

**SISTEM AKUISISI DATA *VIA WEBSITE*
BERBASISKAN MIKROKONTROLLER**

SKRIPSI

Oleh

ANDRI EKO SUWARNO

04 03 03 015 2



**DEPARTEMEN TEKNIK ELEKTRO
FAKULTAS TEKNIK UNIVERSITAS INDONESIA
GENAP 2007/2008**

**SISTEM AKUISISI DATA *VIA WEBSITE*
BERBASISKAN MIKROKONTROLLER**

SKRIPSI

Oleh

ANDRI EKO SUWARNO

04 03 03 015 2



**SKRIPSI INI DIAJUKAN UNTUK MELENGKAPI
SEBAGIAN
PERSYARATAN MENJADI SARJANA TEKNIK**

**DEPARTEMEN TEKNIK ELEKTRO
FAKULTAS TEKNIK UNIVERSITAS INDONESIA
GENAP 2007/2008**

PERNYATAAN KEASLIAN SKRIPSI

Saya menyatakan dengan sesungguhnya bahwa skripsi dengan judul:

SISTEM AKUISISI DATA VIA WEBSITE BERBASISKAN MIKROKONTROLLER

yang dibuat untuk melengkapi sebagian persyaratan menjadi Sarjana Teknik pada Program Studi Teknik Elektro, Departemen Teknik Elektro Fakultas Teknik Universitas Indonesia, sejauh yang saya ketahui bukan merupakan tiruan atau duplikasi dari skripsi yang sudah dipublikasikan dan atau pernah dipakai untuk mendapatkan gelar kesarjanaan di lingkungan Universitas Indonesia maupun di Perguruan Tinggi atau Instansi manapun, kecuali bagian yang sumber informasinya dicantumkan sebagaimana mestinya.

Depok, 24 Juni 2008

Andri Eko Suwarno

NPM. 0403030152

PENGESAHAN

Skripsi dengan judul:

PERANCANGAN SISTEM AKUISISI DATA VIA WEBSITE BERBASISKAN MIKROKONTROLLER

dibuat untuk melengkapi sebagian persyaratan menjadi Sarjana Teknik pada Program Studi Teknik Elektro, Departemen Teknik Elektro Fakultas Teknik Universitas Indonesia dan dinyatakan memenuhi syarat/sah sebagai skripsi pada Program Studi Teknik Elektro, Departemen Teknik Elektro Fakultas Teknik Universitas Indonesia.

Depok, 24 Juni 2008

Dosen Pembimbing

Dr. Ir. Feri Yusivar, M.Eng

NIP. 132 090 912

UCAPAN TERIMA KASIH

Puji syukur kepada Allah SWT atas segala rahmat dan karunia-Nya sehingga skripsi ini dapat diselesaikan. Shalawat dan salam semoga senantiasa tercurahkan kepada Nabi Muhammad SAW. Penulis ingin mengucapkan terima kasih kepada Bapak:

Dr. Ir. Feri Yusivar, M.Eng

selaku dosen pembimbing yang telah banyak meluangkan waktunya untuk memberikan saran, bimbingan, dan pengarahan sehingga skripsi ini dapat diselesaikan dengan baik.

Andri Eko Suwarno
NPM 04 03 03 015 2
Departemen Teknik Elektro

Dosen Pembimbing
Dr. Ir. Feri Yusivar, M.Eng

**SISTEM AKUISISI DATA ELEKTRIS SOLAR CELL
BERBASIS MIKROKONTROLLER**

ABSTRAK

Pengukuran dan pengolahan data, di dalam dunia elektronika merupakan suatu hal yang penting. Pengukuran tersebut biasanya dilakukan untuk mengetahui sejauh mana kinerja ataupun efisiensi dari sistem terpasang. Namun seringkali waktu yang tersedia tidak mencukupi untuk melakukan pengukuran secara langsung, disebabkan karena tempat yang jauh.

Skripsi ini bertujuan untuk merancang suatu sistem pengumpul data yang dapat diakses melalui halaman *website*. Dengan ini diharapkan dapat mengefisienkan waktu yang terpakai untuk pengukuran dan juga mempercepat proses analisis kinerja dari suatu alat/sistem. Parameter yang akan diujikan di dalam sistem ini mencakup kehandalan sistem dalam menyalurkan informasi, dan juga ketelitian pembacaan data-data yang diukur oleh sistem.

Berdasarkan hasil pengujian sistem, komunikasi antara PC dengan master dapat mencapai kecepatan 115200 bps tanpa mengalami *error*. *Database* dapat melakukan *query* dari data-data yang dipilih dengan waktu *query* 0.0021 detik.

Kata Kunci : data akuisisi, komunikasi RS-232, Analog-to-Digital Converter

Andri Eko Suwarno
NPM 04 03 03 015 2
Electrical Engineering Department

Counselor
Dr. Ir. Feri Yusivar, M.Eng

ELECTRICAL DATA ACQUISITION SYSTEM BASED ON MICROCONTROLLER

ABSTRACT

In electronics, data measurement is one of important thing. It usually does to determine efficiency and performance of the system or device. However, because the location of the system being measured located far away makes it impossible to do such thing.

The purpose of this research is to design a system that collect data from the device, and access the collected data through a web page. With this mechanism, the difficulty of measuring electrical data from device that located far away becomes possible. Parameters of the acquisition system which will be analyzed are reliability and accuracy of data that read by system.

From the data obtained from system test, the communication speed between PC and master can reach 115200 bps without error. The average querying time of the database is about 0.0021 sec.

Kata Kunci : data acquisition,RS-232 Communication,Analog-to-Digital Converter

DAFTAR ISI

PERNYATAAN KEASLIAN SKRIPSI	ii
PENGESAHAN	iii
UCAPAN TERIMA KASIH	iv
ABSTRAK	v
DAFTAR ISI	vii
DAFTAR GAMBAR	viii
DAFTAR TABEL	ix
DAFTAR SINGKATAN	x
BAB I PENDAHULUAN	1
1.1 LATAR BELAKANG	1
1.2 TUJUAN PENELITIAN	2
1.3 BATASAN MASALAH	3
1.4 METODOLOGI PENELITIAN	3
1.5 SISTEMATIKA PENULISAN	3
BAB II LANDASAN TEORI	4
2.1 JAVA COMMUNICATION API	4
2.2 RS-232 STANDARD	5
2.2.1 Karakteristik sinyal RS-232	6
2.2.2 Menghubungkan TTL ke RS-232	6
2.2.3 Konektor dan jenis sinyal RS-232	7
BAB III PERANGKAT KERAS DAN PERANGKAT LUNAK SISTEM	9
3.1 PERANCANGAN PERANGKAT KERAS SISTEM	9
3.1.1 Perancangan Perangkat Master	9
3.1.2 Perancangan Perangkat Slave	10
3.1.3 Modul Pengkondisi Sinyal	11
3.1.4 Modul Power Supply	12
3.2 PERANCANGAN PERANGKAT LUNAK SISTEM	12
3.2.1 Perancangan Firmware Perangkat Master	12
3.2.2 Perancangan Firmware Perangkat Slave	14
3.2.3 Perancangan Software Pemroses Data Serial	15
3.2.4 Perancangan Web Server dan Database Server	16
3.2.5 Perancangan Halaman Website	16
3.3 PERANCANGAN PROTOKOL SISTEM	16
3.3.1 <i>Frame</i> Data dari <i>Master</i> ke <i>Slave</i>	16
3.3.2 <i>Frame</i> Data dari <i>Slave</i> ke <i>Master</i>	17
3.3.3 <i>Frame</i> Data dari PC ke <i>Master</i>	18
3.3.4 <i>Frame</i> Data dari <i>Master</i> ke PC	18
BAB IV PEMBAHASAN DAN ANALISIS	20
4.1 METODE PENGUJIAN	20
4.2 PENGUJIAN PERANGKAT KERAS	20
4.2.1 Pengujian Perangkat Master	20
4.2.1.1 Pengujian Komunikasi Data USART0	21
4.2.1.2 Pengujian Komunikasi Data USART1	22

4.2.2 Pengujian Perangkat Slave	22
4.2.3 Pengujian Perangkat Pengkondisi Sinyal	24
4.2.4 Pengujian Perangkat Power Supply	24
4.3 PENGUJIAN PROTOKOL KOMUNIKASI	25
4.3.1 Pengujian Header 0xA1	25
4.3.2 Pengujian Hedaer 0xD1	26
4.3.3 Pengujian Header 0xC1	26
4.3.4 Pengujian Header 0xA0 dan 0xC0	26
4.4 PENGUJIAN PERANGKAT LUNAK DAN SISTEM KESELURUHAN	27
4.4.1 Pengujian Aplikasi Pengambil Data	27
4.4.2 Pengujian Database	29
4.4.3 Pengujian Tampilan Website	30
BAB V KESIMPULAN	32
DAFTAR ACUAN	33
DAFTAR PUSTAKA	34

DAFTAR GAMBAR

	Halaman	
Gambar 1.1	Blok Diagram Sistem Keseluruhan	2
Gambar 2.1	High Level Diagram Java Communiacation API Interface	4
Gambar 2.2	Hubungan antara DTE-DCE-DCE-DTE	5
Gambar 2.3	Level Tegangan RS-232	6
Gambar 2.4	Level Tegangan TTL	7
Gambar 3.1	Blok Diagram Sistem	9
Gambar 3.2	Blok Diagram modul master hardware	10
Gambar 3.3	Blok diagram modul slave hardware	11
Gambar 3.4	Diagram alir proses perangkat master	13
Gambar 3.5	Diagram alir proses perangkat slave	14
Gambar 3.6	Frame Data untuk master ke slave	17
Gambar 4.1	Perangkat master dan slave beserta perangkat pengujian	23
Gambar 4.2	Perangkat <i>Power Supply</i> dan Pengkondisi Sinyal	25
Gambar 4.3	Frame instruksi data request dari PC ke master	25
Gambar 4.4	Frame instruksi data acknowledgement dari PC ke master	26
Gambar 4.5	Frame instruksi data dari master ke PC	26
Gambar 4.6	Frame instruksi data request dari master ke slave.	27
Gambar 4.7	Frame instruksi data dari slave ke master.	27
Gambar 4.8	Tampilan Aplikasi Pengambil Data Serial.	28
Gambar 4.9	Tampilan Aplikasi Pengambil Data Serial ketika berhasil memasukkan data	29
Gambar 4.10	Tampilan dari halaman pembuka website.	31
Gambar 4.11	Tampilan dari hasil pilihan data yang telah dipilih	31

DAFTAR TABEL

	Halaman	
Tabel 2.1	Jenis Sinyal RS-232 yang umum dipakai	8
Tabel 3.1	Penjelasan <i>field</i> data dari <i>frame master</i> ke <i>slave</i>	17
Tabel 3.2	Penjelasan <i>field</i> data frame dari <i>slave</i> ke <i>master</i>	17
Tabel 3.3	Penjelasan field data dari PC ke Master	18
Tabel 3.4	Penjelasan field data dari Master ke PC	19
Tabel 4.1	Hasil pengujian pengiriman data antara PC ke master	21
Tabel 4.2	Hasil pengujian pengiriman data antara PC ke master 115200 bps	22
Tabel 4.3	Hasil Pengujian pengiriman data antara master ke slave	22
Tabel 4.4	Hasil Percobaan waktu query dari database	30

DAFTAR SINGKATAN

PC	Personal Computer
RS-232	Recomended Standard 232
ADC	Analog-to-Digital Converter
API	Application Programing Interface
EIA	Electronic Industry Association
TIA	Telecommunication Industry Association
DTE	Data Terminal Equipment
DCE	Data Circuit-Terminating Equipment
Modem	Modulator-Demodulator
TTL	Transistor-transistor logic
IC	Integrated Circuit
TD	Transmitted Data
RD	Received Data
RTS	Request to Send
CTS	Clear to Send
DSR	DCE Ready
DTR	DTE Ready
RI	Ring Indicator
DCD	Data Carrier Detect
bps	bit per second
STX	Start of Text
ETXB	End of Text
ASCII	American Standard
MSB	Most Significant Bit
LSB	Least Significant Bit

