

## **TUGAS AKHIR**

# **RANCANG BANGUN ALAT PENGUKUR ARUS DAN TEGANGAN PADA SOLAR CELL BERBASIS MIKROKONTROLLER**

*Diajukan Guna Memenuhi Syarat-Syarat Untuk Memperoleh  
Gelar Sarjana Teknik (ST) Pada Program Studi Teknik Elektro  
Fakultas Teknik Universitas Muhammadiyah Sumatera Utara*

Oleh :

**MUHAMMAD RAMADHONI**

**1107220030**



**PROGRAM STUDI TEKNIK ELEKTRO  
FAKULTAS TEKNIK  
UNIVERSITAS MUHAMMADIYAH SUMATERA UTARA  
MEDAN  
2017**

## ABSTRAK

*Sel surya (photovoltaic cell) bekerja dengan menangkap sinar matahari menjadi energi listrik. Indonesia sendiri, sebuah Negara yang dilewati garis khatulistiwa dan menerima panas matahari lebih banyak daripada Negara lain, mempunyai potensi yang sangat besar untuk mengembangkan pembangkit listrik tenaga surya sebagai alternatif batubara dan diesel sebagai pengganti bahan bakar fosil, yang bersih, tidak berpolusi, aman dan persediannya tidak terbatas (Rotib, 2007). Perkembangan teknologi Solar Cell juga memicu sebuah control untuk pengisian baterai dimana jika tidak demikian, baterai yang digunakan tidak akan bisa bertahan dikarenakan selama solar cell mendapat sinar matahari, solar cell tersebut akan terus mengisi baterai yang digunakan. Pada penelitian ini dikonsentrasikan pada perancangan sebuah sistem yang dapat mengontrol pengisian baterai tersebut menggunakan solar cell. Jika tegangan 12 volt yang dihasilkan oleh baterai atau solar cell, maka tegangan yang diterima oleh pin analog arduino adalah 2 volt. Dimana teknologi dari mikrokontroler memungkinkan untuk merancang sebuah sistem tersebut. Menggunakan sebuah sensor yang dapat mengukur arus seperti sensor ACS712 yang pengukurannya cukup akurat untuk menampilkan arus yang digunakan pada beban dan arus yang masuk untuk mengecaskan baterai. Untuk mengontrol tegangan pengisian yang akan diberikan pengontrol menggunakan PWM (Pulse Width Modulation) yang ditentukan oleh perbedaan antara voltase baterai dan tegangan set titik pengisian yang telah ditentukan pada program, frekuensi dasar PWM yang digunakan adalah 490.20 Hz dengan siklus tugas 0-100%.*

**Kata kunci :** *Solar Cell, Mikrokontroler, Sensor ACS712, PWM(Pulse Width Modulation).*

## ABSTRACT

*Solar cells (photovoltaic cells) work by capturing sunlight into electrical energy. Indonesia itself, a country passed by the equator and receiving more solar heat than any other country, has tremendous potential to develop solar power as an alternative to coal and diesel in place of fossil fuels, which are clean, polluted, secure and supplyable not limited (Rotib, 2007). The development of Solar Cell technology also triggers a control for charging the battery where otherwise, the battery used will not survive because as long as the solar cell gets sunlight, the solar cell will continue to charge the battery used. In this study concentrated on the design of a system that can control the charging of these batteries using a solar cell. If a 12 volt voltage is generated by a battery or solar cell, then the voltage received by the arduino analog pin is 2 volts. Where technology from microcontroller makes it possible to design a such system. Using a sensor that can measure current such as ACS712 sensor whose measurement is accurate enough to display the current used on the load and the incoming current to charge the battery. To control the charging voltage that the controller will use the PWM (Pulse Width Modulation) determined by the difference between the battery voltage and the set point voltage of the pre-determined point on the program, the base frequency of PWM used is 490.20 Hz with a 0-100% duty cycle.*

**Keywords :** *Solar Cell, Microcontroller, ACS712 Sensor, PWM (Pulse Width Modulation).*

## KATA PENGANTAR

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

*Assalamu'alaikum Wr. Wb.*

Puji syukur kehadiran ALLAH SWT atas rahmat dan karunianya yang telah menjadikan kita sebagai manusia yang beriman dan insya ALLAH berguna bagi semesta alam. Shalawat berangkaikan salam kita panjatkan kepada junjungan kita Nabi besar Muhammad SAW yang mana beliau adalah suri tauladan bagi kita semua dan telah membawa kita dari zaman kebodohan menuju zaman yang penuh dengan ilmu pengetahuan.

Tulisan ini dibuat sebagai tugas akhir untuk memenuhi syarat dalam meraih gelar kesarjanaan pada Fakultas Teknik Elektro Universitas Muhammadiyah Sumatera Utara. Adapun judul tugas akhir ini adalah “**Rancang Bangun Alat Pengukur Arus Dan Tegangan Pada Solar Cell Berbasis Mikrokontroler**”.

Selesaiannya penulisan tugas akhir ini tidak terlepas dari bantuan dan bimbingan dari berbagai pihak, oleh karena itu penulis menyampaikan rasa terima kasih yang sebesar-besarnya kepada :

1. Ayahanda Agustin dan Ibunda Sri Murti, yang dengan cinta kasih & sayang setulus jiwa mengasuh, mendidik dan membimbing dengan segenap ketulusan hati tanpa mengenal kata lelah, penulis juga mengucapkan terimah kasih kepada Adik saya Hannifathu Sya'diah dan Rivaldi Alwi Ansyari serta segenap keluarga yang telah memberikan do'a dan dukungan kepada penulis sampai saat ini.

2. Bapak Rahmatullah, S.T., MSc. Selaku Dekan Fakultas Teknik Universitas Muhammadiyah Sumatera Utara.
3. Bapak Faisal Irsan Pasaribu, S.T., M.T, Selaku Ketua Program Studi Teknik Elektro di Fakultas Teknik Universitas Muhammadiyah Sumatera Utara.
4. Bapak Partaonan Harahap, S.T., M.T, selaku Sekretaris Program Studi Teknik Elektro Di Fakultas Teknik Elektro Universitas Muhammadiyah Sumatera Utara.
5. Bapak Dr. Ir. Suwarno, M.T, selaku pembimbing I yang telah memberikan wawasan dan arahan yang membangun pada penyusunan tugas akhir ini.
6. Ibu Rohana, S.T., M.T., selaku pembimbing II yang telah memberikan wawasan dan arahan yang membangun pada penyusunan tugas akhir ini.
7. Segenap Bapak & Ibu dosen serta pegawai di Fakultas Teknik Elektro Universitas Muhammadiyah Sumatera Utara.
8. Segenap teman-teman sejawat dan seperjuangan Fakultas Teknik, khususnya Fakultas Teknik Elektro angkatan 2011 yang selalu memberi dukungan dan motivasi kepada penulis.
9. Kawan-kawan seperjuangan yang telah memberikan do'a, dukungan dan motivasi kepada penulis.
10. Segenap sahabat-sahabat seperjuangan saya, Apriandi, S.T, Mahasa Nst, S.T, Yatno, S.T, Abdi Gunawan, S.T, Saat Jati Mulyo, S.T, Zaka Fadhilah Siregar, S.T, yang selalu membantu dan memberikan do'a serta dukungan dan motivasi kepada penulis.

Penulis telah berupaya semaksimal mungkin dalam menyelesaikan skripsi ini, namun penulis menyadari bahwa skripsi ini masih jauh dari kata sempurna,

hal ini di sebabkan keterbatasan kemampuan penulis. Oleh karena itu penulis sangat mengharapkan kritik dan saran yang membangun dari segenap pihak untuk memperbaiki skripsi ini.

Akhir kata penulis mengharapkan semoga skripsi ini dapat menambah dan memperkaya lembar khazanah pengetahuan bagi para pembaca sekalian dan khususnya bagi penulis sendiri. Sebelum dan sesudahnya penulis mengucapkan terima kasih.

Medan, 19 Desember 2017

Penulis

Muhammad Ramadhoni  
1107220030

## DAFTAR ISI

<b>ABSTRAK .....</b>	<b>i</b>
<b>KATA PENGANTAR .....</b>	<b>ii</b>
<b>DAFTAR ISI .....</b>	<b>v</b>
<b>DAFTAR GAMBAR .....</b>	<b>viii</b>
<b>DAFTAR TABEL .....</b>	<b>x</b>
<b>DAFTAR LAMPIRAN .....</b>	<b>xi</b>
<b>BAB I. PENDAHULUAN</b>	
1.1. Latar Belakang.....	1
1.2. Rumusan Masalah .....	4
1.3. Tujuan Penelitian.....	4
1.4. Batasan Masalah.....	4
1.5. Manfaat Penelitian.....	5
1.6. Metode Penelitian.....	5
1.7. Sistematika Penulisan.....	7
<b>BAB II. TINJAUAN PUSTAKA</b>	
2.1. Tinjauan Pustaka Relevan .....	8
2.2. Sejarah Solar Cell .....	8
2.2.1. Solar Cell 10 WP .....	10
2.3. Mikrokontroler ATmega328.....	11
2.3.1. Konfigurasi Pin ATmega328.....	14
2.3.2. Pemrograman .....	16
2.4. Resistor .....	16
2.4.1. Resistor Statis .....	17

2.5. Transistor.....	19
2.6. Kapasitor.....	21
2.6.1. Kapasitor Keramik .....	22
2.6.2. Kapasitor Elektrolit.....	23
2.7. I2C LED ( <i>Light Emiting Diode</i> ) .....	24
2.8. Sensor .....	25
2.8.1. Sensor Arus ACS712 .....	25
2.8.1.1. Konfigurasi Pin Sensor Arus ACS712 .....	27
2.8.2. Sensor LM35.....	28
2.8.2.1. Struktur Sensor LM35 .....	28
2.8.2.2. Karakteristik Sensor LM35 .....	30
2.9. <i>Power Supply</i> .....	32
2.9.1. Cara Kerja <i>Power Supply</i> .....	32
2.9.2. <i>Rectifier</i> (Penyearah).....	33
2.9.3. <i>Filter</i> (Penyaring).....	34
2.9.4. <i>Voltage Regulator</i> .....	34
2.10. <i>Liquid Crystal Display</i> (LCD) .....	35
2.11. i2C LCD PCF8574AT.....	36

### **BAB III METODE PENELITIAN**

3.1. Jadwal dan Lokasi Penelitian .....	38
3.2. Peralatan dan Bahan Penelitian .....	38
3.3. Data Perancangan.....	40
3.3.1. Daftar Input dan Output yang Digunakan.....	40
3.3.2. Perancangan Program Arduino .....	40
3.4. Tahapan Perancangan Alat.....	46

3.4.1. Perancangan Blok Diagram Sistem.....	47
3.4.2. Diagram Alir Perancangan Sistem .....	48
3.5 Prosedur Uji Coba Rangkaian .....	50

#### **BAB IV ANALISA DAN HASIL PEMBAHASAN**

4.1. Hasil Uji Coba Rangkaian. ....	51
4.1.1. Pengujian Sensor Arus ACS712 .....	51
4.1.2. Pengujian Sensor Suhu LM35.....	53
4.2. Analisa Software Arduino-Uno.....	55
4.3. Pengujian Secara Keseluruhan .....	59

#### **BAB V PENUTUP**

5.1. Kesimpulan .....	63
5.2. Saran.....	65

#### **DAFTAR PUSTAKA**

#### **LAMPIRAN**



## DAFTAR GAMBAR

Gambar 2.1 Solar Cell 10 WP.....	11
Gambar 2.2 Blok Diagram Mikrokontroler ATmega328.....	13
Gambar 2.3 Konfigurasi Pin ATmega328.....	14
Gambar 2.4 Resistor Tetap.....	17
Gambar 2.5 Resistor Empat Gelang Warna.....	18
Gambar 2.6 Transistor Bipolar dan Lambang Transistor.....	20
Gambar 2.7 Bias Transistor.....	21
Gambar 2.8 Transistor NPN.....	21
Gambar 2.9 Prinsip Dasar Kapsitor.....	22
Gambar 2.10 Kapasitor Keramik.....	23
Gambar 2.11 Elektrolit Capasitor (Elco).....	23
Gambar 2.12 LED ( <i>Light Emiting Diode</i> ).....	24
Gambar 2.13 Sensor Arus ACS712.....	26
Gambar 2.14 Grafik Perubahan Tegangan pada Sensor Arus ACS712.....	27
Gambar 2.15 Sensor LM35 dan Konfigurasi Pin LM35.....	28
Gambar 2.16 Skematik Rangkaian Dasar Sensor Suhu.....	29
Gambar 2.17 Blok DC <i>Power Supply</i> .....	33
Gambar 2.18 <i>Rectifier</i> .....	34
Gambar 2.19 <i>Filter</i> (Penyaring).....	34
Gambar 2.20 LCD 4x20 CHARS.....	36
Gambar 2.21 Modul i2C dengan Chip PCF8574AT.....	37
Gambar 2.22 Jalur Komunikasi data chip PCF8574.....	37
Gambar 3.1 Software Arduino 1.0.5.....	41
Gambar 3.2 Menu File Baru.....	41
Gambar 3.3 Pemilihan Board Arduino.....	42
Gambar 3.4 Membuat File Projek Baru.....	43
Gambar 3.5 Proses Verify Program.....	44
Gambar 3.6 Proses Upload Program Ke Arduino.....	45
Gambar 3.7 Blok Diagram Sistem.....	47
Gambar 3.8 Flowchart Sistem Perangkat.....	48
Gambar 4.1 Pengujian Sensor Arus ACS712.....	52
Gambar 4.2 Pengujian Sensor Suhu LM35.....	54
Gambar 4.3 Rangkaian Simulasi Keseluruhan.....	59
Gambar 4.4 Hasil Uji Coba Rangkaian Keseluruhan.....	62

## DAFTAR TABEL

Tabel 2.1.a Konfigurasi PORTB.....	14
Tabel 2.1.b Konfigurasi PORTC.....	15
Tabel 2.1.c Konfigurasi PORTD.....	15
Tabel 2.2 Kode Warna Resistor .....	17
Tabel 2.3 Perbandingan Transistor Bipolar dan Unipolar .....	19
Tabel 2.4 Konfigurasi Pin Sensor Arus ACS712 .....	27
Tabel 4.1 Hasil Uji Coba Sensor Suhu LM35.....	43

## DAFTAR LAMPIRAN

### LAMPIRAN PROGRAM

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>

#define SOL_ADC    A0
#define BAT_ADC    A1
#define CURRENT_ADC A2
#define TEMP_ADC   A3
#define AVG_NUM    10
#define BAT_MIN    10.5
#define BAT_MAX    15.0
#define BULK_CH_SP 14.4
#define FLOAT_CH_SP 13.6
#define LVD        11.5
#define PWM_PIN    3
#define LOAD_PIN   2
#define BAT_RED_LED 5
#define BAT_GREEN_LED 6
#define BAT_BLUE_LED 7
#define LOAD_RED_LED 8
#define LOAD_GREEN_LED 9

//=====Deklarasi=====\\
//Pembentuk tampilan karakter khusus\\

byte                                solar[8]                                =
{0b01010,0b10101,0b01010,0b10101,0b01010,0b10101,0b00000};

byte                                battery[8]                            =
{0b01110,0b11011,0b10001,0b10001,0b10001,0b10001,0b10001,0b11111};

byte                                energy[8]                              =
{0b00010,0b00100,0b01000,0b11111,0b00010,0b00100,0b01000,0b00000};

byte                                temp[8]                                =
{0b00100,0b01010,0b01010,0b01110,0b01110,0b11111,0b11111,0b01110};

byte                                charge[8]                             =
{0b01010,0b10001,0b10001,0b10001,0b01110,0b00100,0b00100,};

byte                                not_charge[8]                         =
{0b00000,0b10001,0b01010,0b00100,0b01010,0b10001,0b00000,0b00000,};

//=====Deklarasi Variabel=====\\

float solar_volt = 0;
```

```

float bat_volt = 0;
float load_current = 0;
int temperature = 0;
int temp_change = 0;
float system_volt = 0;
float bulk_charge_sp = 0;
float float_charge_sp = 0;
float charge_status = 0;
float load_status = 0;
float error = 0;
float Ep = 0;
int duty = 0;
float lvd;
float msec = 0;
float last_msec = 0;
float elapsed_msec = 0;
float elapsed_time = 0;
float ampSecs = 0;
float ampHours = 0;
float watts = 0;
float wattSecs = 0;
float wattHours = 0;

LiquidCrystal_I2C lcd(0x38, 2, 1, 0, 4, 5, 6, 7, 3, POSITIVE); //inisialisasi
address I2C
//untuk simulasi pakai 0x38, untuk flash pakai 0x3F

//=====Main Program Start=====\\

void setup() {
  // put your setup code here, to run once:
  Serial.begin(9600);
  pinMode(BAT_RED_LED, OUTPUT);
  pinMode(BAT_GREEN_LED, OUTPUT);
  pinMode(BAT_BLUE_LED, OUTPUT);
  pinMode(LOAD_RED_LED, OUTPUT);
  pinMode(LOAD_GREEN_LED, OUTPUT);
  pinMode(PWM_PIN, OUTPUT);
  pinMode(LOAD_PIN, OUTPUT);
  digitalWrite(PWM_PIN, LOW);
  digitalWrite(LOAD_PIN, LOW);
  lcd.begin(20,4);
  lcd.backlight();
  lcd.createChar(1, solar);
  lcd.createChar(2, battery);
  lcd.createChar(3, energy);
  lcd.createChar(5, temp);
  lcd.createChar(6, charge);

```

```

    lcd.createChar(7, not_charge);
    lcd.clear();
}

void loop() {
    // put your main code here, to run repeatedly:
    read_data();
    system_voltage();
    setpoint();
    charge_cycle();
    power();
    load_control();
    led_indication();
    print_data();
    lcd_display();
}

int read_adc(int adc_parameter){
    int sum = 0;
    int sample;
    for(int i=0; i<AVG_NUM; i++){
        sample = analogRead(adc_parameter);
        sum += sample;
        delayMicroseconds(50);
    }
    return(sum / AVG_NUM);
}

void read_data(void){
    solar_volt = read_adc(SOL_ADC)*0.00488*(120/20);
    bat_volt  = read_adc(BAT_ADC)*0.00488*(120/20);
    load_current = (read_adc(CURRENT_ADC)*.0488 - 25);
    temperature = read_adc(TEMP_ADC)*0.00488*100;
}

void power(void){
    msec = millis();
    elapsed_msec = msec - last_msec;
    elapsed_time = elapsed_msec / 1000.0;
    watts = load_current * bat_volt;
    ampSecs = (load_current*elapsed_time);
    wattSecs = ampSecs * bat_volt;
    ampHours = ampHours + ampSecs/3600;
    wattHours = wattHours + wattSecs/3600;
    last_msec = msec;
}

```

```

void print_data(void)
{
  delay(100);
  Serial.print("Solar Panel Voltage: ");
  Serial.print(solar_volt);
  Serial.println("V");
  Serial.print("Battery Voltage: ");
  Serial.print(bat_volt);
  Serial.println("V");
  Serial.print("System Voltage: ");
  Serial.print(system_volt);
  Serial.println("V");
  Serial.print("Charge Set Point:");
  Serial.println(bulk_charge_sp);
  //Serial.print("Temperature:");
  //Serial.print(temperature);
  //Serial.println("C");
  Serial.print("Load Current: ");
  Serial.print(load_current);
  Serial.println("A");
  Serial.print("Power: ");
  Serial.print(watts);
  Serial.println("W");
  Serial.print("Energy: ");
  Serial.print(wattHours);
  Serial.println("WH");
  Serial.print("Duty Cycle :");
  if (charge_status==1)
  {
    Serial.println("99%");
    Serial.println("BULK CHARGING");
  }
  else if (charge_status==2)
  {
    Serial.print(Ep);
    Serial.println("%");
    Serial.println("FLOAT CHARGING");
  }
  else
  {
    Serial.println("0%");
    Serial.println("NOT CHARGING");
  }
  if(load_status==1)
  {
    Serial.println("LOAD IS CONNECTED");
  }
  else

```

