

BAB III

PERANCANGAN, PEMBUATAN DAN PENGUJIAN

3.1 Perancangan

Sebelum melakukan perancangan terlebih dahulu yang harus dilakukan adalah menganalisa kebutuhan alat yang akan dibuat. Agar dalam pembuatan alat sesuai dengan kebutuhan dan dapat berfungsi dengan baik sehingga tujuan dapat tercapai.

Analisis kebutuhan merupakan batasan masalah pada tujuan yang diharapkan dari sistem yang di bangun yaitu Data logger pada pembangkit kubah masjid. Analisis kebutuhan dari alat yang akan di bangun adalah sebagai berikut:

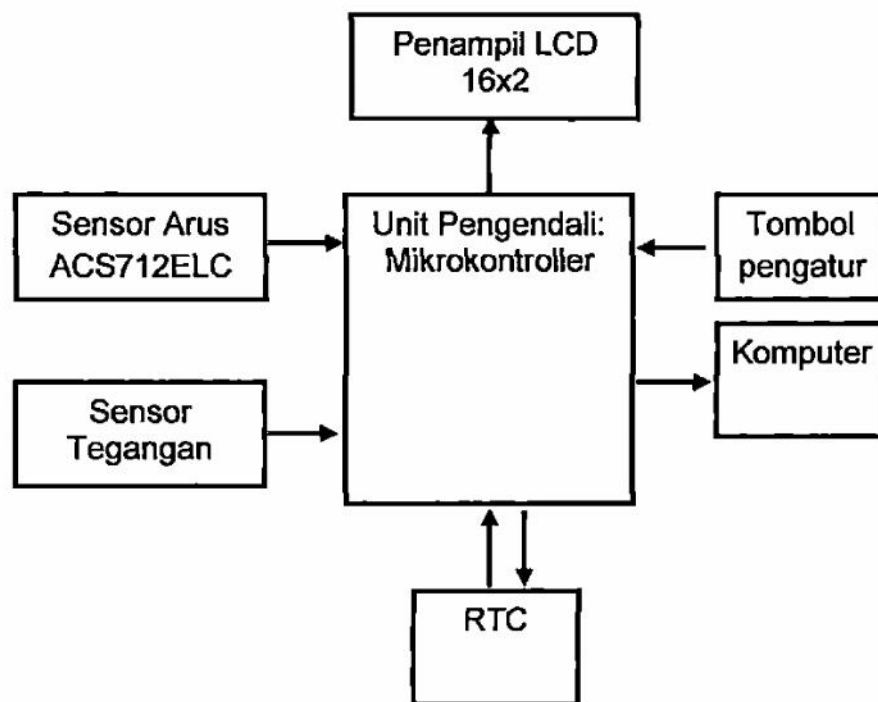
- Parameter yang diukur dan disimpan dalam data logger adalah arus dan tegangan.
- Data ditampilkan di komputer dan dapat tersimpan secara otomatis dalam bentuk file .text dan gambar secara *real time*.

Setelah menganalisis kebutuhan dari alat yang dibuat, kita dapat menentukan spesifikasi alat. Secara umum Data logger pada pembangkit kubah masjid spesifikasi sebagai berikut:

1. Sensor arus dan tegangan untuk membaca besarnya arus dan tegangan dari generator
2. Rangkaian mikrokontroler sebagai nengolah input dari sensor.

3. Tombol digunakan untuk melakukan interaksi antar pengguna dengan mikrocontroller.
4. RTC berfungsi sebagai jam dan tanggal.
5. LCD sebagai penampil.

Penjelasan spesifikasi diatas dapat dilihat pada Gambar 3.1 diagram blok sistem



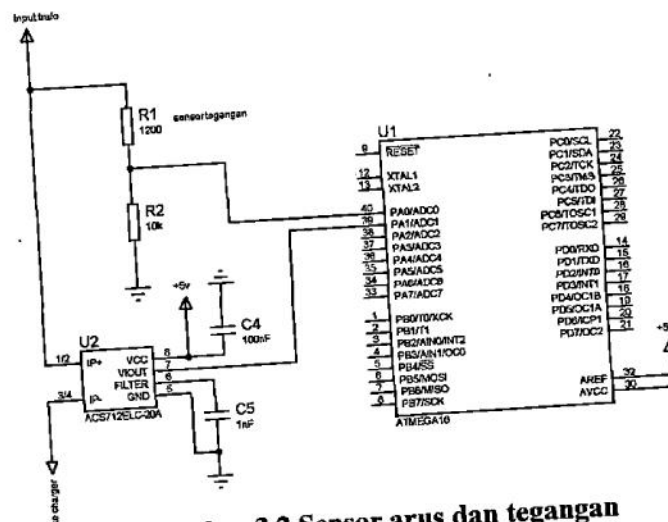
Gambar 3.1 Blok Diagram Sistem

3.1.1 Perancangan Perangkat Keras

3.1.1.1 Sensor Arus ACS712ELC-20A dan Sensor Tegangan

Untuk melakukan proses pengukuran arus yang melewati kabel digunakan suatu piranti sensor arus ACS712ELC-20A. Keluaran dari sensor ini merupakan tegangan analog linear dengan

akurasi sebesar 100 mV/A. Sedangkan untuk sensor tegangan digunakan prinsip pembagi tegangan dengan menggunakan 2 buah resistor yang bernilai 1k2 dan 10k. Tegangan keluaran dari kedua sensor ini dikonversikan ke dalam data numerik oleh ADC internal pada ATMega16. Agar dapat memanfaatkan ADC internal tersebut pin AVCC dihubungkan dengan VCC dan AGND dengan GND *microcontroller*. Tegangan referensi ADC dapat dipilih dari tiga sumber yang tersedia, yaitu AVCC, AREF, dan internal VREF sebesar 5 volt. Pada rangkaian ini digunakan tegangan referensi dari AVCC sebesar 5 volt, dengan akurasi sebesar 10 bit.