BAB I

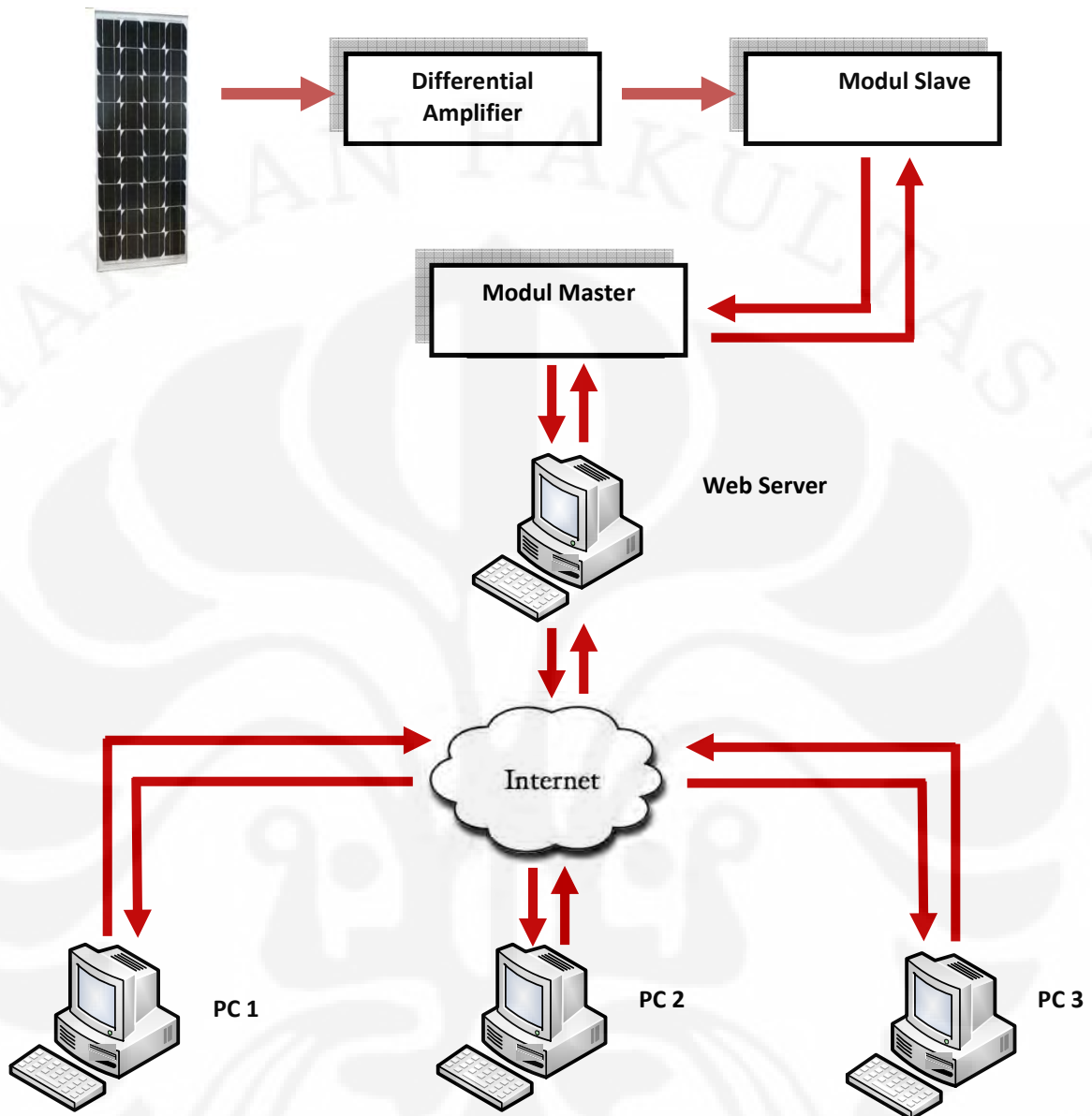
PENDAHULUAN

1.1 LATAR BELAKANG

Pengukuran dan pengolahan data dalam dunia elektronik merupakan hal yang sering dilakukan. Data yang diukur bisa suatu data digital atau analog, untuk mendapatkan data tersebut bisa dengan cara pembacaan secara langsung atau melalui suatu proses pengubahan. Seiring dengan berkembangnya teknologi informasi, maka ketersediaan akan data tersebut menjadi suatu kebutuhan mendasar. Berdasarkan kebutuhan itu maka dalam skripsi ini akan dirancang suatu sistem pengumpul data berbasis *website* sehingga data bisa diakses selama ada koneksi internet.

Sistem pengumpulan data yang dirancang berfungsi untuk membaca data dari berbagai modul sesuai dengan fungsinya masing-masing. Untuk mengumpulkan data tersebut sistem menggunakan mode komunikasi serial. Sistem komunikasi yang dirancang menggunakan prinsip *master-slave*. *Master* menerima perintah yang dikirim oleh PC. Kemudian *master* akan menerjemahkan perintah tersebut untuk diteruskan kepada *slave*. Untuk mengatur komunikasi data antara *master* dan *slave* dibutuhkan suatu protokol komunikasi yang khusus, sehingga perancangan protokol komunikasi menjadi salah satu perancangan dan realisasi yang dilakukan.

Perangkat *master* pada sistem pengumpul data berfungsi memerintah *slave* untuk mengambil data digital perangkat yang terhubung dengan *slave* tersebut. Pada sistem ini *slave* mengambil data dari perangkat *solar cell*. Setelah data tersebut sampai pada *master*, data akan dikirimkan ke PC. Pada PC, kemudian data diolah untuk dimasukkan ke dalam *database* yang akan diakses melalui *website*. Protokol komunikasi antara komputer dan *master* menggunakan protokol buatan. Sistem akuisisi data listrik *solar cell* secara keseluruhan ditampilkan pada gambar 1.1.



Gambar 1.1 Blok Diagram Sistem Keseluruhan

1.2 TUJUAN PENELITIAN

Skripsi ini bertujuan untuk merancang suatu sistem pengumpul data berbasis mikrokontroler, dan membuat antarmuka dengan PC yang akan disimpan ke dalam suatu *database* yang akan diakses oleh halaman *website*. *Website* ini nantinya akan berfungsi sebagai penampil data dari *solar cell*.

1.3 BATASAN MASALAH

Batasan masalah penulisan dalam skripsi ini sebagai berikut :

- a) Uji coba dilakukan dengan memberikan tegangan masukan kepada perangkat *slave* melalui potensiometer.
- b) Komunikasi data antara PC-*Master-Slave* menggunakan sistem komunikasi RS-232.
- c) Data elektrik yang diambil merupakan data hasil pembacaan ADC yang belum dikonversi dalam jangkauan 0 – 1023.
- d) Protokol komunikasi yang digunakan di dalam sistem dibuat sesederhana mungkin
- e) Jarak terjauh yang dapat dicapai oleh komunikasi RS-232 adalah 50 *feet* dengan *data rate* 19.200 bps.

1.4 METODOLOGI PENELITIAN

Metodologi penelitian ini adalah studi literatur yang berkaitan dengan perancangan baik *hardware* maupun *software* yang terkait dengan sistem pengumpul data.

1.5 SISTEMATIKA PENULISAN

Pada bab satu akan dijelaskan mengenai latar belakang serta tujuan yang diperoleh dari penulisan skripsi ini. Bab dua menjelaskan tentang dasar teori yang diperlukan. Bab tiga merupakan perancangan atau pengaplikasian dari teori yang ada. Bab empat adalah analisa dan pengukuran terhadap sistem yang telah dibuat dan bab lima berisi kesimpulan dari sistem yang telah dibuat.

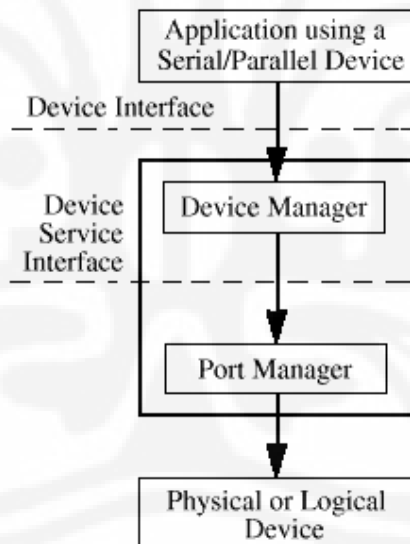
BAB II

LANDASAN TEORI

2.1 JAVA COMMUNICATION API (*Application Programming Interface*)

Java Communication (yang juga dikenal dengan *javax.comm*) API adalah suatu standar yang diusulkan untuk memungkinkan para pembuat aplikasi komunikasi agar dapat membuat suatu *software* berbasis Java yang dapat mengakses *port* komunikasi yang tidak bergantung dengan *platform* yang digunakannya. API ini dapat digunakan untuk membuat suatu aplikasi emulasi terminal, *software fax*, *software* pembaca *smart-card*, dan lain-lain.

Untuk dapat mengembangkan *software* yang baik biasanya kita harus memiliki antarmuka yang secara jelas didefinisikan. *High level diagram* dari lapisan antarmuka API ditunjukkan pada gambar berikut ini.



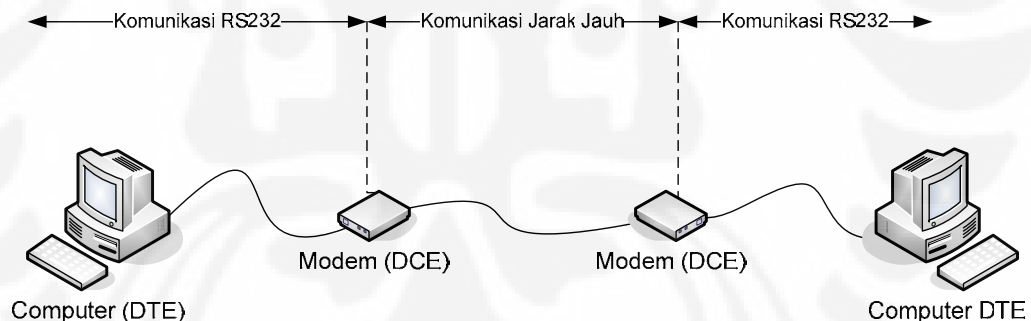
Gambar 2.1 High level diagram Java Communication API interface [1]

Tidak seperti *driver-driver* klasik, yang datang dengan masing-masing model komunikasi dari peristiwa-peristiwa asinkron, *javax.comm* API menyediakan suatu *even-style interface* berdasarkan pada Java event model (*java.awt.event* package). Jika kita ingin mengetahui apakah ada data baru yang berada di *input buffer*. Kita dapat mengetahuinya dengan dua cara yaitu dengan *polling* dan *listening*. Dengan metode *polling*, *processor* memeriksa *buffer* secara periodik untuk melihat apakah terdapat

data baru di input *buffer*. Sedangkan dengan metode *listening*, *processor* menunggu suatu *even* terjadi dalam bentuk data baru di input *buffer*. Ketika data baru muncul di input *buffer*, dia akan mengirimkan pemberitahuan kepada *processor*.

2.2 RS-232 STANDARD

Standard RS232 ditetapkan oleh Electronic Industry Association dan Telecommunication Industry Association pada tahun 1962. Nama lengkapnya adalah *EIA/TIA-232 Interface Between Data Terminal Equipment and Data Circuit-Terminating Equipment Employing Serial Binary Data Interchange*. Standard ini menyangkut komunikasi data antara komputer (*Data Terminal Equipment* – DTE) dengan alat-alat pelengkap komputer (*Data Circuit-Terminating Equipment* – DCE). Dalam banyak literatur, DCE sering diartikan sebagai sebagai *Data Communication Equipment*, hal ini bisa dibenarkan tapi pengertiannya menjadi lebih sempit karena sebagai *Data Communication Equipment* yang dimaksud dengan DTE hanya sebatas peralatan untuk komunikasi, misalnya Modem. Padahal yang dimaksud dengan *Data Circuit-Terminating Equipment* bisa meliputi macam-macam alat pelengkap komputer yang dihubungkan ke komputer dengan standard RS232, misalnya *printer*, *Optical Mark Reader*, *Cash Register*, PABX, dan lain-lain.



