 12 V sampai 13 V lebih. Akan tetapi pada pengujian yang terakhir ini tegangan yang keluar dan terbaca hanya 10,18 V pada saat pengujian baru dimulai.

Kembali pada pernyataan awal pada halaman , yaitu baterai timbal asam yang telah di *charging* hanya akan terisi 85 % dari kapasitas sebenarnya. Selain dari pernyataan tersebut, kita bisa menyatakan bahwa baterai sudah mulai rusak. Alasan dinyatakan rusak bisa didukung dengan data tegangan *start* yang telah diambil pada pengujian-pengujian sebelumnya.



Tabel 4.40 Tampilan GUI *couple* 78 Watt pada saat mulai

Pada gambar 4.40 dapat kita ambil bahwa nilai tegangan saat pengujian dimulai sebesar 10,18 V, dan untuk nilai arus 3,72 A. Sehingga didapat nilai kapasitas dari beban 37,87 Watt, menghasilkan selisih yang cukup besar yaitu 40,13 Watt dari kapasitas tertulis. Bisa dikatakan bahwa selisih ini adalah faktor dari mulai rusaknya baterai.



Tabel 4.41 Tampilan GUI *couple* 78 Watt pada saat selesai

Pada gambar 4.41 adalah saat pengujian *couple* 78 Watt dinyatakan selesai, dapat diambil data akhir dari tegangan sebesar 2,25 V dan untuk arusnya 2,64 A sehingga didapat kapasitas beban akhir 5,94 Watt. Untuk pengujian kali ini membutuhkan waktu selama 5 jam 20 menit.

Pada tabel 4.10 adalah data keseluruhan pengujian dari mulai sampai selesai. Dapat kita lihat bahwa data tegangan awal adalah sebagai acuan untuk mengetahui masalah yang terjadi pada pengujian beban *couple* 78 Watt. Oleh karena itu disini sangat sulit menentukan waktu target yang sebenarnya. Berdasarkan pengujian dan data tabel arus awal 37,87 A waktu target yang diperoleh adalah 12 jam 6 menit. Akan tetapi apabila baterai dalam keadaan *normal* mungkin waktu target akan lebih singkat, dikarenakan arus yang dihasilkan juga akan lebih besar. Kembali untuk

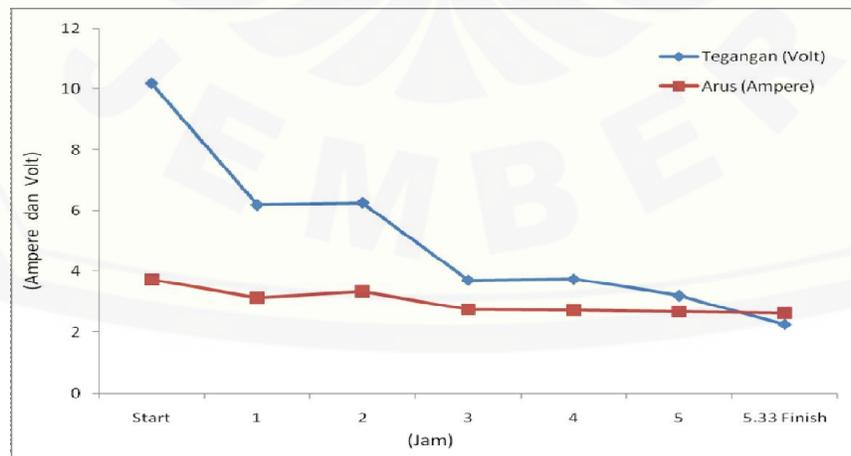
membahas masalah pada baterai, bisa kita lihat pada gambar 4.41. Terlihat grafik arus dan tegangan tidak stabil, tidak seperti pada pengujian-pengujian sebelumnya.

Setelah melihat grafik pada tampilan GUI tersebut, kemudian dapat kita bandingkan dengan data tabel 4.10 yang berisi tentang data keseluruhan pengujian beban 78 Watt dan lama waktu pengujian selesai.

Tabel 4.10 Data pengujian beban *couple* 78 Watt

No.	Beban (Watt)	Waktu (jam)	Tegangan (Volt)	Arus (Ampere)	Daya Beban (Watt)
1.	78	Start	10,18	3,72	37,87
2.		1	6,18	3,14	19,41
3.		2	6,25	3,35	20,94
4.		3	3,7	2,77	10,25
5.		4	3,73	2,74	10,22
6.		5	3,2	2,7	8,64
7.		Finish	2,25	2,64	5,94

Untuk lebih memperjelas data dari tabel 4.10, bisa kita lihat pada gambar grafik 4.42 yang memperlihatkan alur pengujian beban *couple* 78 Watt pada saat mulai sampai selesai..



Gambar 4.42 Grafik arus dan tegangan pada kondisi *couple* 78 Watt

Dapat dilihat pada tabel 4.10 dan gambar grafik 4.42, yaitu nilai data tegangan pada jam ke 1 sampai 4 tidak stabil. Bahkan nilai data tegangan jam ke 2 lebih besar daripada pada jam ke 1, begitu juga yang terjadi pada jam ke 3 terhadap jam ke 4. Dan dari grafik itulah alasan yang cukup menjelaskan untuk kerusakan yang terjadi pada baterai timbal asam.