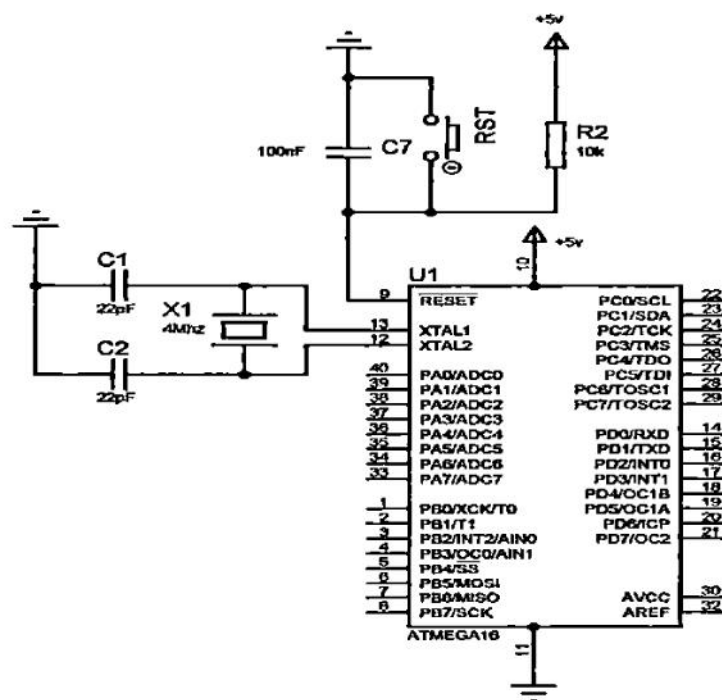


Gambar 3.2 Sensor arus dan tegangan

3.1.1.2 Microcontroller ATMega16

Microcontroller ATMega16 memerlukan minimal catu daya 5V, *clock*, dan reset untuk dapat bekerja. Sumber *clock* diperoleh dari sebuah kristal 4Mhz yang dipasang pada kaki 12 dan

13, seperti terlihat pada Gambar 3.2. Sedangkan tombol reset yang bersifat aktif *low* digunakan untuk me-reset pelaksanaan program dalam *microcontroller* sehingga dimulai dari awal (*restart*). Resistor R2 yang dipasang pada kaki reset dan terhubung pada VCC (+5V) digunakan *pull-up*, yaitu untuk mempertahankan nilai 1 (*high*) pada kaki reset selama tombol reset tidak ditekan.



Gambar 3.3 Sistem Minimum ATmega16

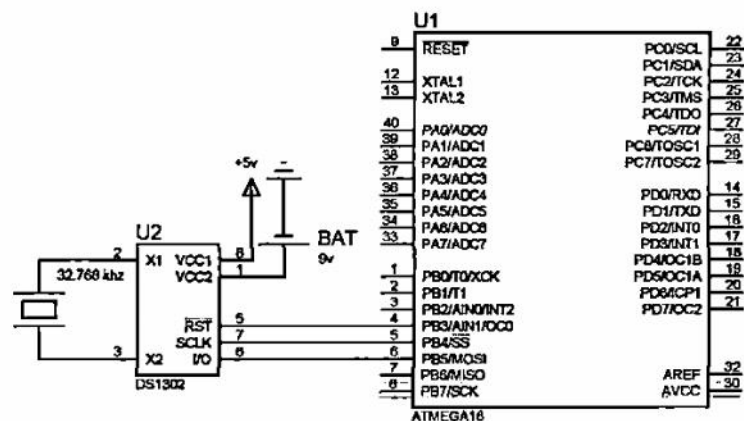
3.1.1.3 RTC

IC RTC (*Real Time Clock*) DS1302 digunakan untuk menjalankan fungsi jam dan penanggalan yang nantinya akan

digunakan sebagai acuan atau pedoman oleh mikrokontroller untuk mengontrol waktu.

Dalam Gambar ditunjukkan kristal 32.768 KHz pada kaki 2 dan 3, agar RTC DS1302 dapat bekerja. Jalur untuk perpindahan data dari dan ke RTC menggunakan RESET, SCLK dan IO. RESET harus berlogika high pada saat melakukan pembacaan dan penulisan. SCLK berfungsi untuk menyelaraskan *clock* antara *master (microcontroller)* dan *slave (RTC)*, sedangkan IO berfungsi sebagai jalur lintas data.

Baterai pada kaki VCC1 digunakan sebagai sumber tenaga cadangan bagi RTC agar tetap mampu bekerja apabila sumber tenaga utama pada kaki VCC2 gagal. Tegangan baterai untuk DS1302 adalah sebesar +3 volt.



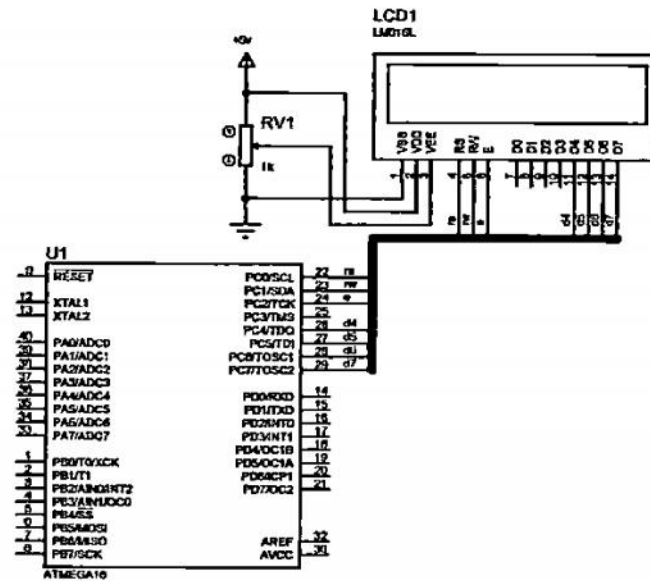
Gambar 3.4 Rangkaian RTC DS1302

3.1.1.4 Rangkaian Penampil LCD 16X2

Rangkaian penampil menggunakan *LCD* pada penampilnya. Rangkaian *LCD* sebagai penampil dihubungkan dengan mikrokontroler. Seperti terlihat pada Gambar 3.4; interaksi antara *microcontroller* dengan modul *LCD* menggunakan sistem pengiriman data 4 bit. Sehingga hanya 4 pin dari *bus data LCD* yang digunakan yaitu D4, D5, D6, dan D7.

Pin E, RS, dan R/W digunakan untuk mengendalikan operasi *LCD*. Untuk semua operasi *LCD*, pin E (*enable*) harus dalam kondisi 1 (*high*). RS digunakan untuk menentukan jenis input, yaitu *Data Input* atau *Instruction Input*. Sedangkan R/W digunakan untuk menentukan jenis operasi yaitu *Read* atau *Write* dengan mengeset *high* atau *low*.