Gambar 2.2 Hubungan antara DTE-DCE-DCE-DTE

Ada 3 hal pokok yang diatur standard RS232, antara lain adalah :

- bentuk sinyal dan *level* tegangan yang dipakai
- penentuan jenis sinyal dan konektor yang dipakai, serta susunan sinyal pada kaki-kaki di konektor
- penentuan tata cara pertukaran informasi antara komputer dan alat-alat pelengkap.

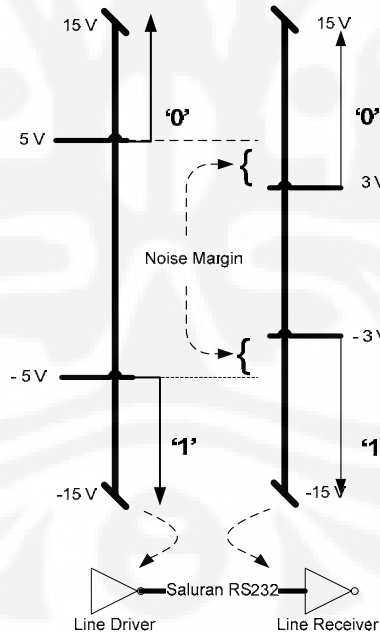
2.2.1 Karakteristik sinyal RS-232

Karakteristik sinyal yang diatur meliputi *level* tegangan sinyal, kecuraman perubahan tegangan (*slew rate*) dari *level* tegangan '0' menjadi '1' dan sebaliknya, serta impedansi dari saluran yang dipakai. RS232 dibuat pada tahun 1962, jauh sebelum IC TTL populer, maka *level* tegangan yang ditentukan untuk RS232 tidak ada hubungannya dengan *level* tegangan TTL, bahkan jauh berbeda.

Dalam standard RS232, tegangan antara +3 sampai +15 Volt pada *input Line Receiver* dianggap sebagai *level* tegangan '0', dan tegangan antara -3 sampai -15 Volt dianggap sebagai *level* tegangan '1'.

Agar *output Line Driver* bisa dihubungkan dengan baik, tegangan *output Line Driver* berkisar antara +5 sampai +15 Volt untuk menyatakan *level* tegangan '0', dan berkisar antara -5 sampai -15 Volt untuk menyatakan *level* tegangan '1'

Beda tegangan sebesar 2 Volt ini disebut sebagai *noise margin* dari RS232.



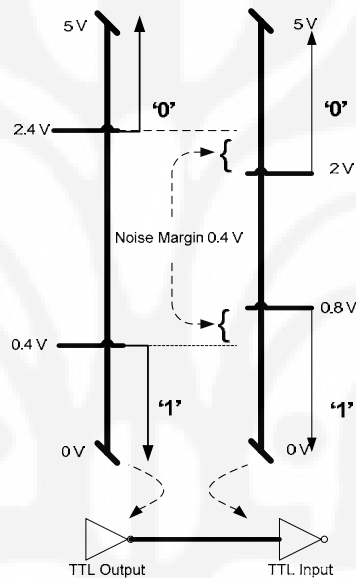
Gambar 2.3 *Level* Tegangan RS232

2.2.2 Menghubungkan TTL ke RS-232

IC digital, termasuk mikrokontroler, umumnya bekerja pada *level* tegangan TTL, yang dibuat atas dasar tegangan catu daya +5 Volt.

- Rangkaian *input* TTL menganggap tegangan kurang dari 0,8 Volt sebagai *level* tegangan '0' dan tegangan lebih dari 2.0 Volt dianggap sebagai *level* tegangan '1'. *Level* tegangan ini sering dikatakan sebagai *level* tegangan TTL.
- Untuk menjamin *output* bisa diumpankan ke *input* dengan baik, tegangan *output* TTL saat *level* '0' dijamin lebih rendah dari 0,4 Volt, atau 0,4 lebih rendah dari tegangan yang dituntut oleh *input* TTL. Sedangkan tegangan *output* TTL pada saat *level* '1' dijamin lebih tinggi dari 2,4 Volt, atau 0,4 Volt lebih tinggi dari tegangan yang dituntut oleh *input* TTL.

Beda tegangan sebesar 0,4 Volt ini disebut sebagai *noise margin* dari TTL.



Gambar 2.4 Level Tegangan TTL

Hampir semua komponen digital bekerja pada *level* tegangan TTL, dengan demikian dalam membentuk saluran RS232 diperlukan perubahan *level* tegangan timbal balik antara TTL-RS232.

Pada IC MAX232 yang berisikan 2 buah RS232 Line Driver dan 2 buah RS232 Line Receiver, kita tidak lagi memerlukan perubahan *level* tegangan karena di dalam IC tersebut dilengkapi pengganda tegangan DC. Sehingga walaupun daya untuk IC MAX232 hanya +5 Volt, tapi sanggup melayani *level* tegangan RS232 antara -10 Volt sampai +10 Volt.

2.2.3 Konektor dan jenis sinyal RS232

Selain mendeskripsikan *level* tegangan seperti yang dibahas di atas, standard RS232 menentukan pula jenis-jenis sinyal yang dipakai mengatur pertukaran

informasi antara DTE dan DCE, semuanya terdapat 24 jenis sinyal tapi yang umum dipakai hanyalah 9 jenis sinyal. Konektor yang dipakai pun ditentukan dalam standard RS232, untuk sinyal yang lengkap dipakai konektor DB25, sedangkan konektor DB9 hanya bisa dipakai untuk 9 sinyal yang umum dipakai. Sinyal-sinyal tersebut ada yang menuju ke DCE ada pula yang berasal dari DCE. Bagi sinyal yang menuju ke DCE artinya DTE berfungsi sebagai *output* dan DCE berfungsi sebagai *input*, misalnya sinyal TD, pada sisi DTE kaki TD adalah output, dan kaki ini dihubungkan ke kaki TD pada DCE yang berfungsi sebagai input. Kebalikan sinyal TD adalah RD, sinyal ini berasal dari DCE dan dihubungkan ke kaki RD pada DTE yang berfungsi sebagai *output*. Susunan sinyal RS232 pada konektor DB9 dan konektor DB25 berlainan, susunan kaki ini dan bahasan di atas semuanya diringkas dalam Tabel 2.1.

Tabel 2.1 Jenis sinyal RS232 yang umum dipakai

Nama Sinyal	Arah Sinyal	Nomor Kaki Konektor	
		DB9	DB25
Signal Common	-	5	7
Transmitted Data(TD)	ke DCE	3	2
Received Data(RD)	Dari DCE	2	3
Request to Send (RTS)	ke DCE	7	4
Clear to Send (CTS)	Dari DCE	8	5
DCE Ready (DSR)	Dari DCE	6	6
DTE Ready (DTR)	ke DCE	4	20
Ring Indicator (RI)	Dari DCE	9	22
Data Carrier Detect (DCD)	Dari DCE	1	8

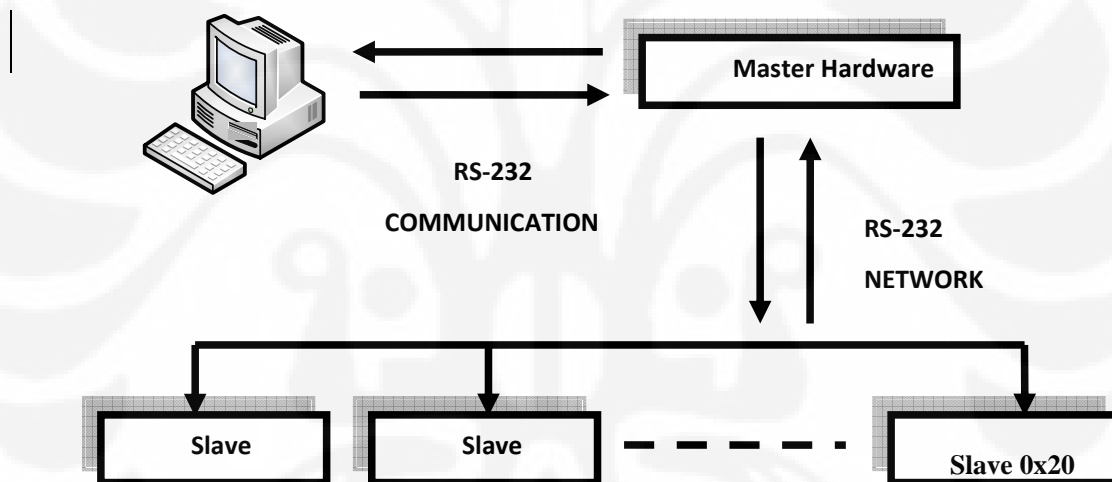
BAB III

PERANGKAT KERAS DAN PERANGKAT LUNAK

SISTEM

3.1. PERANCANGAN PERANGKAT KERAS SISTEM

Perancangan perangkat keras dilakukan untuk setiap modul yang terdapat di dalam sistem ini. Modul yang ada di dalam sistem ini adalah *master*, *slave*, pengkondisi sinyal, dan sistem *power*. Modul-modul yang telah dirancang kemudian akan disusun sebagai suatu sistem seperti pada gambar 3.1.



Gambar 3.1 Blok diagram sistem

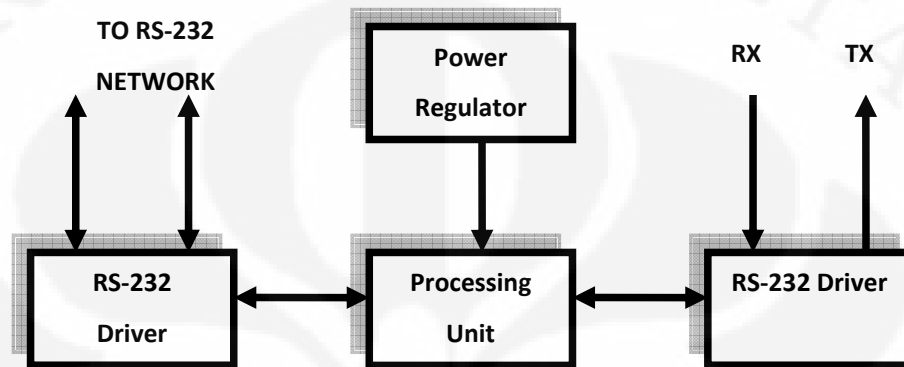
3.1.1 Perancangan Perangkat Master

Perangkat master pada sistem ini bertugas untuk mengumpulkan data dari tiap-tiap *slave* dan mengirimkan data yang diterimanya kepada perangkat lain. Spesifikasi dari perangkat *master* tersebut adalah sebagai berikut :

- Mempunyai suatu *Processing Unit* untuk memproses data. Dalam perancangan ini digunakan mikrokontroler ATMEGA 162.
- Mempunyai saluran komunikasi antara *Master – Slave*. Dalam Perancangan ini digunakan standar komunikasi RS-232 dengan kecepatan 9600 bps.

- Mempunyai saluran komunikasi dengan perangkat lain sebagai pengambil data pada *master*. Dalam perancangan ini digunakan komunikasi standar RS-232 dengan kecepatan 9600 bps.

Rancangan sistem perangkat master yang telah ditentukan spesifikasinya dapat digambarkan sebagai suatu blok diagram seperti pada gambar 3.2.



