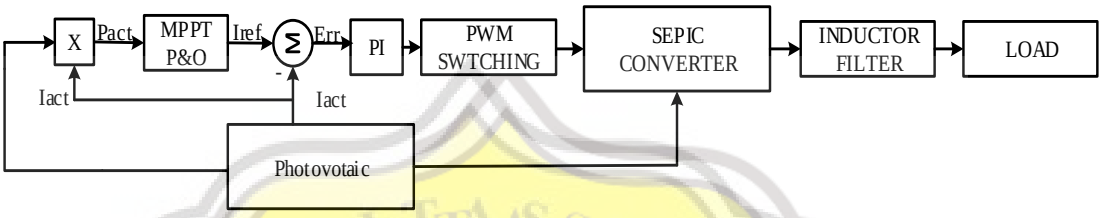
BAB III
DESAIN DAN IMPLEMENTASI

3.1 Pendahuluan



Gambar 3. 1 Diagram blok algoritma MPPT P&O dengan kontrol PI pada SEPIC DC-DC konverter

Pada gambar 3.1 adalah strategi kontrol yang telah diusulkan dan dibuktikan dengan simulasi secara digital menggunakan perangkat lunak pada komputer selanjutnya akan diimplementasikan pada sebuah *hardware* sebagai bukti keberhasilan dari strategi kontrol dengan konverter yang ditampilkan. Desain dari konverter yang ditampilkan disimulasikan menggunakan bantuan aplikasi perangkat lunak PSIM.

Strategi kontrol di atas merupakan algoritma MPPT P&O yang dilengkapi kontrol PI. Dimana daya yang dihasilkan oleh PV akan dideteksi oleh sensor serta digunakan untuk menyuplai SEPIC konverter dan akan digunakan untuk mengisi daya baterai. Tegangan dan arus yang dideteksi oleh sensor akan digunakan sebagai data untuk mikrokontroler STM32F407VET6 dalam proses kalkulasi algoritma MPPT P&O dan menghasilkan arus referensi. Kemudian arus referensi dari mikrokontroler akan dibandingkan dengan arus aktual dari sensor. Hasil perbandingan

tersebut dinamakan sinyal *error* dan kemudian akan diolah pada blok kontrol *Proportional Integral (PI)*. Setelah semua proses terlalui, akan terbentuk PWM yang digunakan untuk mengendalikan saklar elektrik IRFP260 pada SEPIC konverter yang mana PWM sudah sesuai kaidah algoritma MPPT P&O. Maka, output dari SEPIC konverter adalah tegangan DC yang telah terfilter dengan nilai yang maksimal sesuai dengan algoritma.

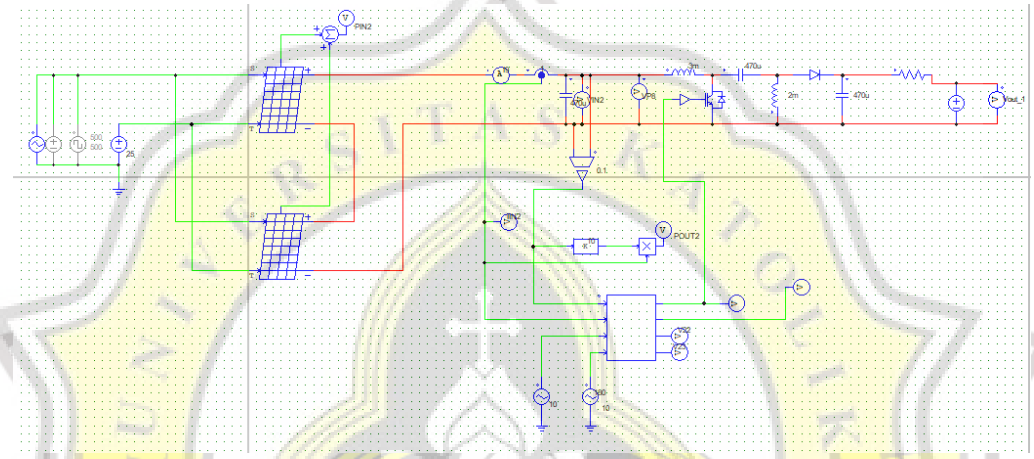
Tabel 1 Parameter simulasi dan implementasi *hardware*

