

0805221307

**SISTEM MONITORING TEGANGAN PLN UNTUK WILAYAH KOTA DEPOK  
BERBASIS MICROCONTROLLER**

Skripsi diajukan sebagai salah satu syarat  
untuk memperoleh gelar Sarjana Fisika

**OLEH**

**FIRDY KUSUMA NURHANDY**

**0305220313**



**DEPARTEMEN FISIKA**

**FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM**

**UNIVERSITAS INDONESIA**

**DEPOK**

**2008**

## LEMBAR PENGESAHAN

Nama : Firdy Kusuma Nurhandy  
NPM : 0305220313  
Jurusan : Fisika  
Peminatan : Instrumentasi Elektronika  
Tanggal Sidang : 16 Mei 2008  
Judul Skripsi : Sistem Monitoring Tegangan PLN Wilayah Kota Depok Berbasis  
Microcontroller

TELAH DIPERIKSA DAN DISETUJUI :

**Drs. Arif Sudarmadji, MT.**  
PEMBIMBING I

**Dr. Agus Salam**  
PEMBIMBING II

**Dr. Prawito**  
PENGUJI I

**Dr. Supriyanto**  
PENGUJI II

## KATA PENGANTAR

*Alhamdulillah robbil aalamiin, puji syukur kehadiran ALLAH SWT, Yang Maha Kuasa, Maha Pengasih lagi Maha Penyayang, karena dengan bantuan, rahmat dan hidayahNYA lah penulis dapat menyelesaikan laporan skripsi ini dengan baik Shalawat serta salam kami panjatkan kepada junjungan Nabi besar Muhammad SAW, serta para keluarga dan sahabatnya. Semoga kita semua termasuk umat yang selalu mendapatkan syafaat hingga hari akhir. Amiin ya robbal aalamiin.*

Laporan skripsi dengan judul **Sistem Monitoring Tegangan RMS PLN Untuk Wilayah Kota Depok Berbasis Microcontroller** ini dibuat sebagai salah satu syarat kelulusan guna mendapatkan Gelar Sarjana Fisika, FMIPA, Universitas Indonesia. Dengan segala daya, usaha dan tantangan serta rintangan yang kami hadapi, akhirnya kami persembahkan laporan skripsi ini. Tentunya laporan skripsi ini masih jauh dari harapan dan sempurna. Namun demikian kami tidak lupa pula mengucapkan terima kasih kepada :

1. Orang tua kami, Bapak Jassin Junus dan Mamah Muminarti, atas doa-doa, bimbingan rohani kepada kami, semoga ALLAH SWT selalu memberikan kesehatan jasmani dan rohani serta rezeki yang cukup dan halal kepada Mamah dan Bapak.
2. Drs.Arif Sudarmadji, MT. - selaku pembimbing I dan penasehat – yang telah memberikan saran dan masukan kepada kami, semoga ALLAH SWT selalu memberikan kemudahan atas segala usaha Bapak dan mendapat pahala atas kebaikan yang Bapak berikan.

3. Dr. Agus Salam - selaku pembimbing II dan penasehat – yang telah memberikan bimbingan kepada kami, semoga ALLAH SWT membalas semua kebaikan Bapak.
4. Dr. Supriyanto – selaku penasehat dan penguji II – semoga ALLAH SWT memberikan jalan kemudahan bagi usaha Bapak dan pahala atas amal ibadah Bapak.
5. Sukarno – best friend forever since 1998 – anytime anywhere Sukarno, kami akan membantu jika memang membutuhkan bantuan, semoga ALLAH SWT memberikan pahala serta kemudahan dan capaian cita-citanya.
6. Iwan Prastyawan dan (Ekstensi 05) – teman seperjuangan dan selalu membantu jika dibutuhkan - Wan Insya Allah semua amal, budi baik Iwan diberikan pahala yang berlipat ganda oleh ALLAH SWT, kalo butuh sharing, info, dll just tell me Wan.
7. Suhendro (Fisika 2006) dan Muhtar S.P. (Fisika 2002) – teman baik seperjuangan juga selalu membantu di setiap waktu – sukses buat kalian dimasa mendatang, semoga amal baik kalian mendapat balasan yang setimpal dari ALLAH SWT, buat Muhtar, jangan sungkan jika memang membutuhkan bantuan jika kami bisa memberikannya.
8. Farhan (Ekstensi 2005), Alfa (Fisika 2002) serta rekan-rekan di Lab. CISCO Fisika, Mas Arif, Dwi Seno – teman baik kami dan selalu membantu pada saat yang tidak diduga – semoga ALLAH SWT membalas semua amal baik kalian, silahkan meminta bantuan jika kami dapat memenuhinya.
9. Rekan-rekan Ekstensi 2005 yang telah lulus sebelumnya – atas bantuan dan dukungannya, semoga ALLAH SWT memberikan bantuan serta pahala atas amal baik kalian semua dan tercapai cita-cita kalian.

10. Firdauzy Kusuma N – adik kami – semoga ALLAH SWT memberikan jalan kemudahan buat Dede Ozy untuk semua usaha dan kerja keras Dede. Jangan nyerah De, pasti selalu ada jalan.
11. Patty Nurfitri Ariati – kekasih tercinta – terima kasih buat semua dukungan moral, doa, perhatian dan pengertian kamu. Aku berdoa semoga cita-cita kamu dan kita tercapai, selalu dalam lindungan ALLAH SWT dalam menjalankan semuanya.
12. Teman-teman dan pihak lain yang tidak dapat kami sebutkan semuanya satu persatu, dikarenakan tidak cukup halamannya ☺ - semoga ALLAH SWT membalas semua kebaikan dan amal ibadah kalian semua.

Akhir kata, semoga laporan skripsi ini dapat menjadi modal dan kemudahan serta inspirasi bagi rekan-rekan mahasiswa Fisika Instrumentasi lainnya untuk dapat menyelesaikan skripsi. Kami akan senantiasa membantu dan menjawab rekan-rekan jika membutuhkannya. Jangan lupa keep in touch sesama alumni Fisika Instrumentasi, Universitas Indonesia.

Depok, Juli 2008

Firdy Kusuma N.

**Sistem Monitoring Tegangan RMS PLN Wilayah Kota Depok berbasis  
Microcontroller**

*(Monitoring System for PLN RMS Voltage at Depok area with Microcontroller basis)*

**Firdy Kusuma N.**

**Departemen Fisika, FMIPA, Universitas Indonesia**

**Depok 16424**

**ABSTRAK**

Pembuatan sistem monitoring dan pengukuran tegangan RMS PLN secara realtime terhadap waktu telah berhasil dilakukan. Sistem tersebut menggunakan sistem rektifikasi sehingga tegangan RMS dari listrik PLN benar-benar aman dan efektif untuk dimonitoring dan diukur secara realtime. Inti sistem menggunakan data logger berbasis microcontroller AVR ATMEGA32 yang dilengkapi dengan RTC, internal ADC, dan menggunakan display pada PC untuk menampilkan data tegangan RMS PLN. Adapun data logger menggunakan salah satu software compiler untuk lebih mengoptimalkan hasil capture monitoring tegangan RMS melalui internal ADC pada microcontroller AVR ATMEGA32, sehingga nampak tren dari monitoring tersebut sesuai interval yang diinginkan. Hasil menunjukkan bahwa tegangan RMS PLN dapat secara efektif diatur interval pengambilan data dan pengkonversian ke bentuk tegangan RMS PLN secara realtime.

*Kata kunci— Realtime, Capture, Microcontroller, Data logger, Monitoring*

## ABSTRACT

Monitoring system of PLN RMS voltage at Depok area have been made which is to monitor and measure PLN RMS voltage as realtime. This system is designed with microcontroller basis which is AVR ATMEGA32 for data logger (included RTC, and internal ADC) also rectification of PLN RMS voltage for safety and effective monitoring and measurements. We used PC Delphi program for output display of the system. With one of compiler program types we have choosed at data logger (AVR ATMEGA32) we could monitoring & capture the RMS voltage as optimally. So as realtime this system could display RMS voltage and made trend depend on interval setting we have made.

*Keywords— Realtime, Capture, Microcontroller, Data logger, Monitoring*

## DAFTAR ISI

	Halaman
Lembar Pengesahan.....	i
Kata Pengantar.....	ii
Abstrak.....	v
Daftar Isi.....	vii
Daftar Gambar.....	ix
Daftar Tabel.....	x
BAB 1. PENDAHULUAN.....	1
1.1. Latar Belakang.....	2
1.2. Tujuan.....	2
1.3. Batasan Masalah.....	3
1.4. Sistematika Penulisan.....	3
BAB 2. TEORI DASAR.....	5
2.1. Tegangan RMS PLN.....	5
2.2. Rangkaian Pembagi Tegangan.....	7
2.3. Rangkaian Rectifier.....	8
2.4. Rangkaian ADC.....	9
2.5. Komunikasi Serial.....	11
2.5.1. Komunikasi Serial Asynchronous.....	11
2.5.2. Bit Start dan Bit Stop.....	12
2.5.3. Parity Bit.....	13
2.5.4. Kecepatan Pemindahan Data.....	13
2.5.5. Standar I/O RS232.....	13
2.6. Konfigurasi Pin RS232 DB-9.....	14
2.7. Mikrokontroler.....	15
2.7.1. USART.....	19
2.8. Rangkaian RTC.....	20
2.9. Rangkaian I2C.....	24
BAB 3. PERANCANGAN DAN CARA KERJA SISTEM.....	33
3.1. Perancangan Hardware.....	33
3.1.1. Perancangan Rangkaian Penyearah dan Pembagi.....	33
3.1.2. Mikrokontroler AVR ATMEGA32.....	35
3.1.3. Rangkaian External RAM.....	36
	Halaman

3.1.4. Rangkaian ADC.....	38
3.1.5. Rangkaian RTC.....	39
3.1.6. Rangkaian Komunikasi Serial RS232.....	40
3.1.7. Rangkaian Catu Daya.....	43
3.2. Perancangan Software.....	43
BAB 4. DATA & PEMBAHASAN.....	50
4.1. Ujicoba Data Logger Mikrokontroler.....	50
4.2. Ujicoba Kalibrasi Alat.....	59
4.3. Analisa Data Grafik AC vs ADC.....	60
BAB 5. PENUTUP.....	63
5.1. Kesimpulan.....	63
5.2. Saran.....	64
DAFTAR PUSTAKA.....	65
LAMPIRAN-LAMPIRAN.....	66

## **BAB I**

### **PENDAHULUAN**

Kebutuhan akan pemakaian listrik PLN baik oleh pemerintah maupun perorangan dan transparansi serta kerancuan mengenai berapa besar tegangan listrik PLN yang dialirkan setiap harinya merupakan alasan mengapa alat ini kami buat. Dengan kita mengetahui seberapa besar tegangan listrik PLN yang dialirkan setiap harinya dan mengapa selalu tidak sesuai dengan kenyataan biaya pemakaian yang dikenakan setiap bulannya. Tegangan yang berasal dari listrik PLN masih bersifat besaran yang berubah-ubah baik nilai tegangan itu sendiri maupun sinyal yang dikirimkan dari suatu terminal ke pengguna listrik PLN.

Sebanding dengan waktu maka akan berakibat adanya perubahan dalam waktu tak tentu baik itu berkurang maupun bertambah dari nilai patokan yang diberikan (ideal). Banyaknya ripple atau gangguan pada gelombang sinyal tersebut disebabkan oleh sifat bahan penghantar, suhu, maupun sumber tegangan PLN itu sendiri mengakibatkan perubahan tegangan yang dikirimkan ke konsumen dari range 220 volt hingga 300 volt menjadi cenderung kurang dari batas minimal dan memberi efek kepada pengurangan masa pemakaian alat-alat listrik dan elektronik. Adapun batasan area tegangan listrik PLN yang diukur berkisar untuk daerah Depok. Dengan mengetahui nilai tegangan listrik AC PLN (220 volt) melalui gelombang sinyal yang cenderung berubah tersebut maka dapat dijadikan sebagai informasi dan bahan pembelajaran serta media untuk memonitoring terhadap biaya pemakaian konsumen setiap hari.

## 1.1 Latar Belakang Permasalahan

Dengan banyaknya keluhan pelanggan listrik PLN akibat sering terjadinya penurunan tegangan listrik dan tarif listrik yang cenderung tidak sesuai dengan pemakaian. Terlebih dengan seringnya terjadi penurunan tegangan listrik PLN maka akan mengakibatkan kerusakan peralatan elektrik seperti pesawat TV, radio, AC dan elektronik seperti generator sinyal dan lain sebagainya baik yang dimiliki oleh perorangan maupun instansi perusahaan.

Oleh karena itu kami mencoba membuat alat untuk memonitor dan memberitahu bagaimana perubahan nilai *tegangan rms* PLN untuk daerah Depok dengan mengandalkan fungsi mikrokontroler sebagai pengatur pengambilan dan pemrosesan data serta mendisplay hasil pemrosesan data tersebut. Adapun pembuatan alat tersebut dilakukan secara bertahap mulai dari rectifier hingga basis mikrocontroller dan alat monitor tegangan serta dilakukan pengambilan data akumulatif selama kurang 20 hari untuk membuat suatu display khususnya informasi hasil penelitian bagi masyarakat akan pentingnya penggunaan sumber energi listrik PLN serta berapa besar tegangan yang disuply dan mengapa selalu tidak sesuai dengan tarif yang dibayarkan oleh konsumen ke PLN beserta software yang digunakan untuk mengatur mikrokontroler tersebut.

## 1.2 Tujuan Penelitian

Tujuan dari penelitian dan pembuatan skripsi tersebut adalah :

1. membuat alat monitor tegangan RMS AC PLN berbasis mikrokontroler

2. mengaplikasikan alat monitoring tegangan RMS tersebut guna secara bertahap dan kontinyu memonitoring nilai tegangan PLN untuk daerah Depok
3. mengetahui cara pengukuran tegangan RMS
4. sebagai syarat kelulusan dari Program Sarjana Ekstensi Fisika.

### 1.3 Batasan Masalah

Permasalahan utama yang akan dibahas dalam penelitian ini adalah masalah monitoring tegangan RMS AC PLN dengan mikrokontroler. Lebih dari itu juga akan dibahas masalah sistem perekaman data secara kontinyu serta menampilkannya dalam bentuk grafis.

### 1.4 Sistematika Penulisan

Dalam penelitian skripsi tersebut penulis membagi langkah pembuatan dalam tulisan dibagi menjadi 5 bab dengan perincian sebagai berikut:

#### BAB I. Pendahuluan

Dalam bab ini akan dijelaskan secara singkat latar belakang permasalahan, tujuan, batasan masalah dan sistematika penelitian.

#### BAB II. Teori Dasar Monitor Tegangan RMS AC PLN

Dalam bab ini akan dijelaskan masalah teori dan pengertian dasar alat monitoring tersebut dengan menjelaskan dasara pengertian sistem mikrokontroler.

#### BAB III. Analisa rangkaian monitoring tegangan RMS AC PLN

Dalam bab tersebut akan dibahas semua rangkaian monitoring tegangan AC

RMS PLN tersebut beserta komponen dan cara kerjanya. Dijelaskan juga bagian perbagian dari rangkaian tersebut.

#### BAB IV. Analisa program dan data pada rangkaian monitoring tegangan AC RMS PLN

Dalam bab ini dibagi menjadi 2 bagian yaitu analisa alat dan data. Pada analisa alat akan dibahas cara kerja dari masing-masing bagian dari rangkaian monitoring. Sedangkan analisa data dibahas guna membuktikan bahwa hasil monitoring tegangan AC RMS PLN tersebut sesuai dengan harapan dan tujuan penulis.

#### BAB V. Penutup

Dalam bab ini akan membahas kesimpulan dari semua analisa program dan data sebagai kelanjutan dari bab sebelumnya dilengkapi dengan saran dari penulis untuk pengembangan alat tersebut lebih jauh lagi.

## BAB 2 TEORI DASAR

Secara garis besar alat monitoring tersebut terdiri atas rangkaian diode bridge untuk rectifier tegangan AC dan data logger yang menggunakan mikrokontroler seri AVR ATMEGA32 untuk pengambilan, penyimpanan serta pengolahan data analog ke digital melalui rangkaian ADC. Pada dasarnya tegangan RMS AC PLN merupakan tegangan yang setiap hari digunakan oleh semua lapisan konsumen dengan tingkat daya yang berbeda-beda sesuai dengan kebutuhan, namun nilai asli tegangan yang berada di setiap konsumen tersebut tidak selalu stabil sesuai dengan biaya pemakaian yang dibayarkan ke PLN.