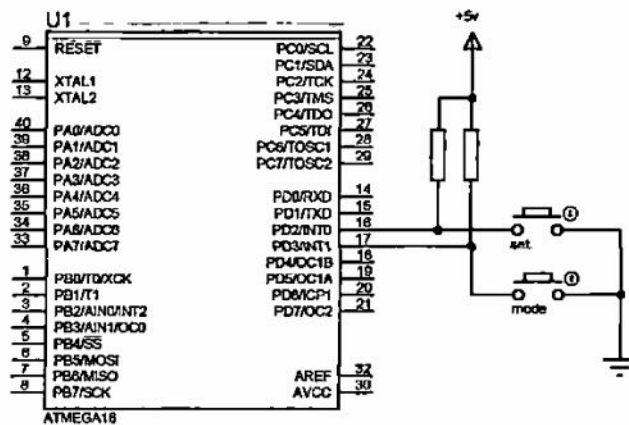
Pin VDD dihubungkan dengan sumber tegangan +5V dan VSS dihubungkan dengan GND. Sedangkan VEE digunakan untuk



Gambar 3.5 Penampil LCD 16x2

3.1.1.5 Tombol push button

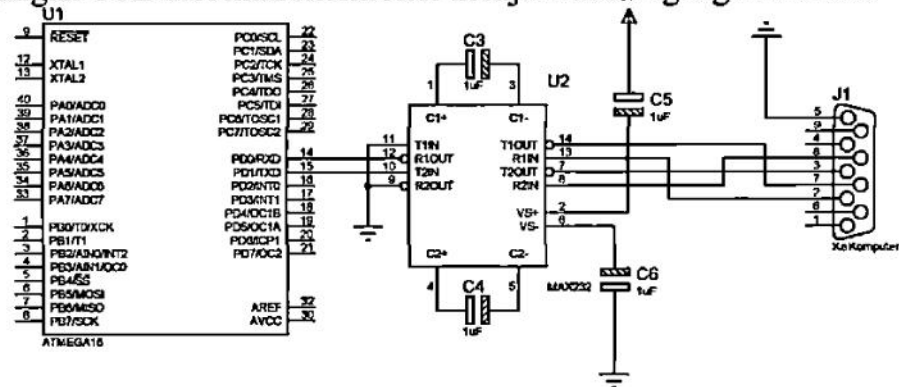
Tombol ini digunakan untuk mengatur waktu, yang terdiri dari 2 buah tombol yang masing-masing berfungsi sebagai mode dan set;



Gambar 3.6 Pushbutton

3.1.1.6 Rangkaian komunikasi serial

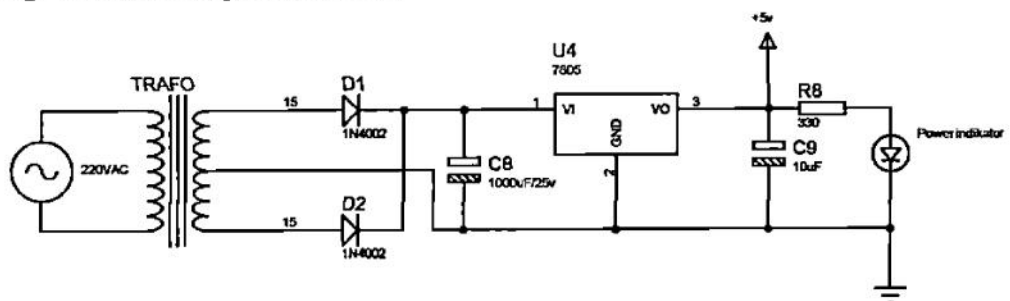
Rangkaian antarmukaserial antara computer dengan mikrokontroller menggunakan sistem komunikasi serial RS232 dan IC yang digunakan MAX232. IC ini berfungsi untuk mengubah tegangan TTL dari mikrokontroller menjadi aras.tegangan RS232.



Gambar 3.7 Komunikasi serial

3.1.1.7 Rangkaian Catu Daya

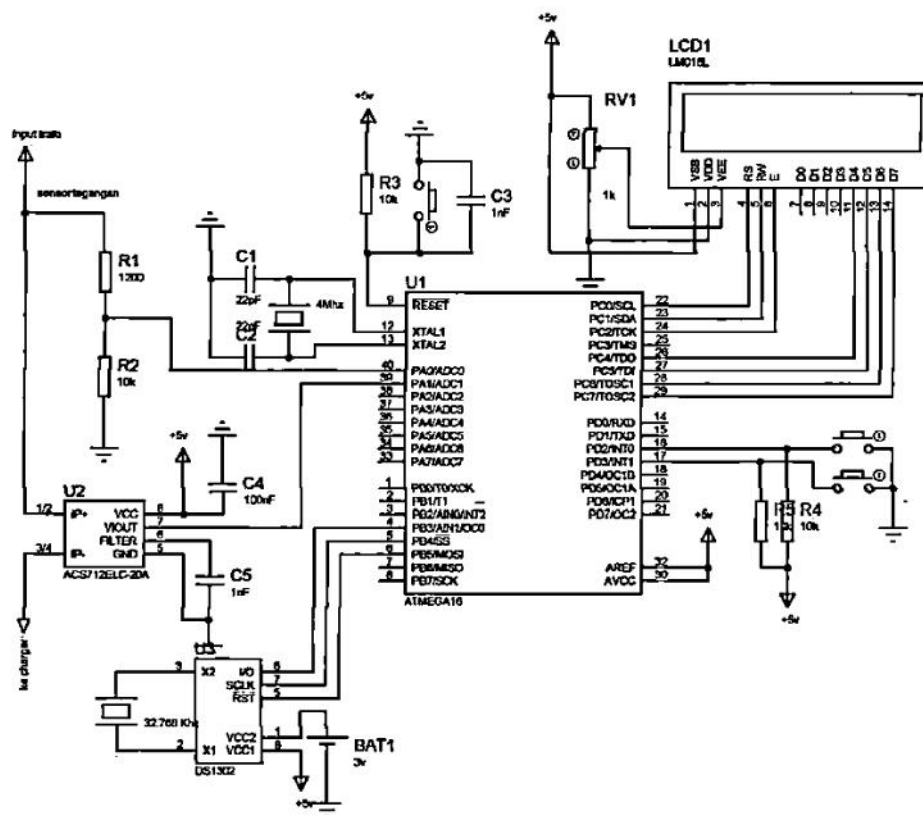
Catu daya merupakan bagian yang sangat penting pada rangkaian listrik karena tanpa catu daya alat ini tidak dapat bekerja. Rangkaian mikrokontroller dan sensor catu daya yang dibutuhkan yaitu tegangan sebesar 5 V sehingga untuk memenuhi hal ini digunakan IC regulator 7805.