```

{
  Serial.println("LOAD IS DISCONNECTED");
}

Serial.println("*****");
}

void system_voltage(void){
  if((bat_volt > BAT_MIN) && (bat_volt < BAT_MAX)){
    system_volt = 12;
  }
  else if((bat_volt > BAT_MIN/2) && (bat_volt < BAT_MAX/2)){
    system_volt = 6;
  }
}

void setpoint(void){
  temp_change = temperature-25.0;
  if(system_volt==12){
    bulk_charge_sp = BULK_CH_SP-(0.030*temp_change);
    float_charge_sp = FLOAT_CH_SP-(0.030*temp_change);
    lvd = LVD;
  }
  else if(system_volt==6){
    bulk_charge_sp = (BULK_CH_SP/2)-(0.015*temp_change);
    float_charge_sp = (FLOAT_CH_SP/2)-(0.015*temp_change);
    lvd=LVD/2;
  }
}

void charge_cycle(void){
  if(solar_volt > bat_volt && bat_volt <= bulk_charge_sp){
    if(bat_volt <= float_charge_sp){
      charge_status = 1;
      duty = 252.45;
      analogWrite(PWM_PIN,duty);
    }
    else if(bat_volt > float_charge_sp && bat_volt <= bulk_charge_sp){
      charge_status = 2;
      error = (bulk_charge_sp - bat_volt);
      Ep = error * 100;
      if(Ep < 0){
        Ep = 0;
      }
      else if(Ep>100){
        Ep = 100;
      }
      else if(Ep>0 && Ep <= 100){

```

```

        duty = (Ep*255)/100;
    }
    analogWrite(PWM_PIN,duty);
}
}
else{
    charge_status = 0;
    duty = 0;
    analogWrite(PWM_PIN,duty);
}
}

void load_control(){
    if(solar_volt < 5){
        if(bat_volt > lvd){
            load_status = 1;
            digitalWrite(LOAD_PIN, HIGH);
        }
        else if(bat_volt < lvd){
            load_status = 0;
            digitalWrite(LOAD_PIN, LOW);
        }
    }
    else{
        load_status = 0;
        digitalWrite(LOAD_PIN, LOW);
    }
}

void led_indication(void){
    battery_led();
    load_led();
}

void battery_led(void){
    if((bat_volt > system_volt) && (bat_volt < bulk_charge_sp)){
        leds_off_all();
        digitalWrite(BAT_GREEN_LED, LOW);
    }
    else if(bat_volt >= bulk_charge_sp){
        leds_off_all();
        digitalWrite(BAT_BLUE_LED, LOW);
    }
    else if(bat_volt < system_volt){
        leds_off_all();
        digitalWrite(BAT_RED_LED, LOW);
    }
}
}

```



```

void load_led(){
  if(load_status==1){
    digitalWrite(LOAD_GREEN_LED, HIGH);
  }
  else if(load_status==0){
    digitalWrite(LOAD_RED_LED, HIGH);
  }
}

```

```

void leds_off_all(void){
  digitalWrite(BAT_RED_LED, HIGH);
  digitalWrite(BAT_GREEN_LED, HIGH);
  digitalWrite(BAT_BLUE_LED, HIGH);
  digitalWrite(LOAD_RED_LED, LOW);
  digitalWrite(LOAD_GREEN_LED, LOW);
}

```

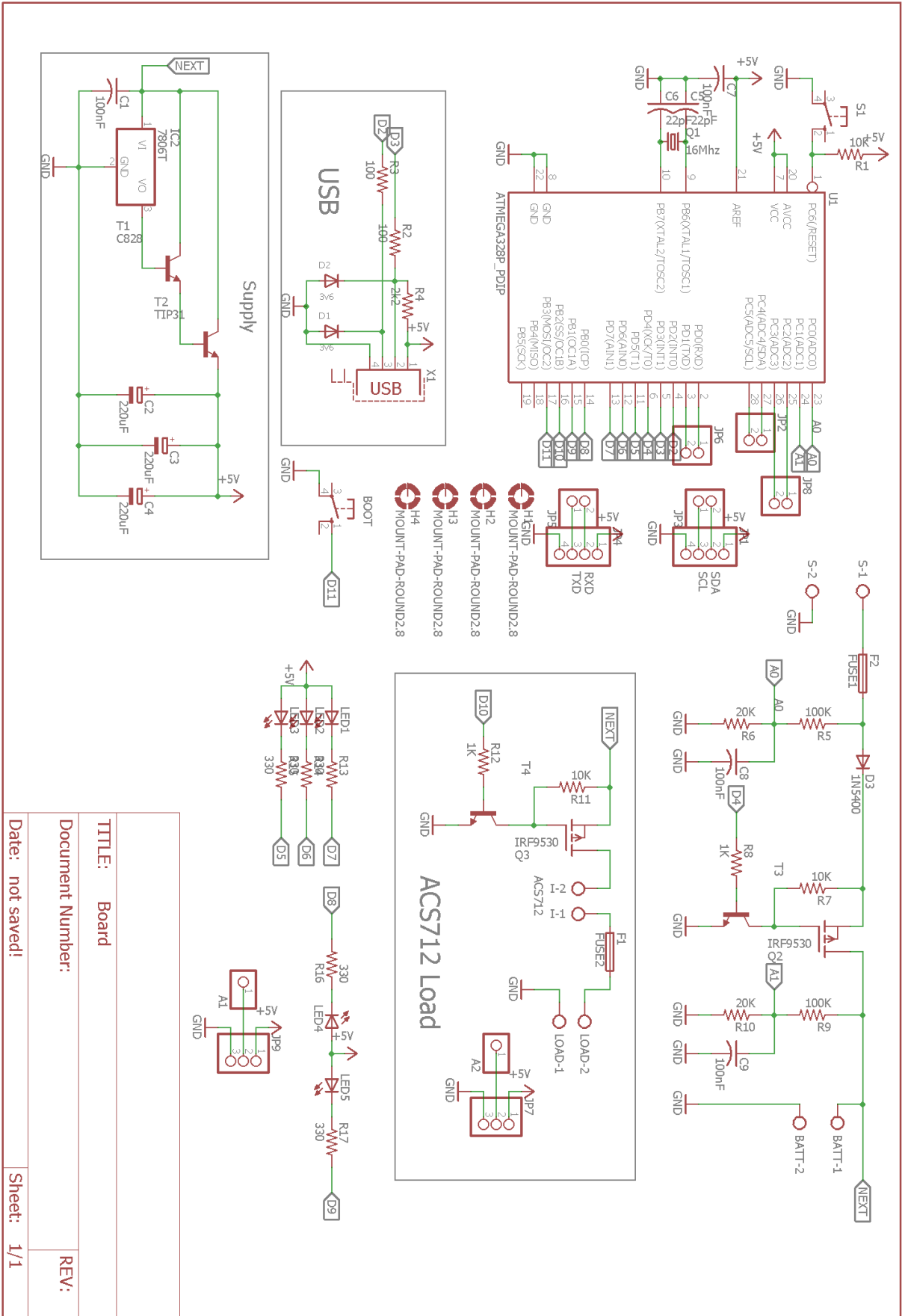
```

void lcd_display(){
  lcd.setCursor(0,0);
  lcd.write(1);
  lcd.setCursor(2,0);
  lcd.print(solar_volt);
  lcd.print("V");
  lcd.setCursor(14,0);
  lcd.write(5);
  lcd.setCursor(16,0);
  lcd.print(temperature);
  lcd.write(0b11011111);
  lcd.print("C");
  lcd.setCursor(0,1);
  lcd.write(2);
  lcd.setCursor(2,1);
  lcd.print(bat_volt);
  lcd.print("V");
  lcd.setCursor(14,1);
  lcd.write(2);
  if((charge_status==1) | (charge_status==2)){
    lcd.write(6);
  }
  else{
    lcd.write(7);
  }
  lcd.setCursor(0,2);
  lcd.write(3);
  lcd.setCursor(2,2);
  lcd.print(load_current);
  lcd.print("A");
}

```

