BAB III

Desain dan Implementasi Digital Maximum Power Point Tracker Berbasis Mikrokontroller ATMEGA8535

3.1 Pendahuluan

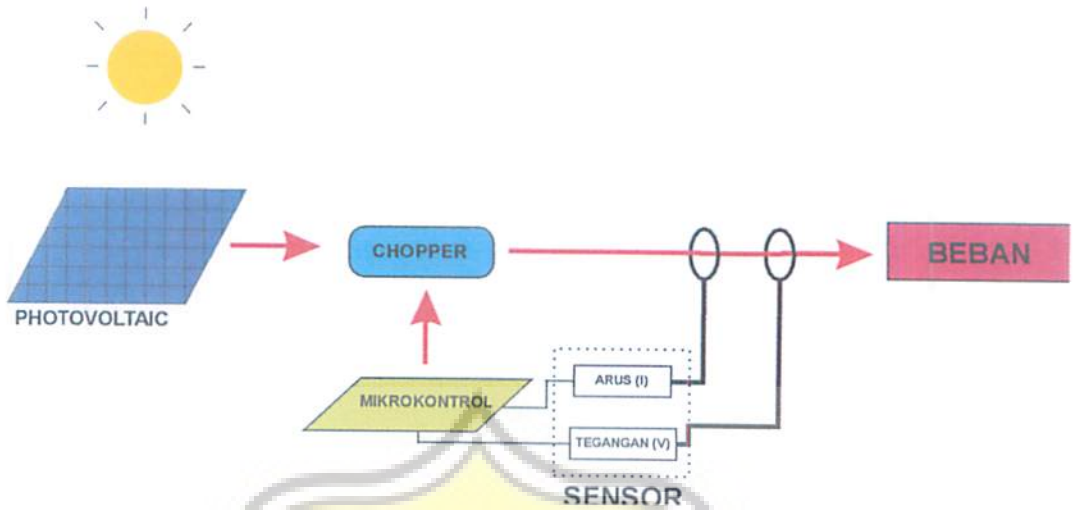
Photovoltaic merupakan perangkat yang mengkonversikan dari energi sinar matahari menjadi energi listrik. *Photovoltaic* tidak menghasilkan suatu tegangan ataupun arus tetapi penggabungan antaran keduanya, yaitu daya. Inilah mengapa sebuah PV memiliki karakteristik jika beban terlalu kecil maka tegangannya besar tetapi arusnya kecil maka daya yang dihasilkan PV tidak maksimal, sebaliknya jika beban terlalu besar maka tegangan pada PV akan drop tetapi arusnya besar maka daya yang dihasilkan juga tidak maksimal.

Untuk memaksimalkan daya yang dihasilkan oleh PV selalu maksimal dibutuhkan suatu rangkaian yang disebut MPPT. Telah dikembangkan bermacam – macam MPPT antara lain dengan kendali *Buck, Boost, Analog* dll.

Pada Tugas Akhir ini akan mendesain dan mengimplementasikan *digital maximum power point tracker* berbasis *mikrokontrol* ATEMEGA8535. Tugas Akhir ini dilaksanakan berdasarkan pada penelitian yang terdahulu yaitu pengembangan dari sistem analog.

3.2 Perancangan Alat

Pada garis besar fungsi dari alat yang telah diimplementasikan digambarkan dalam suatu blok diagram sebagai berikut:



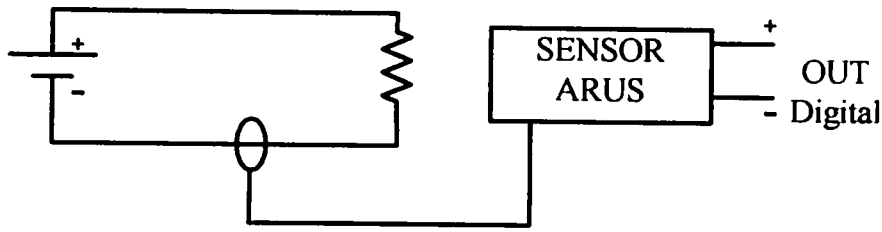
Gambar 3.1 Blok diagram rancangan alat

Pada gambar diatas menujukan tegangan dari PV menuju ke *Chopper*. Tegangan dari PV akan dikontrol oleh *Chopper* supaya daya yang dihasilkan selalu maksimum power point. *Chopper* mendapat inputan kontrol dari mikrokontrol yang dimana kontrol dari mikrokontrol tersebut diolah dari perkalian antara sensor arus dan sensor tegangan.