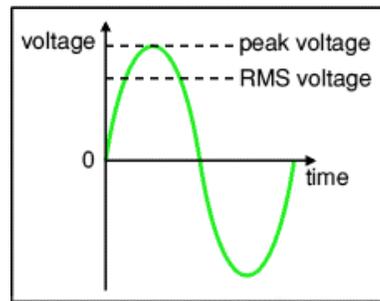
### 1.5 Tegangan PLN dan RMS

Dalam sejarahnya, tegangan AC ditransfer diantara rangkaian terisolasi dengan alat hasil percobaan William Stanley Jr., dengan menggunakan sepasang lilitan koil dari inti besi biasa, desain yang kemudian disebut koil induksi dan sebagai cikal bakal transformer modern. Sistem yang digunakan sekarang didesain ulang pada akhir abad ke 19, oleh Nikola Tesla. Kemudian terdapat pula kontribusi George Westinghouse, Lucien Gaulard, John Dixon Gibbs, Carl Wilhem Siemens, dan Oliver Shallengeter. Sistem AC menjadi pengganti arus searah yang digunakan oleh Thomas Edison untuk mendistribusikan listrik secara efisien pada jarak jauh.

Kata RMS berasal dari kata Root Mean Square yang berarti akar kuadrat rata-rata daya terhadap waktu dibagi hambatannya yang menghasilkan tegangan dan kini menjadi tegangan terhadap waktu, dalam bentuk sinyal sinusoidal dengan harga efektif untuk tegangan AC. Oleh karena itu tegangan AC yang terukur merupakan tegangan RMS. Dimana setiap tegangan AC harus terdiri dari 169 volt nilai puncak tegangan AC dengan output 120 volt (sekitar  $0.707 \times 169$ ). Dengan kata lain nilai tegangan AC yang terbaca pada alat ukur multimeter merupakan nilai

Formatted: Bullets and Numbering

tegangan RMS untuk gelombang sinusoidal AC.



Gambar 2.1. Tegangan Peak dan Tegangan RMS pada Tegangan AC

Jika tegangan AC gelombang sinusoidal memiliki *peak* (puncak) dimana terdapat faktor puncak yang besarnya perbandingan antara faktor puncak dengan nilai RMSnya, nilai faktor bentuk gelombang tegangan AC adalah hasil perbandingan nilai puncak dengan nilai rata-ratanya. Untuk gelombang sinusoidal memiliki faktor puncak 1.414 (hasil dari  $\sqrt{2}$ ) dan faktor bentuk nya 1.571 ( $\pi/2$ ) maka hubungan antara tegangan puncak dan RMS adalah sebagai berikut:

$$V_{rms} = \frac{V_{peak}}{\sqrt{2}} \quad [2.1]$$

Dengan:

$V_{peak}$  = tegangan puncak

$\sqrt{2}$  = faktor puncak gelombang sinusoidal tegangan AC

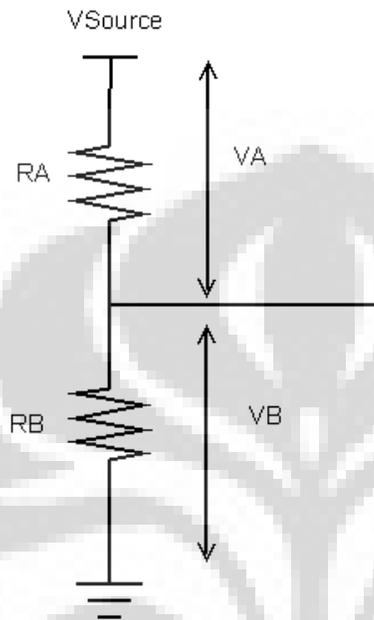
Maka nilai faktor RMS adalah

$$RMS = \frac{\sqrt{2}}{2} \quad [2.2]$$

Yang besarnya 0.707 dari nilai puncaknya sesuai persamaan tersebut.

## 2.2. Rangkaian Pembagi Tegangan

Pembagi tegangan adalah suatu alat yang dipakai untuk membagi tegangan pada tegangan AC maupun pada tegangan DC. Rangkaian pembagi tegangan terdiri atas 2 buah resistor atau lebih yang disusun secara seri.



Gambar 2.2. Rangkaian Pembagi Tegangan

Prinsip kerja dari rangkaian pembagi tegangan mengikuti hukum ohm. Besarnya arus yang mengalir pada kedua resistor adalah sama, dimana dapat dirumuskan:

$$I = \frac{V_{Source}}{R_A + R_B} \quad [2.3]$$

Sedangkan besarnya tegangan pada masing-masing resistor adalah beda, dan bergantung pada besarnya hambatan dari masing-masing resistor. Namun jumlah total dari tegangan VA dan tegangan VB adalah sama dengan Vsource.

$$V_{Source} = V_A + V_B \quad [2.4]$$

Jika arus yang mengalir di kedua resistor adalah sama maka besarnya tegangan  $V_{out}$  sama dengan besarnya tegangan  $V_B$ , dimana dapat dirumuskan:

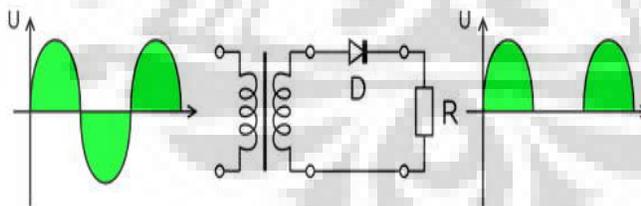
$$V_{out} = \frac{RB}{RA + RB} V_{Source} \quad [2.5]$$

### 2.3 Rangkaian Rectifier

Rangkaian rectifier merupakan rangkaian yang berfungsi untuk mengkonvert tegangan AC menjadi DC (penyearah arus). Rectifier terdiri dari rangkaian *bridge* dioda yang dihubungkan satu sama lain dan terkoneksi langsung ke PLN. Dengan menggunakan dioda *bridge* bekerja memblok bagian gelombang positif atau negatif untuk membedakan waktu rektifikasi dan waktu dioda mengubah gelombang tegangan AC ke DC. Dioda *bridge* juga menyamakan semua polaritas tegangan keluaran untuk polaritas tegangan masukan apapun dan konstan.

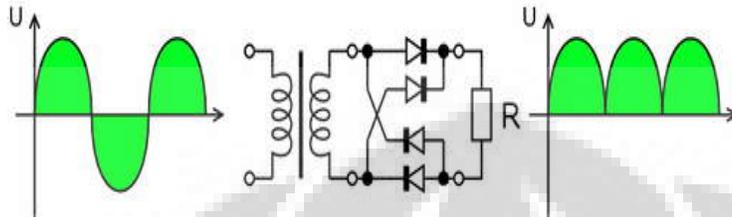
Terdiri dari 2 macam berdasarkan hasil gelombang DC penyearahnya yaitu half wave (penyearah setengah gelombang) dan full wave (penyearah gelombang penuh). Dimana untuk penyearah penuh hanya dapat dihasilkan jika menggunakan transformator *center tap*.

Pada penyearah setengah gelombang baik polaritas positif maupun negative setengah gelombang AC diteruskan sementara setengah gelombang lainnya diblok, tergantung polaritas dari rectifier itu sendiri. Dikarenakan hanya sebagian gelombang yang diteruskan, maka sangat tidak efisien untuk transfer daya. Sebagai contoh pada gambar berikut:



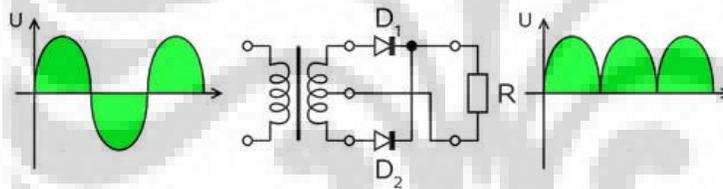
Gambar 2.3. Rangkaian Bridge Half Rectifier

Untuk penyearah gelombang penuh semua polaritas gelombang input AC akan diteruskan menjadi non polaritas pada gelombang DC. Dapat menggunakan transformator non *center tap*, hanya saja dibutuhkan 4 rectifier untuk mengurangi daya yang hilang. Pada bagian positif atau negatif gelombang menyatu dengan bagian sebaliknya dan menghasilkan keseluruhan baik positif/negatif gelombang tegangan/arus. Seperti ditunjukkan pada gambar berikut:



Gambar 2.4. Rangkaian Bridge Full Rectifier Multiphase

Untuk gelombang AC *single phase* jika menggunakan transformator *center tap* maka dioda-dioda yang digunakan dirangkai *back to back* (kutub anoda dengan anoda dan katoda dengan katoda) untuk mengkonvert gelombang AC tersebut menjadi DC. Seperti pada gambar berikut:

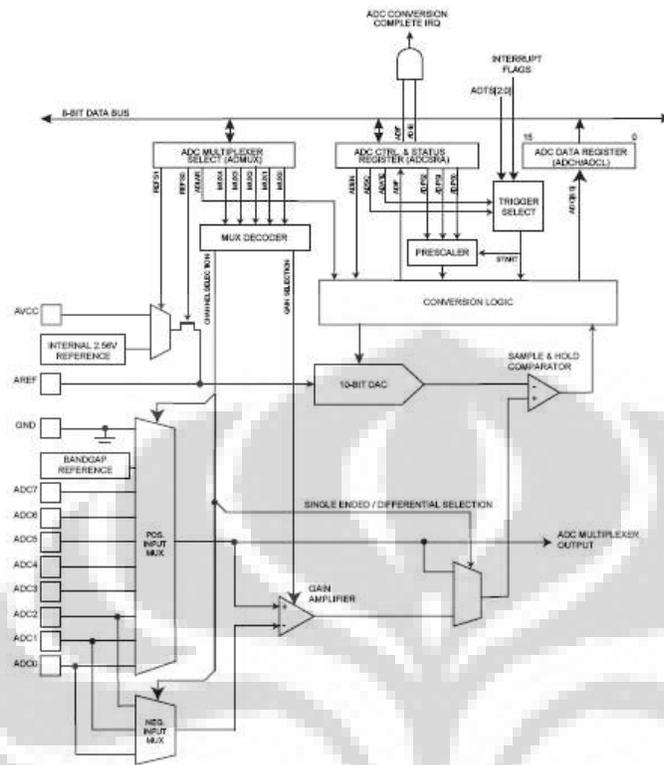


Gambar 2.5. Rangkaian Bridge Full Rectifier Singlephase

## 2.4 Rangkaian ADC

Rangkaian ADC merupakan rangkaian yang berfungsi untuk mengubah nilai analog dari suatu nilai menjadi digital atau lebih dikenal dalam bilangan biner. Biasanya bilangan biner-biner tersebut dikelompokkan ke dalam bit. Sebagai contoh resolusi sebuah ADC 8 bit, maka dapat mengkodekan nilai input analog menjadi 1 dalam 256 bagian dengan  $2^8 = 256$ . Nilai tersebut

mewakili range bilangan 0 sampai 256 atau (-128) sampai 127.



Gambar 2.6 Diagram Blok Successive Approximation ADC pada ATMEGA32

Pada rangkaian tersebut terdapat 8 nilai tegangan yang saling membandingkan dengan nilai tegangan reference.  $V_{ref}$  adalah tegangan referensi stabil yang disupply oleh regulator tegangan presisi sebagai bagian dari rangkaian konverter. Dengan tegangan input analog melebihi tegangan referensi pada setiap komparator, output komparator akan secara berurutan mensaturasi ke keadaan high. Encoder utama menghasilkan bilangan biner sesuai perintah input aktif high, mengindahkan semua input aktif. Kuantitas penting pada ADC adalah rentang tegangan terkecil yang tidak dapat mengubah hasil konversi atau yang lebih dikenal dengan MRV (Minimal Representable Voltage). Dimana  $MRV = LSB$  (Least Significant Bit).

$$MRV = LSB = FS / 2^n \quad (2.4.1)$$

Dimana FS = nilai maksimum dari tegangan referensi

Karena semua tegangan dalam jangkauan ini diwakili oleh bilangan biner yang sama, maka akan terdapat ketidakpastian konversi sebesar  $\pm$  LSB untuk setiap perubahan. Masalah ini dapat dikurangi dengan menambah jumlah bit pada output pengubah.

## 2.5 Komunikasi Serial

Untuk menunjang sistem komunikasi dari alat monitoring tegangan listrik PLN tersebut menggunakan sistem komunikasi serial. Sistem komunikasi serial tidak terlalu rumit dan hanya terdiri dari 2 jenis yaitu komunikasi synchronous dan asynchronous. Untuk sistem yang dirancang pada alat kami yaitu sistem komunikasi serial asynchronous yang memiliki ciri pada start bit dan stop bit nya. Dimana sinyal data dikirim dengan menambahkan bit awal dan bit akhir pada data sebagai penanda. Jadi pada data yang satu dengan lainnya mempunyai tanda pemisah sehingga dapat dibedakan antara keduanya pada saat menerjemahkan data. Untuk sistem komunikasi serial synchronous hanya menggunakan clock oscillator sebagai parameter pembeda bit data yang dikirimkan.

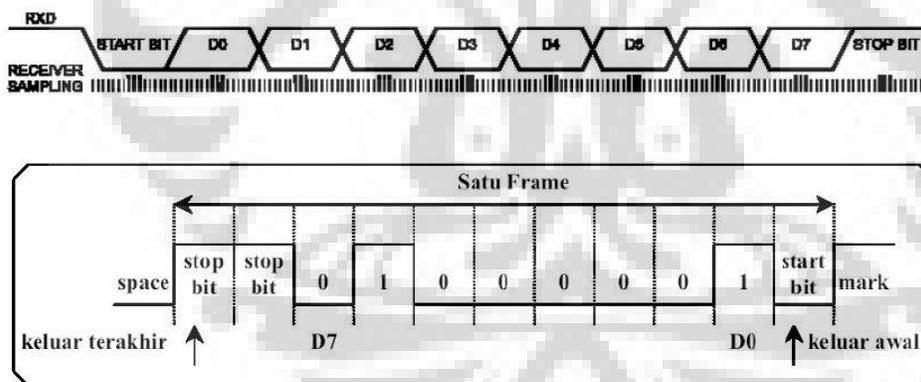
### 2.5.1 Komunikasi Serial Asynchronous

Kedatangan data pada bagian akhir jalur penerimaan data dalam pemindahan data serial seluruhnya adalah 0 dan 1, hal ini sulit untuk membuat pengertian dari sebuah data kecuali kalau pengirim dan penerima menyetujui pada sebuah kumpulan aturan protocol, bagaimana data tersebut dipaketkan, berapa banyak bit merupakan karakter, dan kapan data berawal dan berakhir<sup>[4]</sup>. Beberapa parameter penting yang terdapat pada komunikasi serial asynchronous akan dijelaskan pada sub bahasan berikut.

## 2.5.2 Bit Start dan Bit Stop

Komunikasi data serial asynchronous secara luas digunakan untuk transmisi berorientasi karakter; dan pemindahan orientasi blok data menggunakan metoda synchronous<sup>[4]</sup>. Dalam metoda asynchronous, setiap karakter terletak antara bit start dan bit stop. Hal ini disebut Framing<sup>[4]</sup>. Dalam framing data untuk komunikasi asynchronous, data, seperti karakter ASCII, dipaketkan berada antara sebuah bit start dan bit stop<sup>[4]</sup>. Bit start selalu satu bit tetapi bit stop dapat satu atau dua bit<sup>[4]</sup>. Bit start selalu 0 (low) dan bit stop adalah 1 (high)<sup>[4]</sup>. Sebagai contoh, terlihat pada Gambar 2.5 dimana karakter 20 ASCII “A”, dengan kode biner 010000 01, atau 41H, berada antara bit start dan 2 bit stop<sup>[4]</sup>. Dengan mengingat bahwa LSB dikirim keluar terlebih dahulu.