Dan sesuai ketentuan pengujian, tetap akan dihitung nilai *error* waktu pada pengujian beban *couple* 78 Watt sesuai data yang didapat. Untuk waktu target adalah 12 jam 6 menit, sedangkan untuk waktu pengujian selesai selama 5 jam 20 menit. Sehingga didapat *error* waktunya adalah 5 jam 46 menit.

4.5.5. Perbandingan Pengujian Beban Keseluruhan

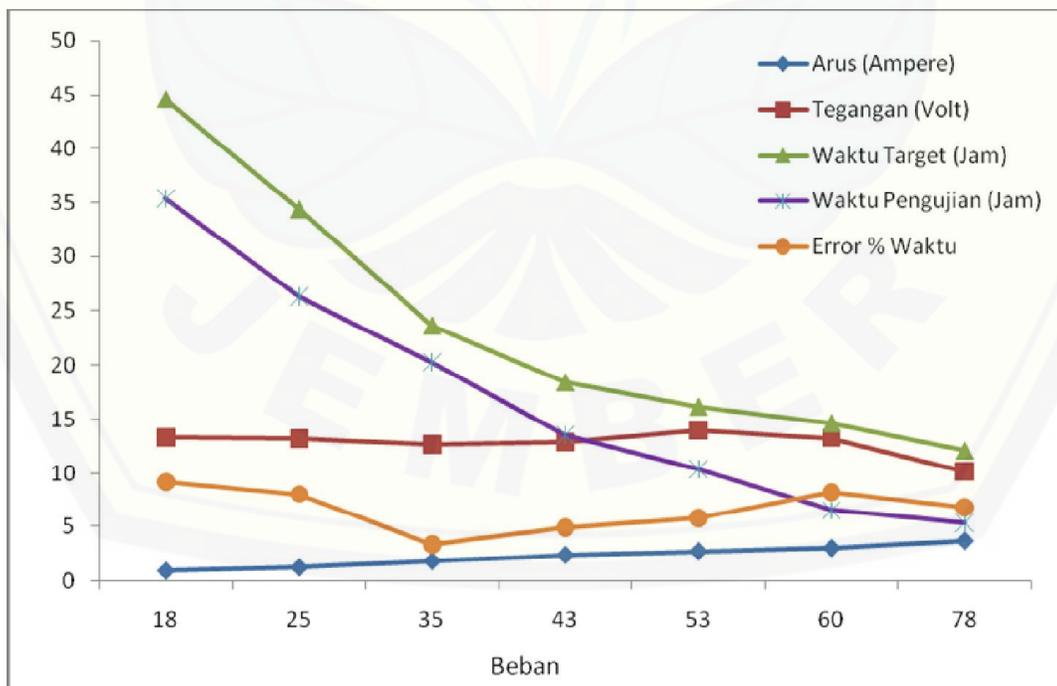
Berdasarkan pengujian sistem secara keseluruhan didapat data seperti pada tabel 4.11, dalam tabel tersebut juga menyertakan dari *error* waktu. Selain itu pada tabel tersebut juga menampilkan kondisi awal tegangan dan arus terhadap semua beban yang digunakan. Sehingga akan didapat waktu target, kemudian bisa dihitung *error* waktu yang dicari. Yang kemudian juga akan ditampilkan dalam bentuk grafik, supaya bisa terlihat jelas alur kerja pada pengujian ini.

Tabel 4.11 Hasil data pengujian secara keseluruhan

No.	Beban (Watt)	Kondisi Awal		Waktu Target (Jam)	Waktu Pengujian (Jam)	<i>Error</i> % Waktu
		Tegangan (Volt)	Arus (Ampere)			
1.	18	13,34	1,01	44,55	35,37	9,18
2.	25	13,25	1,31	34,35	26,32	8,03
3.	35	12,68	1,9	23,67	20,27	3,4
4.	43	12,91	2,44	18,44	13,5	4,93
5.	53	13,98	2,78	16,17	10,35	5,82
6.	60	13,27	3,07	14,65	6,5	8,15
7.	78	10,18	3,72	12,1	5,33	6,77

Dari tabel tersebut dapat dilihat bahwa *error* waktu terbesar adalah pada saat pengujian beban *breaking* menggunakan beban lampu DC 18 Watt dengan selisih waktunya adalah 9,18 atau sekitar 9 jam 11 menit. Untuk *error* waktu terkecil adalah pada beban *starting* dengan 35 Watt yaitu hanya 3,4 atau sekitar 3 jam 24 menit.

Dari data tersebut dapat disimpulkan bahwa baterai adalah faktor terbesar yang menyebabkan terjadinya *error* pada pengujian. Karena baterai merupakan sumber energi inti pada pengujian, oleh sebab itu mempunyai peranan penting terhadap *error* yang didapat. Terutama pada saat pengujian beban terakhir, yaitu pengujian beban *couple* 78 Watt. Yang pada saat itu hanya mampu mengeluarkan tegangan 10,18 V. Faktor dari baterai sendiri bisa disebabkan karena sel-sel yang terkandung didalam larutan baterai timbal asam sudah tidak mampu menyimpan energi 85 % ketika dilakukan *charging*. Atau mungkin dari pemakaian baterai yang terlalu sering digunakan, menyebabkan daya tahan larutan yang terkandung dalam baterai tidak dapat bekerja maksimal. Faktor lain mungkin disebabkan juga oleh umur baterai.