Gambar 3.8 Catu Daya

3.1.1.8 Rangkaian Keseluruhan

Rangkaian keseluruhan merupakan gabungan dari sistem-sistem yang telah dijelaskan diatas. Sehingga dengan penggabungan rangkaian sistem diatas terbentuklah suatu sistem kesatuan yang utuh yaitu Data logger



Cambar 3.0 Rangkaian Keseluruhan

3.1.2 Perancangan Perangkat Lunak Mikrokontroler

3.1.2.1. Spesifikasi Perangkat Lunak

Pemrograman *microcontroller* harus memperhatikan skema rangkaian yang dibuat. Karena program harus sesuai dengan definisi fungsi masing-masing port / pin yang terhubung dengan komponen lain sebagai pendukung operasi *microcontroller*. Apabila program yang dibuat tidak sesuai dengan definisi fungsi port / pin maka sistem tidak akan bekerja dengan benar.

Perangkat lunak untuk sistem ini dibangun dengan bahasa C dan menggunakan *CodeVisionAVR* versi 1.25.9 Standard sebagai kompilernya. Dalam IDE (*Integrated Development Environment*) *CodeVisionAVR* telah disertakan berbagai *library* untuk mendukung kemudahan pemrograman. *CodeVisionAVR* juga dilengkapi dengan *tool* tambahan seperti *CodeWizardAVR* dan Programmer. *CodeWizardAVR* digunakan untuk membuat kerangka program dengan pendefinisian fungsi per langkah (*wizard*). Sedangkan Programmer digunakan untuk mentransfer program hasil kompilasi ke dalam chip AVR.

Pendefinisian port / pin dan fungsi-fungsi *library* yang digunakan dalam pemrograman dijelaskan sebagai berikut:

a. RTC

RTC DS1302 dihubungkan dengan pin 0, 1 dan 1 pada *PORTD* dan menggunakan *library* standar CodeVisionAVR sehingga didefinisikan sebagai berikut:

```
// DS1302 Real Time Clock functions
```

```
#asm
```

```
.equ __ds1302_port=0x12 ;PORTD
```

```
.equ __ds1302_io=1
```

```
.equ __ds1302_sclk=0
```

```
.equ __ds1302_rst=2
```

```
#endasm
```

```
#include <ds1302.h>
```

Fungsi-fungsi *library* ds1302.h yang digunakan dalam operasi *microcontroller* adalah:

- void rtc_init(unsigned char tc_on,unsigned char diodes,unsigned char res);

Fungsi ini digunakan untuk menginisialisasi chip DS1302

- void rtc_get_time(unsigned char *hour, unsigned char *min, unsigned char *sec);

Fungsi ini digunakan untuk mengambil data jam, menit, dan detik yang berjalan dalam RTC. Pointer

**hour, *min, *sec* harus menunjuk variabel jam, menit, dan detik yang digunakan untuk menampung hasil operasi.

- `void rtc_set_time(unsigned char hour, unsigned char min, unsigned char sec);`

Fungsi ini digunakan untuk mengeset jam, menit, dan detik pada RTC. Parameter *hour*, *min*, dan *sec* mewakili nilai jam, menit, dan detik yang hendak dimasukkan.

b. LCD

LCD dipasang pada port C dan menggunakan *library* standar CodeVisionAVR sehingga didefinisikan sebagai berikut:

```
// Alphanumeric LCD Module functions
#asm
    .equ __lcd_port=0x15 ;PORTC
#endasm
#include <lcd.h>
```

Fungsi-fungsi *library* *lcd.h* yang digunakan dalam operasi *microcontroller* adalah:

- `unsigned char lcd_init(unsigned char lcd_columns);`
Fungsi ini digunakan untuk menginisialisasi LCD, membersihkan tampilan dan meletakkan posisi tulis pada baris 0 dan kolom 0. Dalam perintah inisialisasi ini jumlah kolom LCD harus ditentukan; sehingga untuk LCD 16x2 diinisialisasi dengan `lcd_init(16);`. Fungsi inisialisasi ini akan memberikan nilai 1 apabila modul LCD terdeteksi dan memberikan nilai 0 apabila modul LCD tidak berhasil dideteksi. Fungsi inisialisasi ini harus dijalankan sebelum memanggil fungsi LCD yang lainnya.
- `void lcd_clear(void);`
Fungsi ini digunakan untuk membersihkan tampilan dan meletakkan posisi tulis pada baris 0 dan kolom 0.
- `void lcd_puts(char *str);`
Fungsi ini digunakan untuk menampilkan string `str`, yang terletak di SRAM pada posisi *cursor* LCD.

- `void lcd_putsf(char flash *str);`

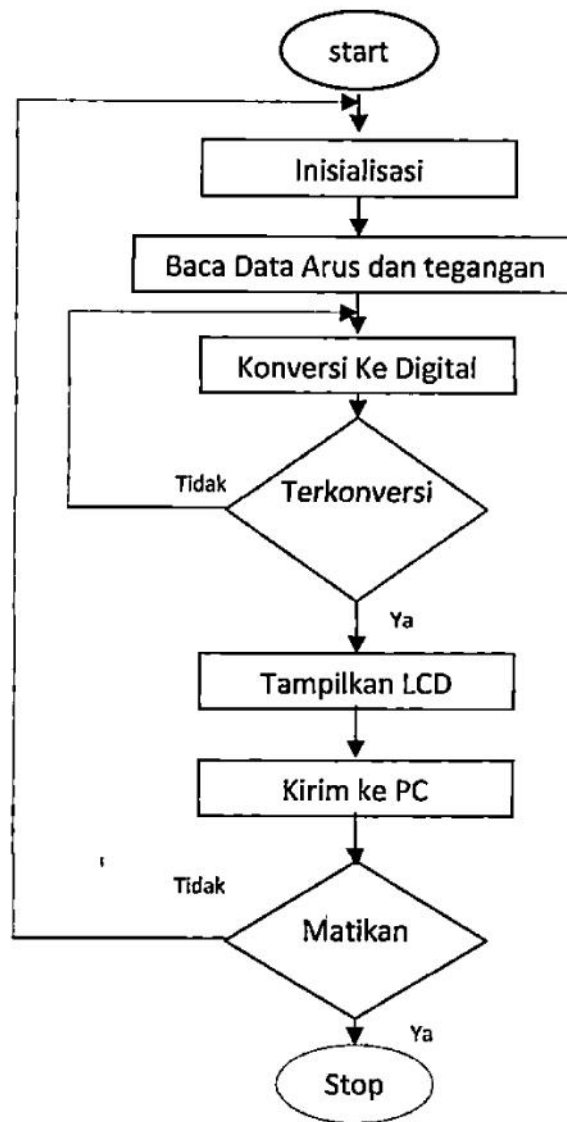
Fungsi ini digunakan untuk menampilkan string *str*, yang terletak di memori Flash, pada posisi *cursor* LCD.