Pada Gambar 2.5 ketika tidak ada perpindahan data, sinyal adalah 1 (high), yang mana ditunjuk sebagai mark<sup>[4]</sup>. 0 (low) ditunjuk sebagai space. Dengan mengingat bahwa transmisi mulai dengan sebuah bit start diikuti oleh D0, bit LSB, dan sisa bit hingga MSB (D7), dan akhirnya, 2 bit stop mengindikasikan akhir dari karakter “A”<sup>[4]</sup>.



Gambar 2.7. Diagram waktu Komunikasi UART

### 2.5.3 Parity Bit

Dalam beberapa sistem untuk memelihara integritas data, parity bit karakter byte terdapat dalam frame data<sup>[4]</sup>. Hal ini berarti untuk tiap-tiap karakter (7 atau 8 bit, bergantung pada sistem) kita memiliki bit parity tunggal dalam penambahan pada bit start dan bit stop<sup>[4]</sup>. Bit parity adalah odd atau even. Dalam kasus ini banyaknya bit data sebuah bit odd-parity, termasuk bit parity, memiliki sebuah nilai ganjil dari banyaknya 1. Dengan cara yang sama, banyaknya bit keseluruhan dalam sebuah bit even-parity, termasuk bit parity, adalah genap. Sebagai contoh, karakter ASCII "A", dengan kode biner 01000001, memiliki 0 untuk bit even-parity<sup>[4]</sup>. Chip UART memperbolehkan pemrograman bit parity untuk pilihan odd, even, dan tanpa parity<sup>[4]</sup>. Jika sistem memerlukan parity, bit parity ditransmisikan setelah MSB, dan diikuti oleh bit stop<sup>[4]</sup>.

### 2.5.4 Kecepatan Pemindahan Data (*Baud Rate*)

Kecepatan pemindahan data dalam komunikasi data serial ditetapkan dalam bps (bits per second)<sup>[4]</sup>. Hal yang lebih luas lainnya lagi digunakan terminologi untuk bps adalah baud rate. Baud rate adalah terminologi pada modem dan didefinisikan sebagai banyaknya perubahan sinyal tiap detik<sup>[4]</sup>. Kecepatan pemindahan data yang diberikan sistem komputer bergantung pada port komunikasi yang tergabung dalam sistem<sup>[3]</sup>. UART yang digunakan dalam sistem ini mempunyai baud rate sebesar 9600 bps<sup>[3]</sup>.

### 2.5.5 Standar Serial I/O RS232

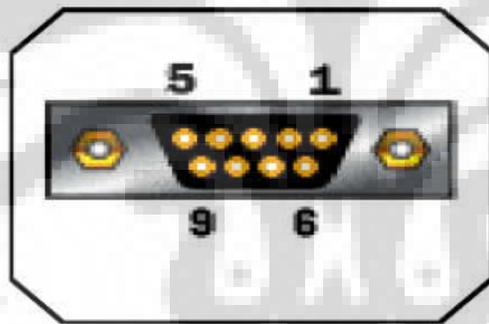
Untuk menyediakan kompatibilitas diantara peralatan komunikasi data yang dibuat oleh berbagai manufaktur, sebuah standar penghubung yang disebut RS232 dibuat oleh EIA (Electronic Industries Associates) pada tahun 1960<sup>[4]</sup>. Saat ini RS232 digunakan secara luas sebagai standar penghubung I/O serial<sup>[4]</sup>.

Standar yang dibuat sebelum kedatangan keluarga logika TTL, level tegangan input dan output

tidak cocok dengan TTL<sup>[4]</sup>. Pada RS232 sebuah logika 1 direpresentasikan oleh -3 hingga -25V, sementara itu bit 0 adalah +3 hingga +25V, sehingga membuat -3 hingga +3 tidak terdefinisi<sup>[4]</sup>. Karena alasan ini, untuk menghubungkan RS232 apapun pada system berbasis mikroprosesor, kita harus mengubah level logika TTL ke level tegangan RS232<sup>[4]</sup>. Fungsi untuk menterjemahkan dari TTL ke level tegangan RS232 disebut line drivers dan fungsi untuk mengkonversi dari RS232 ke level tegangan TTL disebut line receivers<sup>[4]</sup>.

## 2.8. Konfigurasi Pin RS232 DB-9

Pada kenyatannya bahwa tidak semua pin digunakan dalam mikrokomputer modern, diperkenalkan versi DB-9 dari standar I/O serial, yang hanya menggunakan 9 pin<sup>[4]</sup>, seperti yang ditunjukkan pada Tabel 2.1. Port serial 9-pin ditunjukkan pada Gambar 2.6.



Gbr.2.8. Male RS 232 DB 9

Tabel 2.1. Konfigurasi pin DB-9

Pin	Direction of Signal
1	Carrier Detect ( CD ) (from DCE) Incoming Signal From a modem
2	Received data ( RD ) incoming Data from a DCE
3	Transmitted Data ( DT ) Outgoing Data to a DCE
4	Data Terminal Ready ( DTR ) Outgoing Handshaking signal
5	Signal ground Common reference voltage
6	Data Set Ready ( DSR ) Incoming handshaking signal
7	Request To Send ( RTS ) Outgoing flow control signal
8	Clear To Send ( CTS ) Incoming flow control signal
9	Ring Indicator ( RI ) ( From DCE ) Incoming signal from a modem

### 2.7. Microcontroller AVR ATMEGA32

ATMega32 adalah microcontroller dengan single chip computer yang memiliki kemampuan untuk diprogram dan digunakan sebagai pengontrolan. Microcontroller ini termasuk dalam keluarga CMOS tegangan rendah 8-bit berdasarkan AVR arsitektur RISC (Reduced Instruction Set Computer), setiap instruksi dengan menggunakan microcontroller Atmega32 dapat dieksekusi dalam satu siklus clock osilator. Microcontroller ini menghasilkan 1MIPS (Million Instruction Per Second) per MHz sehingga konsumsi daya bisa optimal dan kecepatan proses eksekusi menjadi maksimal.

Microcontroller Atmega32 yang diproduksi oleh ATMEL, memiliki beberapa fasilitas yang mendukung pembuatan system controller ini. Berikut akan diterangkan mengenai fungsi dari masing-masing port dari IC AVR yaitu:

PORT A (A0-A7) : Port A selain sebagai port I/O 8-bit bi-directional, juga melayani input analog sebagai A/D converter.

Tabel 2.2. Fungsi alternatif PORT A

Port Pin	Alternate Function
PA7	ADC7 (ADC input channel 7)
PA6	ADC7 (ADC input channel 6)
PA5	ADC7 (ADC input channel 5)
PA4	ADC7 (ADC input channel 4)
PA3	ADC7 (ADC input channel 3)
PA2	ADC7 (ADC input channel 2)
PA1	ADC7 (ADC input channel 1)
PA0	ADC7 (ADC input channel 0)

Port B (B0-B7): Port B merupakan port I/O 8-bit bi-directional, selain itu port B juga mempunyai beberapa fungsi tambahan seperti terlihat pada tabel berikut:

Tabel 2.3. Fungsi alternatif PORT B

Port Pin	Alternate Function
PB7	SCK (SPI Bus Serial Clock)
PB6	MISO (SPI Bus Master Input/Slave Output)
PB5	MOSI (SPI Bus Master Output/Slave Input)
PB4	SS (SPI Slave Select Input)
PB3	AIN1 (Analog Comparator Negative Input) OC0 (Timer/Counter0 Output Compare Match Output)
PB2	AIN0 (Analog Comparator Positive Input) INT2 (External Interrupt 2 Input)
PB1	T1 (Timer/Counter 1 External Counter Input)
PB0	T0 (Timer/Counter0 External Counter Input) XCK (USART External Clock Input/Output)

Port C (C0-C7): merupakan port I/O 8-bit bi-directional, selain itu port C juga mempunyai fungsi alternatif, yaitu :

Tabel 2.4. Fungsi Alternatif PORT C

Port Pin	Alternate Function
PC7	TOSC2 (Timer Oscillator Pin2)
PC6	TOSC1 (Timer Oscillator Pin1)
PC1	SDA (Two-wire Serial Bus Data Input/Output Line)
PC0	SCL (Two-wire Serial Bus Clock Line)

PORT D (D0-D7): merupakan port I/O 8-bit bi-directional, selain itu port D juga mempunyai fungsi alternatif, yaitu :

Tabel 2.5. Fungsi Alternatif PORT D

Port Pin	Alternate Function
PD7	OC2 (Timer/Counter2 Output Compare Match Output)
PD6	ICP1 (Timer/Counter1 Input Capture Pin)
PD5	OC1A (Timer/Counter1 Output Compare A Match Output)
PD4	OC1B (Timer/Counter1 Output Compare B Match Output)
PD3	INT1 (External Interrupt 1 Input)
PD2	INT0 (External Interrupt 0 Input)
PD1	TXD (USART Output Pin)
PD0	RXD (USART Input Pin)

VCC: Power Supply

GND : Ground

RESET

Reset input. Kondisi low berlangsung lebih panjang dari panjang pulsa minimum akan membuat microcontroller berada pada kondisi reset. Pulsa yang lebih pendek tidak menjamin timbulnya kondisi reset.

AVCC

AVCC adalah tegangan supply untuk input Port analog (PORT A) dan AID converter. Saat ADC tidak dipakai AVCC terhubung ke VCC, dan pada saat ADC dipakai AVCC terhubung ke VCC yang melalui Low Pass filter.

AREF: AREF adalah referensi analog untuk AID converter

XTAL1: Input untuk inverting oscillator amplifier dan input untuk clock internal.

XTAL2: Output inverting oscillator amplifier.

Gabungan dari inti AVR yaitu set instruksi yang banyak dengan register kerja tujuan umum yang kesemuanya sebanyak 32 register secara langsung terhubung dengan Arithmetic Logic Unit (ALU), melewati 2 register yang tetap untuk akses instruksi untuk mengakses dalam satu Instruksi yang dieksekusi dalam satu cycle. Arsitektur menghasilkan kode efisiensi ketika mampu lebih cepat 10 kali daripada Merocontroller konvensional CISC.

Microcontroller Atmega32 memiliki fitur :

- 32 Kbyte dari In System programmable memori flash dengan kemampuan membaca ketika menulis
- 1024 bytes EEPROM
- 2 K bytes internal SRAM
- 32 General Purpose I/O Lines
- 32 General Purpose Working Register
- JTAG interface untuk Boundary-scan
- Interupsi Internal dan eksternal

- Pemrograman serial dengan USART, dengan orientasi bit 2 kabel serial interface dengan 8 channel
- 10 Bit ADC
- 6 mode penyimpan daya

Pada perancangan sistem microcontroller Atmega32 untuk membuat suatu sistem pengendali microcontroller, terlebih dahulu harus mengetahui blok diagram dari microcontoller Atmega32. Adapun blok diagram dari microcontroller Atmega32 dapat dilihat pada bab lampiran.

### 2.7.1 USART

*Universal Synchronous and Asynchronous Receiver Transmitter (USART* merupakan piranti komunikasi serial dengan fleksibilitas yang tinggi. Fasilitas ini berguna agar sistem aplikasi AVR dapat berkomunikasi dengan system lainnya. Fitur-fitur yang dimiliki USART adalah:

- a. Operation *Full duplex* (memiliki register serial Rx dan Tx bebas)
- b. Operasi sinkron dan asinkron
- c. Operasi *Master or Slave Clocked Synchronous*
- d. Generator berkecepatan dan beresolusi tinggi
- e. Didukung *Serial Frame* dengan 5.6.7.8. atau 9 bit data dan 1 atau 2 stop bit
- f. Generasi paritas ganjil atau genap dan pengecekan paritas yang didukung oleh *hardware*
- g. Deteksi data yang berlebih
- h. Deteksi kesalahan frame
- i. Penyaringan *noise* termasuk deteksi kesalahan bit pertama dan *digital low pass filter*
- j. Memiliki tiga buah interrupt yang terpisah. Tx *Complete*, Tx Data Register Empty, dan Rx *Complete*
- k. Mode komunikasi *Multi-Processor*

1. Mode komunikasi Asynchronous dengan kecepatan ganda.

USART harus terlebih dahulu diinisialisasi, tahap penginisialisasian terdiri atas baud rate, *Setting Frame Format*, meng-enable-kan *Transmitter* dan *Receiver* sesuai penggunaan. Untuk pemakaian interrupt pada USART pada saat inisialisasi global *Interrupt Flag* di fungsikan.

Pemakaian *Transmitter* USART, dengan cara mengatur *Transmit Enable* pada register UCSRB. Setelah itu operasi normal port pada pin TxD ditolak oleh USART dan akan difungsikan sebagai *Transmitter's Serial Output*. Pada saat operasi sinkron terjadi *Clock* dari pin XCK tidak digunakan sebagai gantinya digunakan *Clock* pengiriman (transmisi).

Pemakaian *Receiver* USART, dengan cara mengatur *Receive Enable* pada register UCSRB. Setelah itu operasi normal port pada pin RxD ditolak oleh USART dan akan difungsikan sebagai *Receiver's Serial Input*. Pada saat operasi sinkron terjadi *Clock* dari pin XCK digunakan sebagai *Transfer Clock*.

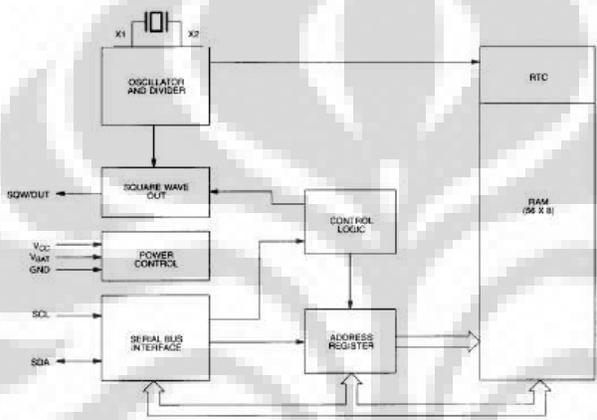
## 2.8. Rangkaian RTC

Rangkaian RTC merupakan rangkaian yang dapat menghitung waktu dalam detik, menit, jam, hari, tanggal dalam suatu bulan, tahun hingga limit tertentu tergantung dari program yang diinginkan (contoh hingga tahun 2100).

Adapun IC yang digunakan sebagai RTC tersebut adalah DS1307, yang mengkonsumsi power rendah, clock full BCD/urutan waktu, plus memiliki RAM NV (Nonvolatile) 56 bytes. Address dan data ditransfer serial melalui 2 kabel, dengan kata lain bi-directional bus. Clocknya sendiri dioperasikan selama 24jam atau 12jam dengan indicator AM/PM. Di dalam DS1307 sendiri sudah terdapat rangkaian pendeteksi kegagalan power sense dan secara otomatis menswitch ke supply batere.

DS1307 beroperasi sebagai perangkat slave pada serial bus. Dapat diakses dengan implementasi kondisi START dan memprovide kode identifikasi sesuai address registernya.

Sub urutan register dapat diakses berurutan sampai kondisi STOP diakifkan. Apabila Vcc berada dibawah  $1.25 \times V_{BAT}$  maka alat akan menghentikan akses yang sedang diproses dan mereset counter address alat dan alat akan menswitch ke mode batere backup arus rendah. Input ke alat tidak akan dapat dikenali untuk mencegah data error yang dipindahkan ke alat dari sistem yang diluar toleransi. Apabila power-up, maka alat akan menswitch kembali dari batere ke Vcc ketika Vcc lebih besar dari  $V_{BAT} + 0.2V$  dan mengenali input ketika VCC lebih  $1.25 \times V_{BAT}$ . Seperti yang ditunjukkan oleh gambar 2.11 sebagai elemen utama dari RTC.



Gbr.2.9. Block Diagram DS1307 / Elemen RTC.