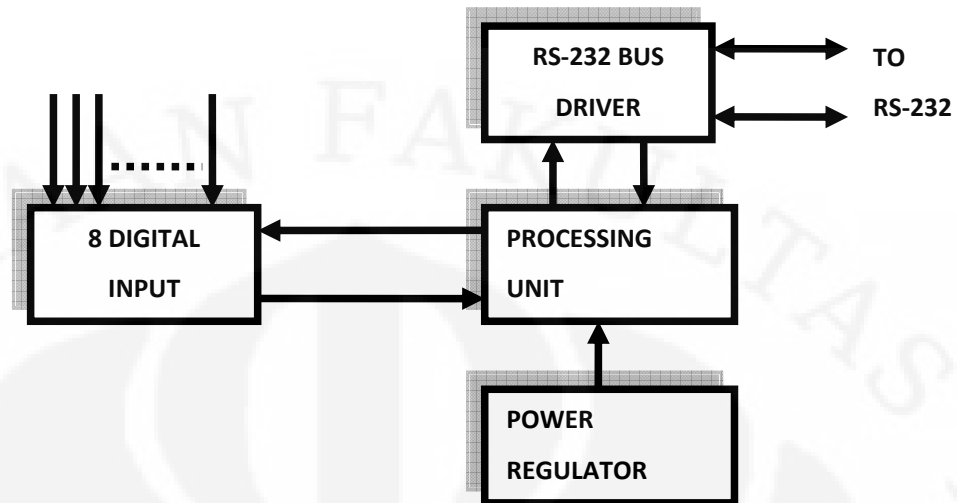
Gambar 3.2 Blok Diagram modul master hardware

3.1.2 Perancangan Perangkat Slave

Perangkat slave merupakan suatu pengumpul data dari suatu parameter yang ingin diketahui besarnya. Perangkat slave dapat membaca data dari 8 input masukan. Spesifikasi dari perangkat slave ini adalah sebagai berikut :

- Mempunyai suatu processing unit untuk memproses data. Dalam perancangan ini digunakan microcontroller ATMEGA 16.
- Mempunyai saluran komunikasi antara Master – Slave. Dalam perancangan ini digunakan standar komunikasi RS-2232 dengan kecepatan 9600 bps.
- Mempunyai input sebanyak 8 kanal. Dalam perancangan ini digunakan ADC yang sudah terdapat di dalam mikrokontroller.
- ADC yang digunakan memiliki ketelitian 10 bit, sehingga terdapat 1024 tingkatan.
- Tegangan referensi maksimum dari ADC adalah tegangan VCC yang dimiliki oleh mikrokontroller.
- Clock ADC yang digunakan adalah 172.800 Hz.

Rancangan sistem tersebut dapat digambarkan sebagai suatu blok diagram seperti pada gambar 3.3.



Gambar 3.3 Blok diagram modul slave hardware.

3.1.3 Modul Pengkondisi Sinyal

Keluaran data dari *solar cell* adalah berupa tegangan yang berada di kisaran 0 – 50 V. Dan arus maksimum yang dapat dihasilkan adalah sebesar 1 A. Agar dapat dibaca oleh *Analog-to-Digital Converter* yang ada pada mikrokontroler, maka tegangan keluaran yang dihasilkan oleh *solar cell* harus diturunkan terlebih dahulu, sedangkan arusnya harus dikonversi menjadi tegangan yang berkisar antara 0 – 5 V.

Untuk menurunkan tegangan, digunakan suatu rangkaian *voltage divider*. Agar sesuai dengan tegangan mikrokontroler maka tegangan diturunkan sampai seperlima tegangan aslinya. Suatu rangkaian *voltage divider* adalah rangkaian yang terdiri atas resistor yang disusun secara seri.

Agar arus yang dihasilkan oleh *solar cell* dapat terbaca, maka arus tersebut harus dilewatkan pada suatu hambatan, untuk kemudian diukur tegangan yang ada pada ujung-ujung hambatan tersebut. Berdasarkan rumus $V = I \times R$, maka kita dapat mendapatkan arus yang lewat sebesar $I = V/R$. Tegangan yang didapatkan tersebut kemudian diturunkan terlebih dahulu sebelum dapat dibaca oleh ADC pada mikrokontroler.

Tegangan-tegangan yang telah diturunkan, sebelum dimasukkan ke dalam mikrokontroler harus kita kondisikan terlebih dahulu agar arus yang memasuki mikrokontroler merupakan arus yang lemah. Untuk itu diperlukan suatu rangkaian pengkondisi. Rangkaian ini terdiri atas suatu *differential amplifier*. Pada rangkaian ini referensi yang digunakan oleh OpAmp adalah sebesar +15 V dan -15 V. Keluaran dari

OpAmp ini kemudian dapat langsung diberikan ke mikrokontroler karena telah sesuai dengan arus-arus yang keluar dari suatu IC TTL.

3.1.4 Modul Power Supply

Tegangan- tegangan yang digunakan di setiap modul berbeda-beda besarnya terutama pada modul pengkondisi sinyal. Modul ini menghasilkan tegangan +5, -6, dan +6 V. Untuk menghasilkan tegangan tersebut dapat digunakan IC regulator yaitu LM 7805 dan 7806 untuk tegangan +5 dan +6, sedangkan untuk tegangan -6 digunakan IC LM7906.

3.2 PERANCANGAN PERANGKAT LUNAK SISTEM

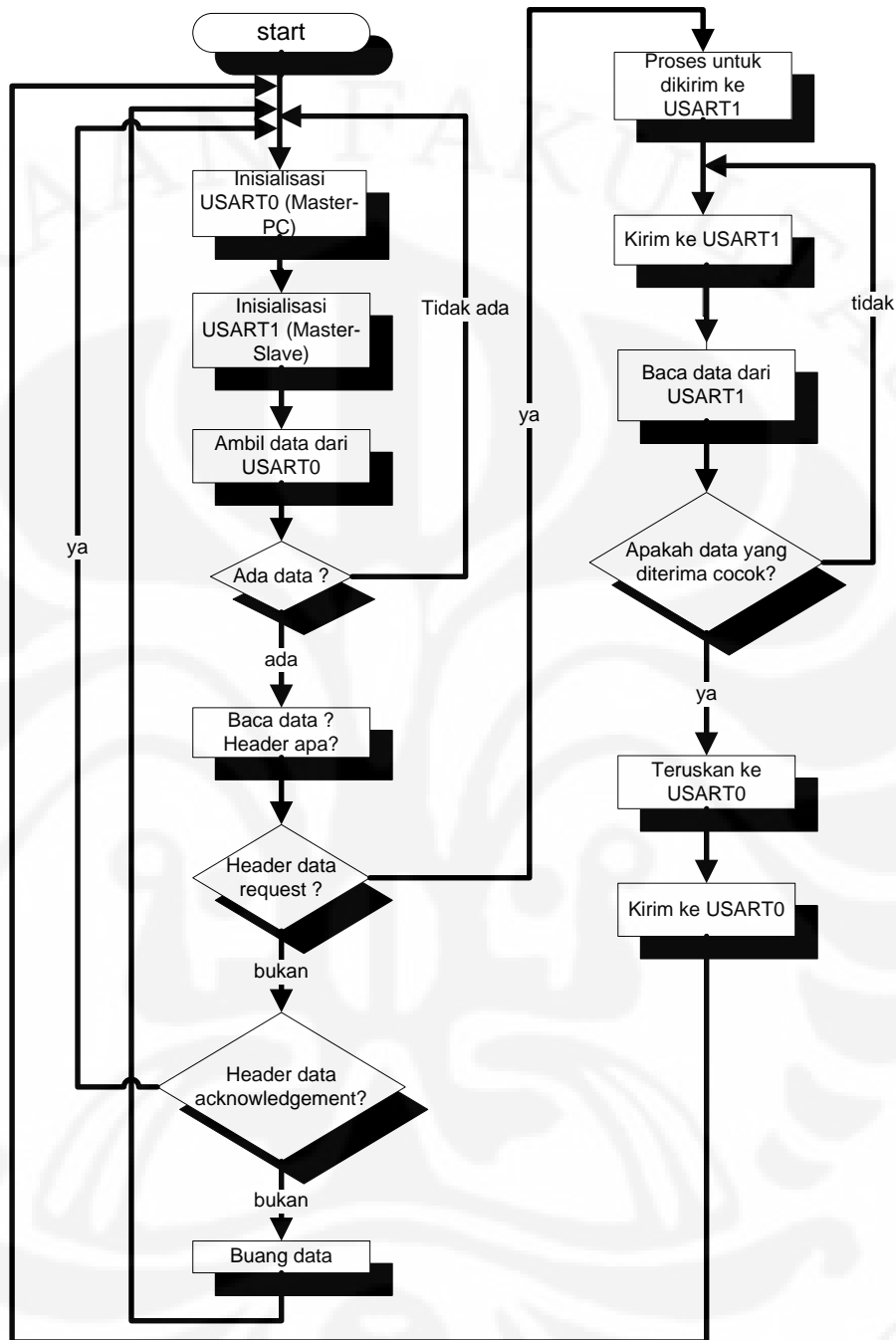
Pada perangkat lunak dari tiap modul didasarkan pada fungsi dari masing-masing modul. Untuk membuat perangkat lunak mikrokontroler digunakan *compiler* CodeVisionAVR yang menggunakan bahasa C. Sedangkan *software* untuk membaca data dari modul master digunakan aplikasi berbasis Java. Aplikasi ini memanfaatkan API dari java yaitu *javacomm* versi 2.0. *Database* yang digunakan untuk penyimpanan data ini adalah *database* MySQL dan untuk tampilan data digunakan suatu halaman *web* yang dibangun menggunakan bahasa PHP. Untuk itu diperlukan suatu aplikasi *webserver* yang menyediakan PHP dan MySQL. Pada sistem ini digunakan *software* XAMPP yang merupakan suatu aplikasi *open source*.

3.2.1 Perancangan Firmware Perangkat Master

Pada perangkat *master*, *firmware* dirancang untuk dapat menjembatani PC dan perangkat-perangkat *slave*. Sehingga master akan mengumpulkan data dan mengirimkan pada PC. Pada *firmware* master fungsi-fungsi yang dilakukan adalah:

- Mampu berkomunikasi dengan PC dengan kecepatan 9600 bps dengan menggunakan standar komunikasi RS-232.
- Mampu berkomunikasi dengan perangkat-perangkat *slave* dengan kecepatan 9600 bps dengan standar komunikasi RS-232.

Pada perangkat ini digunakan metode *interrupt* untuk memproses setiap data yang datang. Berikut adalah diagram alir dari proses yang dilakukan pada perangkat master.



Gambar 3.4 Diagram alir proses perangkat master

3.2.2 Perancangan *Firmware* Perangkat *Slave*

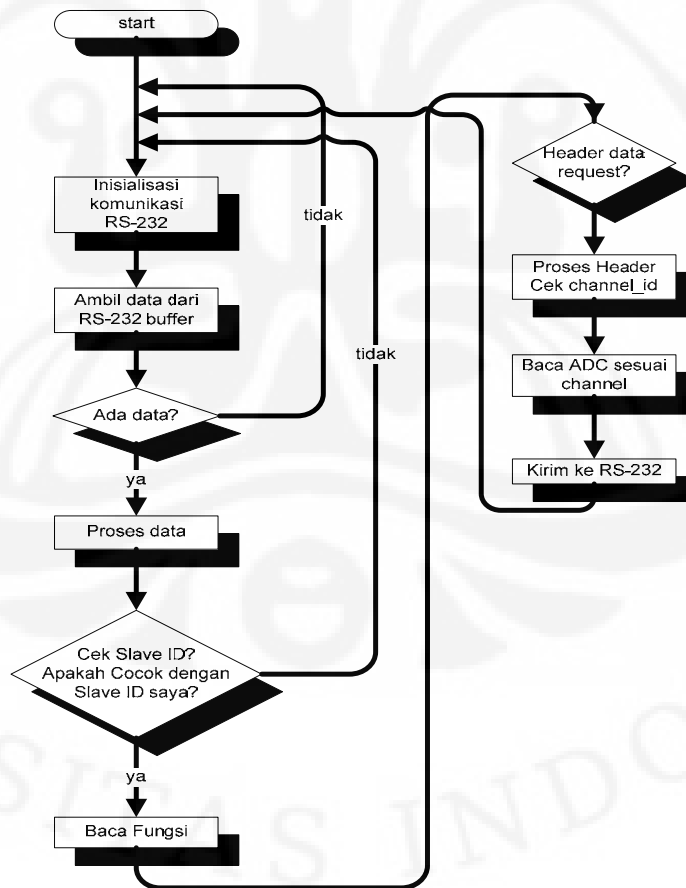
Pada perangkat *slave*, *firmware* dirancang untuk dapat membaca data *analog* dari alat yang sedang diakuisisi datanya. Setiap *slave* yang terkoneksi memiliki alamat yang disimpan di dalam variabel *slave id*. Jumlah maksimum *slave* yang dapat

dihubungkan ke dalam sistem sebanyak 31 buah (32 buah termasuk master). Nilai slave id ini merupakan suatu bilangan *hexadecimal* yang bernilai 0x01 sampai 0x20. Pada skripsi ini digunakan standar komunikasi RS-232, untuk menguji algoritma yang digunakan. Namun untuk dapat berkomunikasi secara *multipoint* dibutuhkan suatu standar komunikasi RS-485 yang mendukung komunikasi *multipoint*, lebih kebal terhadap *noise* dan memiliki jarak tempuh yang lebih jauh dibandingkan RS-232. Panjang kabel maksimum dari komunikasi RS-485 dapat mencapai 4000 *feet*. Dan kecepatan maksimum yang dapat dicapai 10 Mbps.