Gambar 4.43 Grafik *monitoring* sistem pengujian secara keseluruhan

Pada gambar grafik 4.43 adalah bentuk dari jalannya pengujian secara keseluruhan, mulai dari pengujian beban terendah sampai dengan tertinggi. Kemudian dari perhitungan waktu target hingga waktu pengujian selesai, yang kemudian didapatkan *error %* terhadap waktu.

Dari keseluruhan *error %* waktu dapat dijumlah dan diambil rata-rata *error %* tersebut. Yaitu dari data jumlah *error %* keseluruhan adalah 46,28 dibagi dengan jumlah pengujian yaitu 7. Sehingga didapat rata-rata *error %* adalah 6,61 %. Dari keseluruhan pengujian sangat perlu diperhatikan adalah kondisi baterai timbal asam, karena baterai adalah faktor terbesar yang menyebabkan terjadinya keseluruhan *error* pada penelitian.

BAB 5. PENUTUP

5.1 Kesimpulan

Dari hasil pembuatan, pengujian perangkat dan pembahasan dapat diambil kesimpulan sebagai berikut :

1. Pembacaan sensor arus terhadap *output* dari baterai dinyatakan berhasil, karena dari keseluruhan pengujian terhadap data arus *error %* tidak lebih dari 10 %.
2. Bentuk tampilan grafik di komputer untuk *monitoring output* baterai mobil listrik telah berhasil ditampilkan dengan *software* visual studio 10 ultimate, aplikasi tersebut disebut *graphic user interface* yang menampilkan data arus, tegangan dan kapasitas beban.
3. Dari 3 kondisi beban *starting*, *konstan*, *breaking*, arus yang paling besar dihasilkan adalah pada kondisi *starting* dengan nilai 1,9 A. Sedangkan arus terkecil yang dihasilkan adalah pada kondisi *breaking*, dikarenakan besar nilai arus berbanding lurus dengan besar kapasitas nilai beban.

5.2 Saran

Berdasarkan penelitian yang telah dilakukan tentang “*Sistem Monitoring Arus Dan Tegangan Pada Baterai Mobil Listrik Berbasis Mikrokontroler ATmega 16*” penulis memberikan saran berikut dengan harapan untuk penyempurnaan karya ilmiah ini dan lebih memberikan manfaat yang lebih baik di masa yang akan datang :

1. Perlu dilakukan penelitian lebih lanjut dengan menggunakan metode kecerdasan buatan, supaya bisa didapat hasil yang lebih akurat untuk perbandingan dengan pengujian alat.
2. Perlu dilakukan penelitian lebih lanjut menggunakan beban yang lebih bervariasi lagi, dan bisa didapat arus yang lebih besar. Sehingga dapat dilakukan untuk perbandingan penelitian-penelitian sebelum dan selanjutnya.

3. Sangat perlu diperhatikan pada pemilihan baterai yang digunakan pada pengujian, karena baterai disini merupakan komponen penting untuk menentukan keberhasilan dari penelitian.



DAFTAR PUSTAKA

- Agfianto Eko Putra, 2002, *Teknik Antarmuka Komputer: Konsep dan Aplikasi*, Graha Ilmu, Yogyakarta.
- B.G. Pollet, I. Staffell, J.L. Shang, 2012, *Current Status of Hybrid, Battery and Fuel Cell Electric Vehicles*, Electrochemistry to market prospects, Electrochim.
- Bambang Sri Kaloko, 2009, *Lead Acid Battery Modeling for Electric Car Power Sources*, Indonesian Journal of Chemistry.
- Bambang Sri Kaloko, Soebagio, M.H. Purnomo, 2011, *Mapping of Electrochemistry and Neural Network Model for Lead Acid Battery*, Indonesian Journal of Chemistry.
- Dwi Sutadi, 2002, *I/O Bus & Motherboard*, Andi Yogyakarta, Yogyakarta.
- Heri Andrianto, 2013. *Pemrograman Mikrokontroler AVR ATmega16 Menggunakan Bahasa C (CodeVisionAVR)*, Edisi Revisi, Maret 2013, Informatika Bandung, Bandung.
- Hery Kuswanto, 2010. *AC Electrical Measuring Instrument (Current, Voltage, Power) Using Parallel Port*. Final Project. D3 Studies Program Computer Science Faculty of Mathematics and Natural Science University of Sebelas Maret Surakarta.
- Kiehne, H. A., 2003, *Battery Technology Handbook*, Second Edition, Marcel Dekker, NY.
- O'Brien, James A., 2003, *Introduction to Information Systems : Essential for the E-bussiness Enterprise*, 2003, McGraw Hill, Boston.
- Park, S., Savvides, A., Srivastava, M. B., 2001, *Battery Capacity Measurement And Analysis Using Lithium Coin Cell Battery*, ISLPED, August 6-7, 2001, Huntington Beach, California, USA.
- Thomas, B. G., 2000, *Lead-Acid Batteries: Key to Electric Vehicle Commercialization*, The Fifteenth Annual Battery Conference January 13, 2000, Presentation to Long Beach.