Parameter	**Value**
Photovoltaic (PV)	80 WP dan 80 WP 2 buah (seri)
Irradiance	500 W/m^2 – 900 W/m^2
Kapacitor C1, C2, C3	470 Uf
Induktor L1	2,4 mH
Induktor L2	2,2 mH
Switching Frequency	25 kHz
Tegangan Beban Baterai	12 VDC dan 24 VDC
Saklar Elektrik	MOSFET IRFP260

Tabel 3. 1 menyajikan parameter pada implementasi *hardware*. Berdasarkan implementasi *hardware*, PV yang digunakan sebesar 80 WP sebanyak 4 buah, 2 buah untuk alat kami dan 2 buah untuk MPPT *Solar Charge Control*. Daya yang dihasilkan PV akan dihubungkan ke konverter yang berfungsi untuk mengirimkan tegangan menuju baterai untuk pengisian daya secara konstan dan stabil. Dimana nantinya sinyal aktual

yang dihasilkan oleh konverter harus mengikuti sinyal referensi dengan stabil dan konstan sesuai algoritma MPPT P&O.

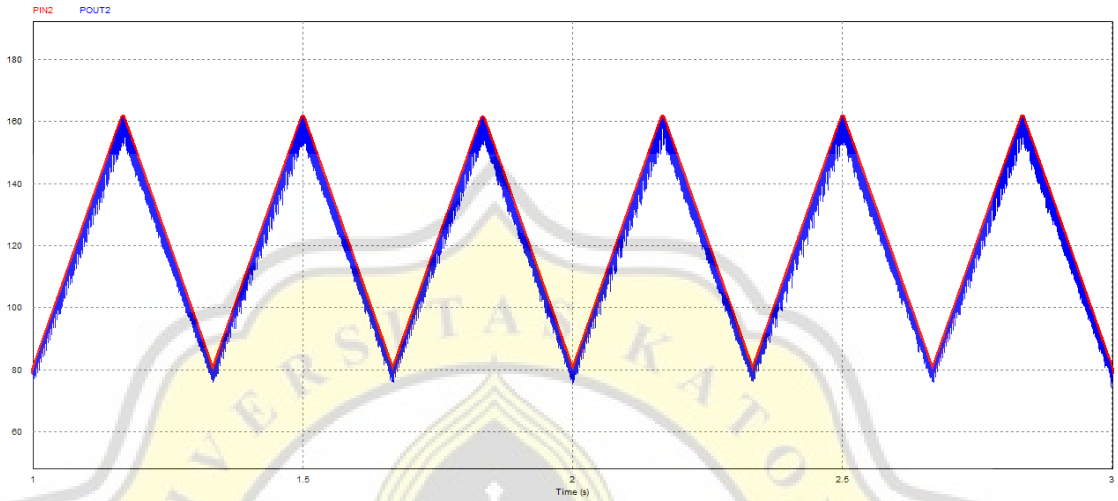
3.2 Simulasi *Hardware* MPPT dengan Algoritma P&O dan PI kontrol pada DC – DC SEPIC Konverter



Gambar 3. 2 Simulasi MPPT beralgoritma P&O pada SEPIC konverter

Sistem perancangan dan kontrol disimulasikan menggunakan perangkat lunak PSIM. Simulasi dilakukan dengan tujuan sebagai langkah awal sebelum implementasi *hardware* agar meminimalisir kesalahan dan kerusakan komponen, serta dapat menganalisa sistem dapat berfungsi sesuai yang ditentukan. Pada aplikasi PSIM sudah tersedia modul PV sehingga memudahkan pengguna untuk mengintegrasikan PV dengan algoritma yang digunakan serta DC-DC konverternya. Pada aplikasi ini juga tersedia blok kontrol bernama *C-block* yang memfasilitasi pengguna untuk mengontrol sistem menggunakan bahasa c. Bahasa c ini merupakan bahasa dasar pemrograman dan banyak digunakan untuk program mikrokontroler termasuk untuk memprogram STM32F407VET6. Berikut pada gambar 3.3

adalah hasil simulasi penghasilan daya dari PV dengan algoritma yang kami rancang.