Jika perbandingan antara nilai sensor arus lebih besar dari nilai sensor tegangan, maka modulasi yang dikeluakan oleh mikrokontrol akan lebih lama pada saat on. Sedangkan jika nilai sensor tegangan lebih besar dari sensor arus, maka modulasi yang dikeluarkan oleh mikrokontol akan lebih lama pada posisi off.

3.3 Sensor Arus

Sensor arus berfungsi untuk mengubah data *analog* dari rangkaian menjadi output digital 5v. Gambar dibawah adalah blok diagram rangkaian sensor arus

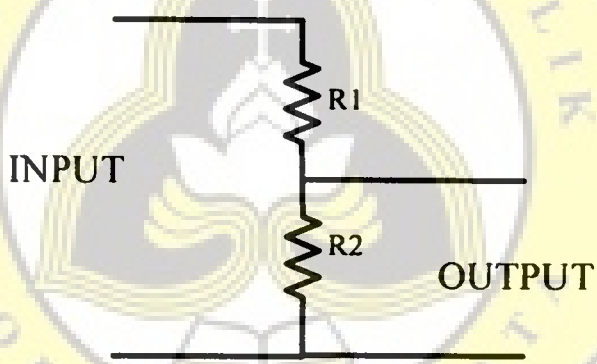


Gamabar 3.2 Blok diagram sensor arus

Arus yang disensor akan diubah menjadi tegangan 5v yang nantinya digunakan sebagai data inputan *mikrokontrol.*

3.4 Sensor Tegangan

Rangkaian sensor tegangan ini terdiri dari resistor yang dirangkai secara seri dengan input, dan outputnya diambil dari titik pertemuan antar resistor seperti pada gambar dibawah.



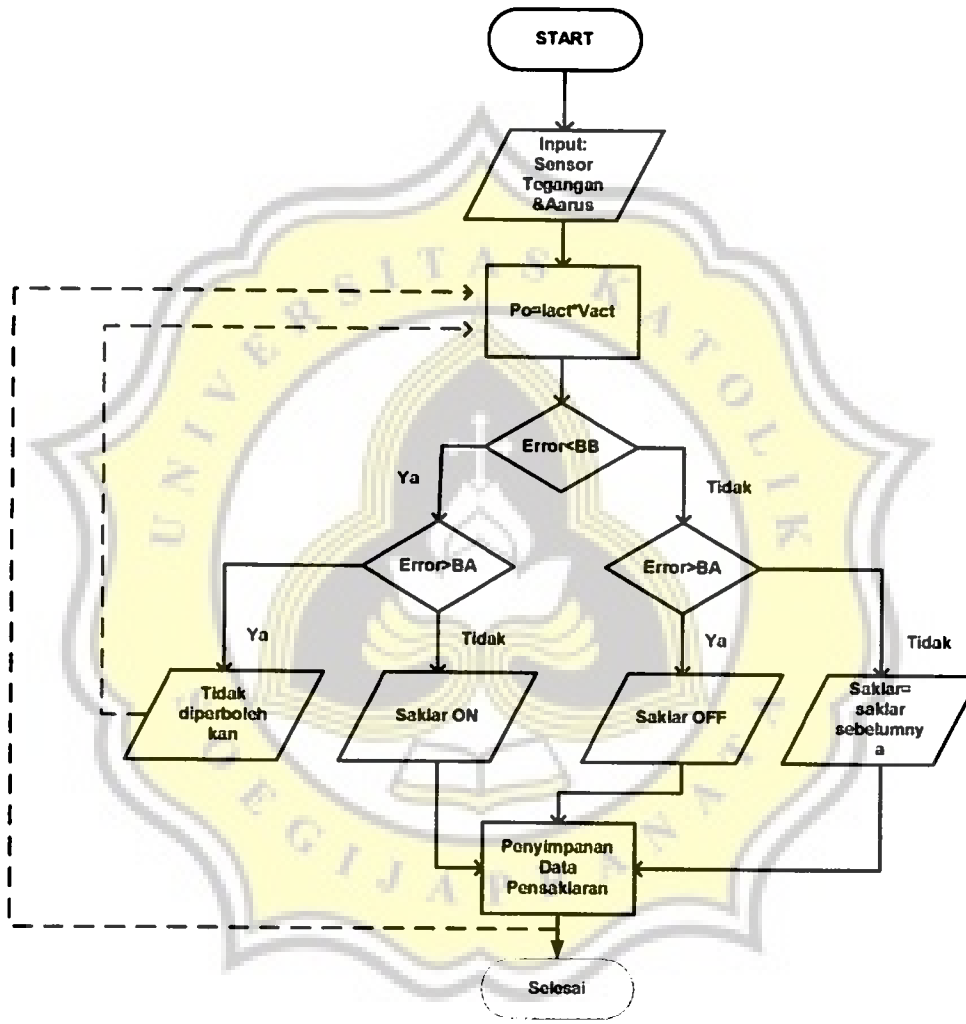
Gambar 3.3 Rangkaian prmbagi tagangan

Dari rangkaian diatas maka digunakan rumus sebagai berikut:

$$V_{OUT} = \frac{R2}{R1 + R2} x V_{IN} \qquad (3.1)$$

3.6 Analisa Program

Sebelum membuat programn perlulah perancangan program dengan cara pembuatan *Flow Chart* Sistem Kontrol. Berikut *Flow Chart* Sistem Kontrol program pada mikrokontrol:



Gambar 3.5 Flow Chart Sistem Kontrol

Program pada mikrokontrol dibuat dengan menggunakan *software Code Vision AVR* yang bahasa programnya menggunakan bahasa C. program yang dibuat ada beberapa tahap yaitu:

1. Membaca sensor
2. Pengolahan data dan pembuatan Last Data
3. Perbangingan Po dengan batas atas dan batas bawah *Hysterisis*
4. *Flip – Flop*

a. Membaca sensor

Sensor yang dihasilkan oleh sensor tegangan merupakan data *analog*. Untuk mendapatkan hasil digital, maka menggunakan fitur dari mikrokontrol yang berfungsi sebagai *ADC (Analog Digital Converter)*. Dengan menggunakan ADC tersebut, maka data analog dari sensor akan diubah menjadi digital.

Untuk mengaktifkan ADC pada ATMEGA8535 diperlukan proses inisiasi atau pemanggilan pada register yang dibutuhkan. Berikut inisiasi *register* ADC:

```
15  int Pembagian;
16  int c;
17  // Read the 8 most significant bits
18  // of the AD conversion result
19  unsigned char read_adc(unsigned char adc_input)
20  {
21  ADMUX=adc_input | (ADC_VREF_TYPE & 0xff);
22  // Delay needed for the stabilization of the ADC input voltage
23  delay_us(10);
24  // Start the AD conversion
25  ADCSRA|=0x40;
26  // Wait for the AD conversion to complete
27  while ((ADCSRA & 0x10)==0);
28  ADCSRA|=0x10;
29  return ADCH;
30  }
```

Gambar 3.6 Inisiasi ADC pada ATMEGA8535

Maka PORTA ATMEGA8535 akan aktif sebagai Pin ADC dan PORTA tersebut hanya dapat digunakan untuk input ADC saja. Berikut list program ADC pada CV AVR:

```
66
67  // ADC initialization
68  // ADC Clock frequency: 691,200 kHz
69  // ADC Voltage Reference: AVCC pin
70  // ADC High Speed Mode: Off
71  // ADC Auto Trigger Source: ADC Stopped
72  // Only the 8 most significant bits of
73  // the AD conversion result are used
74  ADMUX=ADC_VREF_TYPE & 0xff;
75  ADCSRA=0x84;
76  SFIOR&=0xEF;
77
```

Gambar 3.7 Setting ADC pada ATmega 8535

Data dari PORTA digunakan untuk membaca sensor ADC. Pin yang digunaka adalah Pin2 dan Pin3. Berikut perintah membaca program dan kemudian akan dibaca sebagai variabel "Vact" unutk proses lebih lanjut.

```
82  {
83      Lastout=PORTB.6;
84      //MEMBACA SENSOR
85      Vact=read_adc(2)*5;
86      Iact=read_adc(3)*5;
87
```

Gambar 3.8 Pembacaan ADC pada ATmega 8535

b. Pengolahan Data dan Pembuatan Last Data

Data yang dihasilkan oleh ADC akan diolah data dengan cara mengalikan kedua data tersebut, yaitu data I dan data V. perkalian I dan V tersebut didapatkan data, yaitu data Po. Setelah proses tersebut maka dibuatlah data terakhir (Last Data). Berikut listing program:

```
82  {
83  Lastout=PORTB.6;
84  //MEMBACA SENSOR
85  Vact=read_adc(2)*5;
86  Iact=read_adc(3)*5;
87
88  //PENGOLAHAN DATA
89  Po=(float)Vact*Iact;                    //MENGHITUNG NILAI Po
90  delayPo=(Po-lastPo)/a;
91  delayV=(Vact-lastV)/a;
92  Perbagian=delayPo/delayV;          //MENCARI NILAI MODULASI
93
94  //PENGUBAHAN LAST DATA (V(i-1) dan Po(i-1))
95  lastV=Vact;
96  lastPo=Po;
```