LAMPIRAN

A. Program Kalibrasi Sensor Arus Menggunakan CodeVision AVR

This program was produced by the
CodeWizardAVR V2.05.3 Standard
Automatic Program Generator

© Copyright 1998-2011 Pavel Haiduc, HP InfoTech s.r.l.

<http://www.hpinfotech.com>

Project :

Version :

Date : 29/10/2014

Author : ridik

Company : the explorer

Comments:

Chip type : ATmega16

Program type : Application

AVR Core Clock frequency: 12,000000 MHz

Memory model : Small

External RAM size : 0

Data Stack size : 256

*****/

```
#include <mega16.h>
```

```
#include <delay.h>
```

```
#include <stdlib.h>
```

```
int c,d;
float a,b, DataArus,tegangan;
double arus = 0;
int DataTegangan;
int i;

char ridik[30];

// Alphanumeric LCD functions
#include <alcd.h>

// Standard Input/Output functions
#include <stdio.h>

#define ADC_VREF_TYPE 0x00

// Read the AD conversion result
unsigned int read_adc(unsigned char adc_input)
{
    ADMUX=adc_input | (ADC_VREF_TYPE & 0xff);
    // Delay needed for the stabilization of the ADC input voltage
    delay_us(10);
    // Start the AD conversion
    ADCSRA|=0x40;
    // Wait for the AD conversion to complete
    while ((ADCSRA & 0x10)==0);
    ADCSRA|=0x10;
```

```
return ADCW;
}

// Declare your global variables here

void main(void)
{
// Declare your local variables here

// Input/Output Ports initialization
// Port A initialization
// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In Func0=In
// State7=T State6=T State5=T State4=T State3=T State2=T State1=T State0=T
PORTA=0x00;
DDRA=0x00;

// Port B initialization
// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In Func0=In
// State7=T State6=T State5=T State4=T State3=T State2=T State1=T State0=T
PORTB=0x00;
DDRB=0x00;

// Port C initialization
// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In Func0=In
// State7=T State6=T State5=T State4=T State3=T State2=T State1=T State0=T
PORTC=0x00;
DDRC=0x00;
```

```
// Port D initialization
// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In Func0=In
// State7=T State6=T State5=T State4=T State3=T State2=T State1=T State0=T
PORTD=0x00;
DDRD=0x00;

// Timer/Counter 0 initialization
// Clock source: System Clock
// Clock value: Timer 0 Stopped
// Mode: Normal top=0xFF
// OC0 output: Disconnected
TCCR0=0x00;
TCNT0=0x00;
OCR0=0x00;

// Timer/Counter 1 initialization
// Clock source: System Clock
// Clock value: Timer1 Stopped
// Mode: Normal top=0xFFFF
// OC1A output: Discon.
// OC1B output: Discon.
// Noise Canceler: Off
// Input Capture on Falling Edge
// Timer1 Overflow Interrupt: Off
// Input Capture Interrupt: Off
// Compare A Match Interrupt: Off
// Compare B Match Interrupt: Off
TCCR1A=0x00;
```

```
TCCR1B=0x00;
TCNT1H=0x00;
TCNT1L=0x00;
ICR1H=0x00;
ICR1L=0x00;
OCR1AH=0x00;
OCR1AL=0x00;
OCR1BH=0x00;
OCR1BL=0x00;

// Timer/Counter 2 initialization
// Clock source: System Clock
// Clock value: Timer2 Stopped
// Mode: Normal top=0xFF
// OC2 output: Disconnected
ASSR=0x00;
TCCR2=0x00;
TCNT2=0x00;
OCR2=0x00;

// External Interrupt(s) initialization
// INT0: Off
// INT1: Off
// INT2: Off
MCUCR=0x00;
MCUCSR=0x00;

// Timer(s)/Counter(s) Interrupt(s) initialization
```

```
TIMSK=0x00;

// USART initialization
// Communication Parameters: 8 Data, 1 Stop, No Parity
// USART Receiver: On
// USART Transmitter: On
// USART Mode: Asynchronous
// USART Baud Rate: 9600
UCSRA=0x00;
UCSRB=0x18;
UCSRC=0x86;
UBRRH=0x00;
UBRRL=0x4D;

// Analog Comparator initialization
// Analog Comparator: Off
// Analog Comparator Input Capture by Timer/Counter 1: Off
ACSR=0x80;
SFIOR=0x00;

// ADC initialization
// ADC Clock frequency: 750,000 kHz
// ADC Voltage Reference: AREF pin
// ADC Auto Trigger Source: ADC Stopped
ADMUX=ADC_VREF_TYPE & 0xff;
ADCSRA=0x84;

// SPI initialization
```

```
// SPI disabled
SPCR=0x00;

// TWI initialization
// TWI disabled
TWCR=0x00;

// Alphanumeric LCD initialization
// Connections are specified in the
// Project|Configure|C Compiler|Libraries|Alphanumeric LCD menu:
// RS - PORTC Bit 0
// RD - PORTC Bit 1
// EN - PORTC Bit 2
// D4 - PORTC Bit 4
// D5 - PORTC Bit 5
// D6 - PORTC Bit 6
// D7 - PORTC Bit 7
// Characters/line: 16
lcd_init(16);

while (1)
{
// Place your code here
arus=0;

for( i=0;i<1000;i++)
{
a= read_adc(1);
```

```
    arus= arus+(0.049*a-25.45);
}
DataArus = arus/1000;

b= read_adc(2);

tegangan =((float)(b/4.400)/10);

lcd_gotoxy(0,0);
sprintf(ridik,"I = %1.2f A", DataArus);
lcd_puts(ridik);

lcd_gotoxy(0,1);
sprintf(ridik,"V = %1.2f Volt", tegangan);
lcd_puts(ridik);

delay_ms(50);
lcd_clear();

printf("A");printf("%1.2f",DataArus);printf("B");printf("%1.2f",tegangan);printf("C\n"); delay_ms(70);
/* printf("A\n");printf("%1.2f\n",DataArus); delay_ms(70);
printf("B\n");printf("%1.2f\n",tegangan); delay_ms(70);*/
}

/*****
```