Pada *firmware* dari *slave* fungsi-fungsi yang dilakukan adalah :

- Mampu berkomunikasi dengan perangkat master dengan kecepatan 9600 bps menggunakan standar komunikasi RS-232.
- Mampu membaca data *analog* yang diberikan kepadanya.

Pada perangkat ini juga digunakan metode *interrupt* untuk memproses setiap *frame* yang datang. Berikut ini adalah diagram alir dari proses yang dilakukan pada perangkat *slave*.

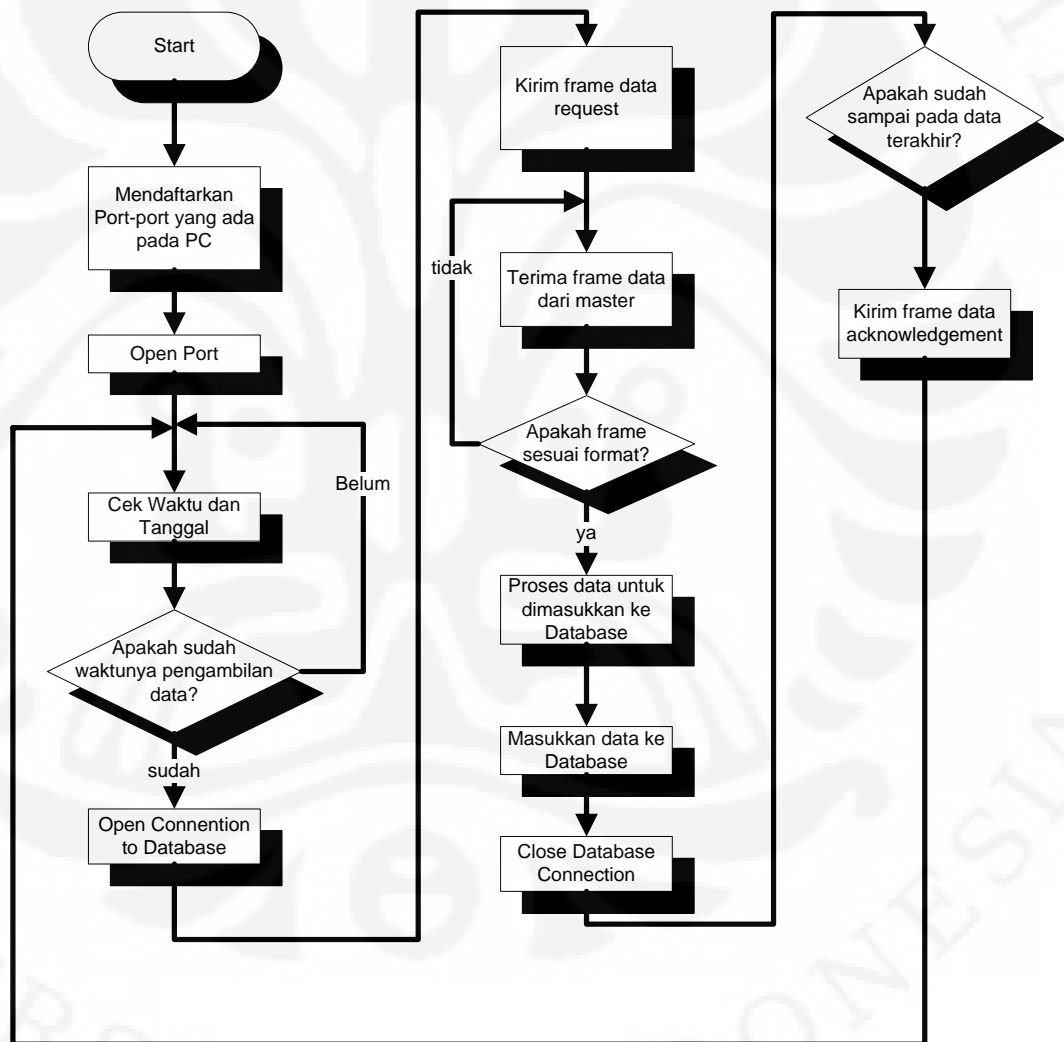


Gambar 3.5 Diagram alir proses perangkat *slave*

3.2.3 Perancangan Software Pembaca Data Serial

Agar data serial yang dikirimkan oleh mikrokontroler dapat dibaca oleh PC, maka dibutuhkan suatu program yang dapat mengubah data serial menjadi data yang dapat dibaca oleh PC. Pada sistem ini digunakan program yang dibangun dari bahasa Java dengan memanfaatkan salah satu API dari Java, yaitu Javacomm2.0.

Data Serial yang diterima kemudian akan diproses untuk dapat dimasukkan ke dalam *database*. Setelah dimasukkan ke dalam *database*, program akan memutuskan hubungan dengan *database*, agar program siap untuk menerima data kembali. Berikut ini adalah diagram alir dari proses yang terjadi pada program.



Gambar 3.6 Diagram Alir proses program penangkap data serial

3.2.4 Perancangan Web Server dan Database Server

Data-data yang telah terkumpul pada *database* dapat kita lihat dengan menggunakan website. Untuk itu diperlukan suatu *Web Server* dan juga *Database Server*. Untuk itu digunakan suatu program *open source* yang bernama XAMPPlite. Pada program ini tersedia fitur-fitur sebagai berikut:

- Apache 2.2.4 (*Web Server*)
- MySQL 5.0.45 (*Database*)
- PHP 5.2.3 (Bahasa Pemrograman Web yang didukung)
- XAMPPlite Control Version 2.5
- XAMPPlite Security 1.0
- SQLite 2.8.15
- OpenSSL 0.9.8e
- phpMyAdmin 2.10.2
- Webalizer 2.01-10
- Zend Optimizer 3.3.0
- eAccelerator 0.9.5.1

3.2.5 Perancangan Halaman Web Site

Pada halaman ini akan ditampilkan data-data elektrik dari *solar cell* yang sedang dipantau. Data-data elektrik tersebut akan ditampilkan dalam bentuk grafik.

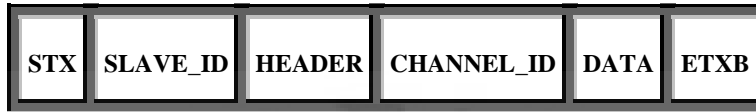
Untuk menampilkannya dalam bentuk grafik digunakan suatu *source* program berbasis PHP yang bernama JGraph-1.22. Program ini mempunyai kemampuan untuk mempermudah dalam membuat grafik.

3.3. PERANCANGAN PROTOKOL SISTEM

Agar komunikasi antar modul di dalam sistem dapat berlangsung dengan baik, diperlukan suatu aturan-aturan di dalam berkomunikasi (protokol). Protokol komunikasi ini terdiri dari *frame-frame* data yang berisi header data *request*, data, dan data *acknowledgement*.

3.3.1 Frame Data dari Master Ke Slave

Frame ini dikirimkan oleh *master* kepada *slave* secara berulang dan dengan menggunakan teknik *broadcast*. Sehingga ketika alamat yang dituju pada *frame* sesuai dengan alamat dari *slave*, maka *slave* tersebut akan merespon. Bentuk dari *frame* yang dikirimkan dari *master* ke *slave* seperti pada Gambar 3.6 dan penjelasannya terdapat pada Tabel 3.1



Gambar 3.6 Frame Data untuk *master* ke *slave*

Tabel 3.1 Penjelasan *field* data dari *frame master* ke *slave*

	Deskripsi	Ukuran
STX	<i>Start of Text</i> Berupa karakter dengan kode 0x02	1 byte
SLAVE_ID	<i>Terminal/slave identification</i> (0x00) Identifikasi terminal yang menghasilkan/akan menerima data	1 byte
Header	Berupa karakter dengan kode 0xA0	1 byte
Channel_ID	<i>ADC Channel Identification</i> (0x00) Identifikasi kanal ADC yang akan diambil datanya	1 byte
Data	0x000000 (kosong)	3 byte
ETXB	<i>End of Text</i> Berupa karakter dengan kode 0xFF	1 byte

3.3.2 *Frame Data dari Slave Ke Master*

Frame ini dikirimkan oleh *slave* kepada *master*, jika alamat yang terdapat pada *frame* yang dikirim oleh *master* sesuai dengan alamat dari *slave*. *Frame* yang dikirimkan oleh *slave* merupakan data pembacaan *analog-to-digital converter* yang terdapat pada *slave*. Bentuk dari *frame* yang dikirimkan dari *slave* ke *master* sama dengan bentuk dari *frame* yang dikirimkan dari *master* ke *slave*, sedangkan penjelasannya terdapat pada Tabel 3.2.

Tabel 3.2 Penjelasan *field* data *frame* dari *slave* ke *master*

	Deskripsi	Ukuran
STX	<i>Start of Text</i> Berupa karakter dengan kode 0x02	1 byte
SLAVE_ID	<i>Terminal/slave identification</i> (0x00 sampai 0x20) Identifikasi terminal yang menghasilkan/akan menerima data	1 byte
Header	Berupa karakter dengan kode 0xC0	1 byte
Channel_ID	<i>ADC Channel Identification</i> (0x00 sampai 0x07) Identifikasi kanal ADC yang akan diambil datanya	1 byte
Data	Data ADC (0x000000 sampai 0x3FFF00)	3 byte
ETXB	<i>End of Text</i> Berupa karakter dengan kode 0xFF	1 byte

3.3.3 Frame Data dari PC Ke Master

Terdapat dua jenis *frame* yang dikirimkan dari PC ke *master* yaitu *frame data request* dan *frame data acknowledgement*. *Frame data request* dikirimkan satu kali kepada *master*, untuk menginformasikan kepada *master* bahwa *master* diharuskan mengirimkan data-data semua *channel* dari *slave-slave* yang terkoneksi kepadanya.

Frame data acknowledgement akan dikirimkan oleh PC jika proses penerimaan data dari *master* ke PC telah selesai. Frame ini juga dikirimkan hanya satu kali kepada *master*, untuk menginformasikan kepada *master* bahwa data yang telah dikirim oleh *master* telah berhasil diterima dengan baik. Jika *master* belum menerima header ini, maka *master* akan terus menerus mengirimkan data-data yang telah berhasil diperolehnya.

Bentuk dari *frame* yang dikirimkan memiliki format yang sama dengan *frame* dari *slave* ke *master*, sedangkan penjelasannya ada pada Tabel 3.3.