Register RTC pada DS1307 berlokasi di alamat 00h sampai 07h. Dan register RAM berada di alamat 08h sampai 3Fh. Selama akses multi-byte, apabila penunjukkan alamat mencapai 3Fh, yaitu akhir alamat register RAM, akan diteruskan ke alamat 00h, sebagai awal tempat clock. Adapun alamat register RTC dan RAM seperti ditunjukkan oleh tabel berikut:

00H	SECONDS
	MINUTES
	HOURS
	DAY
	DATE
	MONTH
	YEAR
07H	CONTROL
08H	RAM
3FH	56 x 8

Gambar 2.10 Susunan Memory Pada DS1307

Informasi waktu dan kalender dapat dibaca melalui byte-byte register. Waktu dan kalender disetting atau di inialisasi dengan menuliskan byte-byte yang sesuai. Isi dari register-register waktu dan kalender dalam bentuk format BCD. Bit 7 dari register 0 adalah bit clock halt (CH). Apabila bit tersebut disetting ke 1, maka osilator akan didisable. Dan ketika di clearkan ke 0 maka osilator akan enable. Selama konfigurasi inisial awal maka penting untuk meng-enable osilator (CH bit = 0). IC DS1307 dapat dijalankan dengan mode baik 12 jam maupun 24 jam. Bit yang menyeleksi mode 12 atau 24 jam tersebut yaitu bit 6. Apabila bit 6 dalam keadaan High maka memilih mode 12 jam, dengan bit 5 sebagai bit AM/PM dengan logika high menjadi PM. Namun pada mode 24 jam, bit 5 merupakan bit 10 jam kedua (20 – 23 jam).

Pada kondisi START 2 jalur, waktu sekarang ditransfer ke set register-register kedua. Informasi waktu dibaca dari register-register kedua, dimana clock akan terus berjalan. Dan dapat menghilangkan keharusan membaca register-register apabila terjadi kondisi mengupdate register-register selama pembacaan.



Gambar 2.11. Register-register Pencatat Waktu

Register control pada DS1307 digunakan untuk mengontrol operasi pin SQW/OUT. Sesuai dengan susunan bit pada register control berikut:

| BIT 7 |
|-------|-------|-------|-------|-------|-------|-------|-------|
| OUT   | 0     | 0     | SQWE  | 0     | 0     | RS1   | RS0   |

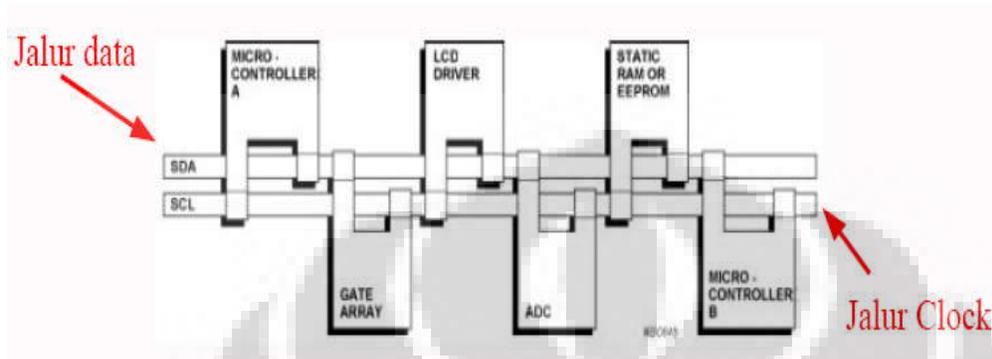
**OUT (Output control):** Bit ini mengontrol level output pin SQW/OUT ketika output gelombang kotak disable. Jika SQWE = 0, level logika di pin SQW/OUT adalah 1, jika OUT = 1 dan level logika pin SQW/OUT = 0 jika OUT = 0.

**SQWE (Square Wave Enable):** Bit tersebut ketika di setting ke logika 0, maka akan mengenable output osilator. Frekuensi output gelombang kotak tergantung nilai bit RS0 dan RS1. Dengan gelombang kotak disetting ke frekuensi 1Hz, register clock akan berubah menurun pada gelombang kotak.

**RS (Rate Select):** Bit ini mengontrol frekuensi output gelombang kotak ketika output gelombang kotak enable.

## 2.9. Rangkaian I2C

I2C merupakan serial protokol yang digunakan untuk berkomunikasi antar IC. Pertama dikembangkan oleh Philips pada era 80 an. Model komunikasi yang dikembangkan adalah



Gambar 2.12. Diagram jalur komunikasi I2C

master-slave. Saat ini I2C sudah mendukung multi-master.

Master biasanya IC yang mengendalikan komunikasi dengan slave. Sedangkan Slave biasanya IC yang melayani permintaan dari master.

Mode I2C dikelompokkan kedalam 3 frekuensi yaitu:

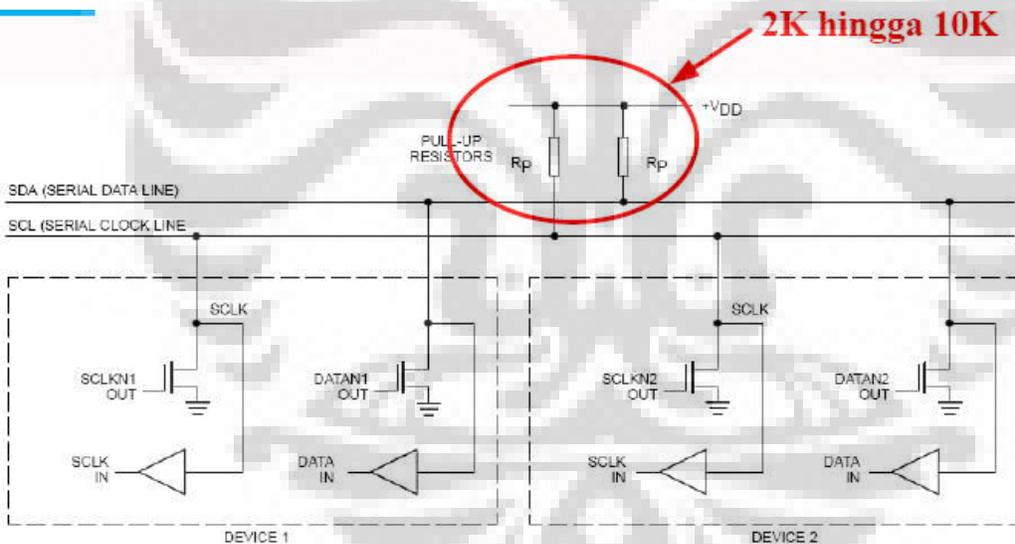
- Mode Standar (frekuensi 100KHz)
- Mode Fast / Cepat (frekuensi 400KHz)
- Mode High Speed / kecepatan tinggi (frekuensi 1.7 MHz dan 3.4 MHz)

Adapun klasifikasi frekuensi sesuai konfigurasi sistem serial bus sebagai berikut :

Tabel 2.6 Model Konfigurasi Sistem Serial Bus

TRANSFER BETWEEN	SERIAL BUS SYSTEM CONFIGURATION			
	Hs + FAST + STANDARD	Hs + FAST	Hs + STANDARD	FAST + STANDARD
Hs ↔ Hs	0 to 3.4 Mbit/s	0 to 3.4 Mbit/s	0 to 3.4 Mbit/s	–
Hs ↔ Fast	0 to 100 kbit/s	0 to 400 kbit/s	–	–
Hs ↔ Standard	0 to 100 kbit/s	–	0 to 100 kbit/s	–
Fast ↔ Standard	0 to 100 kbit/s	–	–	0 to 100 kbit/s
Fast ↔ Fast	0 to 100 kbit/s	0 to 400 kbit/s	–	0 to 100 kbit/s
Standard ↔ Standard	0 to 100 kbit/s	–	0 to 100 kbit/s	0 to 100 kbit/s

Adapun skematik rangkaian hardware I2C sebagai berikut:



Gambar 2.13. Rangkaian Hardware I2C

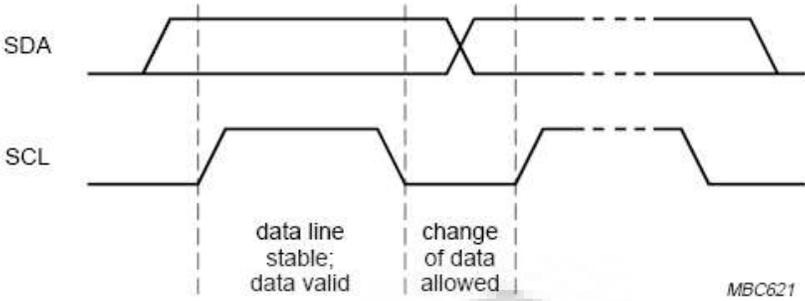
Bagian-bagian dari rangkaian hardware I2C sebagai berikut:

- Transmitter adalah perangkat yang mengirim data ke jalur bus
- Receiver adalah perangkat yang menerima data dari bus
- Master adalah komponen (IC) yang menginisialisasi transfer data, membangkitkan clock, dan memutus transfer data
- Slave adalah komponen yang dituju oleh master
- Multi-master adalah kemampuan untuk memiliki lebih dari dua master dalam satu jalur tanpa ada tabrakan data dan data yang hilang
- Arbitration adalah prosedur yang digunakan untuk memastikan bahwa hanya ada satu master yang boleh mengakses jalur bus
- Synchronization adalah prosedur untuk mensinkronkan clock pada dua master atau lebih

Bentuk transfer Bit pada I2C adalah sebagai berikut:

- Data transfer

# Data Valid ketika SCL High

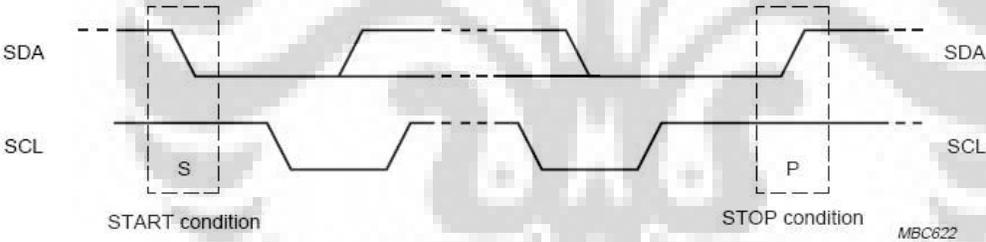


Gambar 2.14. Transfer data pada transfer Bit I2C.

- Bit Start / Stop

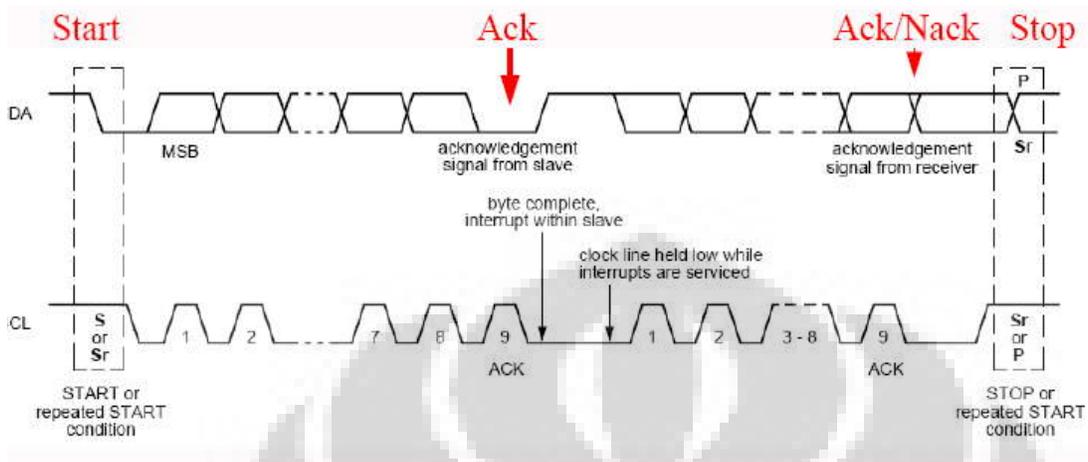
Start = SDA High to Low

Stop = SDA Low to High



Gambar 2.15. Transfer data pada transfer Bit I2C untuk keadaan Start / Stop.

Transfer data I2C dari awal clock START hingga STOP sebagai berikut:



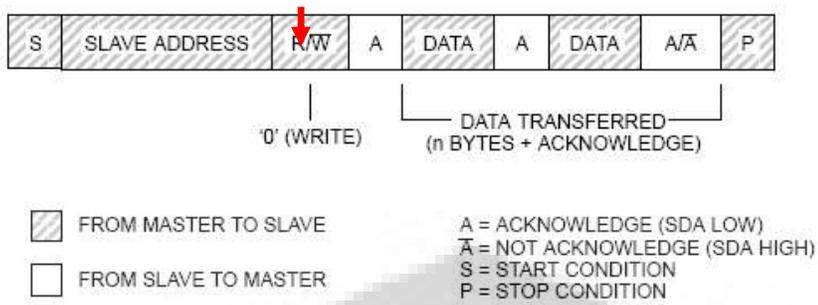
Gambar 2.16. Transfer clock data I2C dari kondisi START hingga STOP.

- Dimana ACK dilakukan oleh Receiver dengan cara membuat LOW jalur SDA
- Tetapi untuk NACK merupakan kebalikan dari ACK

Untuk Write data pada I2C dapat dibagi menjadi 2 kondisi seperti pada tabel-tabel berikut:

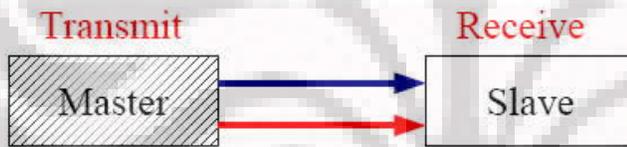
1. Keadaan dimana Bit dibuat 0

### Bit dibuat 0



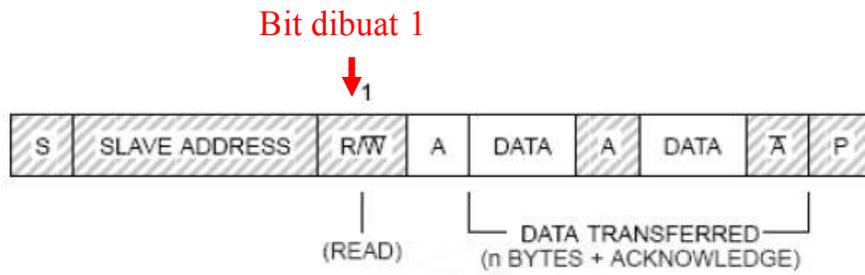
Gambar 2.17. Write Data ke I2C

Dengan komunikasi transmit dan receive antara Master dan Slave digambarkan sebagai berikut:



Gambar .2.18. Komunikasi antara Master dan Slave Bit = 0.

2. Keadaan dimana Bit dibuat 1



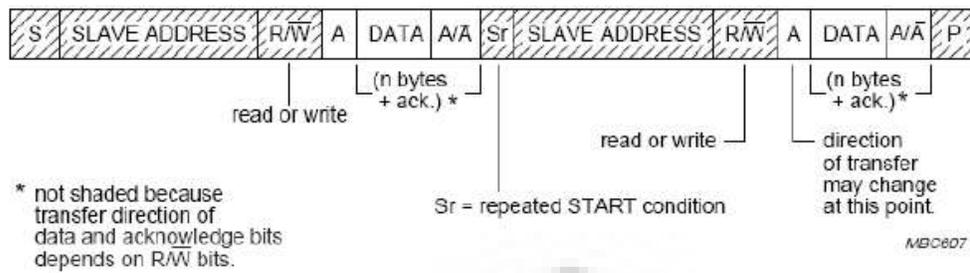
Gambar 2.19. Read Data ke I2C

Dengan komunikasi antara Master dan Slave sebagai berikut:



Gambar .2.20. Komunikasi antara Master dan Slave Bit = 1.

I2C memiliki kombinasi Write dan read sesuai dengan alamat register sebagaimana tabel berikut:

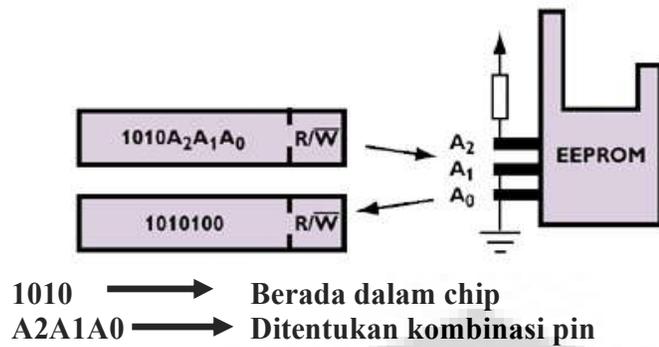


Gambar 2.21. Kombinasi Write dan Read Pada I2C

Untuk membedakan slave satu dengan yang lainnya maka masing-masing slave harus memiliki alamat (address). Adapun alamat yang dapat diberikan terdapat 2 jenis panjangnya yaitu:

- Alamat dengan panjang 7 bit
- Alamat dengan panjang 10 bit

Alamat-tersebut sebagian terprogram dalam perangkat slave dan sebagian lagi dapat ditentukan oleh kombinasi pin.