This program was produced by the
CodeWizardAVR V2.05.3 Standard

Automatic Program Generator

© Copyright 1998-2011 Pavel Haiduc, HP InfoTech s.r.l.

<http://www.hpinfotech.com>

Project :

Version :

Date : 29/10/2014

Author : ridik

Company : the explorer

Comments:

Chip type : ATmega16

Program type : Application

AVR Core Clock frequency: 12,000000 MHz

Memory model : Small

External RAM size : 0

Data Stack size : 256

*****/

```
#include <mega16.h>
```

```
#include <delay.h>
```

```
#include <stdlib.h>
```

```
int c,d;
```

```
float a,b, DataArus,tegangan;
```

```
double arus = 0;
```

```
int DataTegangan;
int i;

char ridik[30];

// Alphanumeric LCD functions
#include <alcd.h>

// Standard Input/Output functions
#include <stdio.h>

#define ADC_VREF_TYPE 0x00

// Read the AD conversion result
unsigned int read_adc(unsigned char adc_input)
{
    ADMUX=adc_input | (ADC_VREF_TYPE & 0xff);
    // Delay needed for the stabilization of the ADC input voltage
    delay_us(10);
    // Start the AD conversion
    ADCSRA|=0x40;
    // Wait for the AD conversion to complete
    while ((ADCSRA & 0x10)==0);
    ADCSRA|=0x10;
    return ADCW;
}
```

```
// Declare your global variables here

void main(void)
{
// Declare your local variables here

// Input/Output Ports initialization
// Port A initialization
// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In
Func0=In
// State7=T State6=T State5=T State4=T State3=T State2=T State1=T
State0=T
PORTA=0x00;
DDRA=0x00;

// Port B initialization
// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In
Func0=In
// State7=T State6=T State5=T State4=T State3=T State2=T State1=T
State0=T
PORTB=0x00;
DDRB=0x00;

// Port C initialization
// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In
Func0=In
// State7=T State6=T State5=T State4=T State3=T State2=T State1=T
State0=T
```

```
PORTC=0x00;
DDRC=0x00;

// Port D initialization
// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In
Func0=In
// State7=T State6=T State5=T State4=T State3=T State2=T State1=T
State0=T
PORTD=0x00;
DDRD=0x00;

// Timer/Counter 0 initialization
// Clock source: System Clock
// Clock value: Timer 0 Stopped
// Mode: Normal top=0xFF
// OC0 output: Disconnected
TCCR0=0x00;
TCNT0=0x00;
OCR0=0x00;

// Timer/Counter 1 initialization
// Clock source: System Clock
// Clock value: Timer1 Stopped
// Mode: Normal top=0xFFFF
// OC1A output: Discon.
// OC1B output: Discon.
// Noise Canceler: Off
// Input Capture on Falling Edge
```

```
// Timer1 Overflow Interrupt: Off
```

```
// Input Capture Interrupt: Off
```

```
// Compare A Match Interrupt: Off
```

```
// Compare B Match Interrupt: Off
```

```
TCCR1A=0x00;
```

```
TCCR1B=0x00;
```

```
TCNT1H=0x00;
```

```
TCNT1L=0x00;
```

```
ICR1H=0x00;
```

```
ICR1L=0x00;
```

```
OCR1AH=0x00;
```

```
OCR1AL=0x00;
```

```
OCR1BH=0x00;
```

```
OCR1BL=0x00;
```

```
// Timer/Counter 2 initialization
```

```
// Clock source: System Clock
```

```
// Clock value: Timer2 Stopped
```

```
// Mode: Normal top=0xFF
```

```
// OC2 output: Disconnected
```

```
ASSR=0x00;
```

```
TCCR2=0x00;
```

```
TCNT2=0x00;
```

```
OCR2=0x00;
```

```
// External Interrupt(s) initialization
```

```
// INT0: Off
```

```
// INT1: Off
```

```
// INT2: Off
MCUCR=0x00;
MCUCSR=0x00;

// Timer(s)/Counter(s) Interrupt(s) initialization
TIMSK=0x00;

// USART initialization
// Communication Parameters: 8 Data, 1 Stop, No Parity
// USART Receiver: On
// USART Transmitter: On
// USART Mode: Asynchronous
// USART Baud Rate: 9600
UCSRA=0x00;
UCSRB=0x18;
UCSRC=0x86;
UBRRH=0x00;
UBRRL=0x4D;

// Analog Comparator initialization
// Analog Comparator: Off
// Analog Comparator Input Capture by Timer/Counter 1: Off
ACSR=0x80;
SFIOR=0x00;

// ADC initialization
// ADC Clock frequency: 750,000 kHz
// ADC Voltage Reference: AREF pin
```

```
// ADC Auto Trigger Source: ADC Stopped
ADMUX=ADC_VREF_TYPE & 0xff;
ADCSRA=0x84;

// SPI initialization
// SPI disabled
SPCR=0x00;

// TWI initialization
// TWI disabled
TWCR=0x00;

// Alphanumeric LCD initialization
// Connections are specified in the
// Project|Configure|C Compiler|Libraries|Alphanumeric LCD menu:
// RS - PORTC Bit 0
// RD - PORTC Bit 1
// EN - PORTC Bit 2
// D4 - PORTC Bit 4
// D5 - PORTC Bit 5
// D6 - PORTC Bit 6
// D7 - PORTC Bit 7
// Characters/line: 16
lcd_init(16);

while (1)
{
```

```
// Place your code here
 arus=0;

for( i=0;i<1000;i++)
{
 a= read_adc(1);
 arus= arus+(0.049*a-25.45);
}
 DataArus = arus/1000;

 b= read_adc(2);

 tegangan =((float)(b/4.400)/10);

 lcd_gotoxy(0,0);
 sprintf(ridik,"I = %1.2f A", DataArus);
 lcd_puts(ridik);

 lcd_gotoxy(0,1);
 sprintf(riki,"V = %1.2f Volt", tegangan);
 lcd_puts(ridik);

 delay_ms(50);
 lcd_clear();
```

```
printf("A");printf("%.2f",DataArus);printf("B");printf("%.2f",tegangan);pri  
ntf("C\n"); delay_ms(70);  
    /* printf("A\n");printf("%.2f\n",DataArus); delay_ms(70);  
    printf("B\n");printf("%.2f\n",tegangan); delay_ms(70);*/  
}  
}
```