```
lcd.setCursor(13,2);  
lcd.print(watts);  
lcd.print("W");  
lcd.setCursor(0,3);  
lcd.print("Energy:");  
lcd.print(wattHours);  
lcd.print("WH");  
}
```

Schematic Diagram :



# **BAB I**

## **PENDAHULUAN**

### **1.1 Latar Belakang**

Memasuki era abad industrialisasi dan diperkirakan pada tahun 2050 kebutuhan energi dunia meningkat dua kali atau hampir mendekati angka tiga kali lipat sebanding dengan pertumbuhan populasi penduduk secara global. Beban ini dirasakan dari tahun ke tahun semakin bertambah berat, terutama ketergantungan terhadap sumber energi fosil sebagai pemasok energi utama masih begitu tinggi, situasi ini menyebabkan kelangkaan dan persediaan sumber energi fosil dari tahun ke tahun yang semakin menipis dan amat terbatas. Kondisi ini turut memperkuat terjadinya krisis energi dan secara tidak langsung membuat harga energi fosil menjadi mahal. Permasalahan ini dirasakan hampir sebagian besar negara-negara di seluruh dunia. Persediaan energi fosil yang terbatas dengan kebutuhan yang meningkat. Selain itu dampak eksploitasi energi fosil yang berlebihan khususnya minyak bumi terbukti telah membawa dampak polusi udara serta pengotoran lingkungan menjadi panorama sebagai ciri khas kota-kota besar negara berkembang saat ini, terutama seperti kota-kota besar di negara-negara berkembang seperti Jakarta, New Delhi dan kota-kota besar di belahan benua Amerika Latin seperti Rio de Janeiro.

Pemanfaatan energi cahaya matahari pada setiap zaman semakin meningkat seiring dengan pengetahuan yang kita dapatkan. Salah satu pemanfaatan energi cahaya matahari adalah Pembangkit Listrik Tenaga Surya (PLTS) yang memanfaatkan energi foton cahaya matahari menjadi energi listrik.

Indonesia sendiri, sebuah Negara yang dilewati oleh garis khatulistiwa dan menerima panas matahari yang lebih banyak daripada Negara lain, mempunyai potensi yang sangat besar untuk mengembangkan pembangkit listrik tenaga surya sebagai alternatif batubara dan diesel sebagai pengganti bahan bakar fosil, yang bersih, tidak berpolusi, aman dan persediannya tidak terbatas (Rotib, 2007).

Sel surya (photovoltaic cell) bekerja dengan menangkap sinar matahari oleh sel-sel semikonduktor untuk diubah menjadi energi listrik. Sel-sel ini termuat dalam panel-panel yang ukurannya dapat disesuaikan dengan keperluannya, apakah untuk rumah tangga, perkantoran atau pembangkit listrik skala besar (Sungkar, 2006).

Sejalan dengan kemajuan industri yang sangat pesat, maka perkembangan akan listrik sebagai sumber energi dalam pelaksanaan industri dan perumahan juga semakin berkembang. Dalam perkembangan barang-barang elektronik sendiri sangat pesat, beberapa factor pendukungnya tentu saja perkembangan alat-alat elektronika yang semakin beragam.

Pada pagi hari pukul 6.00 WIB tingkat kelembaban besar yaitu 88% dan terjadi pengembunan sambil menurunkan partikel-partikel padatan akibat polusi kendaraan bermotor dan industri ke permukaan bumi, sehingga pada saat ini kondisi atmosfir mempunyai kebeningan yang tinggi dan langit biru. Fenomena tersebut mengakibatkan pada pagi hari yang cerah pukul 9.00 WIB sel surya memiliki efisiensi terbesar yaitu dengan efisiensi 10%. Pada siang hari partikel-partikel padatan akibat polusi kembali ke angkasa, dengan meningkatnya temperature udara gerakan partikel semakin hebat, sehingga meningkatkan hamburan radiasi surya yang masuk ke bumi. Hal ini mengakibatkan difusi ratio

membesar dimana jumlah radiasi difusi lebih besar radiasi langsung, dan efisiensi sel surya pada pukul 12.00 WIB adalah sebesar 9%. Lebih rendah dari pagi hari. Pada sore hari akibat terjadi penguapan pada siang hari dan semakin meningkatnya partikel padatan polusi di udara, sehingga indek kecerahan terendah dimana tampak banyak awan. Selain itu radiasi surya global sangat kecil, sehingga pada sore hari sekitar pukul 17.00 WIB dengan efisiensi 3%, kemampuan sel surya menurun secara drastic (yuhardi, 2002).

Solar cell merupakan pembangkit listrik yang mampu mengkonversi sinar matahari menjadi arus listrik. Energi matahari sesungguhnya merupakan sumber energi yang paling menjanjikan mengingat sifatnya yang berkelanjutan (*sustainable*) serta jumlahnya yang sangat besar. Solar cell tidak memiliki eksekusi suara seperti pada pembangkit tenaga angin serta dapat dipasang pada hampir seluruh daerah karena hampir setiap lokasi di belahan dunia ini menerima sinar matahari. Dengan berbagai keunggulan ini maka tidak heran jika negara-negara maju berlomba mengembangkan solar cell agar dapat dihasilkan teknologi pemanfaatan solar cell yang berharga ekonomis.

Pada penelitian dan fenomena yang telah diuraikan pada uraian diatas, dengan keadaan alam yang demikian, saya tertarik untuk membuat penelitian mengenai *charger controller* pada *solar cell* untuk dapat langsung mengetahui daya dan tegangan yang dihasilkan oleh *solar cell*. Penelitian ini memungkinkan untuk mendapatkan data yang akurat akibat terjadinya perubahan cuaca yang sangat mempengaruhi kemampuan kinerja *solar cell*. Untuk itu, saya mengajukan skripsi saya dengan judul **“Rancang Bangun Alat Pengukur Arus dan Tegangan Pada Solar Cell Berbasis Mikrokontroler”**.

## 1.2 Rumusan Masalah

Berdasarkan latar belakang diatas, rumusan masalah dalam skripsi ini adalah sebagai berikut :

1. Merancang sistem Alat Pengukur tegangan dan arus pada *Solar Cell* berbasis mikrokontroler.
2. Bagaimana menganalisa program untuk Alat Pengukur *Solar Cell* menggunakan sensor arus ACS 712.

## 1.3 Tujuan Penelitian

Adapun tujuan penelitian dari perancangan alat pengukur *solar cell* ini adalah :

1. Merancang suatu sistem Alat Pengukur tegangan dan arus untuk digunakan pada *solar cell* yang dirancang berbasis mikrokontroler.
2. Menganalisa sistem kerja alat bantu untuk mengukur penggunaan *solar cell* dan diharapkan perancangan sistem ini dapat memperbaiki tingkat penggunaan *solar cell* berbasis mikrokontroler.

## 1.4 Batasan Masalah

Untuk memperjelas ruang lingkup permasalahan dan mencegah kemungkinan meluasnya masalah ataupun penyimpangan dari fokus pembahasan perancangan alat, maka diperlukan pembatasan masalah sebagai berikut :

1. Perancangan sistem Alat Pengukur tegangan dan arus pada *Solar Cell*.
2. Pembuatan program pengendali sistem dengan menggunakan *Compiler Arduino IDE* untuk bahasa C.

3. Arduino digunakan sebagai pengendalian sistem.
4. Pada purwarupa yang dibuat menggunakan *Solar Cell* dengan daya 10 WP.
5. Menggunakan sensor ACS 712 dengan kapasitas ukur maksimal sebesar 20 Ampere sebagai sensor pengukur arus.
6. Menggunakan *display character 20x4* dan i2C LCD sebagai penampil *display*.

### **1.5 Manfaat Penelitian**

Adapun manfaat dalam penelitian ini adalah :

1. Memberikan kemudahan dan menambah kualitas dari sebuah pembangkit listrik tenaga matahari.
2. Diharapkan kinerja kerja dari pengukur *solar cell* ini bisa lebih maksimal sesuai dengan prinsip kerjanya.
3. Menampilkan perubahan daya dan tegangan sesuai dari sinar matahari yang diterima *solar cell*.
4. Proteksi otomatis setelah *battery* terisi penuh yang dapat menambah masa pemakaian *battery*.

### **1.6 Metode Penelitian**

Dalam penulisan penelitian ini digunakan beberapa metode sebagai berikut

1. Studi Lapangan

Metode ini dilakukan penulis untuk mendesain, merancang, menguji, dan mengimplentasikan alat yang penulis buat dengan melakukan uji coba langsung.



2. Metode Pengumpulan Data yang digunakan untuk mengumpulkan data-data yang diperlukan selama proses penyusunan skripsi yang diawali dengan tahap perancangan sampai dengan implementasi dan pengujian.

Jenis metode yang digunakan oleh penyusun skripsi ini adalah :

- a. Wawancara

Wawancara merupakan pengkajian terhadap masalah yang diambil dengan cara mewawancarai pihak yang telah mengetahui bagaimana penyelesaian masalah yang diinginkan.

- b. Studi Pustaka

Studi yang bisa dijadikan sebagai bahan untuk mengumpulkan data dan mengkaji data dengan membaca berbagai literatur yang ada kaitannya dengan masalah yang akan dibahas seperti buku, skripsi, jurnal maupun bentuk tulisan lainnya yang isinya berkaitan erat dengan masalah yang akan diteliti sebagai bahan referensi tertulis.

- c. Pengujian alat

Data yang diperoleh melalui metode ini dapat setelah alat yang dibuat diuji dan diambil kesimpulan setelah dilakukan pengujian tersebut.

## **1.7 Sistematika Penulisan**

Sistematika penulisan ini terdiri dari 5 bab, dengan tujuan untuk mempermudah dalam pembahasan. Adapun sistematika penulisan tersebut adalah sebagai berikut :

### **BAB I: PENDAHULUAN**

Pada bab ini berisi Latar Belakang, Tujuan Penulisan, Rumusan Masalah, Pembatasan Masalah, Metode Penelitian dan Sistematika Penulisan.

### **BAB II : TINJAUAN PUSTAKA**

Pada bab ini berisi tentang dasar teori-teori penunjang yang berkaitan dengan alat yang penulis buat.

### **BAB III : METODE PENELITIAN**

Pada bab ini akan menerangkan lokasi dilaksanakannya penelitian, peralatan yang dipergunakan pada saat penelitian, data-data penelitian, jalannya penelitian dan jadwal penelitian.

### **BAB IV : ANALISIS DAN HASIL PEMBAHASAN**

Pada bab ini dibahas penjelasan tentang hasil dari uji coba alat dan menganalisa dari hasil yang didapat dari hasil percobaan yang dilakukan beberapa kali.

### **BAB V : PENUTUP**

Pada bab ini berisi tentang kesimpulan dan saran. Kesimpulan didapat dari data-data hasil pengujian yang dibandingkan dengan tujuan tugas akhir ini dan saran bertujuan untuk perbaikan serta pengembangan lebih lanjut.

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **2.1 Tinjauan Pustaka Relevan**

Berikut ini beberapa penelitian yang dilakukan terdahulu untuk mendukung penelitian penulis dalam perancangan alat pengukur arus dan tegangan pada *Solar Cell* berbasis mikrokontroler diantaranya :

Basam Koteswararao (2016), analisis eksperimental efisiensi panel surya dengan modus yang berbeda dari pendinginan. Jurnal ini membahas tentang bagaimana cara meningkatkan efisiensi dari penggunaan solar cell. Meningkatkan tingkat penguatan dari solar cell yang mencapai tingkat 21% dari 16.2%. kemudian juga dapat ditingkatkan kembali dengan menggunakan *filter optic*.

Mahmud Wasfi (2017), *Solar energy dan photovoltaic systems*. Pada jurnal ini membahas tentang bagaimana cara efisiensi *solar cell* dan *battery* yang digunakan. Tiap jenis *battery* yang digunakan untuk pengisian daya dengan *solar cell* mempunyai tingkatan yang berbeda. Didalam jurnal dibahas tentang jenis-jenis *battery* dan bagaimana tingkatan pengisian dengan menggunakan *solar cell*.

Sistem yang dikerjakan akan sangat berpengaruh terhadap cuaca sekitar. Untuk pengisian yang maksimal, akan membutuhkan sinar matahari yang cukup terik agar didapatkan kinerja kerja *solar cell* yang baik.

#### **2.2 Sejarah Solar Cell**

Tenaga listrik dari cahaya matahari pertama kali ditemukan oleh Alexandre Edmund Becquerel seorang ahli fisika pada tahun 1839. Temuannya ini merupakan cikal bakal teknologi *solar cell*. Percobaannya dilakukan dengan

menyinari 2 elektrode dengan berbagai macam cahaya. Electrode tersebut dibalut (*Coated*) dengan bahan yang sensitif terhadap cahaya, yaitu AgCl dan AgBr dan dilakukan pada kotak hitam yang dikelilingi dengan campuran asam. Dalam percobaannya ternyata tenaga listrik meningkat manakala intensitas cahaya meningkat. Selanjutnya penelitian dari Bacquerel dilanjutkan oleh peneliti-peneliti lain. Tahun 1873 seorang insinyur Inggris Willoughby Smith menemukan Selenium sebagai suatu elemen *photo conductivity*. Kemudian tahun 1876, William Grylls dan Richard Evans Day membuktikan bahwa Selenium menghasilkan arus listrik apabila disinari dengan cahaya matahari. Hasil penemuan mereka menyatakan bahwa Selenium dapat mengubah tenaga matahari secara langsung menjadi listrik tanpa ada bagian bergerak atau panas. Sehingga disimpulkan bahwa solar cell sangat tidak efisien dan tidak dapat digunakan untuk menggerakkan peralatan listrik.

Tahun 1894 Charles Fritts membuat *Solar cell* pertama yang sesungguhnya yaitu bahan *semi conductor* (selenium) dibalut dengan lapisan tipis emas. Tingkat efisiensi yang dicapai baru 1% sehingga belum juga dapat dipakai sebagai sumber energi, namun kemudian dipakai sebagai sensor cahaya. Tahun 1905 Albert Einstein mempublikasikan tulisannya mengenai *photoelectric effect* tulisannya ini mengungkapkan bahwa cahaya terdiri dari paket-paket atau "*quanta of energy*" yang sekarang ini lazim disebut "*photon*". Teorinya ini sangat sederhana tetapi revolusioner. Kemudian tahun 1916 pendapat Einstein mengenai *photoelectric effect* dibuktikan oleh percobaan Robert Andrew Millikan seorang ahli fisika berkebangsaan Amerika dan ia mendapatkan Nobel Prize untuk karya *photoelectric effect*. Tahun 1923 Albert Einstein akhirnya juga

mendapatkan Nobel Prize untuk teorinya yang menerangkan *photoelectric effect* yang dipublikasikan 18 tahun sebelumnya.

Hingga tahun 1980 an efisiensi dari hasil penelitian terhadap solar cell masih sangat rendah sehingga belum dapat digunakan sebagai sumber daya listrik. Tahun 1982, Hans Tholstrup seorang Australia mengendarai mobil bertenaga surya pertama untuk jarak 4000 km dalam waktu 20 hari dengan kecepatan maksimum 72km/jam. Tahun 1985 University of South Wales Australia memecahkan rekor efisiensi solar cell mencapai 20% dibawah kondisi satu cahaya matahari. Tahun 2007 University of Delaware berhasil menemukan *solar cell technology* yang efisiensinya mencapai 42.8% hal ini merupakan rekor terbaru untuk "*thin film photovoltaic solar cell*". Perkembangan dalam riset solar cell telah mendorong komersialisasi dan produksi solar cell untuk penggunaannya sebagai sumber daya listrik.

### **2.2.1 Solar Cell 10 WP**

Solar cell atau panel surya adalah sebuah komponen yang dapat mengubah sinar matahari menjadi listrik. Solar cell terdiri dari sejumlah sel silicon (disebut juga solar cells PV) yang disinari matahari/surya, akan menghasilkan photon yang membangkitkan arus listrik.

Sekumpulan solar cell dalam satu sistem disebut array. Banyaknya jumlah modul yang dapat disusun seri atau parallel akan menentukan besarnya total output energi listrik yang dihasilkan.

Pada penelitian ini menggunakan solar cell 10WP. WP adalah singkatan dari *Watt-Peak*. Adalah istilah yang memang biasa digunakan dalam dunia *solar*

energy. WP menggambarkan besarnya nominal Watt tertinggi yang dapat dihasilkan dari sebuah solar sistem.



Gambar 2.1 : Solar Cell 10 WP

### 2.3 Mikrokontroler ATmega328

ATmega328 adalah mikrokontroler keluaran dari atmel yang mempunyai arsitektur RISC (*Reduce Instruction Set Computer*) yang dimana setiap proses eksekusi data lebih cepat dari pada arsitektur CISC (*Completed Instruction Set Computer*).

Mikrokontroler ini memiliki beberapa fitur antara lain :

- a. *130 macam instruksi yang hampir semuanya dieksekusi dalam satu siklus clock.*
- b. *32 x 8 – bit register serba guna.*
- c. *Kecepatan mencapai 16MIPS dengan clock 16 Mhz.*
- d. *32 KB Flash Memory dan pada arduino memiliki bootloader yang menggunakan 2 KB dari flash memori sebagai bootloader.*
- e. *Memiliki EEPROM (Electrically Erasable Programmable Read Only Memory) sebesar 1 KB sebagai tempat penyimpanan data semi permanent*

karena EEPROM tetap dapat menyimpan data meskipun catu daya dimatikan.

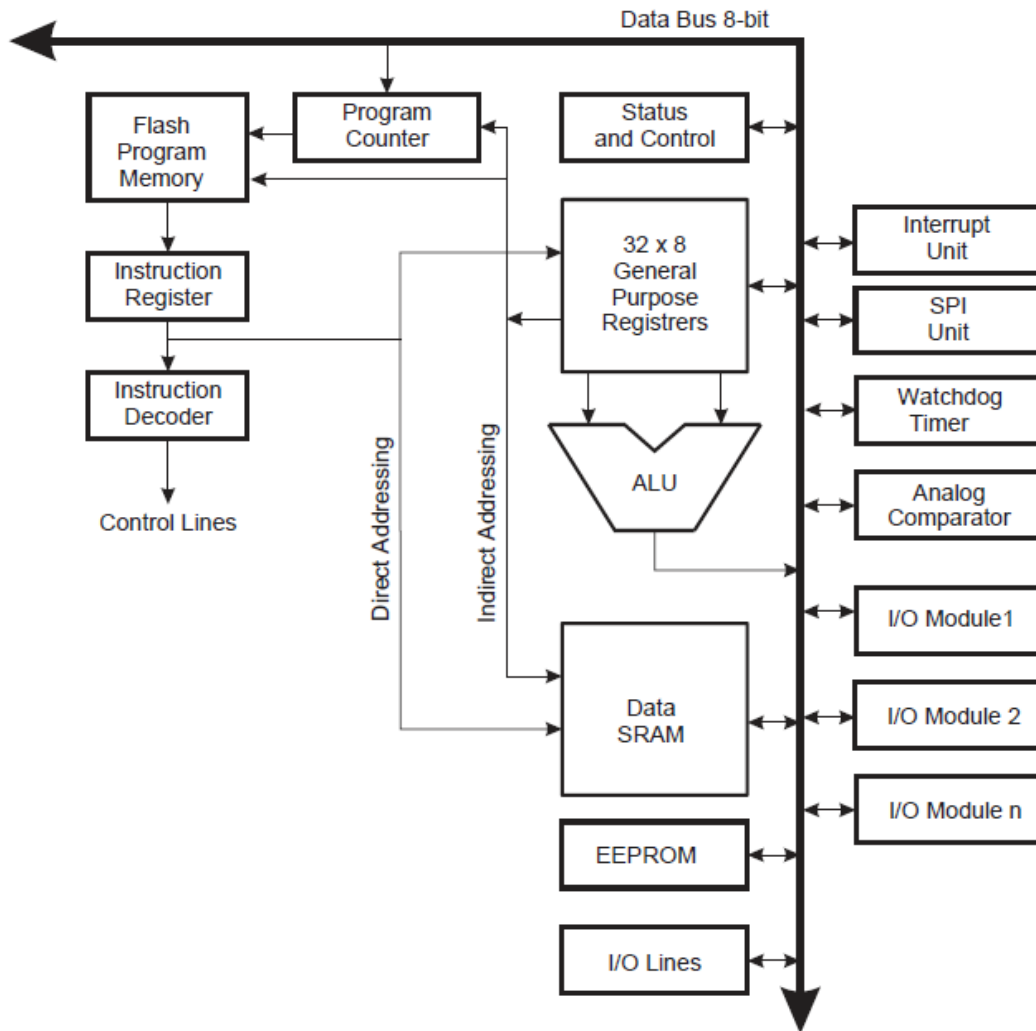