Gambar 2.22. Penentuan address Slave I2C.

Tabel 2.7 Alamat-alamat Slave I2C.

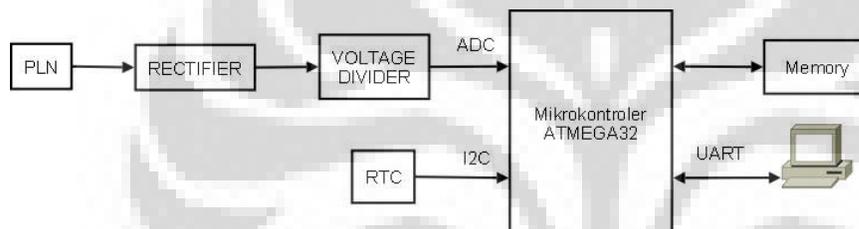
SLAVE ADDRESS	R/W BIT	DESCRIPTION
0000 000	0	General call address
0000 000	1	START byte <sup>(1)</sup>
0000 001	X	CBUS address <sup>(2)</sup>
0000 010	X	Reserved for different bus format <sup>(3)</sup>
0000 011	X	Reserved for future purposes
0000 1XX	X	Hs-mode master code
1111 1XX	X	Reserved for future purposes
1111 0XX	X	10-bit slave addressing

## BAB 3 PERANCANGAN DAN CARA KERJA SISTEM

Pada bab ini akan dibahas mengenai perancangan sistem beserta cara kerja dari masing-masing *hardware* serta *software* yang digunakan penulis dalam pembuatan alat “Perancangan sistem pengukuran dan monitoring tegangan PLN berbasis Microcontroller”. Selain perancangan alat, juga akan dibahas mengenai cara kerja alat.

### 3.1. Perancangan *Hardware*

Blok diagram dari sistem monitoring dan pengukuran tegangan listrik PLN dapat dilihat pada gambar di bawah ini.

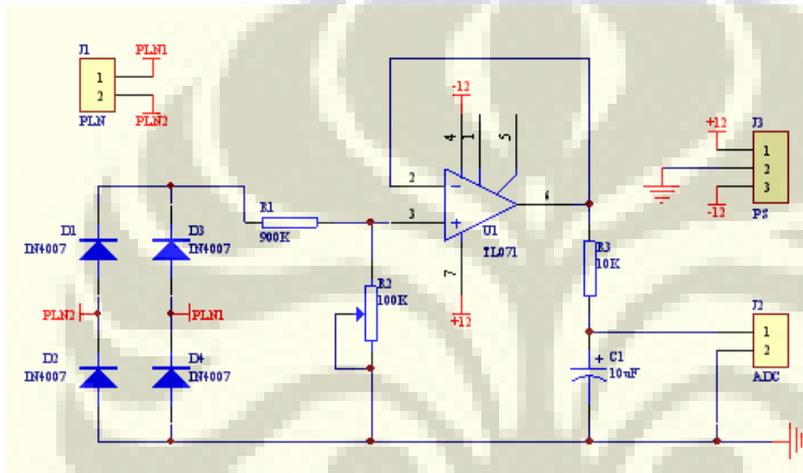


Gambar 3.1. Blok diagram sistem keseluruhan.

#### 3.1.1. Rangkaian Penyearah dan Pembagi Tegangan

Rangkaian penyearah (rectifier) yang digunakan adalah jenis Full Wave Rectifier yang terdiri atas 4 dioda, sebagaimana terlihat pada gambar 3.1. Fungsi dari rangkaian penyearah ini adalah untuk menyearahkan tegangan AC sehingga polaritasnya sama. Pada saat level tegangan PLN bernilai positif maka D1 dan D4 akan dibias maju. Sedangkan pada saat level tegangan PLN bernilai negatif maka D2 dan D3 akan dibias maju.

Keluaran tegangan dari rangkaian penyearah masih sangat tinggi untuk dapat dibaca oleh ADC. Oleh karena itu perlu diturunkan terlebih dahulu dengan menggunakan rangkaian pembagi tegangan. Rangkaian pembagi tegangan ini terdiri atas R1 dan R2. Nilai tahanan dari R2 dapat diubah-ubah untuk menentukan tegangan output yang akan diberikan ADC. Nilai maksimum yang dapat diberikan ADC adalah 5V, namun Vout dari pembagi tegangan diset dibawah 5V yakni 4V. Tujuannya adalah nantinya jika tegangan PLN mengalami kenaikan melebihi 220VAC maka ADC masih bisa membaca perubahan tersebut.



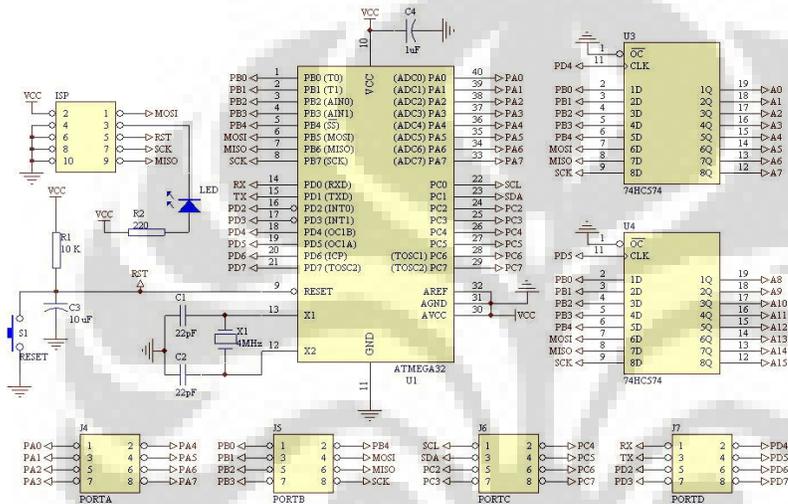
Gambar 3.2. Rangkaian Penyearah dan Penurun Tegangan

ICTL071 digunakan sebagai voltage follower dimana berfungsi untuk meningkatkan impedansi output dari rangkaian penyearah dan pembagi tegangan. Dan C1 dipakai sebagai filter untuk meratakan tegangan DC yang akan masuk ke ADC.

### 3.1.2. Rangkaian *Microcontroller AVR ATMEGA32*

Sebagai inti dari sistem yang kami buat adalah rangkaian Microcontroller AVR dengan menggunakan IC ATMEGA32, yang berfungsi sebagai pengatur kerja sistem untuk memonitoring tegangan PLN melalui internal ADC pada. Sebagai tempat penyimpanan data digunakan eksternal memori 2 buah IC62256 (masing-masing berkapasitas 32Kbyte) dengan menggunakan jalur data 2 buah IC74574 terhadap IC62256 tersebut.

Mikrokontroler AVR ATMEGA32 hanya memiliki 4 buah Port (PortA, PortB, PortC, PortD). Eksternal RAM membutuhkan PortB untuk jalur data dan jalur alamat sekaligus. Dan PortD dipakai sebagai kontrol, sedangkan PORTC dimanfaatkan koneksi ke RTC melalui I2C.



Gambar 3.3. Rangkaian MICROAVR ATMEGA32.

Dari Gambar 3.3 juga terlihat bahwa PORTB juga dihubungkan dengan ISP (In System Programming), dimana fungsi dari ISP adalah untuk menulis program ke dalam flash mikrokontroler. Dan sebagai pembangkit clock maka digunakan crystal oscillator dengan frekuensi 4 MHz.

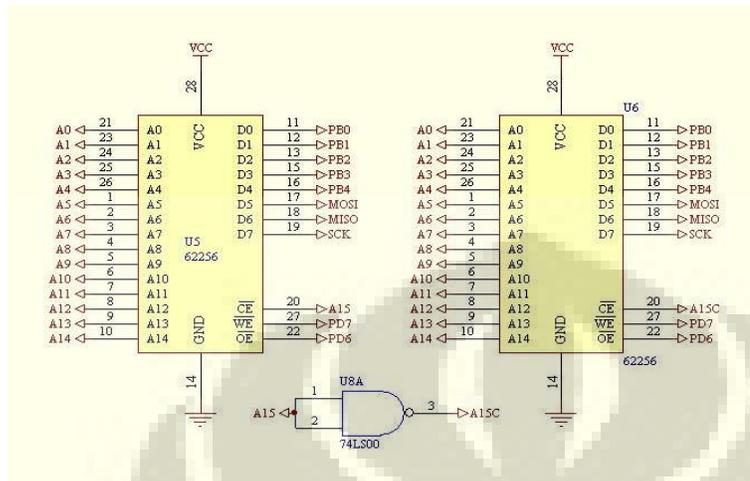
### 3.1.3. Rangkaian External RAM

Mikrokontroler AVR ATMEGA32 memiliki internal SRAM sebesar 2K Byte. Namun karena proses record data tegangan PLN berikut RTC yang menggunakan banyak ruang memori maka diperlukan ukuran SRAM yang mencukupi. Untuk mencukupinya dengan menambah eksternal RAM. RAM yang digunakan adalah 2 buah IC HY62256B produksi Hyundai. Setiap IC HY62256 RAM ini memiliki kapasitas sebesar 32 KB. Penggunaan eksternal RAM sebagai data memori akan memerlukan access time yang lebih lama bagi mikrokontroler. Hal ini dikarenakan ketika mengakses eksternal RAM mikrokontroler membutuhkan tambahan beberapa *clock per byte* dibanding bila menggunakan internal RAM.

IC HY62256B memiliki lebar data 8 bit (D0 – D7) dan lebar alamat 15 bit (A0 – A14). Sedangkan pada mikrokontroler hanya disediakan satu port saja (PORTB dengan lebar data 8 bit) yang akan dipakai untuk mengakses eksternal memori. Agar dapat terkoneksi dengan eksternal memori maka diperlukan 2 buah IC latch 74HC574 seperti pada Gambar 3.3. IC *latch* yang pertama yaitu U3 dipakai untuk menahan alamat low dari eksternal memori. IC U3 akan menahan data ketika PD4 memberikan pulsa. Sedangkan IC U4 dipakai untuk menahan alamat high dari eksternal memori dan IC ini akan aktif menahan data ketika pin PD5 memberikan pulsa.

Semua output dari kedua IC latch tersebut dihubungkan ke jalur alamat dari eksternal memori yaitu IC HY62256B kecuali pin A15. Dari Gambar 3.4 terlihat ada 2 buah IC HY62256B yang digunakan sebagai eksternal memori. Pin A15 dipakai untuk memilih memori yang aktif. Ketika pin A15 low maka IC U5 akan aktif dan ini menunjukkan bahwa IC U5 menyimpan alamat dari 0x0000 sampai dengan 0x7FFF.

Sedangkan jika pin A15 high maka U6 akan aktif dan hal ini menunjukkan bahwa IC U6 mampu menyimpan data pada alamat 0x8000 sampai dengan 0xFFFF.

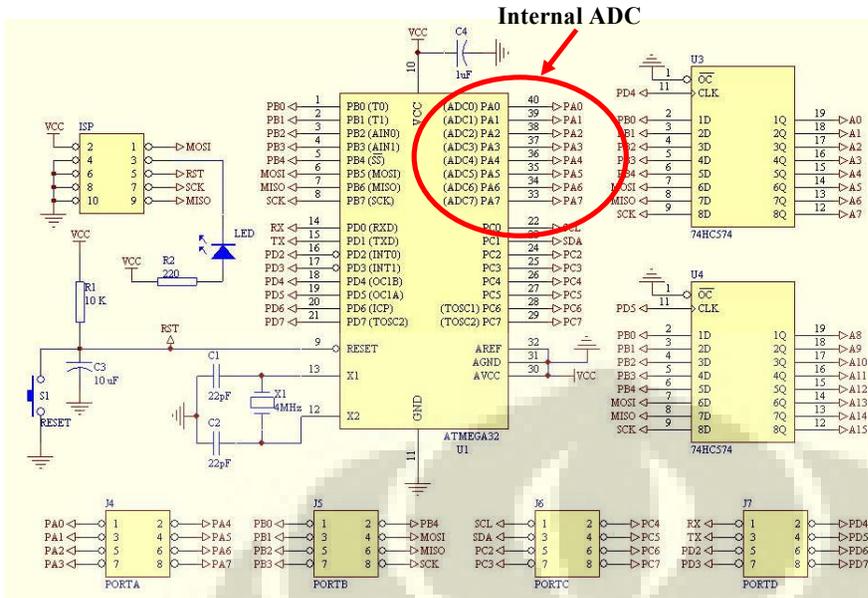


Gambar 3.4. Rangkaian RAM 62256B pada Mikrokontroler.

Untuk aksi membaca data dari eksternal memori ke mikrokontroler maka pin OE harus low dan ini dilakukan oleh pin PD6 pada mikrokontroler. Sedangkan aksi tulis ke eksternal memori dilakukan dengan cara membuat pin WE menjadi low.

### 3.1.4. Rangkaian ADC

ADC adalah sebuah sistem yang dapat mengubah input data analog menjadi keluaran data digital (biner). Adapun ADC yang digunakan dalam sistem tersebut adalah internal ADC dalam microcontroller ATMEGA32. Dengan 10 bit internal ADC pada ATMEGA32 tersebut diharapkan dapat dijadikan data acuan yang nantinya sebagai data perbandingan dengan nilai tegangan PLN yang sesungguhnya.



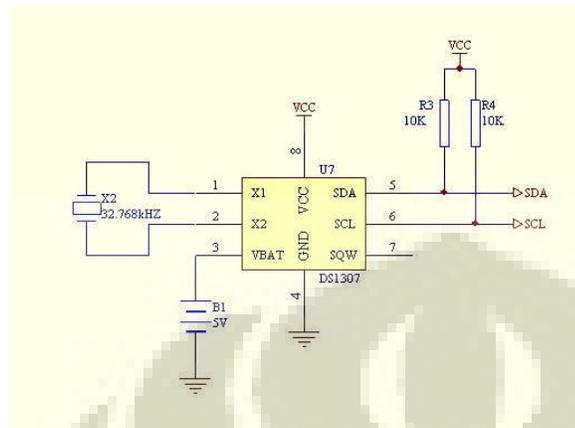
Gambar 3.5. Rangkaian internal ADC.

ADC pada ATMEGA32 mampu mengubah data nilai tegangan yang masuk melalui ATMEGA32 (5V) untuk dijadikan nilai pembanding sebagai referensi keluaran tegangan pada sistem.

### 3.1.4. Rangkaian RTC

RTC merupakan bagian dari sistem yang dapat menghitung dan mengatur waktu baik dalam detik, menit, jam, hari, tanggal, bulan dan tahun. Untuk dapat mengaktifkan clock dari DS1307 digunakan eksternal kristal sebagai pembangkit clock X2. Untuk keluaran maka SDA (Select Data) dan SCL (Select Clock) dengan filter 2 buah resistor

R3 dan R4 setelah dilewati oleh tegangan Vcc 5V. Adapun skematik rangkaian adalah seperti gambar berikut:



Gambar 3.6. Skematik rangkaian RTC

Dengan pengaturan bit pada alamat register-register pada DS1307 tersebut maka register control yang berperan menyeleksi bit yang digunakan untuk mengambil clock dan mengaktifkan RTC (Real Time Clock) didalam IC DS1307 tersebut.

### 3.1.5. Rangkaian Komunikasi Serial RS232

Rangkaian komunikasi serial RS232 berfungsi untuk jalur komunikasi dengan PC (khususnya program tampilan) dari sistem tersebut.