Gambar 3.9 Pengolahan data pada ATmega 8535

c. Perbandingan nilai Po dengan batas atas dan batas bawah Hysterisis

Setelah didapatkan data Po maka data tersebut akan dibandingkan dengan menggunakan kontrol *hysterisis*, yaitu dibatasi batas atas dan batas bawahnya, maka program yang ditulis sebagai berikut:

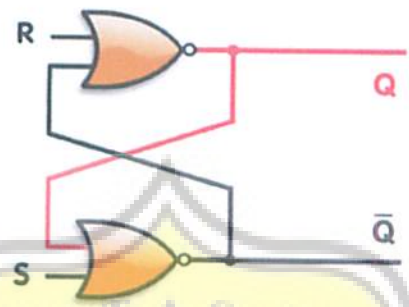
```
97
98   //HYSSRESIS
99
100  e=Perbagian;
101
102  //COMPARATOR
103  if(e>=2.55)
104  {R=1;}
105
106  if(e<2.55)
107  {R=0;}
108
109  if(e<=-2.55)
110  {S=1;}
111
112  if(e>-2.55)
113  {S=0;}
114
```

Gambar 3.10 Listing Program Komparator

d. Flip – Flop

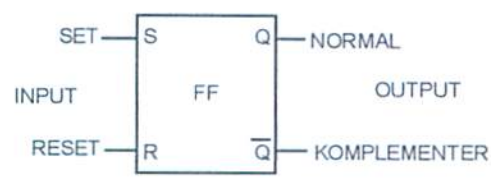
SR-flip flop merupakan gabungan dari beberapa gerbang logika. *Flip Flop SR* memilki dua masukkan S untuk *Set* dan R untuk *Reset*. Gerbang logika NAND biasa digunakan untuk membuat *Flip Flop SR*. Symbol logika

menunjukan dua masukan yang diberi label dengan *Set* dan*Reset*. *Flip Flop SR* ini mempunyai dua keluaran komplementer. Keluaran ini diberi label Q dan $\overline{Q}$. Nilai Q dengan $\overline{Q}$ selalu berlawanan.



Gambar 3.11 Rangkaian SR Flip – Flop

Sinyal SR yang masukan ke dalam *flip flopp* dapat memiliki 4 kemungkinan kondisi yaitu 00, 01, 10, dan 11. Pada saat SR bernilai 00 kemungkinan kondisi flip flop tidak berubah, nilai Q akan sperti sebelumnya. Jika SR bernilai 01 maka keluaran Q akan bernilai 0, kondisi ini akan menyebabkan flip flop reset. Jika SR belnilai 10 maka keluaran Q akan bernilai 1 atau flip flop set. Bagimana kalau SR bernilai 11, ini menarik karena kondisi ini menyebabkan keluaran Q tidak pasti, tegantung sinyal mana yang datang lebih cepat. Kondisi ini disebut kondisi berlomaba (race condition). Karena nilai Q tidak pasti maka kondisi ini tidak digunakan. Kondisi Q$\overline{Q}$ bernilai 00 terjadi pada saat perpindahan dari nilai SR 01 ke -10.



Gambar 3.12 Blok SR Flip – Flop

Berikut listing program SR Flip – Flop:

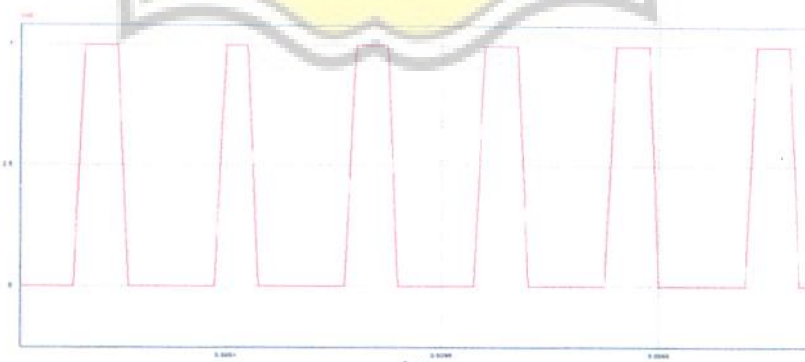
```
114
115         //FLIF-FLOP
116         if((R==1)&&(S==0))
117         {
118           PORTB.6=0;
119         }
120         if((R==0)&&(S==1))
121         {
122           PORTB.6=1;
123         }
124         if((R==0)&&(S==0))
125         {
126           PORTB.6=Lastout;
127         }
128
129         }
```