- f. Memiliki SRAM (Static Random Acces Memory) sebesar 2 KB.
- g. Memiliki pin I/O digital sebanyak 14 pin 6 diantaranya PWM (Pulse Width Modulation) output.
- h. Master / Slave SPI Serial interface.

Mikrokontroler ATmega328 memiliki arsitektur *Harvard*, yaitu memisahkan memori untuk kode program dan memori untuk data sehingga dapat memaksimalkan kerja dan *parallelism*. Instruksi-instruksi dalam memori program dieksekusi dalam satu alur tunggal, dimana pada saat satu instruksi dikerjakan instruksi berikutnya sudah diambil dari memori program. Konsep inilah yang memungkinkan instruksi-instruksi dapat dieksekusi dalam setiap satu siklus clock. 32 x 8-bit register serba guna digunakan untuk mendukung operasi pada ALU (*Arithmetic Logic Unit*) yang dapat dilakukan dalam satu siklus. 6 dari register serbaguna ini dapat digunakan sebagai 3 buah register pointer 16-bit pada mode pengalamatan tidak langsung untuk mengambil data pada ruang memori data.

Ketiga register pointer 16-bit ini disebut dengan register X (gabungan R26 dan R27), register Y (gabungan R28 dan R29), dan register Z (gabungan R30 dan R31). Hampir semua instruksi AVR memiliki format 16-bit. Setiap alamat memori program terdiri dari instruksi 16-bit atau 32-bit. Selain register serba guna diatas, terdapat register lain yang terpetakan dengan teknik *memory mapped I/O* selebar 64 byte. Beberapa register ini digunakan untuk fungsi khusus antara lain sebagai register control Timer/Counter, Interupsi, ADC, USART, SPI, EEPROM,

dan fungsi I/O lainnya. Register-register ini menempati memori pada alamat 0x20h - 0x5Fh.

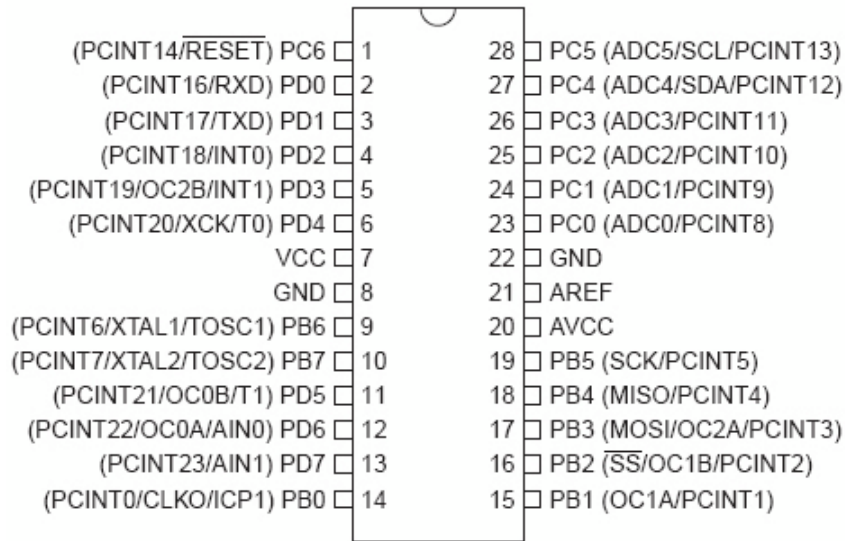
Berikut ini adalah tampilan arsitektur ATmega 328 :



Gambar 2.2 : Blok Diagram Mikrokontroler ATmega328



### 2.3.1 Konfigurasi Pin ATmega328



Gambar 2.3 : Konfigurasi Pin ATmega328

Tabel 2.1.a : Konfigurasi PORTB

Port Pin	Alternate Functions
PB7	XTAL2 (Chip Clock Oscillator pin 2) TOSC2 (Timer Oscillator pin 2) PCINT7 (Pin Change Interrupt 7)
PB6	XTAL1 (Chip Clock Oscillator pin 1 or External clock input) TOSC1 (Timer Oscillator pin 1) PCINT6 (Pin Change Interrupt 6)
PB5	SCK (SPI Bus Master clock Input) PCINT5 (Pin Change Interrupt 5)
PB4	MISO (SPI Bus Master Input/Slave Output) PCINT4 (Pin Change Interrupt 4)
PB3	MOSI (SPI Bus Master Output/Slave Input) OC2A (Timer/Counter2 Output Compare Match A Output) PCINT3 (Pin Change Interrupt 3)
PB2	$\overline{\text{SS}}$ (SPI Bus Master Slave select) OC1B (Timer/Counter1 Output Compare Match B Output) PCINT2 (Pin Change Interrupt 2)
PB1	OC1A (Timer/Counter1 Output Compare Match A Output) PCINT1 (Pin Change Interrupt 1)
PB0	ICP1 (Timer/Counter1 Input Capture Input) CLKO (Divided System Clock Output) PCINT0 (Pin Change Interrupt 0)

**Tabel 2.1.b : Konfigurasi PORTC**

Port Pin	Alternate Function
PC6	RESET (Reset pin) PCINT14 (Pin Change Interrupt 14)
PC5	ADC5 (ADC Input Channel 5) SCL (2-wire Serial Bus Clock Line) PCINT13 (Pin Change Interrupt 13)
PC4	ADC4 (ADC Input Channel 4) SDA (2-wire Serial Bus Data Input/Output Line) PCINT12 (Pin Change Interrupt 12)
PC3	ADC3 (ADC Input Channel 3) PCINT11 (Pin Change Interrupt 11)
PC2	ADC2 (ADC Input Channel 2) PCINT10 (Pin Change Interrupt 10)
PC1	ADC1 (ADC Input Channel 1) PCINT9 (Pin Change Interrupt 9)
PC0	ADC0 (ADC Input Channel 0) PCINT8 (Pin Change Interrupt 8)

**Tabel 2.1.c : Konfigurasi PORTD**

Port Pin	Alternate Function
PD7	AIN1 (Analog Comparator Negative Input) PCINT23 (Pin Change Interrupt 23)
PD6	AIN0 (Analog Comparator Positive Input) OC0A (Timer/Counter0 Output Compare Match A Output) PCINT22 (Pin Change Interrupt 22)
PD5	T1 (Timer/Counter 1 External Counter Input) OC0B (Timer/Counter0 Output Compare Match B Output) PCINT21 (Pin Change Interrupt 21)
PD4	XCK (USART External Clock Input/Output) T0 (Timer/Counter 0 External Counter Input) PCINT20 (Pin Change Interrupt 20)
PD3	INT1 (External Interrupt 1 Input) OC2B (Timer/Counter2 Output Compare Match B Output) PCINT19 (Pin Change Interrupt 19)
PD2	INT0 (External Interrupt 0 Input) PCINT18 (Pin Change Interrupt 18)
PD1	TXD (USART Output Pin) PCINT17 (Pin Change Interrupt 17)
PD0	RXD (USART Input Pin) PCINT16 (Pin Change Interrupt 16)

### **2.3.2 Pemrograman**

Perancangan yang dilakukan pada penelitian ini adalah merancang *board* mikrokontroler dan menambahkan *bootloader* USBASP yang kompatibel dengan *software Arduino IDE*. Caranya adalah dengan mengedit *bootloader* USBASP kemudian menambahkan *board* konfigurasi pada *software* arduino. Kemudian dapat dipilih *board* yang sudah ditambahkan dari *tool* lalu sesuaikan dengan mikrokontroler yang digunakan. Tujuannya adalah untuk mempermudah penggunaan mikrokontroler dan juga meminimalkan penggunaan PCB yang terlampaui banyak karena semuanya dapat dirancang menggunakan 1 buah PCB. Kekurangannya adalah *board* yang dirancang belum dapat berkomunikasi secara *serial* dengan komputer atau laptop dikarenakan belum adanya *chip converter serial to USB* seperti *chip* FT232L dari *maxim*.

### **2.4 Resistor**

Widodo Budiharto dan Sigit Firmansyah (2008:3) menyatakan :” Resistor adalah komponen elektrik yang berfungsi memberikan hambatan terhadap aliran arus listrik. Setiap benda adalah resistor, karena pada dasarnya tiap benda dapat memberikan hambatan listrik. Dalam rangkaian listrik dibutuhkan resistor dengan spesifikasi tertentu, seperti besar hambatan, arus maksimum yang boleh dilewatkan dan karakteristik hambatan terhadap suhu dan panas.

Winarno dan Deni Arifianto (2011:4) menyatakan :” Resistor atau hambatan listrik adalah salah satu komponen elektronik yang digunakan untuk membatasi arus yang mengalir dalam rangkaian tertutup. Lambang komponen

resistor dalam elektronika adalah huruf R dan satuannya adalah ohm ( $\Omega$ ).Berikut adalah jenis-jenis resistor yang biasa digunakan dalam rangkaian elektronik.

### 2.4.1 Resistor Statis

Resistor statis atau resistor bernilai tetap disusun menggunakan cincin-cincin warna.Sebuah resistor statis memiliki 4 atau 5 buah cincin warna.Warna cincin menunjukkan nilai resistor.Di pasaran, resistor dibedakan berdasarkan bahan pembuatnya, yaitu resistor karbon, *wirewound*, dan metafilm.

Pada beberapa resistor berbahan karbon dan metafilm, nilai resistansi ditunjukkan menggunakan kode gelang-gelang warna yang melingkar pada badan resistor.Masing-masing gelang warna memiliki nilai yang berbeda berdasarkan urutannya.Berikut adalah tabel gelang warna pada resistor.



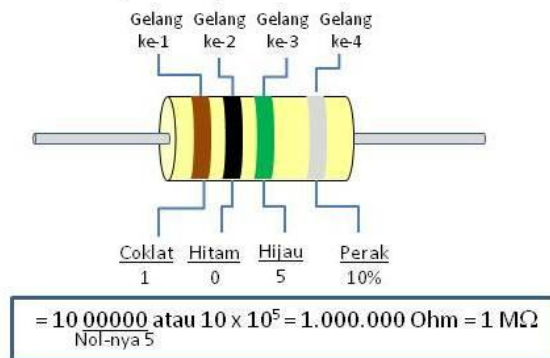
Gambar 2.4 : Resistor Tetap

Tabel 2.2 : Kode Warna Resistor

Warna	Cincin 1	Cincin 2	Cincin 3	Pengali	Toleransi	Koefisien Temperatur (ppm)
Hitam	0	0	0	1	-	
Cokelat	1	1	1	10	1%	100
Merah	2	2	2	100	2%	50
Jingga	3	3	3	1k	-	15

Kuning	4	4	4	10k	-	25
Hijau	5	5	5	100 k	-	-
Biru	6	6	6	1 m	-	-
Ungu	7	7	7	10M	-	-
Abu-abu	8	8	8	100m	-	-
Putih	9	9	9	1 g	-	-
Emas	-	-	-	0.1 g	5%	-
Perak	-	-	-	0.01	10%	-
Tak berwarna	-	-	-	-	20%	-

**Contoh:**  
Resistor dengan 4 Gelang



Gambar 2.5 : Resistor Empat Gelang Warna

- Gelang 1 coklat = 1
- Gelang 2 hitam = 0
- Gelang 3 hijau = x 100.000
- Gelang 4 emas = 10%

Jadi, nilai hambatan resistor tersebut adalah  $1.000.000 \pm 10\%$ , atau

Nilai hambatan maksimum adalah  $1.000.000 + (1.000.000 \times 10\%) = 1.050.000 \Omega$

Nilai hambatan minimum adalah  $1.000.000 - (1.000.000 \times 10\%) = 950.000 \Omega$

## 2.5 Transistor

Richard Blocher, Dipl. Phys (2004:5) menyatakan :”Transistor adalah komponen elektronik yang memiliki tiga sambungan.

Transistor adalah komponen elektronika multitermal, biasanya memiliki 3 terminal. Secara harfiah, kata ‘Transistor’ berarti ‘ Transfer resistor’, yaitu suatu komponen yang nilai resistansi antara terminalnya dapat diatur. Secara umum transistor terbagi dalam 3 jenis :

1. Transistor Bipolar
2. Transistor Unipolar
3. Transistor Unijunction

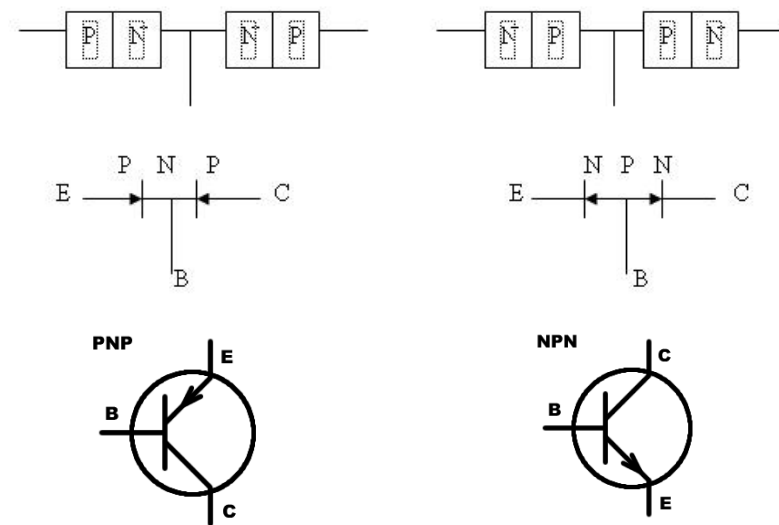
Transistor bipolar bekerja dengan 2 macam *carrier*, sedangkan unipolar satu macam saja, hole atau electron. Beberapa perbandingan transistor bipolar dan unipolar :

Tabel 2.3 : Perbandingan Transistor bipolar dan unipolar

<b>Parameter</b>	<b>Bipolar</b>	<b>Unipolar</b>
Dimensi	Besar	Kecil
Daya	Besar	Kecil
BW	Lebar	Sempit
Respon	Tinggi	Sedang
Input	Arus	Tegangan
Impedansi In	Sedang	Tinggi

Pada transistor bipolar, arus yang mengalir berupa arus lubang (*hole*) dan arus *electron* atau berupa pembawa muatan mayoritas dan minoritas. Transistor dapat berfungsi sebagai penguat tegangan, penguat arus, penguat daya atau sebagai saklar. Ada 2 jenis transistor yaitu PNP dan NPN.

Transistor didesain dari pemanfaatan sifat diode, arus menghantar dari diode dapat dikontrol oleh electron yang ditambahkan pada pertemuan PN diode. Dengan penambahan elektodiode pengontrol ini, maka diode semi-konduktor dapat dianggap dua buah diode yang mempunyai electrode bersama pada pertemuan. Junction semacam ini disebut transistor bipolar dan dapat digambarkan sebagai berikut :



Gambar 2.6 : Transistor Bipolar dan Lambang Transistor

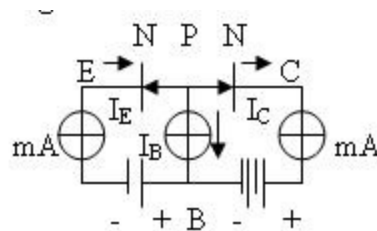
Dengan memilih electrode pengontrol dari type P atau type N sebagai electrode persekutuan antara dua diode, maka dihasilkan transistor jenis PNP dan NPN.

Transistor dapat bekerja apabila diberi tegangan, tujuan pemberian tegangan pada transistor adalah agar transistor tersebut dapat mencapai suatu kondisi menghantar atau menyumbat. Baik transistor NPN maupun PNP tegangan antara emitor dan basis adalah forward bias, sedangkan antara basis dengan kolektor adalah reverse bias.



Gambar 2.7 : Bias Transistor

Dari cara pemberian tegangan muka didapatkan dua kondisi yaitu menghantar dan menyumbat seperti pada gambar transistor NPN dibawah ini.



Gambar 2.8 : Transistor NPN

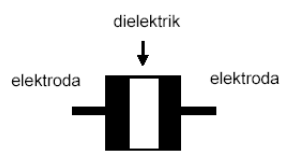
## 2.6 Kapasitor

Kapasitor adalah komponen elektronika yang digunakan untuk menyimpan muatan listrik yang terdiri dari dua konduktor dan dipisahkan oleh bahan penyekat (bahan dielektrik) tiap konduktor disebut keeping. Kapasitor atau sering disebut kondensator merupakan komponen listrik yang dibuat sedemikian rupa sehingga mampu menyimpan muatan listrik. Prinsip sebuah kapasitor pada umumnya sama halnya dengan resistor yang juga termasuk dalam kelompok komponen pasif, yaitu jenis komponen yang bekerja tanpa memerlukan arus panjar. Kapasitor terdiri atas dua konduktor (lempeng logam) yang dipisahkan oleh bahan penyekat (isolator). Isolator penyekat ini sering disebut sebagai bahan (zat) dielektrik.



Satuan nilai kapasitor dinyatakan dalam Farad (F), miliFarad (mF), mikroFarad ( $\mu\text{F}$ ), nanoFarad (nF), atau pikoFarad (pF). Konversi satuan nilai kapasitor sama dengan konversi satuan tahanan listrik.

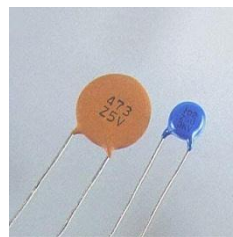
Kapasitor disusun menggunakan dua pelat logam. Kedua pelat logam itu dipisahkan dengan isolator yang disebut dielektrikum. Jenis-jenis dielektrikum antara lain mika, plastik, keramik, tantalum, dan elektrolit.



Gambar 2.9 : Prinsip Dasar Kapasitor

### 2.6.1 Kapasitor Keramik

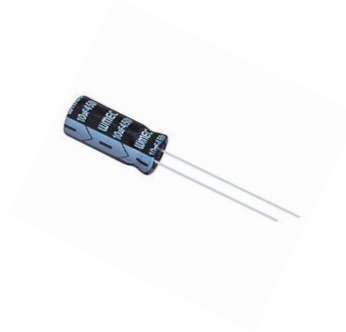
*Bentuk kapasitor keramik bermacam-macam. Karena sifatnya yang stabil, kapasitor keramik bagus digunakan pada frekuensi tinggi. Pemasangan kapasitor keramik pada rangkaian elektronika boleh dibolak-balik, tidak perlu memperhatikan kutub positif dan kutub negatif. Nilai kapasitansi kapasitor keramik sangat kecil, tetapi bagus digunakan pada jangkauan tegangan yang luas yaitu hingga 100 volt.*



Gambar 2.10 : Kapasitor Keramik

### 2.6.2 Kapasitor Elektrolit

Kapasitor elektrolit atau electrolit capasitor (elco) merupakan jenis kapasitor polar yang dipasang pada rangkaian elektronik sesuai dengan jenis-jenis terminalnya. Terminal positif (+) kapasitor dihubungkan dengan potensial tinggi (+) rangkaian elektronik, dan terminal negatif (-) kapasitor dihubungkan dengan potensial rendah(-) rangkaian elektronik. Pemasangan yang salah dapat menyebabkan kapasitor rusak atau meledak. Kutub negatif kapasitor elektrolit berkapasitas besar biasa digunakan dalam *power supply*.

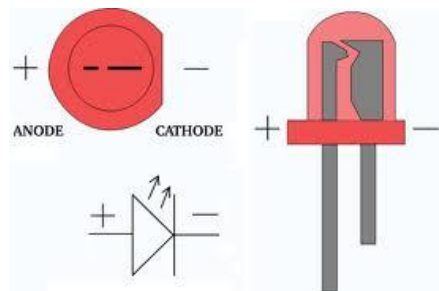


Gambar 2.11 : Elektrolit Capasitor (Elco)

## **2.7 LED (Light Emitting Diode)**

Led merupakan komponen yang dapat mengeluarkan emisi cahaya. LED merupakan produk temuan lain selain dioda. Strukturnya juga sama dengan dioda, tetapi belakangan ditemukan bahwa elektron yang menerjang sambungan P-N juga melepaskan energi berupa energi panas dan energi cahaya. LED dibuat agar lebih efisien jika mengeluarkan cahaya. Untuk mendapatkan emisi cahaya pada semikonduktor, doping yang dipakai adalah galium, arsenik dan fosfor. Jenis doping yang berbeda menghasilkan warna cahaya yang berbeda pula.

Pada saat ini warna-warna cahaya LED yang banyak adalah warna merah, kuning, dan hijau. LED berwarna biru sangat langka. Pada dasarnya semua warna bisa dihasilkan, namun akan menjadi sangat mahal dan tidak efisien. Dalam memilih LED selain warna, perlu diperhatikan tegangan kerja, arus maksimum dan disipasi dayanya. Rumah (*chasing*) LED dan bentuknya juga bermacam-macam, ada yang persegi empat, bulat dan lonjong.



Gambar 2.12 : Led (*Light Emitting Diode*)

## 2.8 Sensor

Sensor adalah jenis transduser yang digunakan untuk mengubah besaran mekanis, magnetis, panas, sinar dan kimia menjadi tegangan dan arus listrik. Sensor sering digunakan untuk pendeteksian pada saat melakukan pengukuran atau pengendalian.

Sensor adalah alat untuk mendeteksi/mengukur sesuatu, yang digunakan untuk mengubah variasi mekanis, magnetis, panas, sinar dan kimia menjadi tegangan dan arus listrik. Dalam lingkungan sistem pengendali dan robotika, sensor memberikan kesamaan yang menyerupai mata, pendengaran, hidung, lidah

yang kemudian akan diolah oleh kontroler sebagai otaknya (Petruzella, 2001). Sensor dalam teknik pengukuran dan pengaturan secara elektronik berfungsi mengubah besaran fisik (misalnya : temperatur, gaya, kecepatan putaran) menjadi besaran listrik yang proposional.

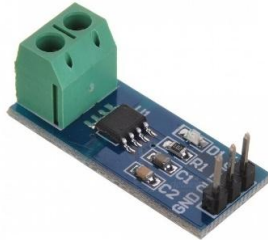
### **2.8.1 Sensor Arus ACS712**

Sensor arus ACS712 adalah sensor arus yang bekerja berdasarkan efek medan. Sensor arus ini dapat digunakan untuk mengukur arus AC atau DC. Modul sensor ini telah dilengkapi dengan rangkaian penguat operasional, sehingga sensitivitas pengukuran arusnya meningkat dan dapat mengukur perubahan arus yang kecil. Sensor ini digunakan pada aplikasi-aplikasi bidang industri, komersial, maupun komunikasi. Contoh aplikasinya antara lain untuk sensor kontrol motor. Deteksi dan manajemen penggunaan daya, sensor untuk daya tersaklar, sensor proteksi terhadap arus lebih, dan lain sebagainya.

Spesifikasi sensor arus ACS712 :

1. *Berbasis ACS712 dengan fitur :*
  - a. *Berbasis kenaikan perubahan luaran = 5  $\mu$ s.*
  - b. *Lebar frekuensi sampai dengan 80 kHz.*
  - c. *Total kesalahan luaran 1.5% pada suhu kerja TA = 25°C.*
  - d. *Tahanan konduktor internal 1.2m $\Omega$ .*
  - e. *Tegangan isolasi minimum 2.1 kVRMS antara pin 1-4 dan pin 5-8.*
  - f. *Sensitivitas luaran 185 mV/A.*
  - g. *Mampu mengukur arus AC atau DC hingga 20 A.*

- h. Tegangan luaran proporsional terhadap masukan arus AC atau DC.
- 2. Tegangan kerja 5 VDC.
- 3. Dilengkapi dengan penguat operasional untuk menambah sensitivitas luaran.



Gambar 2.13 : Sensor arus ACS712

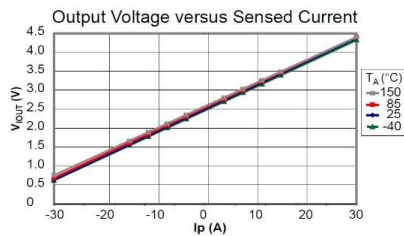