*Port serial* (COM) pada komputer menggunakan slot DB-9 *male* yang terdapat di belakang komputer Skematik rangkaian RS232 menggunakan gambar 3.11 berikut:



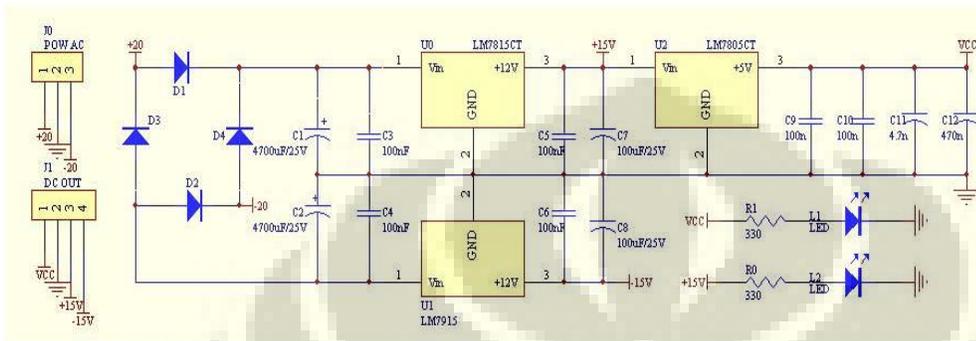
sedangkan port parallel mentransmisikan “0” pada level tegangan 0 Volt dan “1” pada level tegangan 5 Volt. Jumlah kabel yang dibutuhkan pada komunikasi serial jauh lebih sedikit, bisa hanya menggunakan tiga kabel, yaitu saluran transmit data, saluran receive data dan saluran ground selain itu lebih compatible dengan *mikrokontroler*. Hal itu dikarenakan *mikrokontroler* telah dilengkapi dengan SCI (*Serial Communication Interface*) yang dapat digunakan untuk komunikasi dengan port serial computer. Gambar di bawah ini adalah susunan konfigurasi dari pin DB 9. berikut ini:

Tabel 3.1. Susunan Pin *port serial* (COM).

	<b>9 Pin Connector on a DTE device (PC connection)</b>
Male RS232 DB9	
<b>Pin Number</b>	<b>Direction of signal:</b>
1	Carrier Detect (CD) (from DCE) Incoming signal from a modem
2	Received Data (RD) Incoming Data from a DCE
3	Transmitted Data (TD) Outgoing Data to a DCE
4	Data Terminal Ready (DTR) Outgoing handshaking signal
5	Signal Ground Common reference voltage
6	Data Set Ready (DSR) Incoming handshaking signal
7	Request To Send (RTS) Outgoing flow control signal
8	Clear To Send (CTS) Incoming flow control signal
9	Ring Indicator (RI) (from DCE) Incoming signal from a modem

### 3.1.6. Rangkaian Catu Daya

Gambar di bawah ini adalah rangkaian skematik catu daya / regulator tegangan +/- 15V untuk memberikan supply tegangan +Vcc dan -Vcc terhadap IC TL071 pada rangkaian penyearah dan pembagi tegangan. Pada gambar terlihat skematik rangkaian regulator tersebut sebagai berikut:

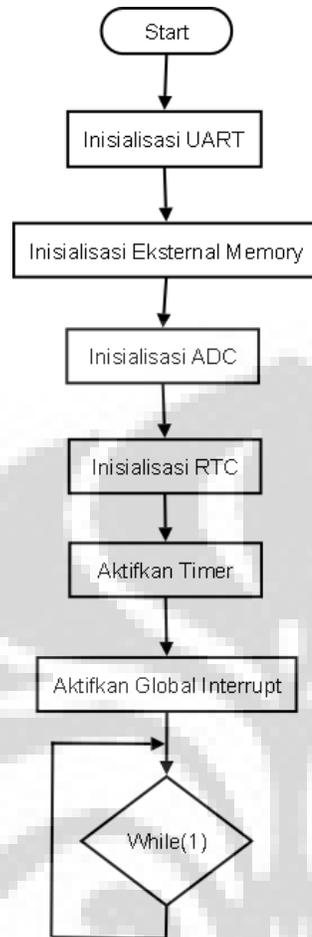


Gambar 3.9. Skematik Rangkaian *catudaya*

### 3.2. Perancangan Software

Seperti dijelaskan sebelumnya *microcontroller* ini digunakan untuk menampilkan dan menyimpan data, menampilkan data nilai tegangan PLN dari tegangan ADC internal serta mengirim data ke eksternal RAM. Adapun program yang digunakan untuk memprogram mikrokontroler adalah WinAvr.

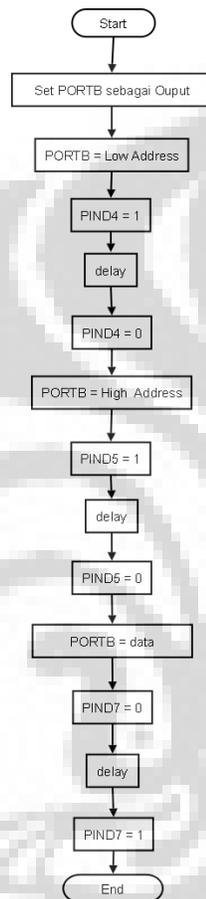
WinAvr menggunakan bahasa C sebagai bahasa utamanya dimana inti dari WinAvr adalah gcc-avr. WinAvr adalah open source program dan tidak dipungut biaya untuk dapat menggunakannya. Bahkan pengguna diberi kesempatan seluas-luasnya untuk berpartisipasi untuk mengembangkannya.



Gambar 3.10. Flowchart Program Utama

Flowchart di atas menunjukkan bahwa sistem akan terus melakukan loop jika dikondisikan untuk monitoring dan pengukuran sampai batas waktu yang ditentukan menuju ke memori eksternal maupun internal mikrokontroler. Pertama program akan menginisialisasi UART yang digunakan untuk menginisialisasi komunikasi serial dan selanjutnya menginisialisasi memori eksternal yang akan digunakan sebagai ruang

penyimpanan data yang di-*capture*. Kemudian menginisialisasi ADC dan *real-time clock* yang masing-masing digunakan untuk mengubah tegangan analog menjadi digital dan digunakan untuk menentukan waktu pengambilan data. Sistem ini menggunakan metode interrupt, sehingga pada dasarnya sistem tidak melakukan apapun hingga terjadi interrupt.



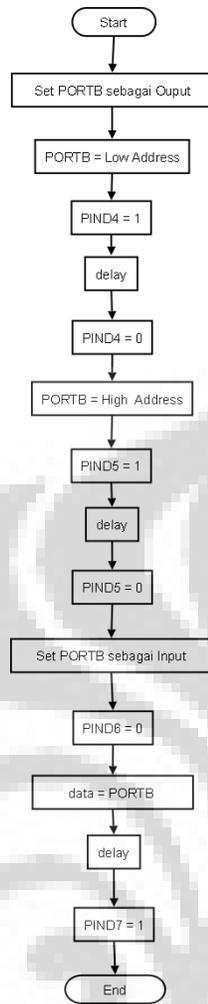
Gambar 3.11. Flowchart Sub Routine Menyimpan data ke eksternal memori

Gambar 3.11 menunjukkan flowchart subrutin yang digunakan untuk menyimpan data ke memori eksternal. Terlebih dahulu kita menginisialisasi Port B sebagai output. Selanjutnya Port B membaca Low Address, sehingga akan membangkitkan sinyal Pin D4 menjadi 1 kemudian didelay selama waktu tertentu dan Pin D4 dikembalikan sinyalnya menjadi 0.

Berikutnya Port B membaca High Address dan membangkitkan sinyal Pin D5 menjadi 1 kemudian didelay selama waktu tertentu dan Pin D5 dikembalikan sinyalnya menjadi 0.

Kemudian Port B membaca data dan membuat sinyal Pin D7 menjadi 0 kemudian didelay selama waktu tertentu dan Pin D7 dikembalikan sinyalnya menjadi 1. Kemudian kejadian menyimpan data selesai dan siap untuk menyimpan data berikutnya.



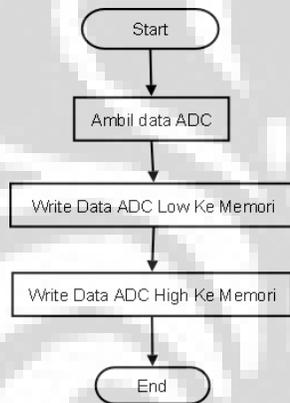


Gambar 3.12. Flowchart Sub Routine Membaca data eksternal memori

Gambar 3.12. menunjukkan flowchart subrutin untuk membaca data dari memori eksternal. Diawal kita menginisialisasi Port B sebagai output. Selanjutnya Port B membaca Low Address, sehingga akan membangkitkan sinyal Pin D4 menjadi 1 kemudian didelay selama waktu tertentu dan Pin D4 dikembalikan sinyalnya menjadi 0.

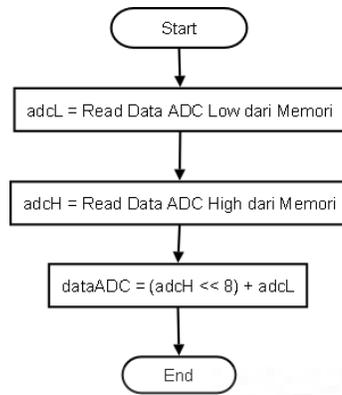
Berikutnya Port B membaca High Address dan membangkitkan sinyal Pin D5 menjadi 1 kemudian didelay selama waktu tertentu dan Pin D5 dikembalikan sinyalnya menjadi 0.

Kemudian, Port B diatur menjadi output karena sebelumnya Port B diatur menjadi input. Sebelum menerima data, sinyal Pin D6 diset menjadi 0, yang artinya siap menerima data. Lalu kita buat variabel untuk menampung data dari Port B dan didelay selama waktu tertentu. Selanjutnya sinyal Pin D7 diset menjadi 1 yang artinya proses membaca data sudah selesai. Kejadian ini dilakukan terus menerus selama membaca data dari memori eksternal.



Gambar 3.13. Flowchart Sub Routine menyimpan data ADC ke eksternal memori

Gambar. 3.13. merupakan flowchart subrutin untuk menyimpan data ADC ke memori eksternal. Sebelum data ADC ditampilkan, terlebih dahulu disimpan di memori eksternal. Karena data ADC sekitar 10-bit sedangkan kemampuan mikrokontroler hanya 8-bit, maka data ADC dibagi menjadi data ADC *High* dan data ADC *Low*.



Gambar 3.14. Flowchart Sub Routine membaca data ADC dari eksternal memori

Gambar 3.14. menunjukkan flowchart subrutin untuk membaca data ADC dari memori eksternal. Setelah data ADC disimpan di memori kemudian dikirimkan dari memori untuk ditampilkan. Karena data ADC terlebih dahulu dipisahkan menjadi bagian data *ADC High* dan data *ADC Low*, maka data *ADC High* dikirim setiap bit-nya kemudian ditambah dengan data *ADC Low* dan hasilnya ditampung ke sebuah variabel. Kemudian data *ADC High* yang sudah dikirim dilanjutkan lagi dengan data bit yang lain, begitu seterusnya hingga data *ADC High* sudah dikirim semuanya dan kejadian mengirimkan data *ADC* selesai dan siap menerima data *ADC* selanjutnya.

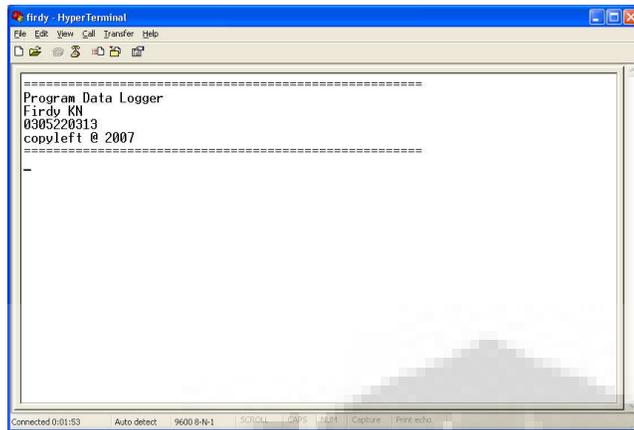
## **BAB 4**

### **DATA DAN PEMBAHASAN**

Setelah pengerjaan keseluruhan sistem, maka perlu dilakukan pengujian alat serta penganalisaan terhadap alat dan data, apakah sistem sudah bekerja dengan baik atau tidak. Pengujian-pengujian tersebut meliputi pengujian terhadap data logger (mikrokontroler), rangkaian bridge rectifier, rangkaian regulator tegangan, serta main program untuk menjalankan perintah pada miktokontroler.

#### **4.1 Uji coba Data Logger Mikrokontroler**

Pada sistem ini kita dapat menjalankan main program pada mikrokontroler menggunakan bahasa C dengan komunikasi serial RS232 dan ditampilkan melalui hyperterminal. Adapun data didapat melalui ADC internal pada mikrokontroler dan menggunakan tegangan sumber 5V dari tegangan input untuk mengaktifkan IC mikrokontroler dan yang lainnya tanpa disambungkan dengan rangkaian bridge rectifier hanya menggunakan rangkaian regulator tegangan 5V. Seperti ditunjukkan pada gambar-gambar berikut:



Gambar 4.1. Data logger aktif. (mikrokontroler)

Gambar tersebut merupakan display awal bahwa data logger sudah aktif dan dapat dilakukan monitoring dan pengukuran.

Adapun spesifikasi dari hyperterminal yang digunakan adalah sebagai berikut :

- ✚ Baudrate = 9600 bps
- ✚ Data Bits = 8
- ✚ Parity = None
- ✚ Stop bits = 1
- ✚ Flow Control = None

```

=====
Program Data Logger
Firdy KN
0305220313
copyleft @ 2007
=====
$#
Command :
$S  -> Set Time
$SD -> Set Date
$SI -> Set Interval
$SA -> Start Capture
$SO -> Stop Capture
$GT -> Get Time and Date
$GM -> Get Data in Memory
$GI -> Get Interval
$H  -> Command List

```

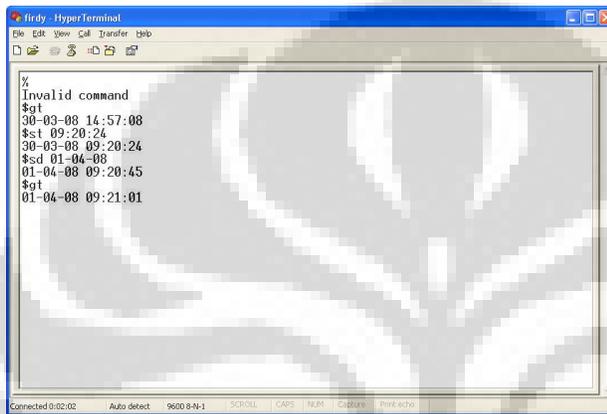
Gambar 4.1.2 List command microcontroller

Pada tahap ini menampilkan daftar perintah terhadap microcontroller dan data logger sendiri. Baik setting waktu, tanggal, serta interval capturing data.

Pada tahap ini berisi command-command yang harus dijalankan selama proses pengambilan data. Seperti pada gambar di atas command-command tersebut adalah :

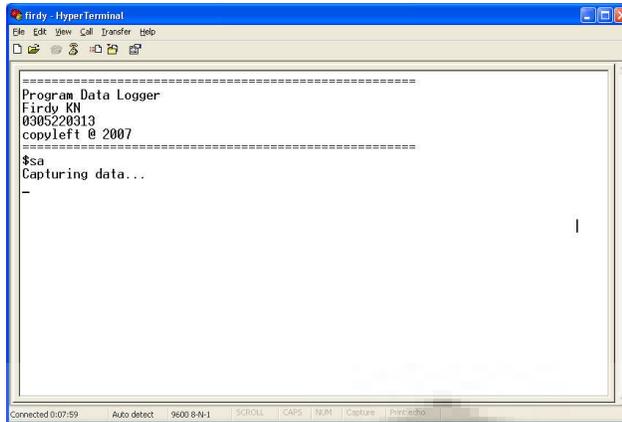
- ✚ \$S = Set Time, yaitu pengaturan jam, menit dan detik pada RTC. Dimana waktu disini merupakan start awal pengambilan data PLN.
- ✚ \$SD = Set Date, yaitu pengaturan tanggal, bulan dan tahun pada RTC. Sama halnya \$S, \$SD harus dikonfirmasi pada start awal pengambilan data PLN.
- ✚ \$SI = Set Interval, waktu yang di tetapkan untuk mengcapture data dalam hal ini sekitar 60 detik dalam waktu 2 hari.
- ✚ \$SA = Start Capture, Perintah untuk mengambil data yang diteima oleh mikrokontroller.
- ✚ \$SO = Stop Capture, Perintah yang diberikan ke mikrokontroller untuk melakukan penghentian pengambilan data.