Tabel 3.3 Penjelasan field data dari PC ke Master

	Deskripsi	Ukuran
STX	<i>Start of Text</i> Berupa karakter dengan kode 0x02	1 byte
SLAVE_ID	<i>Terminal/slave identification</i> (0x00) Identifikasi terminal yang menghasilkan/akan menerima data	1 byte
Header	Berupa karakter dengan kode <ul style="list-style-type: none">➤ 0xA1 (data request)➤ 0xD1 (data acknowledgement)	1 byte
Channel_ID	<i>ADC Channel Identification</i> (0x00) Identifikasi kanal ADC yang akan diambil datanya	1 byte
Data	0x000000	3 byte
ETXB	<i>End of Text</i> (Berupa karakter dengan kode 0xFF)	1 byte

3.3.4 Frame Data dari Master Ke PC

Frame yang dikirimkan dari Master ke PC hanya satu jenis *frame*, yaitu *frame data*. *Frame data* dikirimkan berkali-kali sebagai respon dari *frame data request*. Frame ini memuat data-data dari *analog-to-digital converter* yang akan dimasukkan ke dalam *database*. Bentuk dari *frame* yang dikirimkan dari *master* ke PC sama dengan bentuk dari *frame* yang dikirimkan dari PC ke *master*, sedangkan penjelasannya terdapat pada Tabel 3.4

Tabel 3.4 Penjelasan field data dari Master ke PC

	Deskripsi	Ukuran
STX	<i>Start of Text</i> Berupa karakter dengan kode 0x02	1 byte
SLAVE_ID	<i>Terminal/slave identification</i> (0x00 sampai 0x20) Identifikasi terminal yang menghasilkan/akan menerima data	1 byte
Header	Berupa karakter dengan kode 0xC1	1 byte
Channel_ID	<i>ADC Channel Identification</i> (0x00 sampai 0x07) Identifikasi kanal ADC yang akan diambil datanya	1 byte
Data	Data ADC (0x000000 sampai 0x3FFF00)	3 byte
ETXB	<i>End of Text</i> (Berupa karakter dengan kode 0xFF)	1 byte

BAB IV

PEMBAHASAN DAN ANALISIS

4.1 METODE PENGUJIAN

Sistem yang telah dirancang dan direalisasikan selanjutnya akan dilakukan, pengujian, kemudian hasil pengujian akan digunakan sebagai dasar pembahasan dan analisis. Berdasarkan sistem yang telah dirancang maka pengujian yang dilakukan dapat dikelompokkan sebagai berikut :

1. Pengujian Perangkat Keras
2. Pengujian Protokol Komunikasi
3. Pengujian Perangkat Lunak dan Sistem Keseluruhan

Pengujian dilakukan dengan menggunakan perangkat keras atau perangkat lunak yang umum. Perangkat umum yang dimaksud adalah perangkat yang tidak berfungsi khusus seperti sistem yang telah dirancang. Kemudian setelah sistem yang dirancang bekerja dengan baik maka pengujian yang terintegrasi akan dilakukan untuk seluruh sistem. Perangkat yang digunakan untuk pengujian adalah :

1. Perangkat Lunak CodeVisionAVR, merupakan *compiler* yang digunakan untuk memprogram mikrokontroler yang digunakan. Pada *compiler* tersebut disediakan *tools* berupa terminal untuk memantau komunikasi data yang terjadi pada port-port yang ada pada PC. Tools tersebut dapat menampilkan data berupa hexadecimal dan ASCII.
2. Perangkat keras *master* dan *slave* yang telah dirancang dan telah direalisasikan.

4.2 PENGUJIAN PERANGKAT KERAS

Pengujian pada perangkat keras dilakukan pada perangkat *master* dan perangkat *slave* yang digunakan. Hasil dari pengujian diharapkan dapat sesuai dengan yang rancangan yang telah dibuat.

4.2.1 Pengujian Perangkat Master

Perangkat master diharapkan mampu melakukan komunikasi data melalui jalur USART0 dan USART1. Sehingga dapat menjadi penghubung antara PC dengan *slave* pada sistem perangkat keras. Dengan demikian pengujian yang akan dilakukan adalah pengujian terhadap jalur USART0 dan USART1.

4.2.1.1 Pengujian Komunikasi Data USART0

Pengujian dilakukan dengan cara mengirimkan data dari PC menuju USART0 dari perangkat *master*. Setiap data yang dikirimkan akan dikirimkan kembali ke PC untuk dibaca melalui *terminal* pada CodeVisionAVR. Kemudian data yang diterima akan dicocokkan dengan data yang dikirimkan.

Data yang dikirimkan berupa *frame* dari PC menuju master atau *frame* data dari *master* menuju PC. Data yang dikirimkan telah diatur sebelumnya karena pengujian hanya dilakukan pada perangkat master. Pengujian ini dilakukan untuk mengetahui kehandalan proses transmisi data.

Pengujian dilakukan dengan spesifikasi sebagai berikut :

Baudrate : 9600 bps.
Data bits : 8
Stop bits : 1
Parity : None

Tabel 4.1 Hasil pengujian pengiriman data antara PC ke master 9600 bps

Header	Data Sent			Data Receive		
data request	0x00	0x00	0x00	0x00	0x00	0x00
ack data	0x00	0x00	0x00	0x00	0x00	0x00
data	0x00	0x00 - 0x03	0x00 - 0xFF	0x00	0x00 - 0x03	0x00 - 0xFF

Berdasarkan data hasil pengujian pada tabel 4.1, menunjukkan bahwa dengan menggunakan spesifikasi diatas data dari PC dapat diteruskan kepada perangkat *master*, karena data yang diminta sesuai dengan yang diterima sesuai dengan yang dikirimkan. Untuk meningkatkan respon dari sistem maka dicoba untuk meningkatkan kecepatan *transfer* data dengan meningkatkan *baudrate*. *Baudrate* akan di-*setting* pada kecepatan 115200 bps, tabel hasil pengujian terdapat pada tabel 4.2.

Pengujian selanjutnya akan dilakukan dengan spesifikasi sebagai berikut :

Baudrate : 115200 bps
Data bits : 8
Stop bits : 1
Parity : None

Tabel 4.2 Hasil pengujian pengiriman data antara PC ke master 115200 bps

Header	Data Sent			Data Receive		
data request	0x00	0x00	0x00	0x00	0x00	0x00
ack data	0x00	0x00	0x00	0x00	0x00	0x00
data	0x00	0x00 - 0x03	0x00 - 0xFF	0x00	0x00 - 0x03	0x00 - 0xFF

Berdasarkan data hasil pengujian pada tabel 4.2, pada kecepatan transmisi data yang lebih tinggi, perangkat master masih mampu mengirimkan data yang diterima dengan benar. Dengan demikian saluran komunikasi antara master dengan PC dapat menggunakan kecepatan yang lebih tinggi untuk mempercepat respon dari sistem.

4.2.1.2 Pengujian Komunikasi Data USART1

Pada pengujian ini dilakukan dengan cara mengirimkan data dari PC menuju *master* untuk kemudian diteruskan menuju *slave*. Kemudian *slave* akan mengirimkan kembali data tersebut menuju *master*. Data tersebut kemudian diteruskan ke PC dan dipantau melalui terminal yang ada pada aplikasi CodeVisionAVR.

Berikut ini adalah spesifikasi saluran komunikasi yang digunakan :

Baudrate : 9600 bps.

Data bits : 8

Stop bits : 1

Parity : None

Tabel 4.3 Hasil Pengujian pengiriman data antara master ke slave

Header	Data Sent			Data Receive		
data request	0x00	0x00	0x00	0x00	0x00	0x00
ack data	0x00	0x00	0x00	0x00	0x00	0x00
data	0x00	0x00 - 0x03	0x00 - 0xFF	0x00	0x00 - 0x03	0x00 - 0xFF

Berdasarkan hasil pengujian yang dilakukan pada tabel 4.3, saluran komunikasi USART1 bekerja dengan baik. Hal tersebut menunjukkan PC dapat berkomunikasi dengan *slave* melalui *master*, karena data yang diminta sesuai dengan yang dikirimkan.

4.2.2 Pengujian Perangkat Slave

Pada perangkat *slave* hanya terdapat satu saluran komunikasi yang dapat dipakai. Saluran tersebut menggunakan protokol komunikasi RS-232. Pengujian yang dilakukan untuk perangkat *slave* meliputi pengujian saluran komunikasi dan pengujian kanal ADC.

Hasil pengujian saluran komunikasi pada perangkat *slave* dapat kita lihat hasilnya pada pengujian saluran komunikasi USART1 pada perangkat *master*. Berdasarkan pengujian pada perangkat *master*, USART1 *master* berjalan dengan baik dan mampu berkomunikasi dengan perangkat *slave*. Hal ini menunjukkan saluran komunikasi yang terdapat pada *slave* bekerja dengan baik. Spesifikasi saluran komunikasi yang digunakan dalam pengujian adalah sebagai berikut :

Baudrate : 9600 bps.
Data bits : 8
Stop bits : 1
Parity : None

Pengujian terhadap kanal ADC dilakukan dengan memberikan suatu tegangan kepada ADC dari mikrokontroler untuk dibaca tegangannya. Pengujian dilakukan dengan melakukan pembacaan pada potensiometer yang digunakan sebagai pemberi tegangan. Tegangan yang diberikan berupa tegangan maksimum (VCC) dan tegangan minimum (GND). Berdasarkan pengujian nilai dari ADC untuk tegangan VCC menghasilkan nilai 0x03FF, sedangkan untuk tegangan GND menghasilkan nilai 0x0000. Hal ini sesuai dengan teori dimana nilai maksimum tegangan yang dapat dibaca adalah 0x03FF. Berikut ini adalah spesifikasi ADC yang digunakan :

ADC Clock frequency : 172,800 kHz
ADC Voltage Reference : AVCC pin



Gambar 4.1 Perangkat master dan slave beserta perangkat pengujian

4.2.3 Pengujian Perangkat Pengkondisi Sinyal

Perangkat pengkondisi sinyal bertujuan untuk mengkondisikan sinyal yang masuk menjadi tegangan berkisar antara 0 sampai 5 V. Keluaran dari pengkondisi sinyal ini akan diberikan ke ADC untuk selanjutnya dikonversi menjadi nilai tertentu.

Pengujian yang dilakukan pada perangkat ini dengan memberikan suatu tegangan sebagai masukannya, kemudian keluaran dari modul ini diukur untuk dicocokkan dengan nilai yang dimasukkan. Jika terdapat selisih antara masukan dan keluaran, selisih tersebut harus tidak boleh melebihi ketelitian dari ADC yang bernilai 0,0048828125 V.

Berdasarkan percobaan yang dilakukan, jika perangkat diberi input tegangan sebesar 4.87 V didapatkan keluaran 4.92 V. Hal ini berarti terdapat selisih 0.05 . Selisih ini bernilai 10 kali dari ketelitian ADC.

4.2.4 Pengujian Perangkat *Power Supply*

Untuk menguji perangkat *power supply*, kita akan lihat tegangan keluaran yang dihasilkan dari masing-masing *port*. *Power supply* ini didesain untuk menghasilkan tegangan 15, -15, 5 dan -5.

Berdasarkan percobaan yang dilakukan, ketika perangkat dinyalakan didapatkan hasil sebagai berikut :

- Untuk *power* 5 V didapatkan tegangan keluaran sebesar 4.87 V.
- Untuk *power* -5 V didapatkan tegangan keluaran sebesar -5.02 V.
- Untuk *power* -15 V didapatkan tegangan keluaran sebesar -15.09 V.
- Untuk *power* 15 V didapatkan hasil keluaran sebesar 14.92 V.

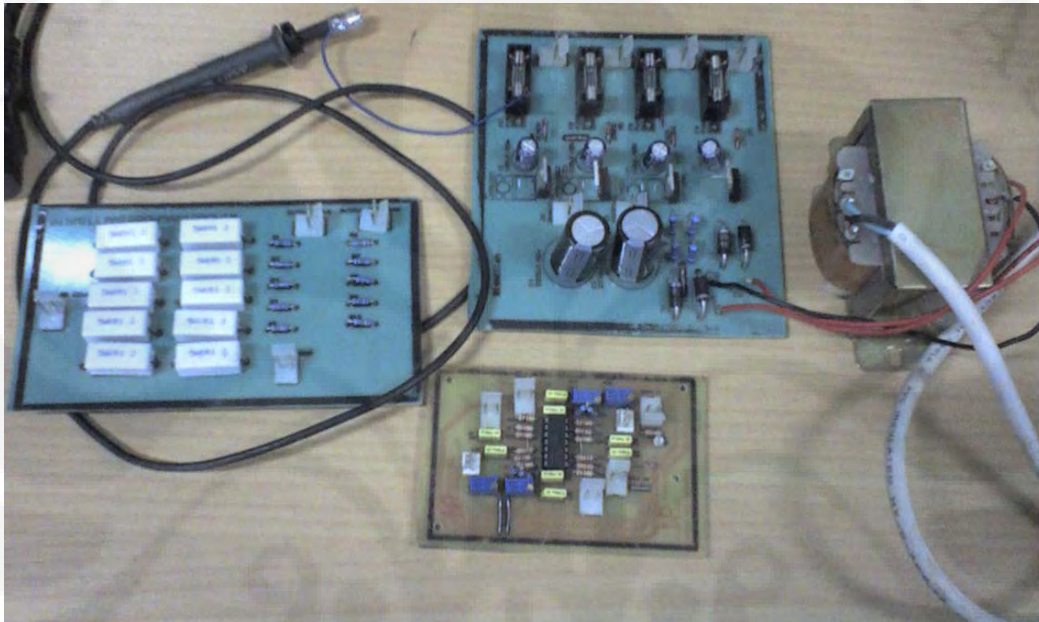
Berdasarkan datasheet dari 7805, IC tersebut dapat menghasilkan tegangan antara 4.8 V hingga 5.2 V. Keluaran yang dihasilkan dari modul *power supply* masih berada dalam rentang keluaran dari IC tersebut, bisa dikatakan IC tersebut bekerja dengan baik.