B. Program *Graphic User Interface* Pada Visual Basic Studio 2010 Ultimate

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using System.IO.Ports;
using System.IO;
using System.Globalization;
using ZedGraph;
//using Microsoft.Office.Interop.Excel;

namespace Monitoring_Tegangan_dan_Arus
{
    public partial class Form1 : Form
    {
        double waktuStart = 100;
        string kode;
        int dd=0;
        Int16 time = 0;
        public Form1()
        {
            InitializeComponent();
        }

        private void Form1_Load(object sender, EventArgs e)
        {
            string[] portNames = SerialPort.GetPortNames();
            for (int i = 0; i <= portNames.Length - 1; i++)
            {
                comboBox1.Items.Add(portNames[i]);
                led1.Enabled = true;
            }

            waktuStart = Environment.TickCount;
            GraphPane grafik = zedGraphControl1.GraphPane;
            grafik.Title.Text = "Grafik";
            grafik.XAxis.Title.Text = "Waktu (Detik)";
        }
    }
}
```

```
grafik.YAxis.Title.Text = "Parameter";

RollingPointPairList listteganggan = new
RollingPointPairList(120);
LineItem kurvategangan =
grafik.AddCurve("Tegangan", listteganggan, Color.Red,
SymbolType.None);
RollingPointPairList listarus = new
RollingPointPairList(120);
LineItem kurvaarus = grafik.AddCurve("Arus",
listarus, Color.Blue, SymbolType.None);

button1.Enabled = true;
button2.Enabled = false;
led1.Value = false;
led2.Value = false;
led3.Value = false;
led4.Value = false;
toolStripStatusLabel1.Text = "ComPort" +
"Disconnect";
if (serialPort1.IsOpen)
{
    serialPort1.Close();
}
button1.Enabled = true;
button2.Enabled = false;
}
/*private void tampil(object sender, EventArgs e)
{
    string aaa = Convert.ToString(dd);
    //richTextBox1.Text = textBox2.Text + '\n' +
textBox3.Text + '\n' + "ak" + '\n' + "ak";
    richTextBox1.AppendText(aaa);
    //textBox1.Text = "\n";
    //richTextBox1.Dock=DockStyle.Fill;
}*/
int ss=0;
private void Detak_Tick(object sender, EventArgs e)
{
    try
    {
        serialPort1.Open();
```

```
-----  
//-----Grafik-----  
-----  
        LineItem kurvategangan =  
zedGraphControl1.GraphPane.CurveList[0] as LineItem;  
        IPointListEdit listtategangan =  
kurvategangan.Points as IPointListEdit;  
        LineItem kurvaarus =  
zedGraphControl1.GraphPane.CurveList[1] as LineItem;  
        IPointListEdit listarus = kurvaarus.Points as  
IPointListEdit;  
  
        double waktu = (Environment.TickCount -  
waktuStart) / 1000.0;  
        int menit = (int)waktu / 60;  
        int jam = (int)waktu / 3600;  
        double dat = jam * 3600;  
        int mod = (int)waktu - (int)dat;  
        int det = (int)waktu-(menit*60);  
        int men = menit - (jam * 60);  
        kode = serialPort1.ReadLine();  
  
        float Ars =  
float.Parse(kode.Substring(kode.IndexOf('A') + 1,  
kode.IndexOf("B") - 1));  
        float Teg =  
float.Parse(kode.Substring(kode.IndexOf('B') + 1,  
kode.IndexOf("C") - kode.IndexOf("B") - 1));  
  
        string aaa = Convert.ToString(Ars);  
        string ddd = Convert.ToString(Teg);  
        string sss = Convert.ToString(jam);  
        string fff = Convert.ToString(menit);  
        string ggg = Convert.ToString(mod);  
  
        if (mod < 3)  
        {  
            if (ss == 0)  
            {  
                ss = -1;  
                richTextBox1.AppendText("Jam ke-" +  
sss + " : " + ddd + " V " + aaa + " A" +'\n');  
            }  
        }  
    }  
}
```