- ✚ \$GT = Get Time and Date, Perintah yang diberikan ke microcontroller untuk memberikan data waktu setiap saat.
- ✚ \$GM = Get Data in Memory, Perintah yang diberikan untuk mengambil data yang tersimpan di EEPROM.
- ✚ \$GI = Get Interval, Perintah yang diberikan untuk mendapatkan interval pada saat capturing data.



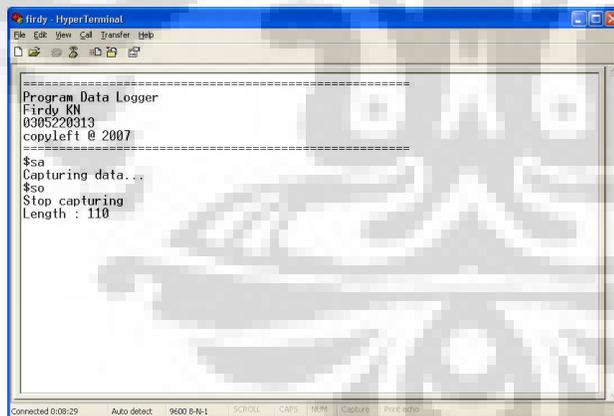
Gambar 4.1.3 Setting dan getting time data logger mikrokontroler

Seperti pada gambar di atas pada saat setting time dan getting time hal tersebut menunjukkan adanya pengaturan bit oleh microcontroller melalui jalur SDA dan SCL pada alamat register-register pada DS1307 tersebut maka register control yang berperan menyeleksi bit yang digunakan untuk mengambil clock dan mengaktifkan RTC (Real Time Clock) didalam IC DS1307 tersebut.



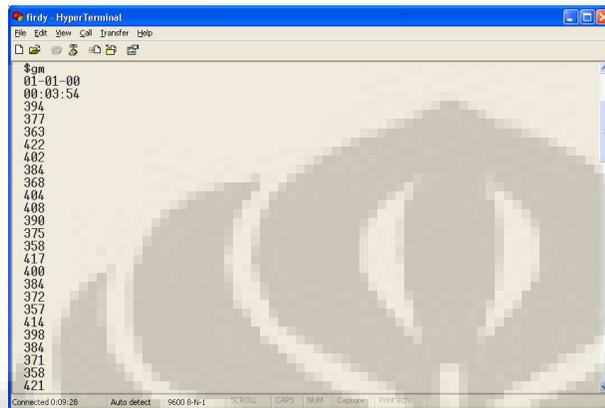
Gambar 4.1.4. Mulai mengcapture data

Pada tahap ini mikrokontroler bersiap menerima data dari rangkaian pengkondisi sinyal. Adapun output dari rangkaian tersebut akan menjadi input bagi mikrokontroler. Data kemudian masuk ke pin mikrokontroler yang berfungsi khusus sebagai ADC (Analog to Digital Converter). ADC pada ATMEGA32 mampu mengubah data nilai tegangan yang masuk untuk dijadikan nilai referensi keluaran tegangan pada sistem.



Gambar 4.1.5 Stop capturing data mikrokontroler

Seperti pada gambar di atas menunjukkan banyaknya data yang diambil pada saat memulai dan pada saat menghentikan pengambilan data. Dimana pada saat data mulai diambil dan pada saat data dihentikan semuanya telah tersimpan di EEPROM. Adapun banyaknya data yang tersimpan selama interval tersebut sebanyak 110 data.



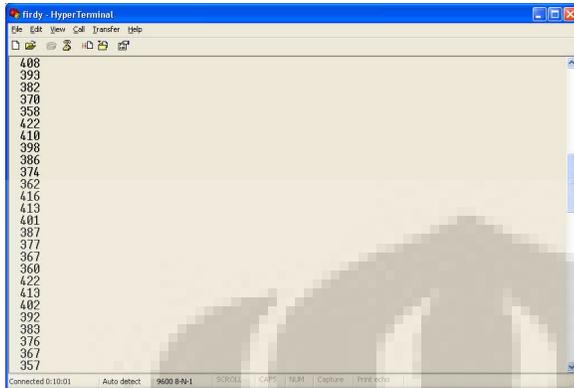
Gambar 4.1.6 Get data capture mikrokontroler dari memori

Seperti pada gambar di atas nilai-nilai tersebut merupakan representasi dari nilai-nilai PLN yang terbaca, baik pada saat terjadi kenaikan tegangan maupun pada saat terjadi penurunan tegangan PLN. Adapun Nilai-nilai ADC yang terbaca direpresentasikan dengan nilai tegangan 0 – 5 volt.

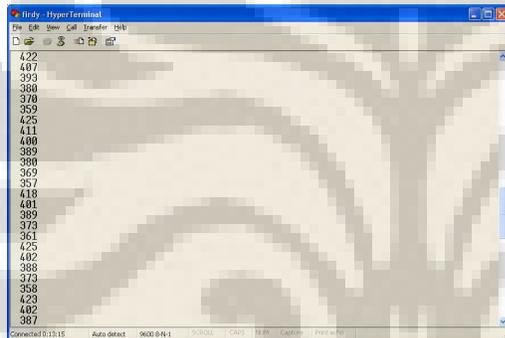
Dengan 10 bit internal ADC pada ATMEGA32 tersebut diharapkan dapat dijadikan data acuan / referensi sebagai data perbandingan dengan nilai tegangan PLN yang sesungguhnya .

Kemudian nilai ADC yang didapat dari rangkaian pengkondisi sinyal disimpan menggunakan dua buah eksternal memori yaitu menggunakan 2 buah IC62256 (masing-

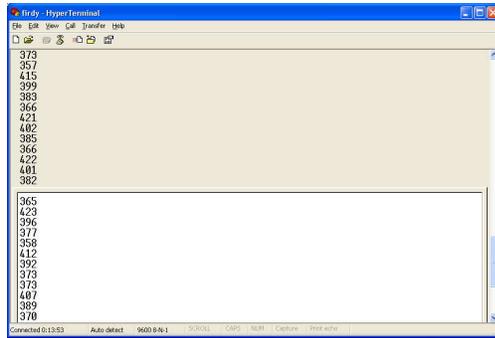
masing berkapasitas 32Kbyte) dengan menggunakan jalur data 2 buah IC74574 terhadap IC62256 itu sendiri.



Gambar 4.1.7 Lanjutan hasil capturing data mikrokontroler (1)



Gambar 4.1.8 Lanjutan hasil capturing data mikrokontroler (2)

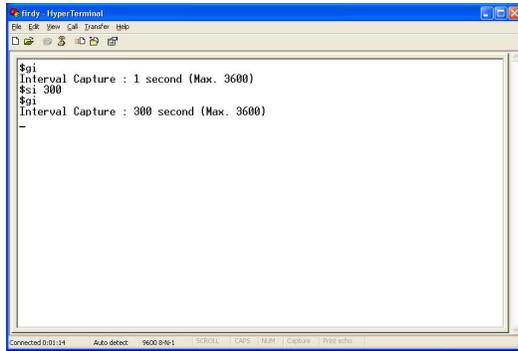


Gambar 4.1.9 Lanjutan hasil capturing data mikrokontroler (3)



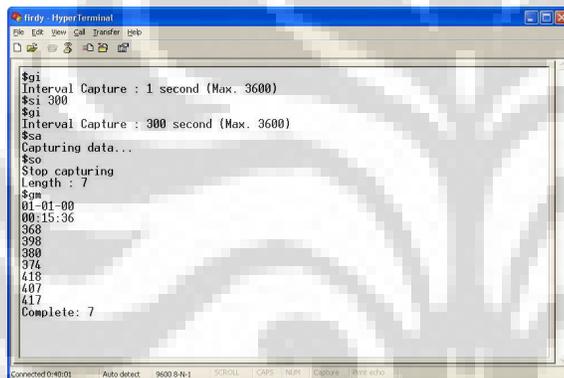
Gambar 4.1.10 Hasil terakhir semua capturing data mikrokontroler (4)

Seperti pada tampilan hyperterminal di atas maka banyaknya data yang tersimpan sebanyak 110 data. Hal tersebut dapat disesuaikan pada saat setting interval dimana banyaknya data yang ingin diambil dapat disesuaikan sesuai kebutuhan.



Gambar 4.1.11 Setting interval capturing data mikrokontroler

Pada tahap ini pengaturan jumlah banyaknya data yang dapat diambil bisa diatur, pada penelitian kali ini maksimal 3600 data.



Gambar 4.1.12 Capturing data mikro setelah perubahan interval capturing

Untuk pengujian data logger mikrokontroler menggunakan tegangan input 5V berhasil dan memunculkan nilai ADC hasil dari capturing data tersebut. Adapun interval pengambilan data menggunakan skala mulai dari setiap 1 second hingga maksimum 3600 second (1 jam), tergantung kebutuhan data yang akan disimpan dalam SRAM.

## 4.2 Uji Kalibrasi Alat

Pada bagian ini terdapat nilai-nilai pengkalibrasian, termasuk di dalamnya sistem pengkonversian nilai-nilai ADC ke tegangan AC dan sistem konversi nilai-nilai DC ke nilai tegangan AC.

Berikut adalah persamaan yang digunakan :

$$y = 0,273x - 8,217 \text{ (ADC ke AC)}$$

$$y = 55,43x - 8,906 \text{ (DC ke AC)}$$

Sebagai contoh disini kita akan mengambil salah satu hasil dari pengambilan data. Pada data terbaca data ADC sebesar 852 (nilai ADC). Hal itu berarti kita harus mengkonversinya menurut rumus di atas, yaitu sebagai berikut :

$$y = 0,273(852) - 8,217$$

$$y = 232,596 - 8,217$$

$$y = 224,379$$

Kemudian dengan membandingkan nilai tegangan DC yang didapat dengan menggunakan persamaan di atas. Pada data terbaca data tegangan DC sebesar 4,209 volt. Hal itu berarti kita harus mengkonversinya menurut rumus di atas.

$$y = 55,43(4,209) - 8,906$$

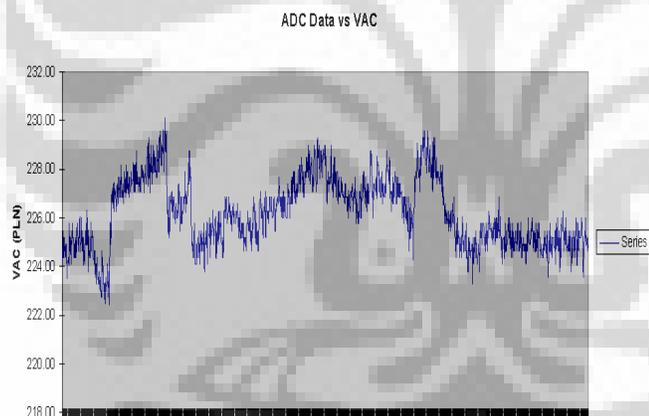
$$y = 233,30487 - 8,906$$

$$y = 224,398$$

Berdasarkan hasil kedua perhitungan tersebut maka didapatkan hasil tegangan AC yang sama yaitu didapat tegangan AC sebesar 224,398 volt.

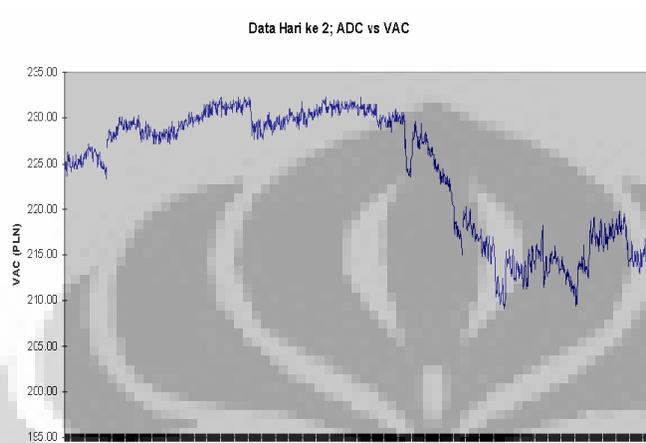
### 4.3 Analisa Data Grafik tegangan AC Vs ADC

Pada penelitian ini dilakukan pengambilan data selama 2 hari. Berikut adalah grafik yang didapat pada pengambilan data pada hari pertama.



Gambar 4.2.1 Grafik Tegangan pada hari pertama

Pada saat awal pengambilan data hari pertama, nilai tegangan yang terbaca sebesar 224,11 volt kemudian nilai tersebut akan mengalami fluktuasi / naik turun terhadap waktu hingga mencapai nilai maksimal sebesar 230,11 volt. dan nilai tegangan minimum yang terbaca sebesar 222,47 volt.



Gambar 4.2.2 Grafik Tegangan pada hari kedua

Pada saat pengambilan data pada hari kedua, nilai tegangan yang terbaca sebesar 224,38 volt dan akan terus mengalami fluktuasi / naik turun terhadap waktu hingga mencapai nilai maksimal sebesar 232,30 volt dan nilai tegangan minimum sebesar 209,09 volt.

Dari kedua grafik tersebut ada beberapa faktor yang mempengaruhi naik turunnya besar nilai tegangan yaitu :

- ✚ Banyaknya beban yang menggunakan suply tegangan rms PLN
- ✚ Besar kecilnya sumber tegangan AC PLN

## BAB 5

### KESIMPULAN DAN SARAN

Pada bab ini berisi kesimpulan yang diperoleh penulis setelah melakukan penelitian tugas akhir serta saran-saran untuk perbaikan sistem dan hasil yang lebih baik lagi dimasa yang akan datang.

#### 5.1 Kesimpulan

Setelah menyelesaikan perancangan sistem serta melakukan pengujian terhadap sistem tersebut, maka penulis dapat menarik suatu kesimpulan bahwa:

1. Perbandingan nilai-nilai ADC yang terbaca berbanding lurus dengan nilai tegangan AC. Begitupun dengan nilai-nilai tegangan DC yang terbaca berbanding lurus dengan tegangan AC.
2. Semakin besar nilai ADC / DC maka nilai tegangan AC akan semakin besar pula.
3. Beberapa faktor yang mempengaruhi besarnya tegangan AC yaitu besar kecilnya sumber tegangan AC PLN
4. Pengaruh internal ADC mempengaruhi kecepatan dari pengambilan data tegangan rms AC.
5. Pembacaan data serial yang dikirim dari PC baru dilakukan pada saat data tersebut diterima oleh mikrokontroler.

#### 5.2 Saran

Berikut ini adalah saran dari penulis agar dimasa yang akan datang sistem ini dapat dibuat jauh lebih baik lagi :

- ✚ Pengambilan data pada penelitian diharapkan dilakukan dengan waktu yang lebih lama untuk ke depannya. Sehingga didapatkan dperbandingan data yang lebih bervariasi.
- ✚ Pengambilan data untuk kedepannya diharapkan tidak hanya dilakukan di satu tempat saja. Baik itu di tempat-tempat dengan sumber tegangan PLN yang baik maupun ditempat dengan Sumber tegangan PLN yang kurang baik.
- ✚ Perancangan sistem penyimpanan data tegangan untuk kedepannya diharapkan sudah menggunakan database di komputer yang memiliki kapasitas lebih besar bila dibandingkan kita menyimpan melalui memori eksternal dalam rangka pengaplikasian yang lebih jauh.

## DAFTAR PUSTAKA

- [1] Atmel, 2007, *8-Bit AVR<sup>®</sup> Microcontroller with 8K byte in-system programmable flash AT Mega8535*, Atmel.inc.(<http://www.atmel.com>), 12 Januari 2007, pk.15.30.
- [2] Jan Axelson, 21 agustus 1995, *Networks for Monitoring and Control Using an RS-232 Interface*, Microcomputer journal.
- [3] Frank D.Petruzella, 1996, *Elektronik industri*, Terj.dari *Industrial electronic*, oleh Sumanto, Penerbit Andi, Yogyakarta.
- [4] Fredrick W. Hughes, 1981, *Panduan Op-amp*, Terj.dari *Op-Amp Handbook*, oleh Ignatius Hartono, PT. Elek Media Komputindo.
- [5] Mazidi, Muhammad Ali, *The 8086 IBM PC & Compatible Computers*, Prentice Hall, New York, 1995.
- [6] *Data Sheet Maxim RS-232 Transceiver*  
<http://www.alldatasheet.com>, 5 Februari 2007, pk.09.00 WIB.
- [7] RS 232  
<http://www.wikipedia.com/rs232>, 4 Juni 2007, pk.10.00 WIB.