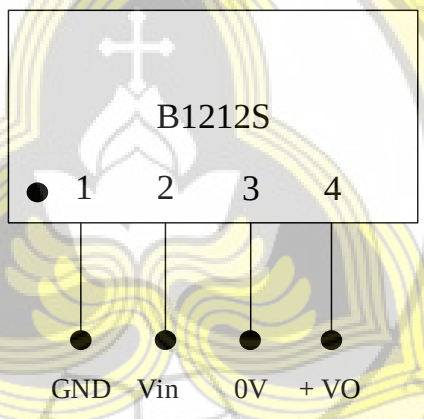
Gambar 3. 3 Simulasi penghasilan daya PV melalui PSIM

Gambar 3.3 di atas adalah grafik penghasilan daya PV, dimana PIN adalah daya masukan yang berasal dari intensitas cahaya matahari. Sedangkan POUT adalah daya keluaran atau tegangan yang dihasilkan oleh PV. Berdasarkan hasil simulasi tersebut, MPPT yang kami rancang sudah sesuai dan siap diimplementasikan ke *hardware.* Karena sinyal POUT sudah mengikuti perubahan dari PIN dan menghasilkan daya maksimal.

Setelah dilakukannya simulasi dan menghasilkan keluaran yang sesuai dengan ketentuan, desain dapat diimplementasikan dalam bentuk *hardware*.

3.3 Komponen B1212

Rangkaian B1212 adalah rangkaian regulator yang digunakan untuk menyuplai tegangan ke driver. Seperti yang telah dijelaskan pada bab 2.5, TLP250 digunakan sebagai *gate driver* saklar daya yang memiliki sinyal keluaran yang terisolasi oleh rangkaian optocoupler. Namun, komponen ini tidak dapat berfungsi sendiri dan membutuhkan suplai tegangan menggunakan catu daya eksternal B1212. Berikut pada gambar 3.4 adalah komponen B1212.

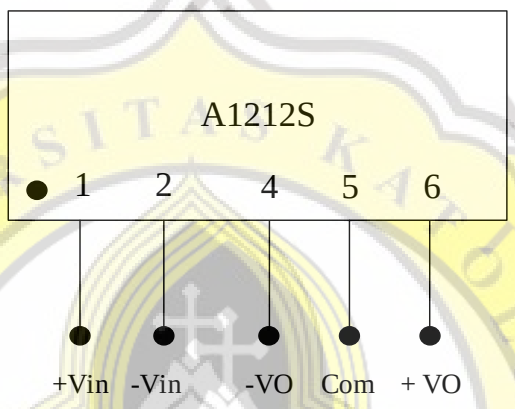


Gambar 3. 4 Gambar komponen dan fungsi pin B1212

Fungsi dari komponen B1212 adalah regulator dan menghasilkan tegangan 12 VDC murni. Pin 1 merupakan ground yang didapat dari catu daya, lalu pin 2 merupakan Vin yang mendapatkan tegangan positif catu daya tersebut. Perlu dicatat, tegangan pada catu daya harus diatur pada tegangan 12-15 VDC. Kemudian pada pin 3 dan pin 4 adalah bagian output, pin 3 adalah bagian ground dan pin 4 adalah +12 VDC.

3.4 Komponen A1212

Rangkaian A1212 adalah regulator tegangan DC mirip dengan rangkaian B1212. Bedanya, komponen A1212 memiliki 5 pin dan pada bagian outputnya memiliki 3 pin, yaitu; -V, COM, dan +V. berikut adalah gambar dari komponen A1212 beserta konfigurasi pinnya.