### 2.8.1.1 Konfigurasi Pin Sensor Arus ACS712

Tabel 2.4 : Konfigurasi Pin Sensor Arus ACS712

Pin Sensor ACS712	Fungsi
IP +	Terminal yang mendeteksi arus, terdapat sekering di dalamnya
IP-	Terminal yang mendeteksi arus, terdapat sekring di dalamnya
GND	Terminal sinyal <i>ground</i>
FILTER	Terminal untuk kapasitor eksternal yang berfungsi sebagai <i>bandwith</i>
VIout	Terminal keluaran sinyal analog

Vcc	Terminal masukan catu daya
-----	----------------------------

Hambatan dalam penghantar sensor sebesar 1.2 mΩ dengan daya yang rendah. Jalur terminal konduktif secara kelistrikan diisolasi dari sensor timah mengarah (pin 5 sampai pin 8). Hal ini menjadikan sensor arus ACS712 dapat digunakan pada aplikasi-aplikasi yang membutuhkan isolasi listrik tanpa menggunakan opto-isolator atau teknik isolasi lainnya yang mahal.



Gambar 2.14 : Grafik Perubahan Tegangan pada Sensor Arus ACS712

Gambar diatas menunjukkan IC yang digunakan adalah versi 20A, yaitu versi sensor yang dapat mengukur arus maksimal sebesar -20A sampai dengan 20A dengan sensitivitas 100mV/A.

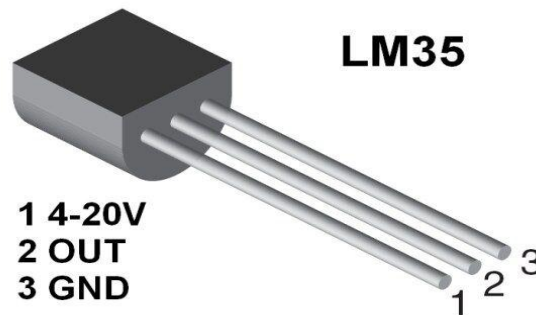
### 2.8.2 Sensor LM35

Sensor suhu LM35 adalah komponen elektronika yang memiliki fungsi untuk mengubah besaran suhu menjadi besaran listrik dalam bentuk tegangan. Sensor Suhu LM35 yang dipakai dalam penelitian ini berupa komponen elektronika yang diproduksi oleh *National Semiconductor*. LM35 memiliki keakuratan tinggi dan kemudahan perancangan jika dibandingkan dengan sensor suhu yang lain, LM35 juga mempunyai keluaran impedansi yang rendah dan

linieritas yang tinggi sehingga dapat dengan mudah dihubungkan dengan rangkaian kendali khusus serta tidak memerlukan penyetelan lanjutan.

Meskipun tegangan sensor ini dapat mencapai 30 volt akan tetapi yang diberikan kesensor adalah sebesar 5 volt, sehingga dapat digunakan dengan catu daya tunggal dengan ketentuan bahwa LM35 hanya membutuhkan arus sebesar 60  $\mu\text{A}$  hal ini berarti LM35 mempunyai kemampuan menghasilkan panas (*self-heating*) dari sensor yang dapat menyebabkan kesalahan pembacaan yang rendah yaitu kurang dari 0,5  $^{\circ}\text{C}$  pada suhu 25  $^{\circ}\text{C}$  .

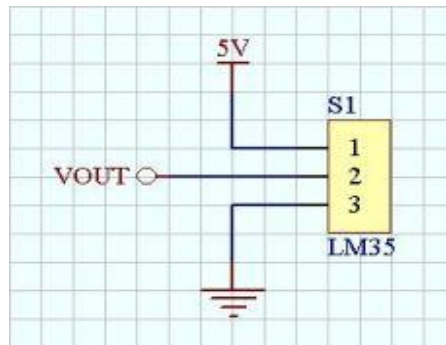
### 2.8.2.1 Struktur Sensor LM35



Gambar 2.15 : Sensor LM35 dan Konfigurasi Pin LM35

Gambar diatas menunjukkan bentuk dari LM35. 3 pin LM35 menunjukan fungsi masing-masing pin diantaranya, pin 1 berfungsi sebagai sumber tegangan kerja dari LM35, pin 2 atau tengah digunakan sebagai tegangan keluaran atau  $V_{\text{out}}$  dengan jangkauan kerja dari 0 Volt sampai dengan 1,5 Volt dengan tegangan operasi sensor LM35 yang dapat digunakan antar 4 Volt sampai 30 Volt. Keluaran sensor ini akan naik sebesar 10 mV setiap derajat *celcius* sehingga diperoleh persamaan sebagai berikut :

$$V_{\text{LM35}} = \text{Suhu} * 10 \text{ mV}$$



Gambar 2.16 : Skematik Rangkaian Dasar Sensor Suhu

Gambar diatas kanan adalah gambar skematik rangkaian dasar sensor suhu LM35-DZ. Rangkaian ini sangat sederhana dan praktis. Vout adalah tegangan keluaran sensor yang terskala linear terhadap suhu terukur, yakni 10 milivolt per 1 °C. Jadi jika  $V_{out} = 530\text{mV}$ , maka suhu terukur adalah  $53^{\circ}\text{C}$ . Dan jika  $V_{out} = 320\text{mV}$ , maka suhu terukur adalah  $32^{\circ}\text{C}$ . Tegangan keluaran ini bisa langsung diumpankan sebagai masukan ke rangkaian pengkondisi sinyal seperti rangkaian penguat operasional dan rangkaian *filter*, atau rangkaian lain seperti rangkaian pembanding tegangan dan rangkaian *Analog-to-Digital Converter*.

### 2.8.2.2 Karakteristik Sensor LM35

Adapun karakteristik sensor LM35 yaitu :

1. Memiliki sensitivitas suhu, dengan faktor skala linier antara tegangan dan suhu  $10\text{ mVolt}/^{\circ}\text{C}$ , sehingga dapat dikalibrasi langsung dalam *celcius*.
2. Memiliki ketepatan atau akurasi kalibrasi yaitu  $0,5^{\circ}\text{C}$  pada suhu  $25^{\circ}\text{C}$  seperti terlihat pada gambar 2.2.
3. Memiliki jangkauan maksimal operasi suhu antara  $-55^{\circ}\text{C}$  sampai  $+150^{\circ}\text{C}$ .



4. Bekerja pada tegangan 4 sampai 30 volt.
5. Memiliki arus rendah yaitu kurang dari 60  $\mu\text{A}$ .
6. Memiliki pemanasan sendiri yang rendah (*low-heating*) yaitu kurang dari 0,1  $^{\circ}\text{C}$  pada udara diam.
7. Memiliki impedansi keluaran yang rendah yaitu 0,1  $\Omega$  untuk beban 1 mA.
8. Memiliki ketidaklinieran hanya sekitar  $\pm 1/4^{\circ}\text{C}$ .

Sensor LM35 bekerja dengan mengubah besaran suhu menjadi besaran tegangan. Tegangan ideal yang keluar dari LM35 mempunyai perbandingan 100 $^{\circ}\text{C}$  setara dengan 1 volt. Sensor ini mempunyai pemanasan diri (*self heating*) kurang dari 0,1 $^{\circ}\text{C}$ , dapat dioperasikan dengan menggunakan *power supply* tunggal dan dapat dihubungkan antar muka (*interface*) rangkaian control yang sangat mudah.

IC LM 35 sebagai sensor suhu yang teliti dan terkemas dalam bentuk *Integrated Circuit* (IC), dimana output tegangan keluaran sangat linear terhadap perubahan suhu. Sensor ini berfungsi sebagai penguah dari besaran fisis suhu ke besaran tegangan yang memiliki koefisien sebesar 10 mV / $^{\circ}\text{C}$  yang berarti bahwa kenaikan suhu 1 $^{\circ}\text{C}$  maka akan terjadi kenaikan tegangan sebesar 10 mV.

IC LM 35 ini tidak memerlukan pengkalibrasian atau penyetelan dari luar karena ketelitiannya sampai lebih kurang seperempat derajat celsius pada temperature ruang. Jangka sensor mulai dari - 55 $^{\circ}\text{C}$  sampai dengan 150 $^{\circ}\text{C}$ , IC LM35 penggunaannya sangat mudah, difungsikan sebagai kontrol dari indikator tampilan catu daya terbelah. IC LM 35 dapat dialiri arus 60  $\mu\text{A}$  dari supplay sehingga panas yang ditimbulkan sendiri sangat rendah kurang dari 0  $^{\circ}\text{C}$  di dalam suhu ruangan.

Untuk mendeteksi suhu digunakan sebuah sensor suhu LM35 yang dapat dikalibrasikan langsung dalam C (*celcius*), LM35 ini difungsikan sebagai basic temperature sensor.

Adapun keistimewaan dari IC LM 35 adalah :

1. Kalibrasi dalam satuan derajat celcius.
2. Lineritas +10 mV/ ° C.
3. Akurasi 0,5 ° C pada suhu ruang.
4. Range +2 ° C – 150 ° C.
5. Dioperasikan pada catu daya 4 V – 30 V.
6. Arus yang mengalir kurang dari 60 µA

Kelebihan dan kelemahan sensor LM35 adalah :

*Kelebihan :*

1. Rentang suhu yang jauh, antara -55 sampai + 150 °C
2. Low self-heating, sebesar 0,08 °C
3. Beroperasi pada tegangan 4 sampai 30V
4. Rangkaian tidak rumit
5. Tidak memerlukan pengkondisian sinyal

*Kekurangan :*

*Mebutuhkan sumber tegangan untuk beroperasi.*

## **2.9 Power Supply**

*Power supply* ini digunakan untuk memberikan pasokan daya ke seluruh bagian arduino dan motor kapasitor. Kebanyakan Arduino bekerja pada daya 3 – 5

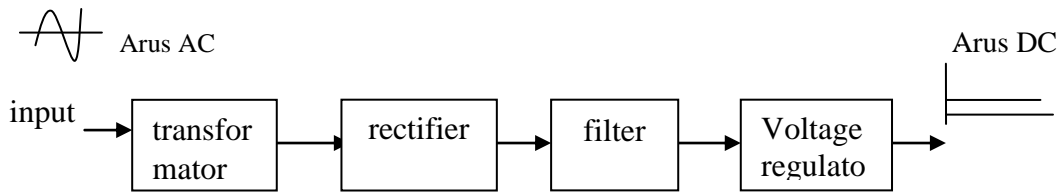
VDC, sehingga di perlukan catu daya untuk mengkonversikan tegangan AC menjadi DC yang memungkinkan untuk disalurkan ke arduino uno.

### **2.9.1 Cara Kerja *Power Supply***

Prinsip Kerja DC *Power Supply* (Adaptor) – Arus Listrik yang kita gunakan di rumah, kantor dan pabrik pada umumnya adalah dibangkitkan, dikirim dan didistribusikan ke tempat masing-masing dalam bentuk Arus Bolak-balik atau arus AC (*Alternating Current*). Hal ini dikarenakan pembangkitan dan pendistribusian arus Listrik melalui bentuk arus bolak-balik (AC) merupakan cara yang paling ekonomis dibandingkan dalam bentuk arus searah atau arus DC (*Direct Current*).

Akan tetapi, peralatan elektronika yang kita gunakan sekarang ini sebagian besar membutuhkan arus DC dengan tegangan yang lebih rendah untuk pengoperasiannya. Oleh karena itu, hampir setiap peralatan Elektronika memiliki sebuah rangkaian yang berfungsi untuk melakukan konversi arus listrik dari arus AC menjadi arus DC dan juga untuk menyediakan tegangan yang sesuai dengan rangkaian Elektronika-nya. Rangkaian yang mengubah arus listrik AC menjadi DC ini disebut dengan DC Power Supply atau dalam bahasa Indonesia disebut dengan Catu daya DC. DC Power Supply atau Catu Daya ini juga sering dikenal dengan nama “Adaptor”.

Sebuah DC *Power Supply* atau Adaptor pada dasarnya memiliki 4 bagian utama agar dapat menghasilkan arus DC yang stabil. Keempat bagian utama tersebut diantaranya adalah *Transformer*, *Rectifier*, *Filter* dan *Voltage Regulator*. Sebelum kita membahas lebih lanjut mengenai Prinsip Kerja DC *Power Supply*,

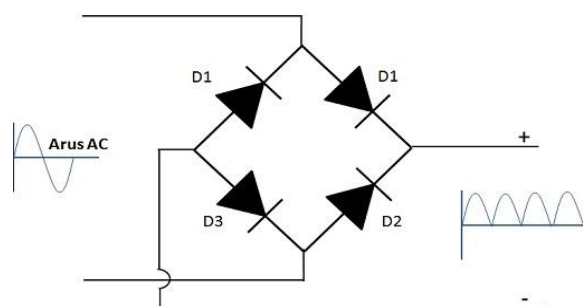


Gambar 2.17 : Blok DC *Power Supply*

Berikut ini adalah penjelasan singkat tentang prinsip kerja DC *Power Supply* (Adaptor) pada masing-masing blok berdasarkan Diagram blok diatas.

### 2.9.2 Rectifier (Penyearah)

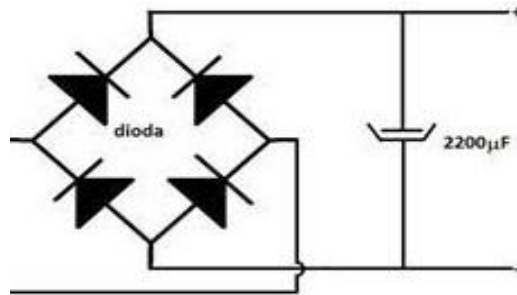
*Rectifier* atau penyearah gelombang adalah rangkaian Elektronika dalam *Power Supply* (catu daya) yang berfungsi untuk mengubah gelombang AC menjadi gelombang DC setelah tegangannya diturunkan oleh *Transformer Step down*. Rangkaian *Rectifier* biasanya terdiri dari komponen Dioda. Terdapat 2 jenis rangkaian *Rectifier* dalam *Power Supply* yaitu "*Half Wave Rectifier*" yang hanya terdiri dari 1 komponen Dioda dan "*Full Wave Rectifier*" yang terdiri dari 2 atau 4 komponen dioda.



Gambar 2.18 : *Rectifier*

### 2.9.3 Filter (Penyaring)

Dalam rangkaian *Power supply* (Adaptor), *Filter* digunakan untuk meratakan sinyal arus yang keluar dari *Rectifier*. *Filter* ini biasanya terdiri dari komponen Kapasitor (Kondensator) yang berjenis Elektrolit atau ELCO



Gambar 2.19 : Filter (penyaring)

#### 2.9.4 Voltage Regulator

Untuk menghasilkan Tegangan dan Arus DC (arus searah) yang tetap dan stabil, diperlukan *Voltage Regulator* yang berfungsi untuk mengatur tegangan sehingga tegangan *Output* tidak dipengaruhi oleh suhu, arus beban dan juga tegangan input yang berasal *Output Filter*. *Voltage Regulator* pada umumnya terdiri dari Dioda *Zener*, Transistor atau IC (*Integrated Circuit*).

Pada DC *Power Supply* yang canggih, biasanya *Voltage Regulator* juga dilengkapi dengan *Short Circuit Protection* (perlindungan atas hubung singkat), *Current Limiting* (Pembatas Arus) ataupun *Over Voltage Protection* (perlindungan atas kelebihan tegangan).

#### 2.10 Liquid Crystal Display (LCD)

LCD adalah sebuah display dot matrik yang difungsikan untuk menampilkan tulisan berupa angka atau huruf sesuai dengan yang di inginkan (sesuai dengan program yang digunakan untuk mengontrolnya). Pada tugas akhir

ini menggunakan LCD dot matrix dengan 4 x 20, sehingga kaki-kakinya berjumlah 16 pin.

LCD sebagaimana output yang dapat menampilkan tulisan sehingga lebih mudah dimengerti , dibanding jika menggunakan LED saja, dalam modul ini menggunakan LCD character untuk menampilkan tulisan atau character saja. Tampilan LCD terdiri dari dua bagian, yakni bagian panel LCD yang terdiri dari banyak titik. LCD dan sebuah arduino yang menempel dipanel dan berfungsi mengatur titik-titik LCD tadi menjadikan huruf atau angka yang terbaca.

Huruf atau angka yang akan ditampilkan dikirim ke LCD dalam bentuk kode ASCII, kode ASCII ini diterima dan diolah oleh arduino didalam LCD menjadi titik-titik LCD yang terbaca sebagai huruf atau angka. Dengan demikian tugas arduino pemakai tampilan LCD hanya mengirimkan kode-kode ASCII untuk ditampilkan.

Spesifikasi LCD secara umum:

1. *Jumlah baris*
2. *Jumlah karakter perbaris*
3. *Tegangan kerja*

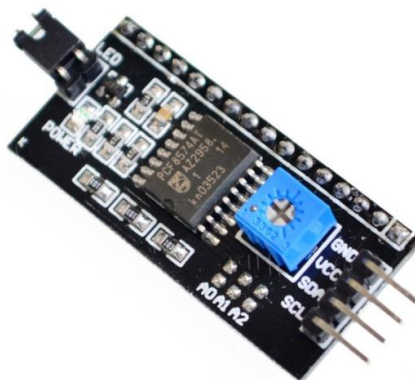
*Berikut ini gambar LCD 4x20 CHARS:*



Gambar 2.20 : LCD 4x20 CHARS

### 2.11 i2C LCD PCF8574AT

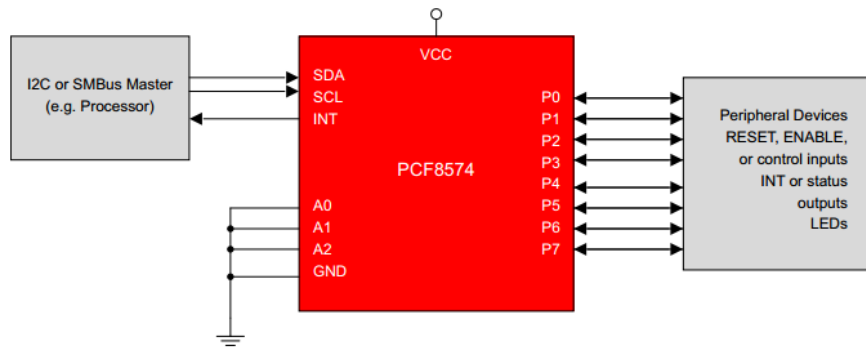
Penggunaan modul LCD i2C ini dimaksudkan untuk menghemat penggunaan pin-pin mikrokontroler. i2C LCD adalah modul LCD yang dikendalikan secara serial sinkron dengan *protocol* i2C/IIC (*Inter Integrated Circuit*) atau TWI (*Two Wire Interface*). Normalnya, modul LCD dikendalikan secara parallel baik untuk jalur data maupun kontrolnya.



Gambar 2.21 : Modul i2C dengan Chip PCF8574AT

Modul i2C diatas adalah converter yang menggunakan chip PCF8574AT produk dari NXP sebagai kontrollernya. IC ini adalah sebuah *8 bit I/O expander for i2C bus* yang pada dasarnya adalah sebuah *shift register*.

Jalur komunikasi datanya dapat terlihat pada gambar dibawah :



Gambar 2.22 : Jalur Komunikasi data chip PCF8574



## **BAB III**

### **METODE PENELITIAN**

#### **3.1 Jadwal dan Lokasi Penelitian**

Penelitian ini dilaksanakan diluar ruangan terbuka untuk mendapatkan sinar matahari yang cukup. Perubahan suhu panas matahari juga diukur dengan menggunakan sensor suhu LM35. Perubahan tingkat sinar matahari diukur untuk mendapatkan arus listrik dan pencuplikan perubahan arus dan tegangan terus diukur untuk mendapatkan nilai yang cukup baik. Waktu penelitian direncanakan berlangsung selama 5 (lima) bulan, dimulai dari perencanaan alat, pembuatan alat, pengujian alat, pengambilan data hingga pengolahan data.

#### **3.2 Peralatan dan Bahan Penelitian**

Adapun alat dan bahan yang digunakan pada penelitian kali ini adalah sebagai berikut:

- a. Alat – Alat
  1. Multimeter
  2. Bor PCB dan bor listrik
  3. Solder
  4. Attractor
  5. Hands tool (alat tangan)
  6. Software Arduino
  7. Project board
  8. Larutan FeC13
- b. Bahan - Bahan

1. Solar Cell panel
2. Sensor arus ACS712
3. Sensor LM35
4. Mikrokontroler ATmega328
5. Battery kering 12 volt 7 Ampere
6. Timah
7. LCD  $20 \times 4$
8. I2C modul PCF8574AT
9. Kabel pelangi
10. Terminal male dan female
11. PCB
12. Capacitor
13. Transistor
14. Dioda.
15. Resistor

### **3.3 Data Perancangan**

#### **3.3.1 Daftar Input dan Output yang Digunakan**

Perancangan alat ini menggunakan beberapa input dan output perangkat yang akan bekerja dengan perintah dari sebuah controller yakni Arduino-Uno. Perangkat input berupa Sensor arus ACS712 sebagai pengukur arus dan tegangan, sensor LM35 sebagai detektor pengukur panas matahari, dan bahasa pemrograman Arduino-Uno. Outputnya berupa LCD display dan beberapa buah LED.

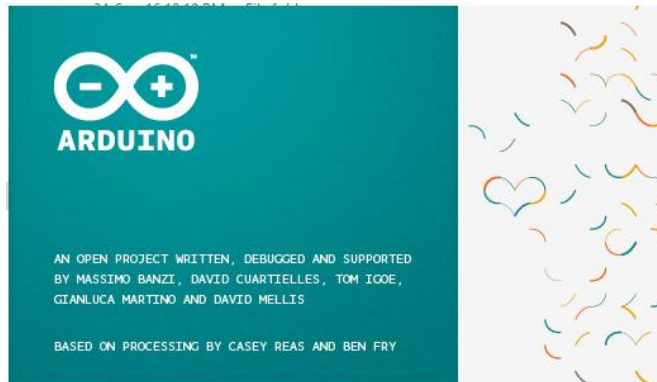
### 3.3.2 Perancangan Program Arduino

Persiapan yang akan dilaksanakan dalam memasukan program ke dalam board Arduino-Uno adalah sebagai berikut :

1. Merakit seluruh rangkaian.
2. Memasukkan program *bootloader* agar mikrokontroler dapat memprogram dirinya sendiri dan dapat diprogram dengan menggunakan *software* arduino.
3. Mengetik program menggunakan software Arduino (dalam penelitian ini penulis menggunakan versi 1.0.5).
4. Melakukan pengecekan (Verify) program yang telah ditulis, untuk mengetahui apakah ada kesalahan dalam penulisan atau tidak.
5. Mengupload program ke board Arduino
6. Menjalankan program

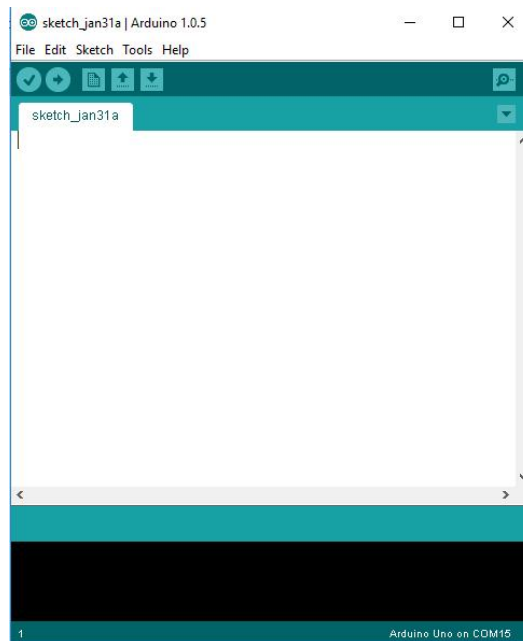
Adapun Langkah-langkah yang dilakukan:

1. Klik *Local Disk C* → *Program Files* → *arduino-nightly* → *arduino.exe*



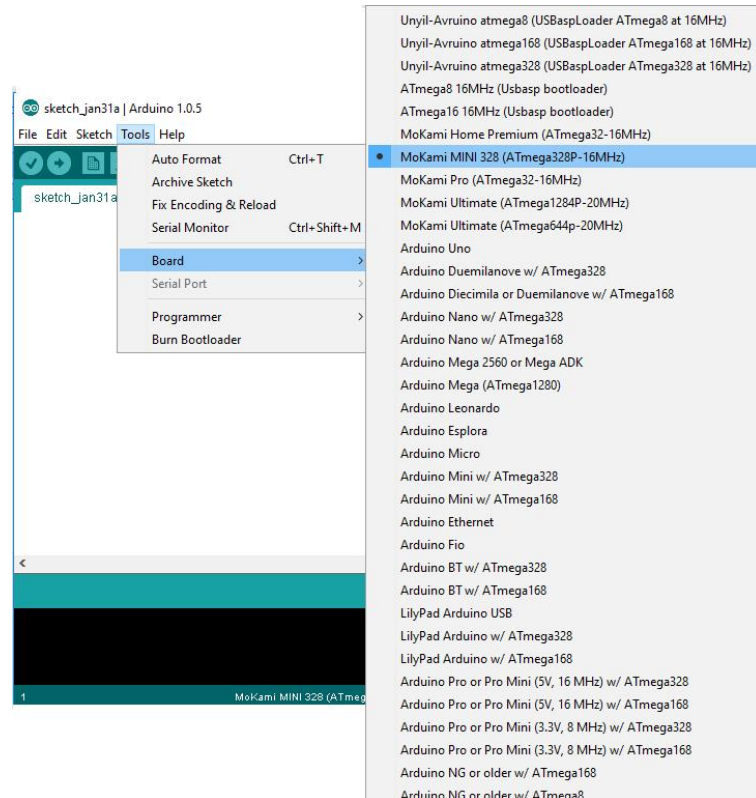
Gambar 3.1: Software Arduino 1.0.5

2. Pada software Arduino, *Klik File* → *New*
3. Muncul kotak dialog seperti gambar dibawah ini:



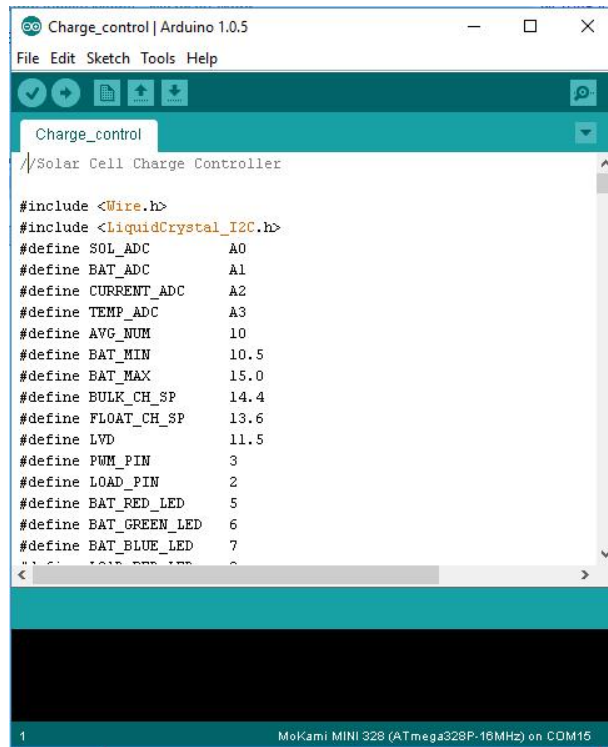
Gambar 3.2: Menu File Baru

4. Sebelum mulai menuliskan sintax, pilih dahulu jenis board Arduino yang akan di gunakan (penulis menggunakan Arduino-Uno). Klik *Tools* → *Board* → *Arduino Uno*.



Gambar 3.3: Pemilihan Board Arduino

5. Setelah board dipilih, untuk membuat proyek baru, langsung masukkan syntax pemrograman pada kotak dialog Arduino



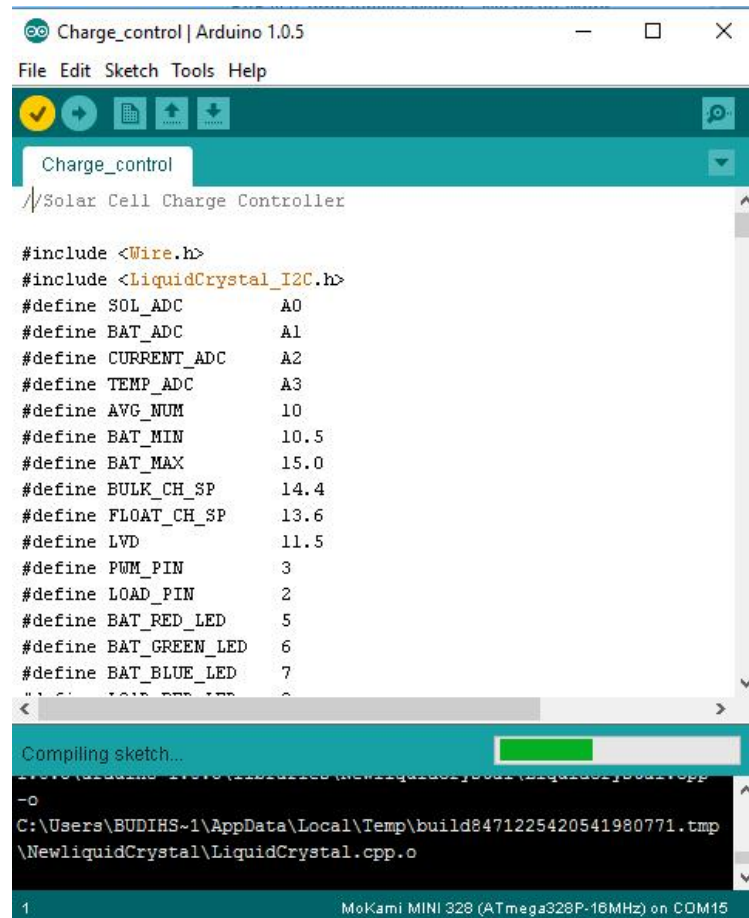
```
Charge_control | Arduino 1.0.5
File Edit Sketch Tools Help