# LAMPIRAN



Listing program microcontroller dengan bahasa C sebagai berikut:

```
/******  
Program untuk logger  
Nama : Firdy KN  
NPM : 0305220313  
copyleft @ 2007  
*****/  
  
#include <avr/interrupt.h>  
#include <stdlib.h>  
#include "uart.h"  
#include "rtc.h"  
#include "global.h"  
#include "adc.h"  
#include "extmem.h"  
  
#define MAX_DATA 65536  
  
char iv[4], time[10], date[10], cmd[20];  
unsigned char data, cmdln, startflag, i, j;  
unsigned short w_addr, r_addr, ln, tmp, interval, tick, tiv;  
unsigned char adch, adcl;  
unsigned char adcbuff[2];  
  
void invalid_error(void)  
{  
    send_str("Invalid command\n\r");  
}  
  
void incomplite_error(void)  
{  
    send_str("Incomplete command\n\r");  
}  
  
void help_me(void)  
{  
    send_str("Command :\n\r");  
    send_str("$ST -> Set Time\n\r$SD -> Set Date\n\r$SI -> Set Interval\n\r$SA ->  
Start Capture\n\r$SO -> Stop Capture\n\r");  
    send_str("$GT -> Get Time and Date\n\r$GM -> Get Data in Memory\n\r$GI ->  
Get Interval\n\r");  
}
```

```

    send_str("$H -> Command List\n\r");
}

ISR(USART_RXC_vect )
{
    data = DATA_REG;
    if (data != 13){
        cmd[cmdln] = data;
        send_char(data);
        cmdln++;
    }else {
        send_str("\n\r");
        cmd[cmdln] = '\0';

        if (cmd[0] != '$') {
            invalid_error();
        }else {
            if (cmd[1] == 'S' || cmd[1] == 's') {
                //Perintah untuk seting
                if (cmd[2] == 'T' || cmd[2] == 't') {
                    //Setting Waktu
                    if (cmdln > 3) {
                        i = 0;
                        j = 4;
                        while ( i < 8) {
                            time[i] = cmd[j];
                            i++;
                            j++;
                        }
                        settime(&time[0]);
                        getrtc(&time[0], &date[0]);
                        send_str(&date[0]);
                        send_str(" ");
                        send_str(&time[0]);
                        send_str("\n\r");
                    } else {
                        incomplite_error();
                    }
                }
            }else if (cmd[2] == 'D' || cmd[2] == 'd') {
                //Setting tanggal
                if (cmdln > 3) {
                    i = 0;
                    j = 4;
                    while ( i < 8) {

```

```

        date[i] = cmd[j];
        i++;
        j++;
    }
    setdate(&date[0]);
    getrtc(&time[0],&date[0]);
    send_str(&date[0]);
    send_str(" ");
    send_str(&time[0]);
    send_str("\n\r");
} else {
    incomplite_error();
}

} else if (cmd[2] == 'A' || cmd[2] == 'a') {
    //Start
    w_addr = 0;

    //Ambil tanggal dan jam
    getrtc(&time[0],&date[0]);

    //simpan tanggal di memori
    i = 0;
    while (i < 8) {
        savemem(w_addr,date[i]);
        i++;
        w_addr++;
    }

    //simpan jam di memori
    i = 0;
    while (i < 8) {
        savemem(w_addr,time[i]);
        i++;
        w_addr++;
    }

    startflag = 1;
    ln = 0;
    tick = 0;
    send_str("Capturing data... \n\r");
} else if (cmd[2] == 'O' || cmd[2] == 'o') {
    //Stop
    startflag = 0;
    send_str("Stop capturing \n\r");
    send_str("Length : ");
}

```

```

        send_num((long)ln);
        send_str("\n\r");

    }else if (cmd[2] == 'I'|cmd[2] == 'i' ){
        //setting interval
        if (cmdln > 3) {
            i = 0;
            j = 4;
            while (cmd[j] != '\0') {
                iv[i] = cmd[j];
                i++;
                j++;
            }
            iv[i] = '\0';

            tiv = (unsigned short) atoi(&iv[0]);
            if (tiv < 3601) {
                interval = tiv;
            } else {
                send_str("Maximum 3600\n\r");
            }

        } else {
            incomplite_error();
        }
    }else {
        invalid_error();
    }
}

}else if (cmd[1] == 'H'|cmd[1] == 'h'){
    help_me();

}else if (cmd[1] == 'G'|cmd[1] == 'g'){

    if (cmd[2] == 'T'|cmd[2] == 't' ){
        //Get Waktu sekarang
        getrtc(&time[0],&date[0]);
        send_str(&date[0]);
        send_str(" ");
        send_str(&time[0]);
        send_str("\n\r");

    }else if (cmd[2] == 'M'|cmd[2] == 'm' ){
        //Get data di memori
        if (w_addr > 0){
            //tanggal pembacaan

```

```

i = 0;
r_addr = 0;
while ( i < 8 ) {
    date[i] = readmem(r_addr);
    i++;
    r_addr++;
}
date[i] = '\0';
send_str(&date[0]);
send_str("\n\r");
ms_delay(400);

//jam pembacaan
i = 0;
while ( i < 8 ) {
    time[i] = readmem(r_addr);
    i++;
    r_addr++;
}
time[i] = '\0';
send_str(&time[0]);
send_str("\n\r");
ms_delay(400);

//membaca data di memori
while ( r_addr < w_addr){
    adcl = readmem(r_addr);
    r_addr++;
    adch = readmem(r_addr);
    r_addr++;
    tmp = (adcl|(adch<<8));
    send_num((long)tmp);
    send_str("\n\r");
    ms_delay(400);
}

send_str("Complete: ");
//ms_delay(400);

//send_str("Length : ");
send_num((long)ln);
send_str("\n\r");
ms_delay(400);
} else {
    send_str("Memory is empty \n\r");
}

```

```

    }
    }else if (cmd[2] == 'I'||cmd[2] == 'i') {
        send_str("Interval Capture : ");
        send_num((long)interval);
        send_str(" second (Max. 3600)\n\r");
    }else {
        invalid_error();
    }
    }else {
        invalid_error();
    }
    }
    cmdln = 0;
}
}

ISR(TIMER1_OVF_vect){
    if (startflag) {
        tick++;
        if (tick == interval) {
            adc_get_data(&adcbuf[0]);
            savemem(w_addr,adcbuf[0]);
            w_addr++;
            savemem(w_addr,adcbuf[1]);
            w_addr++;
            ln++;
            tick = 0;
        }
        //if (w_addr == MAX_DATA) startflag = 0;
    }

    TCNT1L = 0xf7; //ubah sesuai kebutuhan anda
    TCNT1H = 0xc2; //ubah sesuai kebutuhan anda
}

int main(void)
{
    //Matikan semua interrupt
    cli();

    //Inisialisasi Hardware
    send_str("Inisialisasi\n\r");
    uart_init();
    ms_delay(50);
    initxtmem();
}

```

```

ms_delay(50);
adc_init();
ms_delay(50);
initrtc();
ms_delay(50);
//setrtc("14:00:30","27-10-07");
ms_delay(50);

//Inisialisasi timer
TIMSK |= (1<<TOIE1);

//interupsi akan terjadi setiap 1 detik sekali
TCNT1L = 0xf7; //ubah sesuai kebutuhan anda
TCNT1H = 0xc2; //ubah sesuai kebutuhan anda

TCCR1A = 0;
TCCR1B = 4; //set prescaler 256 - anda boleh merubahnya

//Inisialisasi variable
cmdln = 0;
startflag = 0;
w_addr = 0;
r_addr = 0;
ln = 0;
tick = 0;
interval = 1;

//Aktifkan semua interrupt
sei();

//Mulai program
send_str("=====\n\r");
send_str("Program Data Logger\n\r");
send_str("Firdy KN\n\r");
send_str("0305220313\n\r");
send_str("copyleft @ 2007\n\r");
send_str("=====\n\r");

while(1);

return 0;
}

```

Sample hasil pengambilan data hari ke 1 dan 2 sebagai berikut:

No.	ADC	VAC	No.	ADC	VAC	No.	ADC	VAC
1	851	224.11	43	856	225.47	85	856	225.47
2	856	225.47	44	855	225.20	86	855	225.20
3	852	224.38	45	854	224.93	87	854	224.93
4	852	224.38	46	855	225.20	88	851	224.11
5	855	225.20	47	853	224.65	89	853	224.65
6	855	225.20	48	856	225.47	90	853	224.65
7	855	225.20	49	857	225.74	91	851	224.11
8	852	224.38	50	858	226.02	92	849	223.56
9	853	224.65	51	853	224.65	93	851	224.11
10	854	224.93	52	855	225.20	94	848	223.29
11	853	224.65	53	858	226.02	95	849	223.56
12	852	224.38	54	853	224.65	96	852	224.38
13	854	224.93	55	856	225.47	97	851	224.11
14	849	223.56	56	853	224.65	98	849	223.56
15	850	223.83	57	855	225.20	99	848	223.29
16	852	224.38	58	855	225.20	100	849	223.56
17	852	224.38	59	851	224.11	101	848	223.29
18	851	224.11	60	852	224.38	102	850	223.83
19	857	225.74	61	853	224.65	103	848	223.29
20	854	224.93	62	853	224.65	104	848	223.29
21	853	224.65	63	851	224.11	105	847	223.01
22	855	225.20	64	856	225.47	106	849	223.56
23	855	225.20	65	854	224.93	107	846	222.74
24	851	224.11	66	857	225.74	108	846	222.74
25	853	224.65	67	855	225.20	109	846	222.74
26	855	225.20	68	854	224.93	110	853	224.65
27	854	224.93	69	859	226.29	111	850	223.83
28	857	225.74	70	857	225.74	112	852	224.38
29	855	225.20	71	854	224.93	113	850	223.83
30	855	225.20	72	856	225.47	114	848	223.29
31	855	225.20	73	850	223.83	115	848	223.29
32	854	224.93	74	855	225.20	116	847	223.01
33	852	224.38	75	853	224.65	117	848	223.29
34	852	224.38	76	852	224.38	118	847	223.01
35	854	224.93	77	853	224.65	119	845	222.47
36	852	224.38	78	852	224.38	120	846	222.74
37	855	225.20	79	852	224.38	121	849	223.56
38	854	224.93	80	857	225.74	122	847	223.01
39	852	224.38	81	856	225.47	123	848	223.29
40	853	224.65	82	855	225.20	124	851	224.11
41	855	225.20	83	853	224.65	125	847	223.01
42	857	225.74	84	856	225.47	126	846	222.74

No.	ADC	VAC	No.	ADC	VAC	No.	ADC	VAC
133	854	224.93	181	865	227.93	229	864	227.66
134	857	225.74	182	863	227.38	230	864	227.66
135	855	225.20	183	862	227.11	231	863	227.38
136	860	226.56	184	863	227.38	232	865	227.93
137	862	227.11	185	861	226.84	233	867	228.47
138	861	226.84	186	864	227.66	234	867	228.47
139	860	226.56	187	863	227.38	235	868	228.75
140	861	226.84	188	864	227.66	236	869	229.02
141	863	227.38	189	862	227.11	237	863	227.38
142	861	226.84	190	862	227.11	238	865	227.93
143	859	226.29	191	863	227.38	239	870	229.29
144	863	227.38	192	865	227.93	240	866	228.20
145	862	227.11	193	863	227.38	241	864	227.66
146	862	227.11	194	865	227.93	242	864	227.66
147	861	226.84	195	866	228.20	243	862	227.11
148	862	227.11	196	862	227.11	244	869	229.02
149	860	226.56	197	866	228.20	245	865	227.93
150	859	226.29	198	866	228.20	246	865	227.93
151	860	226.56	199	862	227.11	247	867	228.47
152	862	227.11	200	862	227.11	248	865	227.93
153	863	227.38	201	866	228.20	249	863	227.38
154	863	227.38	202	863	227.38	250	866	228.20
155	860	226.56	203	863	227.38	251	869	229.02
156	864	227.66	204	868	228.75	252	866	228.20
157	861	226.84	205	863	227.38	253	866	228.20
158	864	227.66	206	863	227.38	254	868	228.75
159	861	226.84	207	865	227.93	255	865	227.93
160	861	226.84	208	863	227.38	256	865	227.93
161	865	227.93	209	864	227.66	257	867	228.47
162	866	228.20	210	863	227.38	258	865	227.93
163	861	226.84	211	862	227.11	259	867	228.47
164	864	227.66	212	863	227.38	260	864	227.66
165	861	226.84	213	864	227.66	261	864	227.66
166	861	226.84	214	864	227.66	262	864	227.66
167	862	227.11	215	862	227.11	263	869	229.02
168	863	227.38	216	862	227.11	264	865	227.93
169	860	226.56	217	862	227.11	265	868	228.75
170	862	227.11	218	865	227.93	266	865	227.93
171	862	227.11	219	866	228.20	267	870	229.29
172	865	227.93	220	864	227.66	268	867	228.47
173	863	227.38	221	866	228.20	269	869	229.02
174	862	227.11	222	865	227.93	270	871	229.57
175	864	227.66	223	864	227.66	271	870	229.29
176	861	226.84	224	865	227.93	272	869	229.02
177	866	228.20	225	864	227.66	273	866	228.20
178	862	227.11	226	863	227.38	274	866	228.20
179	863	227.38	227	862	227.11	275	869	229.02
180	865	227.93	228	866	228.20	276	868	228.75

No.	ADC	VAC	No.	ADC	VAC	No.	ADC	VAC
277	868	228.75	325	858	226.02	373	855	225.20
278	869	229.02	326	856	225.47	374	855	225.20
279	871	229.57	327	858	226.02	375	854	224.93
280	865	227.93	328	861	226.84	376	852	224.38
281	865	227.93	329	861	226.84	377	853	224.65
282	871	229.57	330	857	225.74	378	856	225.47
283	871	229.57	331	859	226.29	379	851	224.11
284	868	228.75	332	857	225.74	380	854	224.93
285	873	230.11	333	860	226.56	381	854	224.93
286	868	228.75	334	861	226.84	382	854	224.93
287	869	229.02	335	861	226.84	383	856	225.47
288	871	229.57	336	860	226.56	384	853	224.65
289	868	228.75	337	861	226.84	385	853	224.65
290	860	226.56	338	865	227.93	386	855	225.20
291	857	225.74	339	859	226.29	387	854	224.93
292	856	225.47	340	861	226.84	388	855	225.20
293	861	226.84	341	860	226.56	389	853	224.65
294	857	225.74	342	862	227.11	390	857	225.74
295	855	225.20	343	860	226.56	391	852	224.38
296	858	226.02	344	860	226.56	392	850	223.83
297	856	225.47	345	862	227.11	393	852	224.38
298	861	226.84	346	863	227.38	394	855	225.20
299	860	226.56	347	863	227.38	395	854	224.93
300	861	226.84	348	863	227.38	396	854	224.93
301	862	227.11	349	868	228.75	397	856	225.47
302	859	226.29	350	863	227.38	398	856	225.47
303	859	226.29	351	864	227.66	399	857	225.74
304	861	226.84	352	868	228.75	400	851	224.11
305	860	226.56	353	864	227.66	401	854	224.93
306	858	226.02	354	864	227.66	402	856	225.47
307	861	226.84	355	864	227.66	403	857	225.74
308	861	226.84	356	861	226.84	404	855	225.20
309	859	226.29	357	854	224.93	405	858	226.02
310	859	226.29	358	855	225.20	406	856	225.47
311	857	225.74	359	854	224.93	407	852	224.38
312	857	225.74	360	853	224.65	408	857	225.74
313	860	226.56	361	854	224.93	409	854	224.93
314	860	226.56	362	852	224.38	410	856	225.47
315	859	226.29	363	852	224.38	411	856	225.47
316	858	226.02	364	854	224.93	412	856	225.47
317	859	226.29	365	857	225.74	413	856	225.47
318	858	226.02	366	856	225.47	414	858	226.02
319	859	226.29	367	854	224.93	415	859	226.29
320	862	227.11	368	851	224.11	416	855	225.20
321	863	227.38	369	851	224.11	417	854	224.93
322	860	226.56	370	854	224.93	418	856	225.47
323	862	227.11	371	852	224.38	419	856	225.47
324	859	226.29	372	852	224.38	420	857	225.74