- `void lcd_gotoxy(unsigned char x, unsigned char y);`

Fungsi ini digunakan untuk meletakkan posisi tulis LCD pada kolom *x* dan baris *y*. Nomor kolom dan baris dimulai dari 0 (nol).

3.1.2.2. Operasional Perangkat Lunak

Saat pertama kali sistem dinyalakan, *microcontroller* akan menjalankan program dari awal, yaitu dari inisialisasi hingga proses pengiriman data ke LCD. Urutan kerja program pada saat pertama kali dijalankan ditunjukkan oleh arah panah dalam Gambar 3.8. Pada proses tersebut dilakukan seluruh inisialisasi dan pengecekan semua komponen library yang dipergunakan. Setelah proses inisialisasi selesai dilanjutkan dengan pembacaan sensor arus dan tegangan. Dari data ini akan dilakukan proses perhitungan oleh mikrokontroler sehingga diperoleh nilai arus dan tegangan yang sebenarnya, nilai tegangan ini kemudian akan ditampilkan ke lcd dan komputer yang kemudian disimpan secara realtime melalui data logger. proses ini berlangsung terus-menerus selama tombol



Gambar 3.10 Alur Program

3.1.3 Perancangan Perangkat Lunak Delphi

1. Spesifikasi Perangkat Lunak

Pembuatan software dimulai dengan merancang tampilan yang diinginkan.

Untuk menambah keandalan sistem program delphi yang dirancang memiliki fitur

a. Pembacaan data serial

Perintah ini dikerjakan oleh delphi dengan statemen sebagai berikut

```
procedure TForm1.ComPortRxChar(Sender: TObject; Count: Integer);  
begin  
    ComPort.ReadStr(Data, Count);  
    Label1.Caption := Data;  
    val(Data,Conv,Kode);  
    Edit1.Text:=floattostr(Conv);  
    HConv:=Conv;  
    Edit2.Text:=floattostr(HConv);  
    //series1.AddXY(nomor,time);  
end;
```

b. Rekam data

Perintah ini dikerjakan oleh delphi dengan statemen sebagai berikut:

```
procedure TForm1.Button1Click(Sender: TObject);  
begin  
    if button1.Caption = 'Run' then  
        begin  
            button1.Caption := 'Stop';  
            timer1.Interval := StrToInt(edit3.Text)*1000;  
            timer1.Enabled := true;  
            comport.Connected:= true;  
            ComPort.Open;  
            save1.Enabled:= True;  
            open1.Enabled := True;  
        end
```



```

memo1.Lines.Insert(0,'Data Pemantauan Arus dan tegangan');
memo1.Lines.Insert(1,'Tanggal : '+datetostr(date));
memo1.Lines.Insert(2,'No Tegangan Arus Waktu');
memo1.Lines.Insert(3,'=====');
end
else if button1.Caption = 'Stop' then
begin
//Memo1.Lines.Clear;
button1.Caption := 'Run';
timer1.Enabled := false;
comport.Connected:=false;
ComPort.Close;
save1.Enabled:= False;
Open1.Enabled := False;
end;
end;

```

c. Grafik

Perintah ini dikerjakan oleh delphi dengan statemen sebagai berikut:

```

procedure TForm1.Timer1Timer(Sender: TObject);
begin
inc(nomor);
series1.AddXY(nomor,Hconv);
//form1.Series1.Add(nomor,"cired);
memo1.Lines.Add(inttostr(nomor)+' '
+floattostrf(Hconv,ffnumber,6,1)+' '
+timetostr(time));
if nomor = maksimum then

```

```

begin
  Chart1.BottomAxis.minimum:=maksimum;
  maksimum:= maksimum*2;
  Chart1.BottomAxis.maximum:=maksimum;
end;
Chart1.Refresh;
end;

```

d. Simpan data

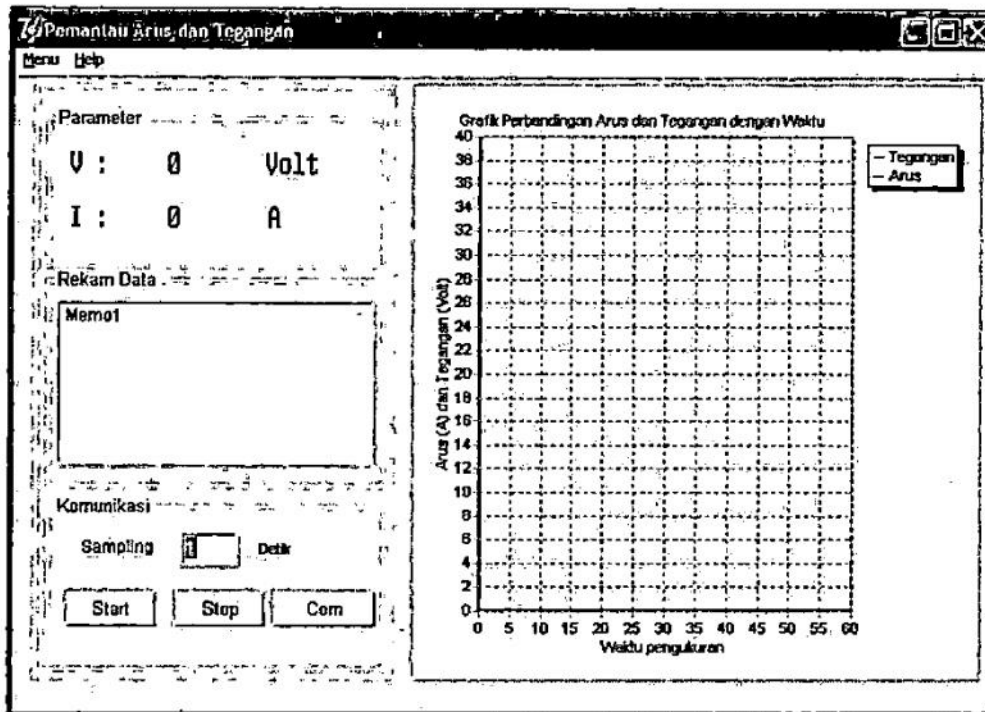
Perintah ini dikerjakan oleh delphi dengan statemen sebagai berikut:

```

procedure TForm1.Save1Click(Sender: TObject);
begin
  SaveDialog1.FileName := Form1.Caption;
  if savedialog1.Execute then
  begin
    memo1.Lines.SaveToFile(savedialog1.FileName+ '.txt');
    form1.Caption := savedialog1.FileName;
    chart1.SaveToMetafile(savedialog1.FileName+'Grafik.jpg');
    series1.Clear;
  end;
end;

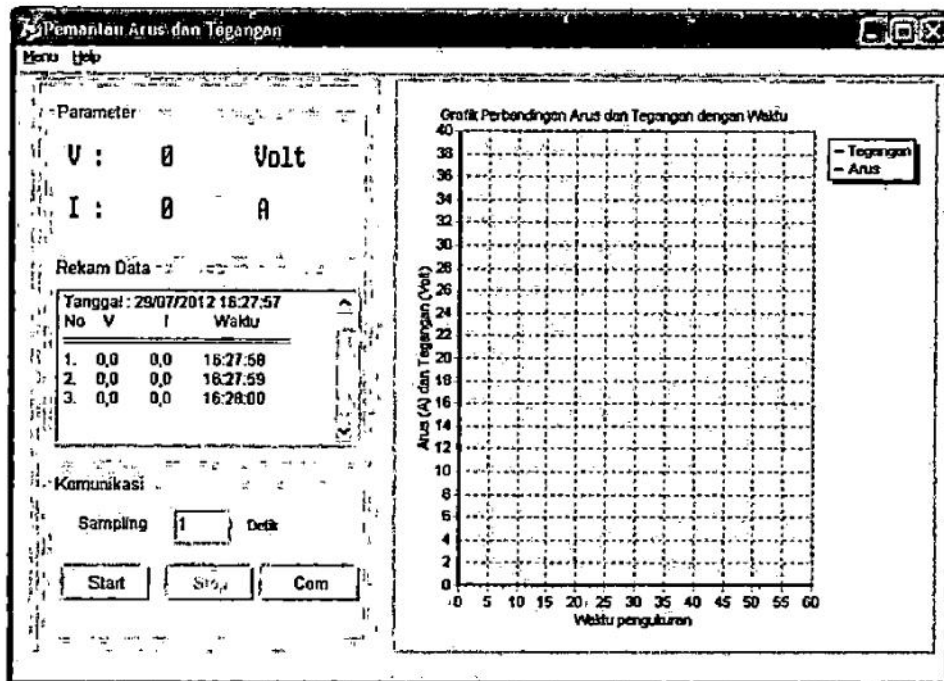
```