Charge_control
//Solar Cell Charge Controller

#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#define SOL_ADC      A0
#define BAT_ADC      A1
#define CURRENT_ADC  A2
#define TEMP_ADC     A3
#define AVG_NUM      10
#define BAT_MIN      10.5
#define BAT_MAX      15.0
#define BULK_CH_SP   14.4
#define FLOAT_CH_SP  13.6
#define LVD          11.5
#define PWM_PIN      3
#define LOAD_PIN     2
#define BAT_RED_LED  5
#define BAT_GREEN_LED 6
#define BAT_BLUE_LED 7
// #define BAT_ORANGE_LED 6
```

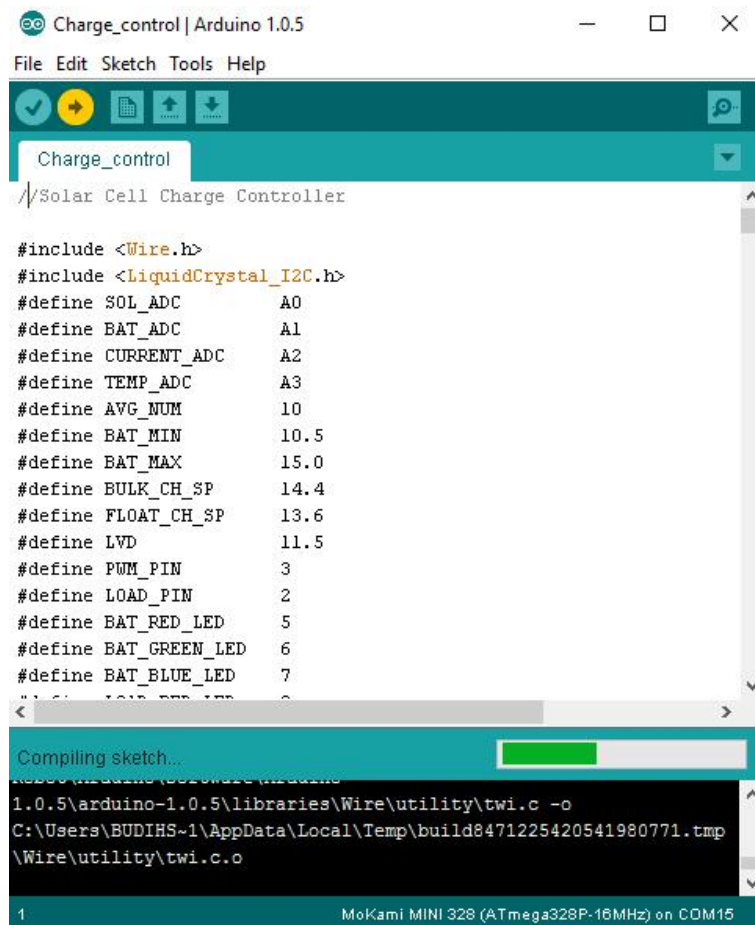
Gambar 3.4: Membuat File Proyek Baru

6. Setelah syntax pemrograman selesai dibuat, maka langkah berikutnya adalah mengecek (Verify) program tersebut dengan cara mengklik button Verify berlogo centang (✓) di kiri atas Menu Bar software Arduino.



Gambar 3.5: Proses Verify program

- Setelah proses Verify berhasil dan penulisan program dinyatakan benar oleh software arduino, maka langkah berikutnya adalah mengupload program ke board Arduino. Caranya adalah dengan menghubungkan board Arduino ke PC / Laptop menggunakan kabel USB, kemudian mengklik button Upload berlogo (  $\Rightarrow$  ) pada Menu Bar software Arduino.



Gambar 3.6: Proses Upload program ke Arduino

- Setelah selesai di Upload, simpan syntax pemrograman yang telah dibuat dengan cara *File* → *Save As* atau *Ctrl+Shift+S*, kemudian pilih lokasi penyimpanan yang diinginkan. Lalu lepas board Arduino dari PC / Laptop kemudian jalankan rangkaian sistem yang telah dirakit sebelumnya.

Setelah membuat program, sebelum program diuji coba pada rangkaian yang sebenarnya, program diuji dahulu pada *proteus 7 professional* untuk

menghindari kesalahan-kesalahan yang terjadi dan untuk memastikan keseluruhan program dapat berjalan dengan baik dan benar. Untuk itu dibawah adalah hasil simulasi dengan *proteus 7 professional*.

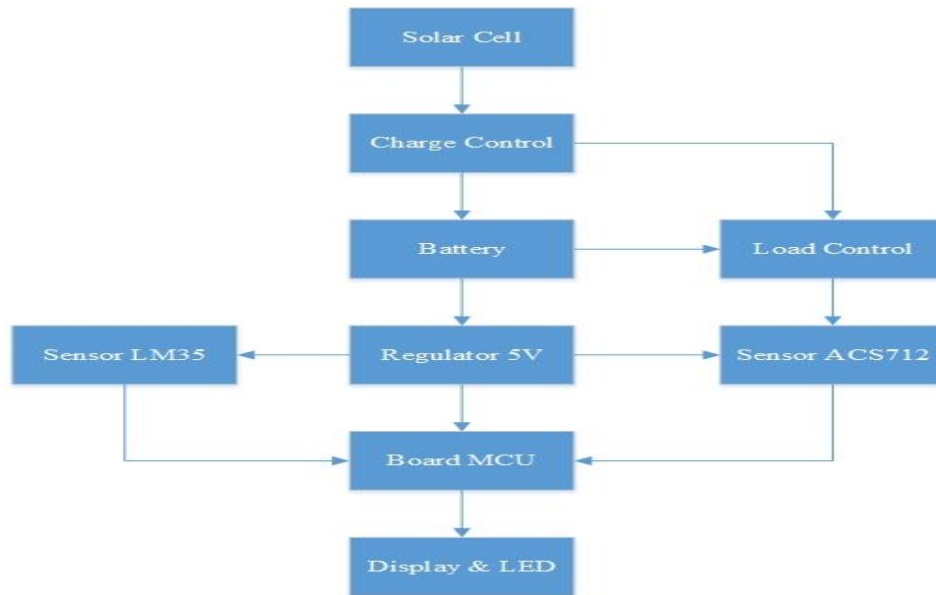




### **3.4 Tahapan Perancangan Alat**

Tahapan ini meliputi perencanaan ke depan dimana kita membutuhkan blok diagram fungsi yang mana terdiri dari komponen-komponen yang mendukung kinerja kerja perancangan alat yang akan dikerjakan. Selanjutnya adalah *flowchart* program. Dimana *flowchart* ini berfungsi untuk merancang program yang akan ditulis nantinya.

#### **3.4.1 Perancangan Blok Diagram Sistem**



Gambar 3.7: Blok Diagram Sistem

Pada gambar diatas terdapat beberapa blok yang masing-masing berfungsi membentuk suatu koordinasi supaya tercapai tujuan yang diinginkan, yaitu input, proses, dan output. Input adalah merupakan *setpoint* sistem, yaitu suatu nilai atau besaran yang dimasukkan agar diperoleh output yang diinginkan.

Perencanaan regulator DC berfungsi untuk memberi input tegangan ke sensor arus ACS712, sensor LM35, dan Display LCD. Sedangkan untuk kontroller Arduino-Uno menggunakan suplay 12 volt langsung dari power suplay.

Pada rangkaian sensor arus ACS712 berfungsi sebagai pemberi sinyal analog yang nantinya akan diubah oleh mikrokontroler ke dalam satuan ampere dan tegangan. Sedangkan sensor LM35 berfungsi untuk mengukur panas matahari. Hasil pengukuran dari kedua sensor ditunjukkan pada layar LCD 20x4.

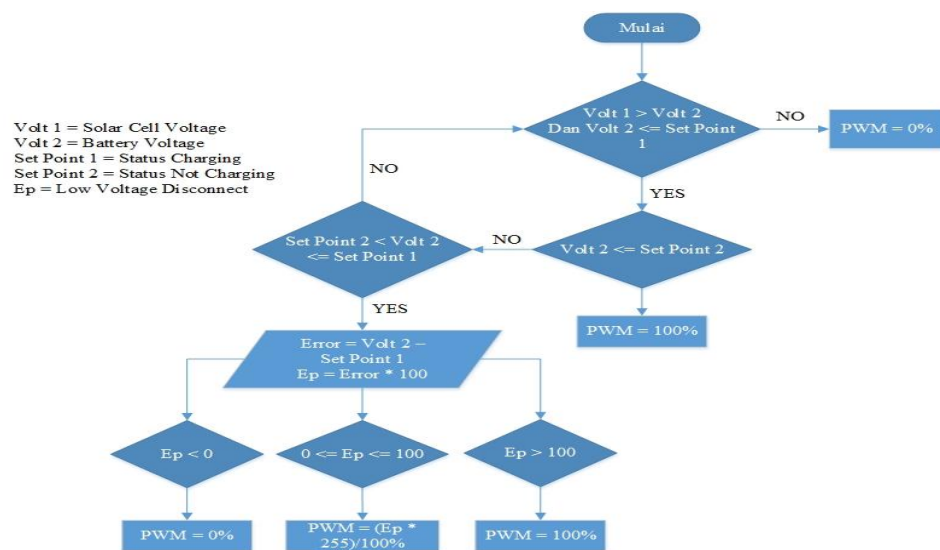
Board kontroller adalah bagian pemroses utama dalam alat ini yang mengatur fungsi kerja dari keseluruhan sistem. Dalam perancangan rangkaian ini, mikrokontroler ATmega328 digunakan sebagai sistem kontrol input dan output.

Input dalam rangkaian ini adalah sensor arus ACS712 dan sensor suhu LM35. Sedangkan output (keluaran) dari mikrokontroler dihubungkan ke piranti monitor dan indikator.

Rangkaian LCD digunakan untuk menampilkan kondisi bacaan sensor terhadap kondisi pembacaan arus dan tegangan dari solar Cell dan suhu disekitarnya.

### 3.4.2 Diagram Alir Perancangan Sistem

Adapun diagram alir (*flowchart diagram*) untuk mempermudah memahami perancangan alat ini dan juga mempermudah dalam pembuatan program adalah sebagai berikut:



Gambar 3.8 Flowchart sistem perangkat

Pada pemrograman C, pembacaan program dimulai dari baris pertama hingga baris terakhir. Biasanya awal pembacaan program dimulai dari dimulai dari void main dan diikuti dengan program selanjutnya. Pada *Flowchart* diatas,

pembacaan program diawali dengan inisialisasi *input/output*, ini adalah proses dimana program mendaftarkan pin-pin mikrokontroler yang akan digunakan. Selanjutnya adalah pembacaan data dari sensor, pada bagian ini banyak variable-variabel program yang dibuat untuk kepentingan perhitungan matematika yang nantinya akan digunakan untuk mengkonversi nilai-nilai pembacaan data dari sensor yang masih hanya dikenali oleh mesin pemroses itu sendiri, variabel-variabel tersebut kemudian mengolah data-data yang didapatkan kemudian menampilkannya dalam nilai arus, tegangan, suhu dan lain-lain.

Proses selanjutnya adalah menampilkan data-data yang telah dikonversi ke dalam tampilan LCD 20x4. Pada proses ini data tidak langsung ditampilkan ke dalam LCD, tetapi diolah kembali menjadi data I2C. Data I2C tersebut kemudian dikirimkan ke modul I2C untuk dikembalikan menjadi data awal dan ditampilkan ke LCD. Selanjutnya program diulang kembali ke baris awal untuk dibaca dari awal kembali dan begitu seterusnya.

### **3.5 Prosedur Uji Coba Rangkaian**

Setelah rangkaian selesai dikerjakan maka penulis perlu melakukan pengujian terhadap rangkaian secara keseluruhan dengan bergantian

Adapun hal-hal yang perlu penulis lakukan untuk pengujian sebagai berikut:

1. Siapkan alat atau rangkaian yang akan di uji coba.
2. Menyiapkan catu daya
3. Menghubungkan rangkaian atau alat dengan catu daya
4. Setelah rangkaian atau alat terhubung dengan catu daya, maka alat telah siap di uji.
5. Untuk memberikan inputan pada Arduino-Uno, dengan cara memberi inputan melalui sensor ACS712, yaitu dengan cara membaca arus yang berjalan pada rangkaian.
6. LCD digunakan sebagai tampilan dari pembacaan tegangan, arus, dan suhu yang akan diukur.

## **BAB IV**

### **ANALISIS DAN HASIL PEMBAHASAN**

Pengujian dan pengukuran dilakukan untuk membuktikan apakah rangkaian yang sudah di buat bekerja sesuai dengan yang direncanakan . pertama sekali pengujian dilakukan pada setiap bloknnya dan pengujian beberapa blok yang saling berkaitan. Dalam setiap pengujian dilakukan dengan pegukuran yang nantinya akan digunakan untuk menganalisa hardware pendukungnya. Setelah semua komponen dipasang dan semua instalasi selesai, lalu dilakukan pemeriksaan ulang terhadap jalur PCB, solderan dan pengawatan agar pengujian dan pengukuran dapat dilaksanakan dengan cepat dan baik.

#### **4.1 Hasil Uji Coba Rangkaian**

Untuk menguji apakah rangkaian yang dirancang berjalan dengan baik, maka perlu dilakukan pengujian. Pengujian rangkaian meliputi pengujian tiap-tiap komponen dan sensor yang akan diuji satu persatu. Agar tidak terjadi kesalahan yang fatal maka pengujian secara satu-persatu harus dilakukan.

##### **4.1.1 Pengujian Sensor Arus ACS712**

Pengujian pada sensor ACS712 adalah dengan cara memberikan beban kepada sensor lalu sensor akan melakukan pengukuran dengan cara mengkonversi arus yang terukur dan mengubahnya menjadi data yang akan ditampilkan ke dalam LCD. Tetapi pada tahap ini pengujian sensor ACS712 tidak langsung pada rangkaian sebenarnya dikarenakan untuk memastikan apakah program dasar untuk menggunakan sensor arus ACS712 yang dihubungkan pada mikrokontroler dapat

bekerja dengan baik atau tidak. Untuk menutup kemungkinan kesalahan yang fatal, maka digunakan simulasi terlebih dahulu. Simulasi ini juga menggunakan software Proteus 7 professional untuk membuktikan apakah program yang dijalankan telah benar atau masih terdapat kesalahan. Program dasarnya adalah sebagai berikut :

```

const int analog_pin = A0;
int mVperAmp = 100; //use 185 for 10 A
int RawValue = 0;
int ACSoffset = 2500;
double Voltage = 0;
double Amps = 0;