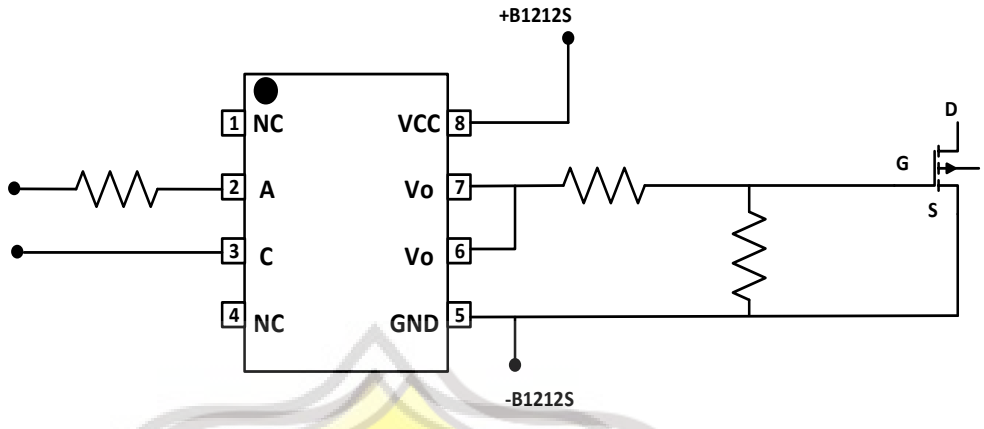


Gambar 3. 5 Gambar komponen dan fungsi pin A1212

Rangkaian A1212 ini digunakan untuk mensuplai tegangan pada sensor. Baik sensor tegangan maupun sensor arus. Pin 1 adalah pin input positif dan pin 2 adalah pin input negatif. Dan bagian output memiliki 3 pin dan ketiganya dihubungkan ke sensor arus sesuai dengan pin catu daya sensor arus.

3.5 Driver TLP250

Dan berikut adalah rangkaian TLP250 sebagai kendali saklar daya dapat dilihat pada gambar 3. 6 berikut ini.



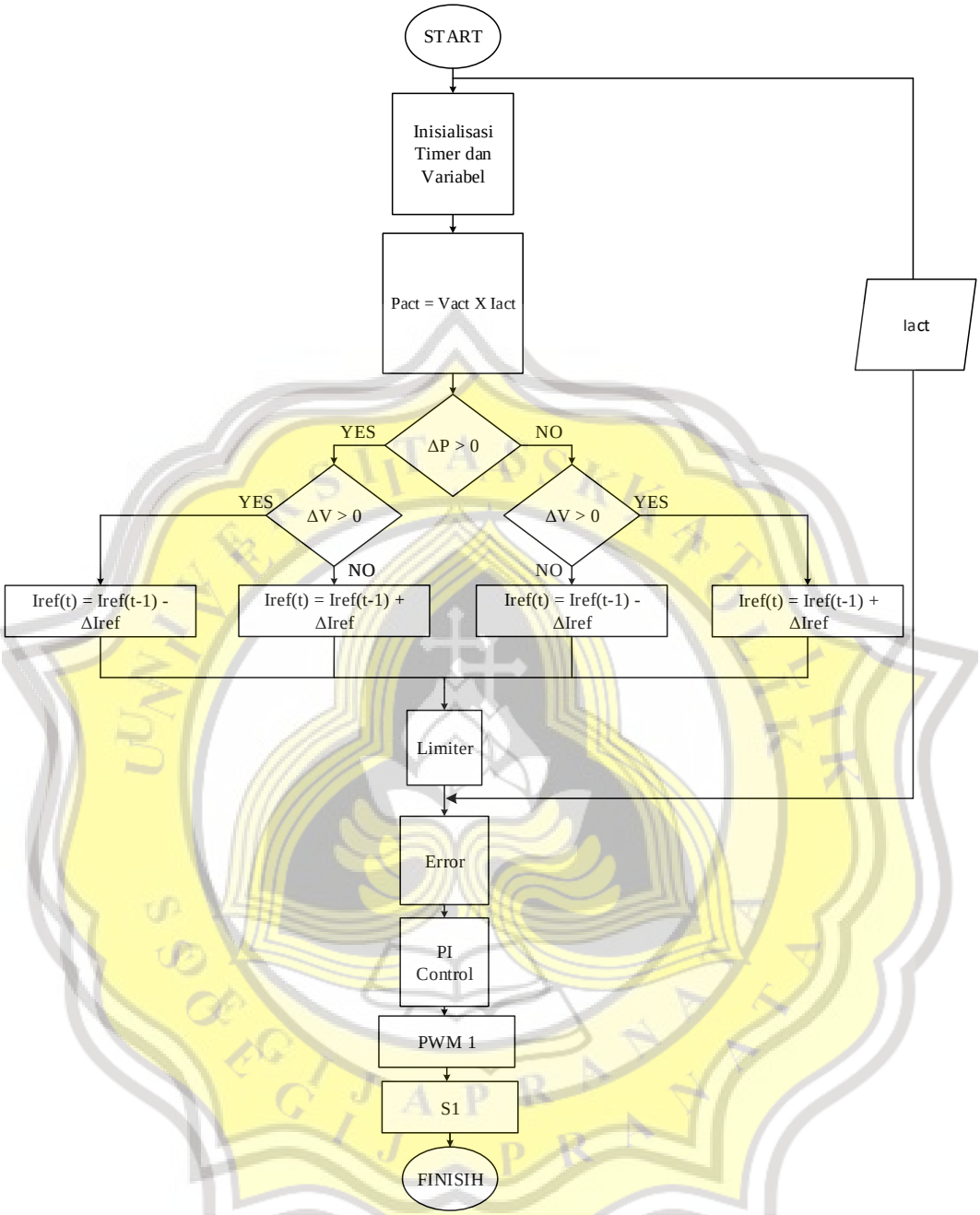
Gambar 3. 6 Rangkaian TLP250 sebagai d*river* MOSFET