Urutan kerja program pada saat pertama kali dijalankan ditunjukkan oleh urutan tampilan – tampilan sebagai berikut :



Gambar 3.11 Tampilan Utama

Pada saat tombol run ditekan maka program akan menjalankan timer dan menampilkan nilai hasil pengukuran arus dan tegangan. Tampilan pengukuran arus dan tegangan dapat dilihat pada Gambar 3.12 ketika program *running*



Gambar 3.12 Program Running

Kolom rekam data berfungsi untuk menyimpan data arus dan tegangan jika save pada menu diklik sedangkan grafik akan menampilkan grafik nilai arus

6. Software pendukung AvrCodevision

- **Bahan**

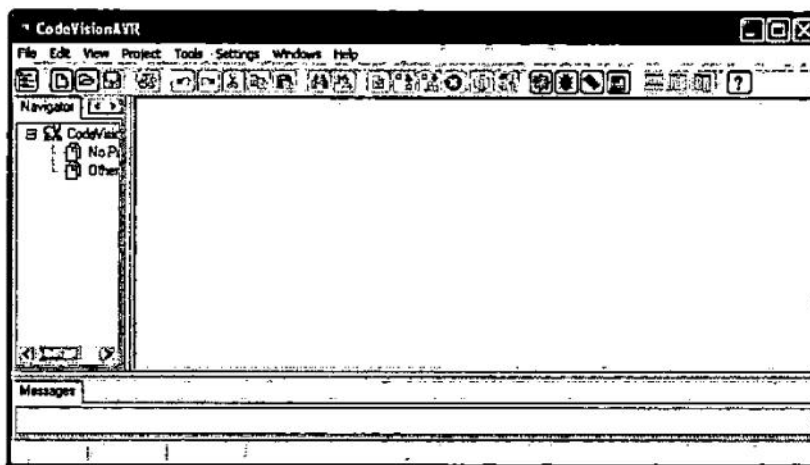
1. Sensor Arus ACS712ELC-20A
2. Catu Daya
3. AT Mega 16
4. LCD
5. RTC
6. Max232
7. Komponen pendukung (Elco, resistor, dioda, Kapasitor, dll)

3.2.2 Proses Pengerjaan

1. Membuat rangkaian pada program Proteus-ISIS
2. Membuat rangkaian PCB pada Proteus-ARES
3. Mengkonversi gambar PCB dari proteus ARES kedalam PCB
Yaitu dengan cara mencetak gambar kedalam kertas glosy kemudian gambar tersebut disetrika pada papan PCB polos, dengan demikian jalur PCB yang dibuat dapat menempel dalam papan PCB
4. Melarutan papan PCB yang telah tertempel papan dengan Ferrv

3.2.3 Proses Pengerjaan Perangkat Lunak AVRCodevison 1.25.9

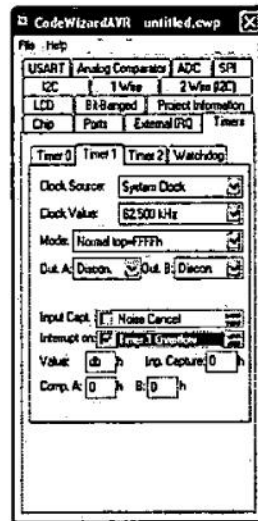
Untuk memulai pemrograman C dengan AVR ,Buka program CodeVisionAVR caranya klik Start- AllPrograms- CodeVisionAVR- CodeVisionAVR C Compiler.



Gambar 3.13. Jendela CodevisonAVR

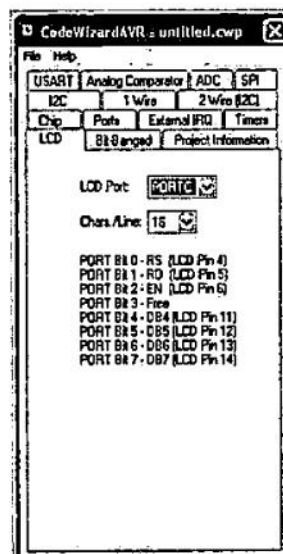
Untuk memulai membuat project baru klik File-New maka akan keluar 2 pilihan yaitu source atau project. Karena kita akan membuat project maka pilih Project. Setelah itu akan keluar kotak konfirmasi apakah kita akan menggunakan fasilitas CodeWizartAVR atau tidak. Pilih Yes.