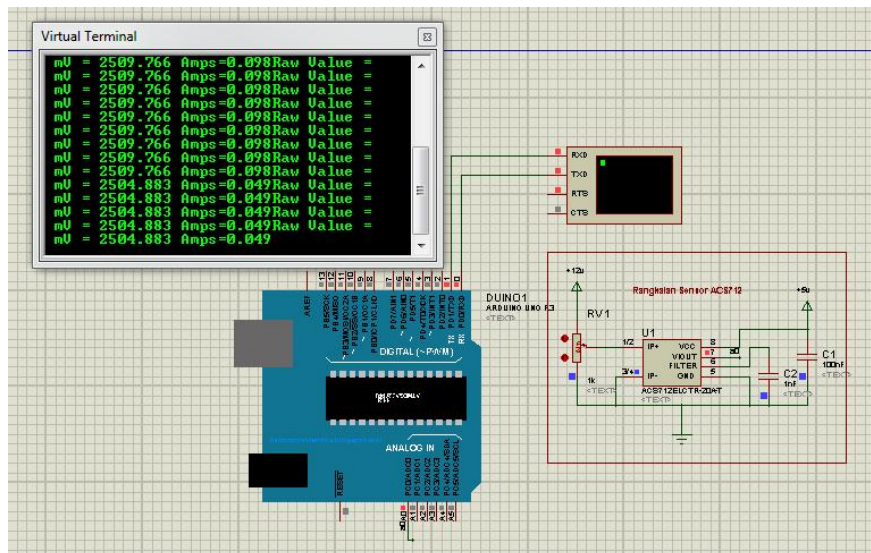
void setup(){
  Serial.begin(9600);
}

void loop(){
  RawValue = analogRead(analog_pin);
  Voltage = (RawValue/1024.0)*5000;
  Amps = ((Voltage - ACSoffset)/mVperAmp);

  Serial.println("Raw Value = ");
  Serial.print("\t mV = ");
  Serial.print(Voltage,3);
  Serial.print("\t Amps=");
  Serial.print(Amps,3);
  delay(1000);
}

```

Simulasi dapat terlihat pada gambar dibawah :



Gambar 4.1 : Pengujian Sensor Arus ACS712

Pada gambar diatas, digunakan *virtual terminal* untuk mengetahui nilai yang terukur oleh sensor ACS712. Untuk mengubah nilai tegangan digunakan sebuah *potensiometer* agar didapat nilai yang beragam dari pengukuran sensor. Hasil yang didapatkan adalah program dasar yang dibuat dapat langsung dimodifikasi untuk aplikasi lebih lanjut karena program tersebut dapat bekerja dengan baik.

#### 4.1.2 Pengujian Sensor Suhu LM35

Untuk melakukan pengukuran, maka lebih dahulu input dari sensor suhu LM35 dihubungkan ke probe positif dan ground ke probe negatif dari voltmeter. Pada rangkaian sensor LM35 terdapat tiga kaki sensor yang utama yaitu Vin, Vout, dan Ground.

Tabel 4.1 Hasil Uji Coba Sensor Suhu LM35

Suhu (Celsius)	Output (V)
31°-33°	0,30V
34°-36°	0,34V

Dari pengujian diketahui tegangan keluaran sensor naik sebesar 0,1V untuk setiap 3°Celsius, maka sensor telah bekerja dengan baik. Untuk membuktikan lebih dalam dikarenakan tidak didapat suhu yang beragam pada sebuah lingkungan, maka pengujian dibuat dengan menggunakan simulasi dengan program dasar sebagai berikut :



```

int val;
int tempPin = 0;

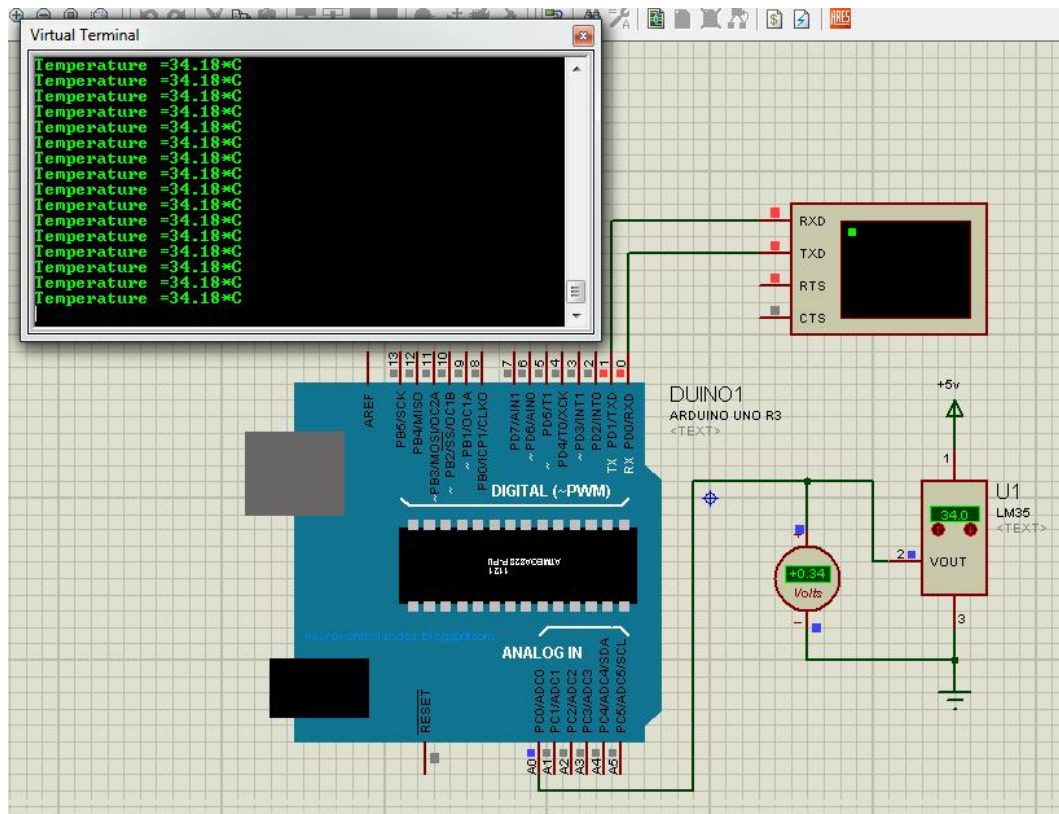
void setup() {
  // put your setup code here, to run once:
  Serial.begin(9600);
}

void loop() {
  // put your main code here, to run repeatedly:
  val = analogRead(tempPin);
  float mv = (val/1024.0*5000);
  float cel = mv/10;
  float farh = (cel*9)/5+32;

  Serial.print("Temperature =");
  Serial.print(cel);
  Serial.print("°C");
  Serial.println();
  delay(1000);
}

```

Dengan simulasi seperti gambar dibawah :



Gambar 4.2 : Pengujian Sensor Suhu LM35

Maka didapatkan hasil yang cukup akurat untuk dilanjutkan pada aplikasi berikutnya.

## 4.2 Analisa *software* Arduino Uno

Di dalam program, LCD yang digunakan adalah LCD 20x4 dengan komunikasi data i2c. Pada bab sebelumnya telah dijelaskan tentang i2c dan keuntungan menggunakan komunikasi data i2c. Untuk menggunakan komunikasi data i2c, diperlukan tambahan *library* agar komunikasi i2c dapat digunakan. Caranya adalah dengan menambahkan *library NewliquidCrystal* pada *folder libraries* yang ada di dalam *software arduino*. *Library* ini dapat di *download* dari situs resmi arduino. Untuk menggunakan *library* tersebut cukup dengan deklarasi nama *library* di awal program sebagai berikut :

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
```

Wire.h berfungsi untuk memudahkan penempatan *address* dan data-data yang akan digunakan. Selanjutnya adalah pengenalan variabel-variabel yang akan digunakan di dalam program seperti di bawah :

```
#define SOL_ADC      A0
#define BAT_ADC      A1
#define CURRENT_ADC  A2
#define TEMP_ADC     A3
#define AVG_NUM      10
#define BAT_MIN      10.5
#define BAT_MAX      15.0
#define BULK_CH_SP   14.4
#define FLOAT_CH_SP  13.6
#define LVD          11.5
#define PWM_PIN      3
#define LOAD_PIN     2
#define BAT_RED_LED  5
#define BAT_GREEN_LED 6
#define BAT_BLUE_LED 7
#define LOAD_RED_LED 8
#define LOAD_GREEN_LED 9
```

Inisialisasi ini sebagai pengenalan set pin komponen-komponen yang terhubung pada *board*. Didalam pin A0 terhubung pengukuran tegangan *solar cell*, pin A1

terhubung ke pengukuran tegangan *battery*. Pin A2 terhubung ke sensor ACS712, dan pin A3 terhubung ke sensor LM35. Inisialisasi yang lain adalah pemberian nilai untuk perhitungan pengukuran.

```
byte solar[8] = {0b11111,0b10101,0b10101,0b11111,0b10101,0b11111,0b00000};
byte battery[8] = {0b01110,0b11011,0b10001,0b10001,0b10001,0b10001,0b10001,0b11111};
byte energy[8] = {0b00010,0b00100,0b01000,0b11111,0b00010,0b00100,0b01000,0b00000};
byte temp[8] = {0b00100,0b01010,0b01010,0b01110,0b01110,0b11111,0b11111,0b01110};
byte charge[8] = {0b01010,0b10001,0b10001,0b10001,0b01110,0b00100,0b00100,};
byte not_charge[8] = {0b00000,0b10001,0b01010,0b00100,0b01010,0b10001,0b00000,0b00000,};
```

Program diatas berfungsi untuk membentuk karakter-karakter khusus akan digunakan pada program, untuk membentuk karakter-karakter khusus tersebut adalah dengan membentuk bit-bit bilangan biner sampai di dapatkan bentuk karakter yang diinginkan.

```
float solar_volt = 0;
float bat_volt = 0;
float load_current = 0;
int temperature = 0;
int temp_change = 0;
float system_volt = 0;
float bulk_charge_sp = 0;
float float_charge_sp = 0;
float charge_status = 0;
float load_status = 0;
float error = 0;
float Ep = 0;
int duty = 0;
float lvd;
float msec = 0;
float last_msec = 0;
float elapsed_msec = 0;
float elapsed_time = 0;
float ampSecs = 0;
float ampHours = 0;
float watts = 0;
float wattSecs = 0;
float wattHours = 0;
```

Program diatas berfungsi untuk pemberian nilai awal dan tipe data di dalam pengukuran.

```
LiquidCrystal_I2C lcd(0x3F, 2, 1, 0, 4, 5, 6, 7, 3, POSITIVE); //inisialisasi address I2C
```

Program diatas adalah inisialisasi address i2C yang digunakan. *Chip* i2C yang digunakan adalah PCF8574AT yang dapat diakses dengan *address* 0x3F. selanjutnya adalah inisialisasi pin dan tipe *backlight* LCD yang digunakan adalah positif.

```
void setup() {  
  // put your setup code here, to run once:  
  Serial.begin(9600);  
  pinMode(BAT_RED_LED, OUTPUT);  
  pinMode(BAT_GREEN_LED, OUTPUT);  
  pinMode(BAT_BLUE_LED, OUTPUT);  
  pinMode(LOAD_RED_LED, OUTPUT);  
  pinMode(LOAD_GREEN_LED, OUTPUT);  
  pinMode(PWM_PIN, OUTPUT);  
  pinMode(LOAD_PIN, OUTPUT);  
  digitalWrite(PWM_PIN, LOW);  
  digitalWrite(LOAD_PIN, LOW);  
  lcd.begin(20,4);  
  lcd.backlight();  
  lcd.createChar(1, solar);  
  lcd.createChar(2, battery);  
  lcd.createChar(3, energy);  
  lcd.createChar(5, temp);  
  lcd.createChar(6, charge);  
  lcd.createChar(7, not_charge);  
  lcd.clear();  
}
```

Didalam pemrograman bahasa C, eksekusi program dimulai dari void main. Di dalam void main terdapat variabel-variabel yang di daftarkan kembali untuk memulai program, dimulai dari *baudrate* atau kecepatan *transfer* data yang digunakan untuk komunikasi serial, selanjutnya adalah mode pin yang akan digunakan apakah *input* atau *output*, banyaknya karakter LCD yang digunakan dan pendaftaran karakter-karakter khusus yang digunakan.

```

void loop() {
  // put your main code here, to run repeatedly:
  read_data();
  system_voltage();
  setpoint();
  charge_cycle();
  power();
  load_control();
  led_indication();
  print_data();
  lcd_display();
}

```

*Void loop* adalah subrutin program. Yaitu program yang akan diulang secara terus-menerus. Didalam *void loop*, hanya variabel yang telah ditulis pada program yang dibawahnya yang dipanggil ulang untuk dikerjakan secara terus-menerus (diulang). Isi dari variabel di dalam *void loop* dapat dilihat pada bagian bawah program.

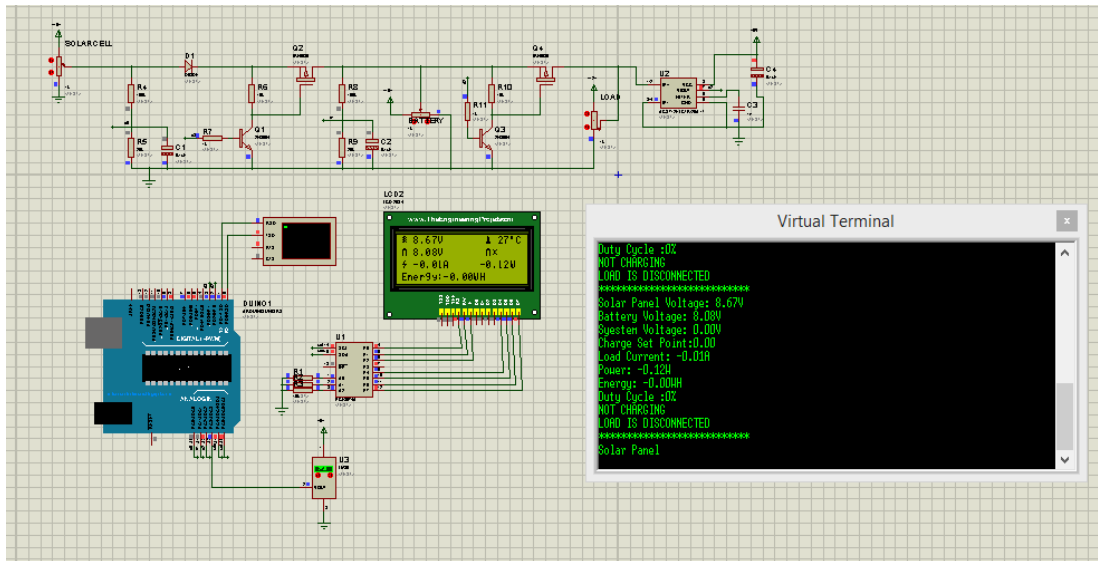
```

int read_adc(int adc_parameter){
  int sum = 0;
  int sample;
  for(int i=0; i<AVG_NUM; i++){
    sample = analogRead(adc_parameter);
    sum += sample;
    delayMicroseconds(50);
  }
  return(sum / AVG_NUM);
}

```

Program diatas digunakan untuk perhitungan ADC yang lebih akurat dikarenakan pembacaan ADC yang digunakan lebih dari 1.

Setelah dilakukan pengujian satu-persatu pada program yang telah dianalisa, maka seluruh program yang telah dibuat dilakukan uji simulasi. Rangkaian simulasi pada *software proteus 7 professional* dapat dilihat pada gambar dibawah :



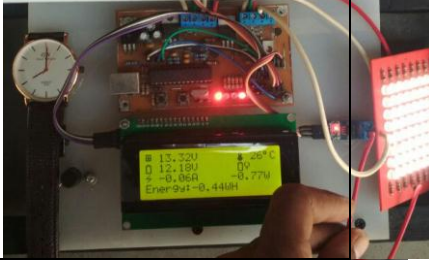


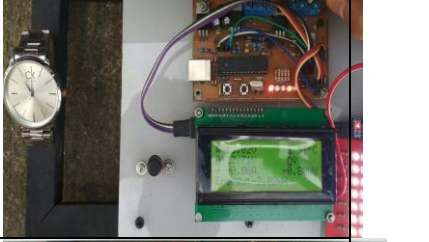


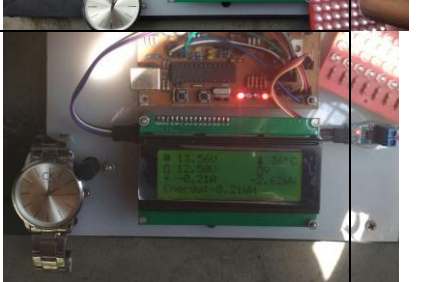
Gambar 4.3 : Rangkaian Simulasi Keseluruhan

### 4.3 Pengujian Secara Keseluruhan

Untuk mendapatkan hasil yang akurat, perlu dilakukan pengujian rangkaian keseluruhan. Pengujian dilakukan disaat matahari terbit dan terbenam. Diambil sampel tiap-tiap jam perubahan sinar matahari dan perubahan tegangan yang didapat dari *solar cell*. Pengujian ini dilakukan untuk lebih memfokuskan apakah alat yang telah dibuat dapat bekerja sebagaimana mestinya.