```
        else
        {
            ss++;
        }
    }

    listtegangan.Add(waktu,
Convert.ToDouble(Teg));
    listarus.Add(waktu, Convert.ToDouble(Ars));

    Scale xscale =
zedGraphControl1.GraphPane.XAxis.Scale;
    if (waktu > xscale.Max - xscale.MajorStep)
    {
        xscale.Max = waktu + xscale.MajorStep;
        xscale.Min = xscale.Max - 30.0;
    }
    zedGraphControl1.AxisChange();
    zedGraphControl1.Invalidate();
    led1.Value = true;
    led2.Value = true;
    led3.Value = true;
    led4.Value = true;
    segmentGauge1.Value = Teg;
    segmentGauge2.Value = Ars;
    angularGauge1.Value = Teg;
    angularGauge2.Value = Ars;
    float arsmil = Ars * 1000;
    float daya = Ars * Teg;
    segmentGauge4.Value = arsmil;
    segmentGauge3.Value = daya;
    angularGauge3.Value = daya;
    segmentGauge5.Value = waktu;
    segmentGauge6.Value = men;
    segmentGauge7.Value = jam;
    segmentGauge8.Value = det;
    //label5.Text =
Convert.ToString(Convert.ToString(waktu));
    //richTextBox1.Text = Convert.ToString(dd);
}
}
```

```
        catch (Exception gagal)
        {
            if (serialPort1.IsOpen)
            {
                serialPort1.Close();
            }
        }
    }

e)    private void button1_Click_1(object sender, EventArgs
    {
        button1.Enabled = false;
        button2.Enabled = true;
        serialPort1.PortName = comboBox1.Text;
        serialPort1.BaudRate = int.Parse(comboBox2.Text);
        serialPort1.Parity = Parity.None;
        serialPort1.DataBits = 8;
        serialPort1.StopBits = StopBits.One;
        Detak.Enabled = true;
        Detak.Start();
        //textBox1.Text = ;
        toolStripStatusLabel1.Text = comboBox1.Text +
"Connected";
    }

e)    private void button2_Click_1(object sender, EventArgs
    {
        button1.Enabled = true;
        button2.Enabled = false;
        Detak.Enabled = false;
        Detak.Stop();
        toolStripStatusLabel1.Text = comboBox1.Text +
"DisConnected";
        if (serialPort1.IsOpen)
        {
            serialPort1.Close();
        }
    }