Untuk 7815, keluaran yang seharusnya dikeluarkan berada di dalam rentang 14.4 sampai 15.6 V. Keluaran yang dihasilkan dari modul *power supply* berada di dalam rentang keluaran IC tersebut, bisa dikatakan modul tersebut bekerja dengan baik.

Untuk 7905, keluaran yang seharusnya dikeluarkan berada di dalam rentang -4.75 sampai -5.2 V. Keluaran yang dihasilkan dari modul *power supply* berada di

dalam rentang keluaran IC tersebut, bisa dikatakan modul tersebut bekerja dengan baik.

Untuk 7915, keluaran yang seharusnya dikeluarkan berada di dalam rentang -14.4 sampai -15.6 V. Keluaran yang dihasilkan dari modul *power supply* masih berada di dalam rentang keluaran IC tersebut, bisa dikatakan modul tersebut bekerja dengan baik.



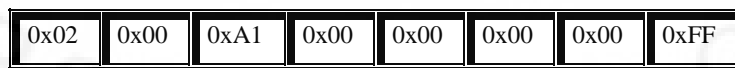
Gambar 4.2 Perangkat *Power Supply* dan Pengkondisi Sinyal

4.3 PENGUJIAN PROTOKOL KOMUNIKASI

Protokol komunikasi yang akan diuji merupakan protokol yang digunakan dalam komunikasi antar perangkat keras dan antara PC dengan perangkat *master*.

4.3.1 Pengujian Header 0xA1 (data request)

Dalam pengujian ini, terminal pada PC akan mengirimkan *frame data request* kepada *master*. Hasil dari pengujian diharapkan *master* akan mengirimkan *data request* ke jaringan RS-485. *Request* yang dikirimkan akan dikirimkan satu per satu ke masing-masing *slave*.

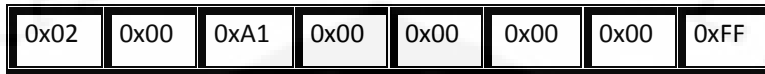


Gambar 4.3 Frame instruksi data request dari PC ke master

Respon yang dihasilkan dari pengiriman frame ini adalah *master* mengirimkan *frame data request* ke masing-masing *slave* secara berurutan satu per satu.

4.3.2 Pengujian Header 0xD1 (data acknowledgement)

Dalam pengujian ini, terminal pada PC akan mengirimkan *frame data acknowledgement* kepada *master*. Hasil dari pengujian diharapkan *master* akan menampilkan karakter 0x6A, pada terminal di PC.

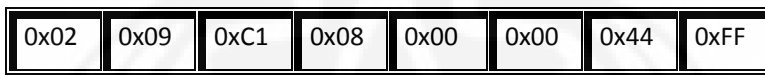


Gambar 4.4 Frame instruksi data acknowledgement dari PC ke master

Respon yang dihasilkan dari pengiriman *frame* ini adalah *master* menampilkan karakter 0x6A ke terminal pada PC, yang berarti proses pengambilan data telah selesai.

4.3.3 Pengujian Header 0xC1 (data)

Dalam pengujian ini PC akan mengirimkan *frame data request* kepada *master*, kemudian *master* akan menindaklanjuti *frame* tersebut kepada *slave* dengan mengirimkan *frame data request* ke masing-masing *slave*. *Slave* tersebut akan merespon *frame data request* dari *master* dengan mengirimkan data dari masing-masing *channel* dengan data yang telah diambilnya. *Slave* disini disimulasikan dengan PC 2 yang mengirimkan data ke master dengan data yang telah dibuat sebelumnya. Untuk kemudian data tersebut dicocokkan dengan data yang diterima oleh PC yang mengirimkan *frame data request*.



Gambar 4.5 Frame instruksi data dari master ke PC

Respon yang dihasilkan dari pengujian ini, *master* mengirimkan data yang sama dengan yang dikirimkan oleh PC 2. Gambar 4.3 merupakan *frame* yang dikirim oleh *master* ke PC setelah mendapatkan data tersebut dari PC 2. Arti dari *frame* tersebut adalah berasal dari *slave* 9 *channel* ke-8. Dengan data ADC yang berhasil dibaca bernilai 0x00 untuk MSB dan 0x44 untuk LSB.

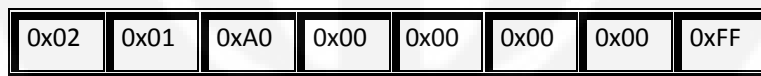
4.3.4 Pengujian Header 0xA0 dan 0xC0

Pengujian untuk *header* 0xA0 atau *data request* dilakukan dengan mengirimkan *header* 0xA0 ke perangkat *slave* melalui jalur RS-232. Pada kenyataannya nanti *header* ini akan melalui jalur RS-232 menuju saluran yang ada di

perangkat *master*, sehingga yang diuji sekarang merupakan kerja dari perangkat *slave* itu bukan saluran komunikasi.

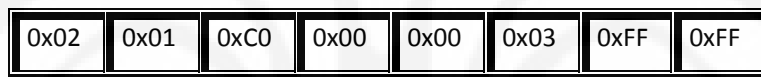
Apabila *slave* menerima *frame* ini, dia akan membalas dengan mengirimkan *frame* data 0xC0 dari kanal-kanal ADC yang dimilikinya.

Dalam pengujian ini hanya digunakan satu kanal ADC sedangkan yang lainnya hanya memberikan nilai nol. Untuk memberikan suatu nilai tegangan digunakan suatu potensiometer. Pada pengujian ini potensiometer diset memiliki hambatan maksimum, sehingga tegangan yang dihasilkan merupakan tegangan maksimum (VCC).



Gambar 4.6 Frame instruksi data request dari master ke slave.

Respon yang dihasilkan dari pengujian ini, master mengirimkan data ADC dari kanal 1 sesuai dengan tegangan yang dikonversi.



Gambar 4.7 Frame instruksi data dari slave ke master.

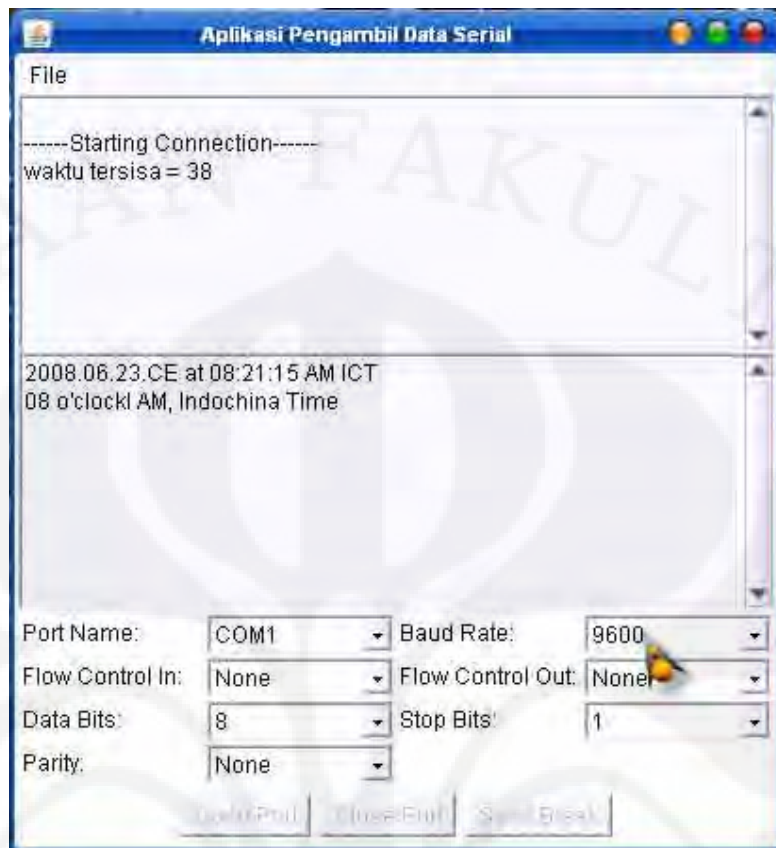
Berdasarkan frame yang diterima sebagai balasan dari frame data request, nilai dari MSB ADC sebesar 0x03 dan LSB ADC 0xFF, yang merupakan nilai maksimum dari ADC tersebut. ADC yang digunakan memiliki panjang 10 bit sehingga jika semua bitnya di set 1 maka menghasilkan nilai 0x03FF.

4.4 PENGUJIAN PERANGKAT LUNAK DAN SISTEM KESELURUHAN

Perangkat lunak yang akan diuji disini meliputi aplikasi pengambil data yang berbasiskan Java, *database* yang digunakan, dan tampilan dari *website* itu sendiri.

4.4.1 Pengujian Aplikasi Pengambil Data

Aplikasi pengambil data yang akan diuji merupakan aplikasi komunikasi serial berbasiskan java. Aplikasi ini akan melakukan hubungan dengan *port* serial yang digunakan dan *database* MySQL.



Gambar 4.8 Tampilan Aplikasi Pengambil Data Serial.

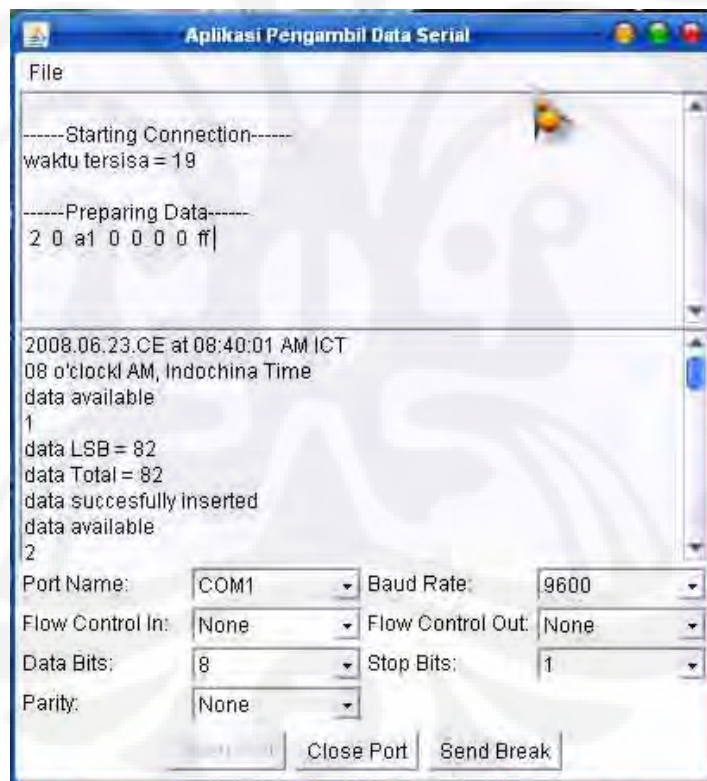
Untuk berkomunikasi dengan *port* serial, PC harus mengenali terlebih dahulu tipe *port* yang akan dibuka. Setelah PC mengetahui tipe *port* yang akan dibuka, kemudian aplikasi akan mencoba membuka *port* tersebut dan men-*setting* parameter-parameter yang diinginkan. Setelah melewati tahap ini maka *port* yang dibuka siap untuk digunakan. Sedangkan untuk menerima data aplikasi ini menggunakan suatu *event listener* yang akan memberi tanda kepada program jika terdapat data serial yang masuk. Data serial yang masuk bertipe *byte* yang bernilai 0 -255. Namun dalam pembacaannya, data ini harus dikondisikan terlebih dahulu. Karena tipe data *byte* dalam Java mempunyai nilai antara -128 sampai 127.