Hasil pengujian dapat dilihat dari tabel berikut :

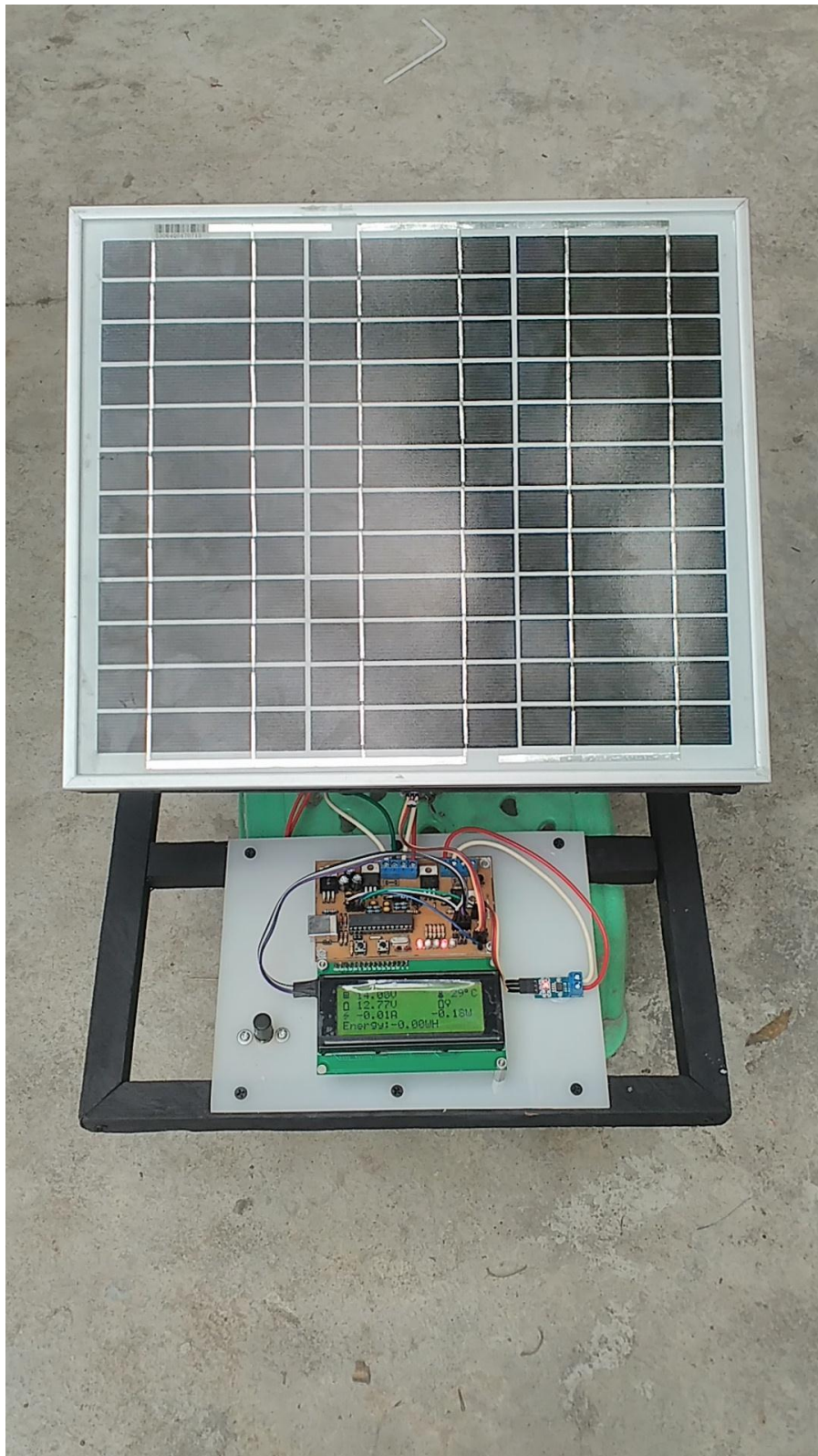
No.	Percobaan (Jam)	Hasil Pengukuran				Gambar Hasil Pengujian
		Tegangan (Volt)	Arus (mA)	Cuaca (Celcius)	Energi terkumpul	
1	07-00	13.29 V	0.06 A	25°C	0.05 WH	

2	08-00	13.32 V	0.06 A	26°C	0.44 WH	
3	09-00	14.2 V	0.06 A	28°C	0.34 WH	
4	10-00	13.91 V	0.03 A	30°C	1.33 WH	
5	11-00	13.82 V	0.06 A	31°C	2.02 WH	
6	12-00	13.85 V	0.65 A	31°C	3.58 WH	
7	13-00	14.44 V	0.36 A	33°C	5.20 WH	
8	14-00	13.46 V	0.21 A	36°C	8.21 WH	

9	15-00	13.64 V	0.16 A	33°C	11.17 WH	
10	16-00	13.47 V	0.06 A	36°C	13.47 WH	
11	17-00	06.03 V	0.06 A	27°C	14.33 WH	
12	18-00	02.20 V	0.11 A	25°C	15.09 WH	



Untuk hasil keseluruhan dapat dilihat pada gambar di bawah :



Gambar 4.4 : Hasil Uji Coba Rangkaian Keseluruhan

## BAB V PENUTUP

### 5.1 Kesimpulan

Berdasarkan perancangan, pengujian dan analisa yang telah dilakukan pada penelitian ini, maka dapat diperoleh beberapa kesimpulan sebagai berikut :

1. Sensor tegangan digunakan untuk mengetahui voltase tegangan pada *solar cell* dan baterai. Diimplementasikan dengan menggunakan dua rangkaian pembagi tegangan. Terdiri dari dua resistor  $R1 = 100k\Omega$  dan  $R2 = 20 k\Omega$  untuk membagi tegangan yang didapat dari *solar cell* dan untuk membagi tegangan dari baterai digunakan 2 resistor yang sama seperti resistor untuk membagi tegangan pada *solar cell*. Resistor ini kemudian dihubungkan secara seri dimana salah satu pin resistor  $100k\Omega$  terhubung ke sumber tegangan baterai atau *solar cell* dan salah satunya lagi terhubung ke *ground*. Diantara pin resistor yang saling terhubung secara seri dihubungkan ke pin analog arduino. Jika dirumuskan menjadi

$$V_0 = \frac{R_2}{(R_1 + R_2)} \times V_{in} . \text{ Maka jika dihitung } \frac{20}{(100 + 20)} \times 12 \text{ volt} , \text{ maka}$$

$$\frac{20}{120} \times 12 \text{ dan hasilnya adalah } 2 \text{ volt. Kesimpulannya adalah jika tegangan}$$

12 volt yang dihasilkan oleh baterai atau *solar cell*, maka tegangan yang diterima oleh pin analog arduino adalah 2 volt.

2. Sensor arus yang digunakan adalah sensor ACS712 dengan arus maksimum yang dapat dilewati adalah sebesar 20 Ampere.

3. Sensor suhu digunakan untuk memantau suhu pada luar ruangan ketika baterai sedang dalam proses pengisian. Dikarenakan suhu yang tinggi akan mempengaruhi reaksi kimia baterai.
4. Untuk kalibrasi sensor di dalam program didapatkan perhitungan sebagai berikut :  $5V = \text{ADC count } 1024$ . Maka  $1 \text{ ADC count} = (5/1024) \text{ volt} = 0.0048828 \text{ Volt}$ .  $V_{out} = V_{in} * R_2/(R_1+R_2)$ .  $V_{in} = V_{out} * (R_1+R_2)/R_2$  dimana  $R_1 = 100k\Omega$  dan  $R_2 = 20 k\Omega$ .  $V_{in} = \text{ADC count} * 0.00488 * (120/20) \text{ Volt}$ .
5. Sesuai dengan *datasheet* dari sensor ACS712 yang sensitivitasnya adalah  $= 100\text{mV/A} = 0.100 * I$  (daya saat ini), maka jumlah ADC yang didapatkan dalam program adalah  $\text{ADC count} = 1024/5 * V_{in}$  dan  $V_{in}$  dari keadaan tidak ada beban didapat dari  $V_{CC}/2 = 2.5$  dimana  $V_{CC}$  adalah 5 Volt. Maka jumlah ADC  $= 204.8(2.5+0.1*I) = 512+20.48*I$ .  
 $20.48 * I = (\text{ADC count} - 512)$  maka  $I = (\text{ADC yang didapat} / 20.48) - 512 / 20.48$ . Saat ini  $(I) = 0.04882 * \text{ADC} - 25$ .
6. PWM (Pulse Width Modulation) merupakan modulasi sinyal pulsa untuk teknik cerdas dalam mengontrol tegangan analog melalui teknik digital. PWM (Pulse Width Modulation) ini digunakan untuk mengontrol tegangan pengisian yang akan diberikan oleh pengontrol. Jumlah arus pengisian ditentukan oleh perbedaan antara voltase baterai dan tegangan set titik pengisian yang telah ditentukan pada program. Pengontrol menggunakan 2 tahap algoritma pengisian daya. Menurut algoritma pengisian, sinyal PWM frekuensi tetap diberikan ke salah satu transistor p-MOSFET. Frekuensi dasar PWM yang digunakan adalah 490.20Hz

(frekuensi *default* pada pin 3 arduino. Siklus tugas 0 – 100% disesuaikan dengan sinyal kesalahan. Pengontrol memberikan perintah *HIGH* (1) atau *LOW* (0) ke sisi beban p-MOSFET sesuai dengan sinar yang didapat *solar cell* dan tegangan yang dikonversi *solar cell* tersebut.

## **5.2 Saran**

Penulis dalam merancang sistem ini menyadari banyak kekurangan yang terdapat pada sistem ini, dan untuk itu penulis memberikan beberapa saran agar sistem ini lebih baik dan efektif, pemutus arus seperti saklar pada *solar cell* dapat digunakan agar sistem dapat bekerja dengan baik dan lebih aman. Selanjutnya adalah *packing* yang baik agar sistem dapat bekerja di luar ruangan secara efektif tanpa khawatir akan adanya hujan yang dapat mengakibatkan sistem rusak. Selanjutnya adalah adanya sistem *converter* yang dapat mengubah arus DC 12 volt menjadi arus AC yang dapat digunakan untuk lampu *emergency* jika listrik PLN mati.

## DAFTAR PUSTAKA

Budiharto, Widodo dan Sigit Firmansyah.2008, *Elelektronika Digital + Mikroprosesor*, Penerbit, Andi Offset Yogyakarta.

Ardianto, Heri. Juli 2008. *Pemrograman Mikrokontroller AVR ATMEGA16 Menggunakan bahasa C (Code Vision AVR)*. Bandung

Blocher, Richard. 2004, *Dasar Elektronika*, Penerbit, Andi Offset Yogyakarta.

Sidik Nurcahyo. 2012, *Mikrokontroler AVR Atmel*, Penerbit. Andi Yogyakarta

Abdul Kadir. 2014, *Buku Pintar Pemrograman Arduino*, Penerbit. MediaKom Yogyakarta

Subandi, Slamet Hani, *Korelasi Suhu dan Intensitas Cahaya Terhadap Daya pada Solar Cell*, 15 November 2014

Mahmud Wasfi, *Solar Energy and Photovoltaic Systems*, February Edition 2011

Lingga, Wardana. 2006. *Belajar Sendiri Mikrokontroller AVR Seri ATMEGA16, Simulasi Hardware dan aplikasi*. Yogyakarta

Bishop, Owen.2004.*Dasar-Dasar Elektronika* Jakarta:Erlangga

Usman, 2008. *Teknik Antar muka + Pemrograman Mikrokontroller AT89S52*, Andi Yograkarta

<http://teundiksha.files.wordpress.com/2010/04/sekilas20codevisionavr.pdf>

Lampiran Program :

```
/*Solar Cell Charge Controller Project
V.1*/
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#define SOL_ADC    A0
#define BAT_ADC    A1
#define CURRENT_ADC  A2
#define TEMP_ADC   A3
#define AVG_NUM    10
#define BAT_MIN    10.5
#define BAT_MAX    15.0
#define BULK_CH_SP 14.4
#define FLOAT_CH_SP 13.6
#define LVD        11.5
#define PWM_PIN    3
#define LOAD_PIN   2
#define BAT_RED_LED  5
#define BAT_GREEN_LED 6
#define BAT_BLUE_LED 7
#define LOAD_RED_LED 8
#define LOAD_GREEN_LED 9

//=====Deklarasi=====\\
//Pembentuk tampilan karakter khusus\\

byte solar[8] = {0b01010,0b10101,0b01010,0b10101,0b01010,0b10101,0b00000};

byte                                     battery[8]                                     =
{0b011110,0b11011,0b10001,0b10001,0b10001,0b10001,0b10001,0b11111};
```

```
byte energy[8] =
{0b00010,0b00100,0b01000,0b11111,0b00010,0b00100,0b01000,0b00000};
```

```
byte temp[8] =
{0b00100,0b01010,0b01010,0b01110,0b01110,0b11111,0b11111,0b01110};
```

```
byte charge[8] = {0b01010,0b10001,0b10001,0b10001,0b01110,0b00100,0b00100,};
```

```
byte not_charge[8] =
{0b00000,0b10001,0b01010,0b00100,0b01010,0b10001,0b00000,0b00000,};
```

```
//=====Deklarasi Variabel=====\\
```

```
float solar_volt = 0;
```

```
float bat_volt = 0;
```

```
float load_current = 0;
```

```
int temperature = 0;
```

```
int temp_change = 0;
```

```
float system_volt = 0;
```

```
float bulk_charge_sp = 0;
```

```
float float_charge_sp = 0;
```

```
float charge_status = 0;
```

```
float load_status = 0;
```

```
float error = 0;
```

```
float Ep = 0;
```

```
int duty = 0;
```

```
float lvd;
```

```
float msec = 0;
```

```
float last_msec = 0;
```

```
float elapsed_msec = 0;
```

```
float elapsed_time = 0;
```

```
float ampSecs = 0;
```

```
float ampHours = 0;
float watts = 0;
float wattSecs = 0;
float wattHours = 0;
```

```
LiquidCrystal_I2C lcd(0x27, 2, 1, 0, 4, 5, 6, 7, 3, POSITIVE); //inisialisasi address I2C
```

```
//=====Main Program Start=====\\
```

```
void setup() {
  // put your setup code here, to run once:
  Serial.begin(9600);
  pinMode(BAT_RED_LED, OUTPUT);
  pinMode(BAT_GREEN_LED, OUTPUT);
  pinMode(BAT_BLUE_LED, OUTPUT);
  pinMode(LOAD_RED_LED, OUTPUT);
  pinMode(LOAD_GREEN_LED, OUTPUT);
  pinMode(PWM_PIN, OUTPUT);
  pinMode(LOAD_PIN, OUTPUT);
  digitalWrite(PWM_PIN, LOW);
  digitalWrite(LOAD_PIN, LOW);
  lcd.begin(20,4);
  lcd.backlight();
  lcd.createChar(1, solar);
  lcd.createChar(2, battery);
  lcd.createChar(3, energy);
  //lcd.createChar(5, temp);
  lcd.createChar(6, charge);
  lcd.createChar(7, not_charge);;
  lcd.clear();
}
```



```

void loop() {
    // put your main code here, to run repeatedly:
    read_data();
    system_voltage();
    setpoint();
    charge_cycle();
    power();
    load_control();
    led_indication();
    print_data();
    lcd_display();
}

int read_adc(int adc_parameter){
    int sum = 0;
    int sample;
    for(int i=0; i<AVG_NUM; i++){
        sample = analogRead(adc_parameter);
        sum += sample;
        delayMicroseconds(50);
    }
    return(sum / AVG_NUM);
}

void read_data(void){
    solar_volt = read_adc(SOL_ADC)*0.00488*(120/20);
    bat_volt = read_adc(BAT_ADC)*0.00488*(120/20);
    load_current = (read_adc(CURRENT_ADC)*.0488 - 25);
    temperature = read_adc(TEMP_ADC)*0.00488*100;
}

```

```
void power(void){
    msec = millis();
    elapsed_msec = msec - last_msec;
    elapsed_time = elapsed_msec / 1000.0;
    watts = load_current * bat_volt;
    ampSecs = (load_current*elapsed_time);
    wattSecs = ampSecs * bat_volt;
    ampHours = ampHours + ampSecs/3600;
    wattHours = wattHours + wattSecs/3600;
    last_msec = msec;
}
```

```
void print_data(void)
{
    delay(100);
    Serial.print("Solar Panel Voltage: ");
    Serial.print(solar_volt);
    Serial.println("V");
    Serial.print("Battery Voltage: ");
    Serial.print(bat_volt);
    Serial.println("V");
    Serial.print("System Voltage: ");
    Serial.print(system_volt);
    Serial.println("V");
    Serial.print("Charge Set Point:");
    Serial.println(bulk_charge_sp);
    //Serial.print("Temperature:");
    //Serial.print(temperature);
    //Serial.println("C");
    Serial.print("Load Current: ");
```

```
Serial.print(load_current);
Serial.println("A");
Serial.print("Power: ");
Serial.print(watts);
Serial.println("W");
Serial.print("Energy: ");
Serial.print(wattHours);
Serial.println("WH");
Serial.print("Duty Cycle :");
if (charge_status==1)
{
Serial.println("99%");
Serial.println("BULK CHARGING");
}
else if (charge_status==2)
{
Serial.print(Ep);
Serial.println("%");
Serial.println("FLOAT CHARGING");
}
else
{
Serial.println("0%");
Serial.println("NOT CHARGING");
}
if(load_status==1)
{
Serial.println("LOAD IS CONNECTED");
}
else
{
```

```

Serial.println("LOAD IS DISCONNECTED");
}

Serial.println("*****");
}

void system_voltage(void){
    if((bat_volt > BAT_MIN) && (bat_volt < BAT_MAX)){
        system_volt = 12;
    }
    else if((bat_volt > BAT_MIN/2) && (bat_volt < BAT_MAX/2)){
        system_volt = 6;
    }
}

void setpoint(void){
    temp_change = temperature-25.0;
    if(system_volt==12){
        bulk_charge_sp = BULK_CH_SP-(0.030*temp_change);
        float_charge_sp = FLOAT_CH_SP-(0.030*temp_change);
        lvd = LVD;
    }
    else if(system_volt==6){
        bulk_charge_sp = (BULK_CH_SP/2)-(0.015*temp_change);
        float_charge_sp = (FLOAT_CH_SP/2)-(0.015*temp_change);
        lvd=LVD/2;
    }
}

void charge_cycle(void){
    if(solar_volt > bat_volt && bat_volt <= bulk_charge_sp){

```

```

if(bat_volt <= float_charge_sp){
    charge_status = 1;
    duty = 252.45;
    analogWrite(PWM_PIN,duty);
}
else if(bat_volt > float_charge_sp && bat_volt <= bulk_charge_sp){
    charge_status = 2;
    error = (bulk_charge_sp - bat_volt);
    Ep = error * 100;
    if(Ep < 0){
        Ep = 0;
    }
    else if(Ep>100){
        Ep = 100;
    }
    else if(Ep>0 && Ep <= 100){
        duty = (Ep*255)/100;
    }
    analogWrite(PWM_PIN,duty);
}
}
else{
    charge_status = 0;
    duty = 0;
    analogWrite(PWM_PIN,duty);
}
}

void load_control(){
    if(solar_volt < 5){
        if(bat_volt > lvd){

```

```
    load_status = 1;
    digitalWrite(LOAD_PIN, HIGH);
}
else if(bat_volt < lvd){
    load_status = 0;
    digitalWrite(LOAD_PIN, LOW);
}
}
else{
    load_status = 0;
    digitalWrite(LOAD_PIN, LOW);
}
}
```

```
void led_indication(void){
    battery_led();
    load_led();
}
```

```
void battery_led(void){
    if((bat_volt > system_volt) && (bat_volt < bulk_charge_sp)){
        leds_off_all();
        digitalWrite(BAT_GREEN_LED, LOW);
    }
    else if(bat_volt >= bulk_charge_sp){
        leds_off_all();
        digitalWrite(BAT_BLUE_LED, LOW);
    }
    else if(bat_volt < system_volt){
        leds_off_all();
        digitalWrite(BAT_RED_LED, LOW);
    }
}
```

```
    }  
}
```

```
void load_led(){  
    if(load_status==1){  
        digitalWrite(LOAD_GREEN_LED, HIGH);  
    }  
    else if(load_status==0){  
        digitalWrite(LOAD_RED_LED, HIGH);  
    }  
}
```

```
void leds_off_all(void){  
    digitalWrite(BAT_RED_LED, HIGH);  
    digitalWrite(BAT_GREEN_LED, HIGH);  
    digitalWrite(BAT_BLUE_LED, HIGH);  
    digitalWrite(LOAD_RED_LED, LOW);  
    digitalWrite(LOAD_GREEN_LED, LOW);  
}
```

```
void lcd_display(){  
    lcd.setCursor(0,0);  
    lcd.write(1);  
    lcd.setCursor(2,0);  
    lcd.print(solar_volt);  
    lcd.print("V");  
    //lcd.setCursor(14,0);  
    //lcd.write(5);  
    //lcd.setCursor(16,0);  
    //lcd.print(temperature);  
    //lcd.write(0b11011111);
```

```
//lcd.print("C");  
lcd.setCursor(0,1);  
lcd.write(2);  
lcd.setCursor(2,1);  
lcd.print(bat_volt);  
lcd.print("V");  
lcd.setCursor(14,1);  
lcd.write(2);  
if((charge_status==1) | (charge_status==2)){  
    lcd.write(6);  
}  
else{  
    lcd.write(7);  
}  
lcd.setCursor(0,2);  
lcd.write(3);  
lcd.setCursor(2,2);  
lcd.print(load_current);  
lcd.print("A");  
lcd.setCursor(13,2);  
lcd.print(watts);  
lcd.print("W");  
lcd.setCursor(0,3);  
lcd.print("Energy:");  
lcd.print(wattHours);  
lcd.print("WH");  
}
```