    private void button3_Click(object sender, EventArgs e)
```

```
{
    try
    {
        using (StreamWriter tulis = new
StreamWriter(@"D:\fileku.txt", false))
        {
            tulis.Write(richTextBox1.Text);
        }
    }
    catch (Exception error)
    {
        MessageBox.Show("Terjadi kesalahan\n" +
error.ToString());
    }
}
}
```

C. Data Lengkap Pengujian *Starting* Beban 35 Watt

No.	Beban (Watt)	Waktu (jam)	Tegangan (Volt)	Arus (Ampere)	Daya Beban (Watt)
1.	35	Start	12,68	1,9	24,09
2.		1	12,43	1,85	23
3.		2	12,32	1,84	22,67
4.		3	12,25	1,85	22,66
5.	35	4	12,14	1,84	22,34
6.		5	9,75	1,64	16
7.		6	9,89	1,64	16,22
8.		7	9,91	1,62	16,04
9.		8	9,84	1,62	15,94
10.		9	9,82	1,62	15,91
11.		10	9,73	1,62	15,76
12.		11	9,61	1,62	15,57
13.		12	9,52	1,6	15,23
14.		13	9,09	1,55	14,09
15.		14	8,5	1,51	12,84
16.		15	6,89	1,36	9,49
17.		16	6,55	1,31	8,58
18.		17	6,16	1,28	7,88
19.		18	5,98	1,26	7,34
20.		19	5,34	1,19	6,35
21.		20	3,77	1,01	3,81
22.	35	Finish	1,95	0,78	1,52

D. Data Lengkap Pengujian Konstan Beban 25 Watt

No.	Beban (Watt)	Waktu (jam)	Tegangan (Volt)	Arus (Ampere)	Daya Beban (Watt)
1.	25	Start	13,25	1,31	17,36
2.		1	12,2	1,26	15,37
3.		2	12,2	1,27	15,49
4.		3	12,18	1,26	15,34
5.		4	12,18	1,26	15,34
6.		5	12,14	1,25	15,18
7.		6	12,07	1,26	15,21
8.		7	11,89	1,29	15,34
9.		8	9,8	1,17	11,47
10.		9	9,77	1,16	11,33
11.		10	9,66	1,17	11,3
12.		11	9,77	1,15	11,24
13.		12	9,77	1,14	11,14
14.		13	9,77	1,13	11,04
15.		14	9,77	1,13	11,04
16.		15	9,69	1,12	10,85
17.		16	9,66	1,1	10,63
18.		17	9,56	1,11	10,61
19.		18	9,52	1,11	10,57
20.		19	9,48	1,1	10,43
21.	25	20	9,36	1,1	10,3
22.		21	9,27	1,09	10,1
23.		22	8,25	1,03	8,5
24.		23	6,57	0,91	5,98
25.		24	5,89	0,88	5,18
26.		25	4,75	0,8	3,8
27.		26	3,18	0,68	2,16
28.	25	Finish	1,93	0,56	1,08

E. Data Lengkap Pengujian *Breaking* Beban 18 Watt

No.	Beban (Watt)	Waktu (Jam)	Tegangan (Volt)	Arus (Ampere)	Daya Beban (Watt)
1.	18	Start	13,34	1,01	13,47
2.		1	12,39	0,98	12,14
3.		2	12,45	0,99	12,33
4.		3	12,43	1	12,43
5.		4	12,43	0,99	12,31
6.		5	12,41	0,99	12,29
7.		6	12,39	0,99	12,27
8.		7	12,34	0,99	12,22
9.		8	12,3	0,99	12,18
10.	18	9	10,02	0,89	8,92
11.		10	10,09	0,89	8,98
12.		11	10,09	0,89	8,98
13.		12	10,09	0,89	8,98
14.		13	10,02	0,9	9,02
15.		14	10,05	0,89	8,94
16.		15	10,02	0,89	8,92
17.		16	10	0,89	8,9
18.		17	10	0,88	8,8
19.		18	9,95	0,88	8,76
20.		19	9,89	0,89	8,8
21.		20	9,87	0,88	8,69
22.		21	9,84	0,88	8,66
23.		22	9,82	0,88	8,64
24.		23	9,7	0,88	8,54
25.		24	9,59	0,87	8,34
26.		25	9,49	0,87	8,26
27.		26	9,34	0,86	8,03
28.		27	9,06	0,87	7,88
29.		28	8,86	0,86	7,62
30.		29	8,64	0,84	7,26
31.		30	8	0,79	6,32
32.	18	31	5,09	0,64	3,26
33.		32	4,8	0,62	2,98
34.		33	4,73	0,63	2,98
35.		34	4,2	0,58	2,44
36.		35	3	0,52	1,56
37.	18	Finish	2,36	0,47	1,11

F. Data Lengkap Pengujian Couple Beban 43 Watt

No.	Beban (Watt)	Waktu (jam)	Tegangan (Volt)	Arus (Ampere)	Daya Beban (Watt)
1.	43	Start	12.91	2.44	31.5
2.		1	10.91	2.12	23.13
3.		2	10.86	2.12	23.02
4.		3	10.75	2.11	22.68
5.		4	10.75	2.11	22.68
6.		5	10.68	2.1	22.43
7.		6	10.55	2.1	22.16
8.		7	9,52	2.03	19.33
9.		8	5.45	1.56	8.5
10.		9	5.45	1.56	8.5
11.		10	5.43	1.54	8.36
12.		11	5.11	1.52	7.77
13.		12	3.18	1.25	3.98
14.		13	3.05	1.23	3.75
15.	43	Finish	2.3	1.13	2.6

G. Data Lengkap Pengujian Couple Beban 53 Watt

No.	Beban (Watt)	Waktu (jam)	Tegangan (Volt)	Arus (Ampere)	Daya Beban (Watt)
1.	53	Start	13.98	2.78	38.68
2.		1	11.18	2.49	27.84
3.		2	11.07	2.48	27.45
4.		3	11	2.39	26.29
5.		4	8,55	2.1	17.96
6.		5	8.55	2.08	17.78
7.		6	5.98	1.85	11.06
8.		7	5.73	1.79	10.26
9.		8	3.07	1.38	4.24
10.		9	2.86	1.36	3.89
11.		10	2.68	1.28	3.43
12.	53	Finish	2.39	1.22	2.92

H. Data Lengkap Pengujian Couple 60 Watt

No.	Beban (Watt)	Waktu (jam)	Tegangan (Volt)	Arus (Ampere)	Daya Beban (Watt)
1.	60	Start	13.27	3.07	40,74
2.		1	10.8	2.69	29,05
3.		2	8.36	2.33	19,48
4.		3	7.25	2.1	15,23
5.		4	6.18	1.97	12,17
6.		5	3.91	1.57	5,94
7.		6	3.3	1.46	4.82
8.	60	Finish	2.32	1.3	2.65

I. Data Lengkap Pengujian Couple 78 Watt

No.	Beban (Watt)	Waktu (jam)	Tegangan (Volt)	Arus (Ampere)	Daya Beban (Watt)
1.	78	Start	10.18	3.72	37,87
2.		1	6.18	3.14	19,41
3.		2	6.25	3.35	20,94
4.		3	3.7	2.77	10,25
5.		4	3.73	2.74	10,22
6.		5	3.2	2.7	8,64
7.	78	Finish	2.25	2.64	5,94