Gambar 3.13 Listing Program Flip – Flop

3.7 Modulasi Swiching Chopper

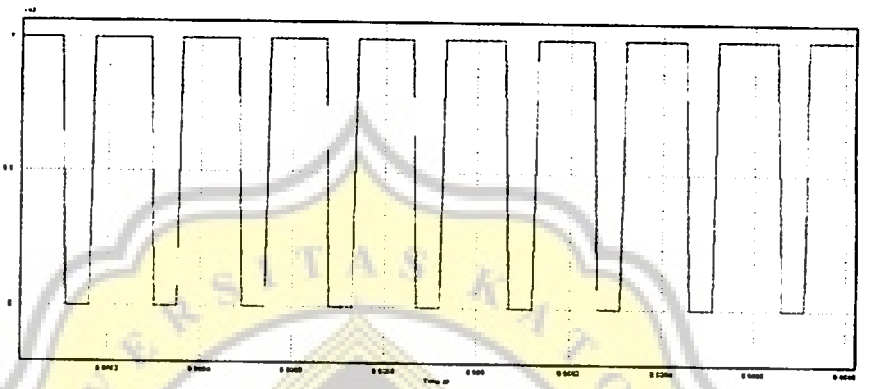
Pada kendali *switching chopper* tersebut menggunakan sistem modulasi dimana ada saat on yang lebih lama, dan ada saat off yang lebih lama. Sistem modulasi tersebut akan berubah – ubah seiring dengan beratnya beban yang dibebani, kecerahan cahaya matahari yang berubah ubah, dan hal – hal yang lain.

- Jika beban lebih kecil dari tegangan sumber maka bekerja sebagai buck chooper. Dengan demikian modulasi sinyal yang dihasilan pada posisi off lebih lama dari pada posisi on. Seoerti pada gambar dibawah ini.



Gambar 3.14 Modulasi Duty Cycle

- Jika beban lebih besar dari tegangan sumber maka bekerja sebagai *boost chopper*. Dengan demikian modulasi sinyal yang dihasilkan pada posisi on lebih lama dar pada posisi off. Seperti pada gambar dibawah ini.

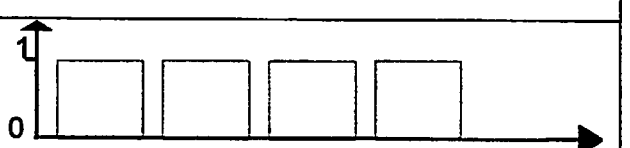


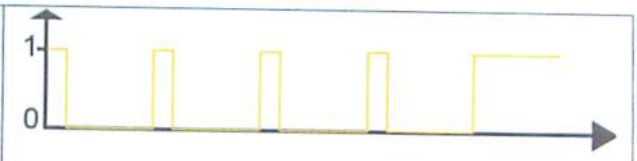
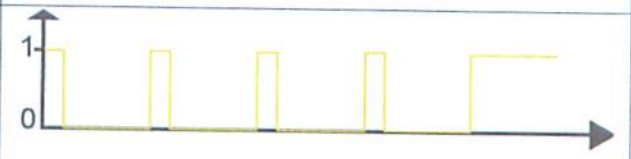
Gambar 3.15 modulasi duty cycle

3.8 Kerja Modulasi Chopper

Pada pengujian yang dilakukan ada beberapa mode modulasi yang digunakan. Modulasi ini akan berubah menyesuaikan dengan beban yang dibebankan dan keceralian cahaya. Perubahan dari sinyal modulasi tersebut digunakan untuk mengatur tegangan dan arus yang dihasilkan oleh PV agar daya yang dihasilkan oleh PV selalu MPP.

Tabel 3.1 Sinyal modulasi

Kondisi	Sinyal Modulasi
◦ Sinar matahari redup	

• Sinar matahari cerah	
• Beban ringan	
• Beban tengah	
• Beban berat	

Jika sinyal modulasi pada posisi on lebih lebar dari posisi off, maka pada keluaran akan terukur tegangan mendekati dengan tegangan input. Karena tegangan yang terpotong hanya sedikit.