Pilih chip yang digunakan yaitu ATMega16 dan clock yang digunakan 4MHz



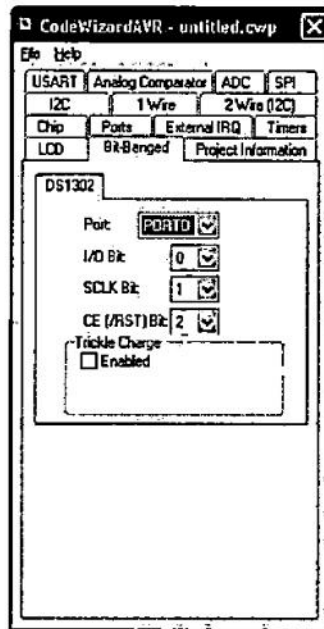
Gambar 3.16. Setting Timer yang digunakan

Pilih tab LCD pilih Port C

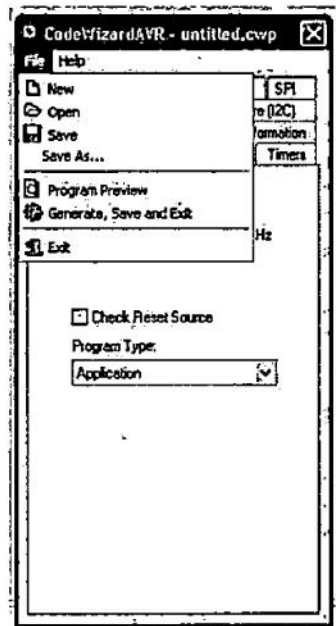


Gambar 3.17. Setting LCD

Pilih tab Bit Banged untuk mengatur RTC DS1302



Gambar 3.18. Setting RTC DS1302



Gambar 3.19. Penyimpanan Project

Setelah semua konfigurasi/ Setting Klik File- Generate and Save

and Exit. Simpan Project pada folder yang telah di tentukan

Kemudian akan dihasilkan tampilan sebagai berikut

```

1  //*****
2  This program was produced by the
3  Embedded Wizard FL22.2 Standard
4  Automatic Program Generator
5  © Copyright 1998-2008 Pavel Machar, WZ Software s.r.l.
6  http://www.wz-software.com
7
8  Project : DMA logger
9  Version :
10 Name    : 17/04/2012
11 Author  : Evi
12 Company : IIT Tech
13 Comment:
14
15 Chip type      : Altera
16 Program type  : Application
17 Clock frequency : 4,000000 MHz
18 Memory model  : Small
19 Embedded SWP size : 0
20 Main SWP size  : 128
21
22 *****
23
24 #include "hwapi.h"
25
26 // Declare your global variables here
27
28 void main(void)
29 {
30     // Declare your local variables here
  
```

Gambar 3.20. Project yang siap dikerjakan

3.3 Pengujian

3.3.1 Pengujian Catu Daya

Tegangan yang dibutuhkan untuk pembuatan alat adalah 5 volt. Untuk tegangan 5 volt menggunakan IC *regulator* LM7805.

Pengujian dilakukan dengan menggunakan multimeter dan adaptor

Tabel 3.1. Hasil Validasi Catu Daya untuk *Regulator* LM7805

No	Tegangan Input (DC)	Tegangan Output	Keterangan
1	2,79 volt	1.34 volt	Gagal
2	4,58 volt	3.21 volt	Gagal
3	5,87 volt	4,32 volt	OK
4	7,26 volt	4,95 volt	OK
5	9,45 volt	4,96 volt	OK

Berdasarkan data pengujian menunjukkan bahwa hubungan antara tegangan *input* dan tegangan *output* LM7805 bersifat stabil. Tegangan keluaran yang berbeda disebabkan oleh banyak hal seperti kondisi pengukuran yang berbeda saat pengambilan data atau dapat pula disebabkan oleh kondisi IC LM7805 itu sendiri karena hasil produksi pabrikan tak ada yang sempurna tepat mencapai tegangan 5 Volt. Tetapi berdasarkan data hasil pengukuran prosentase nilai *error* masih jauh dari batas nilai *error* yang diperbolehkan untuk IC LM7805 yaitu sebesar 4%

3.3.2 Pengujian Sensor Arus ACS712ELC-20A

Sensor ini memiliki tingkat akurasi yang cukup tinggi. Keluaran sensor berupa tegangan analog dengan akurasi 100mV/A. Tujuan pengujian yaitu untuk mengetahui tegangan keluaran dari sensor yang dari tegangan ini nantinya akan dijadikan acuan untuk kalibrasi alat. Hasil pengujian dapat dilihat pada tabel 3.2.

Tabel 3.2 Pengujian sensor ACS712ELC20A

No	Besar Arus (A)	Tegangan keluaran (Volt)
1	0	2.55
2	0.1	2.56
3	0.2	2.57
4	0.3	2.58
5	0.4	2.59
6	0.5	2.60
7	0.6	2.61
8	0.7	2.62
9	0.8	2.63
10	0.9	2.64
11	0.1	2.65

3.3.3 Pengujian Penampil

Pengujian penampil bertujuan untuk mengetahui bekerja tidaknya display pada LCD, bagian yang diuji termasuk ketajaman tampilan dan lampu backlight sehingga dapat bekerja maksimal

Cara pengujian dilakukan dengan cara memutar trimpot yang terhubung pada kaki VEE / Kontras LCD pada pin 3 untuk mengatur ketajaman karakter.