Setelah data serial didapatkan, proses selanjutnya yang dilakukan adalah memasukkan data tersebut ke dalam *database*. Untuk dapat berkomunikasi dengan *database* dibutuhkan suatu *database driver* yang dinamakan JDBC. Karena tipe data serial sebelumnya telah disesuaikan, maka data tersebut bisa langsung dimasukkan ke dalam *database*. Setelah memasukkan data ke dalam *database*, jangan lupa untuk

segera memutuskan hubungan, karena hal tersebut dapat menyebabkan suatu *infinite looping* yang dapat menyebabkan aplikasi menjadi *error*.

Pengujian yang dilakukan pada aplikasi ini untuk melihat apakah aplikasi ini mampu berkomunikasi dengan *port* serial dan kemudian memasukkan data tersebut ke dalam *database*. Untuk dapat melakukan pengujian diperlukan perangkat *master* yang bertugas mengirimkan data kepada aplikasi tersebut. Data pada awalnya telah dimasukkan ke dalam perangkat *master*, namun data yang dimasukkan hanyalah data yang berasal dari *slave* 0x01.

Berdasarkan pengujian yang dilakukan, aplikasi berhasil memasukkan data ke dalam *database*. Hal ini terlihat dapat diketahui dari jumlah *record* yang terdapat pada jendela phpMyAdmin yang menunjukkan pertambahan jumlah *record*, selain itu pada jendela aplikasi pengambil data serial juga ditunjukkan juga bahwa aplikasi berhasil memasukkan data ke dalam *database*.



Gambar 4.9 Tampilan Aplikasi Pengambil Data Serial ketika berhasil memasukkan data

4.4.2 Pengujian Database

Database yang digunakan disini merupakan *database* MySQL versi 5.0.45 yang berada dalam paket dari XAMPP. Pengujian bertujuan untuk mengetahui berapa kecepatan *query* yang dapat dilakukan oleh *database*. Kecepatan *query* dapat dilihat

pada halaman phpMyAdmin. Pengujian dilakukan dengan cara memasukkan data ke ke dalam database satu per satu. Kemudian akan dilihat waktu *query* dari database, apakah *database* sangat terpengaruh oleh struktur tabel.

Tabel 4.4 Hasil Percobaan waktu query dari database

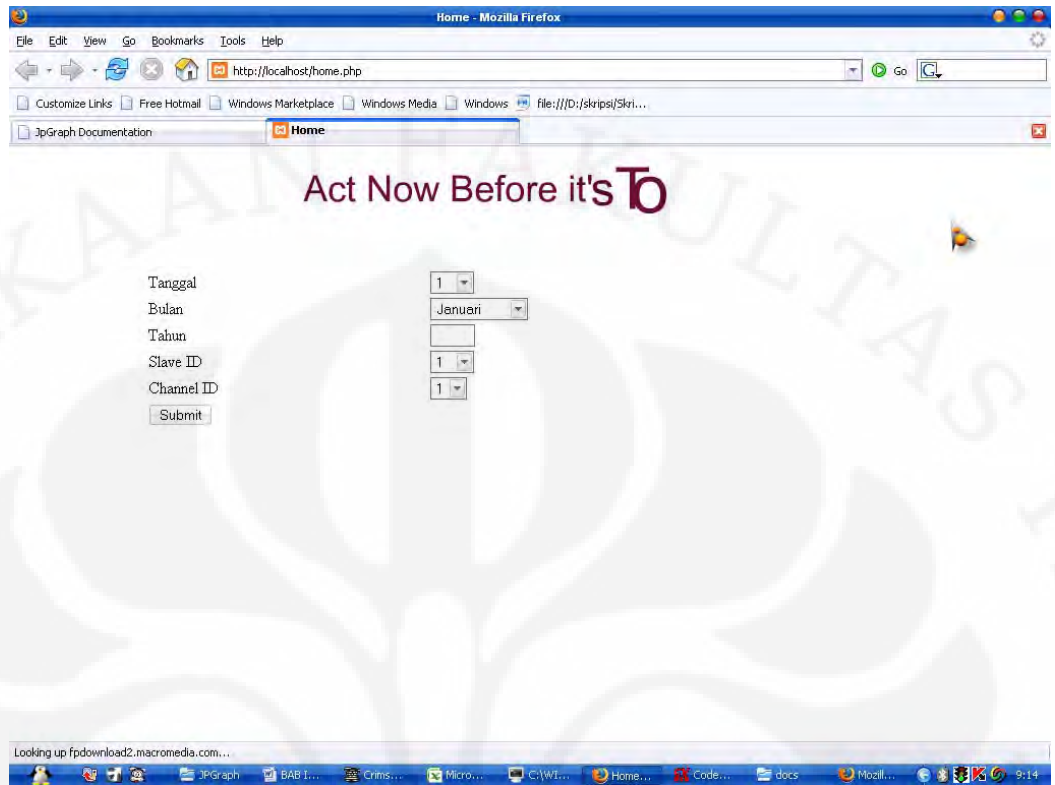
No	Jumlah record	waktu Query
1	29	0,0021 sec
2	48	0,0072 sec
3	72	0,0030 sec
4	96	0,0022 sec
5	120	0,0039 sec
6	144	0,0021 sec
7	168	0,0021 sec
8	192	0,0021 sec
9	216	0,0021 sec
10	240	0,0021 sec

Berdasarkan hasil dari pengujian, kecepatan *query* dari *database* cukup stabil yaitu dengan waktu *query* rata-rata 0.0021 detik. Waktu akses tersebut tidak berubah terlalu jauh seiring dengan bertambahnya data.

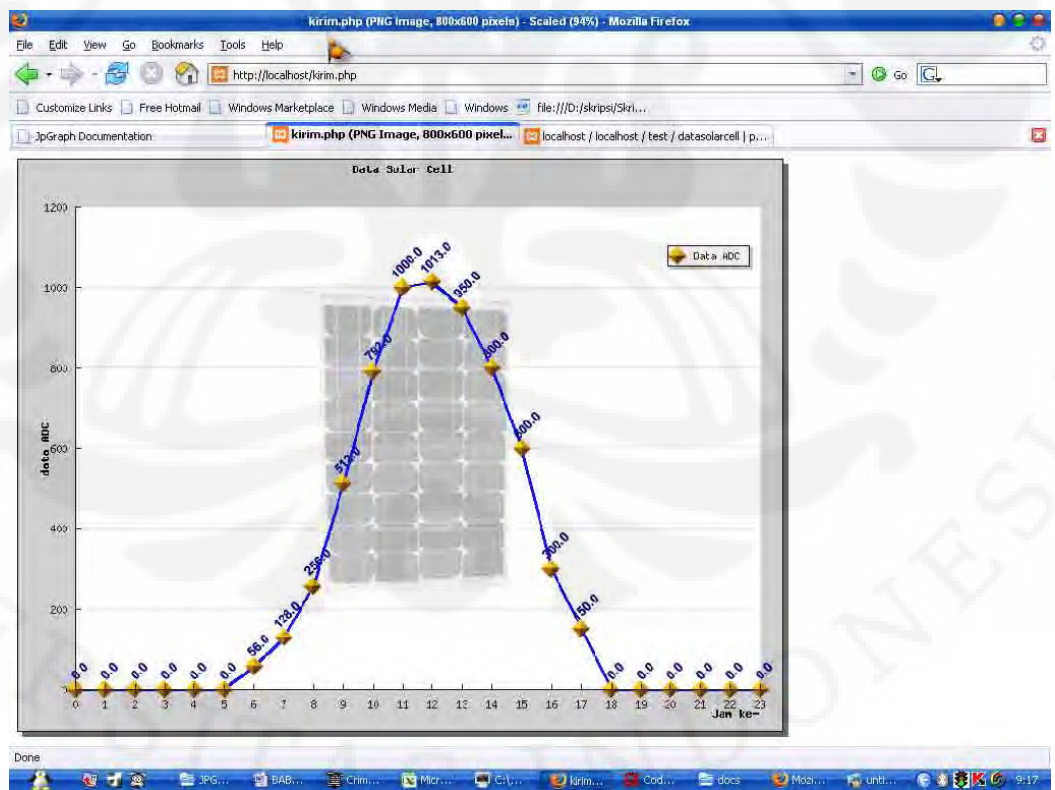
4.4.3 Pengujian Tampilan Website

Website yang digunakan untuk penampilan data dibangun dengan menggunakan bahasa PHP. Pengujian dilakukan dengan cara mencobanya pada *server localhost*. Dalam pengujian ini data tidak tersedia semuanya. Data yang tersedia hanyalah data yang berasal dari *slave 1* dan *channel 1* dari jam 1 sampai jam 24 pada hari-hari tertentu. Pengujian bertujuan untuk melihat seberapa cepat halaman dapat ditampilkan.

Berdasarkan pengujian yang dilakukan, penampilan dari database sangat lambat. Hal ini disebabkan karena data yang ditampilkan berupa gambar. Dengan ukuran yang relatif besar, Hal ini dapat dipercepat dengan merubah format gambar dan memperkecil resolusi dari gambar. Namun hal ini dapat mengurangi kenyamanan pengguna karena data yang ditampilkan memiliki ukuran yang kecil.



Gambar 4.10 Tampilan dari halaman pembuka website.



Gambar 4.11 Tampilan dari hasil pilihan data yang telah dipilih

BAB V

KESIMPULAN

Berdasarkan hasil perancangan, realisasi dan pengujian terhadap sistem, maka dapat diambil dan disimpulkan beberapa hal sebagai berikut :

1. Sistem yang dibuat masih memiliki kelemahan dari sisi kehandalan pengiriman data. Kehandalan yang dimaksud disini adalah tingkat keberhasilan pengiriman data sampai ke tujuan. Agar sistem transmisi data menjadi lebih handal dibutuhkan suatu *acknowledgement* dari pengirim maupun penerima. Namun kehandalan tersebut harus dibayar dengan meningkatnya *traffic* komunikasi yang dapat menyebabkan sistem bekerja menjadi lebih lambat.
2. Untuk menjamin data dapat diterima dengan baik oleh PC maka perangkat *master* harus memberikan jeda waktu yang cukup untuk PC memproses data sebelum menerima data kembali. Jeda waktu minimal yang dibutuhkan oleh aplikasi pembaca data serial sebelum dapat menerima data yang masuk sebesar 500 ms.
3. Waktu akses dari halaman website penampil data tergolong lama karena data yang ditampilkan berupa gambar.
4. Waktu *query* dari *database* walaupun *record*-nya terus bertambah menunjukkan waktu yang sama, dan tergolong cepat.

DAFTAR ACUAN

- [1]Shivaram H.Mysore, Rinaldo Di Giorgio (1998) “Java gets serial support with new javax.comm package” dari JavaWorld.com diakses 16 November 2007 dari JavaWorld.com
<http://www.javaworld.com/>



DAFTAR PUSTAKA

- Barnett, Richard, Larry O’Cull, Sarah Cox., *Embedded C Programming and the Atmel AVR* (New York : Thomson Learning, Inc., 2003)
- Hardiawan, Doni. “Penerapan Protokol Modbus pada Sistem Pengumpul dan Pengirim Data.” Skripsi, Program Sarjana Fakultas Teknik UI, Depok, 2005.
- Deitel, H.M., *Java How to Program, Sixth Edition* (New Jersey: Prentice Hall, 2004).
- Harris, Andi. *PHP/MySQL Programming for the Absolute Beginner* (Boston : Premier Press, 2003)
- _____, “KA78XX/KA78XXA 3-Terminal 1A Positive Voltage Regulator”, FairChild Semiconductor Datasheet, 2001
- _____, “LM79XX Linear Integrated Circuit 3-Terminal 1A Negative Voltage Regulator”, Contek Microelectronics Co.,Ltd.
- _____,”OpAmp Circuit Collection”, National Semiconductor Application Note, 1978.
- _____,”8-bit AVR Microcontroller with 16k Bytes In-System Programmable Flash ATmega162/ATmega162V Datasheet”, Atmel Corporation, 2003.
- _____,”8-bit AVR Microcontroller with 16k Bytes In-System Programmable Flash ATmega32/ATmega32L Datasheet”, Atmel Corporation, 2003.