Gambar 3.21 Tampilan LCD

3.3.4 Pengujian Keseluruhan Alat

Pengujian kali ini melibatkan keseluruhan rangkaian yang telah dirancang. Pengujian rangkaian mikrontroler adalah pengujian terhadap semua fungsi yang ada dalam rangkaian tersebut yang telah terintegrasi dalam suatu rangkaian yang utuh, meliputi pengujian LCD , pengujian sensor, catudaya, penampil, dan pengujian tombol. Hasil dari pengujian ini secara keseluruhan dapat dilihat pada tabel sebagai berikut :



Gambar 3.22 Alat Keseluruhan

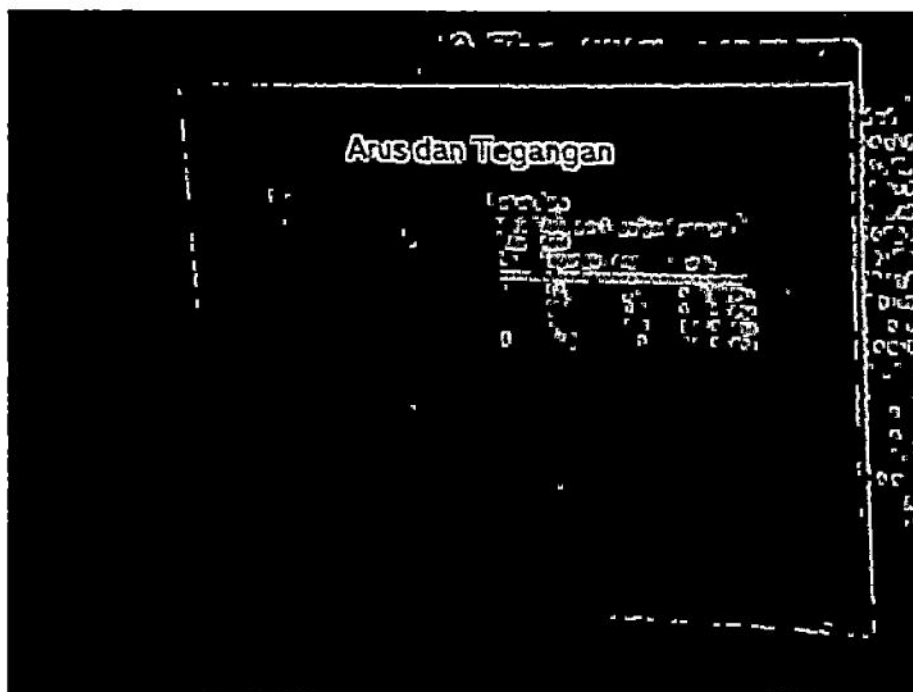
Tabel 3.3. Hasil Validasi Terhadap Fungsi Bagian-bagian Sistem

No	Kerja Alat	Kondisi	Deskripsi kerja	Status
1	Saklar reset	OFF	Alat bekerja normal	OK
		ON	Mereset <i>microcontroller</i> dan mengulang pelaksanaan program dari awal (<i>restart</i>)	OK
2	Push button	Tombol SET	Untuk memasukkan nilai jam dan menit	OK
		Tombol Mode	Memindah fungsi pengaturan dari normal, atur jam , atur menit dan kembali ke normal	OK

3	LCD sebagai penampil menu dan informasi		<ul style="list-style-type: none"> • Menampilkan jam • Menampilkan tegangan dan arus 	OK
4	Rangkaian serial		Mengirimkan data dari mikrokontroler ke komputer	OK
5	RTC	Perbarui nilai	Memperbarui nilai waktu, yaitu jam, menit, detik, tanggal, bulan,	OK
		<i>Run</i>	Menjalankan RTC dan memberikan data waktu ke sistem	OK
		Sumber tegangan utama terputus	Berjalan dengan sumber tenaga cadangan dari baterai.	OK

3.3.5 Analisis Perhitungan dan Pengukuran Alat

Pengujian alat dilakukan dengan simulasi menggunakan generator ac yang dapat diatur frekuensinya.



Gambar 3.23 Pengujian data logger

Tabel 3.4 Tabel hasil pengujian komputer dan penampil mikrokontroller

T (menit)	Tampilan PC				Penampil LCD	
	Vin	I (ampere)	Frekuensi (hz)	RPM	Arus (I)	V (V)
1	24	1.3	60	1787	1.3	24
2	24	1.3	60	1787	1.3	24
3	24	1.3	60	1786	1.3	24
4	23	1.2	59	1757	1.2	23
5	23	1.1	59	1757	1.1	23
6	22	0.8	58	1728	0.8	22
7	22	0.8	58	1728	0.8	22
8	22	0.8	58	1728	0.8	22
9	21	0.7	57	1699	0.7	21
10	21	0.7	57	1698	0.7	21
11	21	0.7	56	1669	0.7	21
12	21	0.7	55	1639	0.7	21
13	21	0.7	55	1639	0.7	21
14	20	0.7	54	1610	0.7	20
15	20	0.8	53	1581	0.8	20
16	20	0.7	52	1551	0.7	20
17	20	0.7	52	1551	0.7	20
18	20	0.6	52	1551	0.6	20
19	21	0.7	50	1490	0.7	21
20	27	0.7	48	1432	0.7	27
21	17	0.7	48	1435	0.7	17
22	24	0.7	60	1787	0.7	24
23	24	0.6	60	1786	0.6	24
24	23	0.7	59	1757	0.7	23
25	23	0.7	58	1728	0.7	23
26	24	0.7	58	1728	0.7	24
27	23	0.7	56	1669	0.7	23
28	22	0.6	56	1668	0.6	22
29	22	0.7	55	1640	0.7	22
30	22	0.7	55	1640	0.7	22
31	22	0.7	54	1610	0.7	22
32	23	0.7	53	1581	0.7	23
33	22	0.7	53	1581	0.7	22
34	22	0.7	54	1609	0.7	22
35	21	0.7	50	1491	0.7	21

Tampilan PC				Penampil LCD		
T (menit)	Vin	I (ampere)	Frekuensi (hz)	RPM	Arus (I)	V (V)
36	17	0.6	48	1435	0.6	17
37	16	0.4	47	1402	0.4	16
38	16	0.4	46	1372	0.4	16
39	15	0.3	45	1342	0.3	15
40	14	0.3	44	1316	0.3	14
41	14	0.3	43	1285	0.3	14
42	13	0.2	43	1285	0.2	13
43	13	0.2	43	1285	0.2	13
44	12	0.2	43	1287	0.2	12
45	20	0.5	50	1490	0.5	20
46	19	0.5	50	1491	0.5	19
47	24	0.5	60	1784	0.5	24
48	23	0.5	59	1757	0.5	23
49	23	0.5	58	1728	0.5	23
50	23	0.5	57	1699	0.5	23
51	23	0.5	56	1669	0.5	23
52	23	0.5	58	1728	0.5	23
53	23	0.5	57	1701	0.5	23
54	22	0.5	55	1643	0.5	22
55	22	0.5	54	1610	0.5	22
56	22	0.5	52	1550	0.5	22
57	21	0.4	50	1493	0.4	21
58	20	0.4	50	1492	0.4	20
59	20	0.4	50	1491	0.4	20
60	21	0.4	50	1492	0.4	21

Dari tabel 3.4 diketahui bahwa antara mikrokontroller dan PC memiliki data yang sama sehingga proses komunikasi dapat disimpulkan berjalan dengan normal