TLP250 memiliki total pin sebanyak 8 buah, namun yang dapat difungsikan hanya 6 pin. Pin 2 merupakan pin yang nantinya dikoneksikan pada *port* I/O mikrokontroler, memiliki fungsi sebagai anoda LED yang ada di dalam TLP250. Dikarena pin 2 merupakan anoda dari LED diperlukan sebuah hambatan untuk menghambat arus yang masuk. Pin 3 merupakan katoda dari LED didalam TLP250 dikoneksikan menuju port GND pada mikrokontroler. Pin 8 merupakan sumber tegangan dari TLP250 yang dikoneksikan menuju pin positif B1212 selanjutnya akan mensuplai tegangan menuju pin *gate* pada MOSFET. Pin 5 merupakan GND yang terhubung pada pin negatif B1212 dan pin *source* pada MOSFET. Pin 6 dan 7 merupakan VO atau keluaran dari komponen TLP250 yang saling terhubung dan memiliki kesamaan pada fungsinya yang terkoneksi pada PIN *gate* sebagai pemberi sinyal pensaklaran pada MOSFET. Resistor pada rangkaian *driver* TLP250 ini berfungsi sebagai hambatan untuk mencegah arus berlebih yang masuk menuju pin *gate* MOSFET dan resistor vertikal

sebagai pengaman ketika MOSFET dalam kondisi off saat tidak ada sinyal pensaklaran yang masuk.

3.6 Alur Pemrograman

Alur pemrograman adalah suatu sistem kendali digital yang diaplikasikan pada suatu program untuk mengendalikan jalanya prosedur kerja alat agar sesuai dengan hasil yang diinginkan. Program akan diimplementasikan pada *hardware* dan akan diproses menggunakan mikrokontroler. Gambar 3.7 mendeskripsikan alur pemrograman dari MPPT dengan algoritma P&O yang dilengkapi kontrol PI pada SEPIC DC-DC konverter.



Gambar 3. 7 *Flowchart* program MPPT P&O dan kontrol PI pada SEPIC DC-DC konverter

Flowchart pada gambar 3.7 berbeda dengan flowchart yang ada pada bab 2.9, MPPT P&O pada alat kami telah dilengkapi dengan kontrol *Proportional Integral (PI)* dan strategi kontrol.

Mula – mula sistem pada mikrokontroler menyiapkan pin – pin, timer, serta fungsi - fungsi yang akan digunakan. Selanjutnya mikrokontroler membaca sinyal aktual tegangan dan sinyal aktual arus yang didapat dari sensor. Sinyal ini akan dikonversi menggunakan fungsi ADC yang tertanam pada mikrokontroler. Alur ini terus diulang setiap sampling data yang diambil. Dari data yang didapat dari sensor, akan didapat pula suatu daya aktual dengan mengalikan tegangan aktual dengan arus aktual. Setelah nilai – nilai tersebut diperoleh, maka nilai tersebut akan dibandingkan dengan data yang terbaru. Kemudian mikrokontroler akan menganalisis apakah data yang baru dan data yang lama terdapat perbedaan nilai. Perbedaan nilai ini merupakan arus referensi yang akan dibandingkan dengan arus aktual yang didapat oleh sensor dan menghasilkan suatu sinyal *error*. Sinyal ini akan diolah lebih lanjut di blok kontrol PI. Setelah semua proses ini selesai, terbentuk suatu PWM yang digunakan untuk mengendalikan saklar elektrik pada SEPIC konverter.

Program yang digunakan pada sistem ini menggunakan bahasa C/C++ dengan bantuan perangkat lunak Arduino IDE yang akan dijelaskan di bawah ini.

#include <STM32F4ADC.h>	Pemanggil header library
#define S1 PB0 double D, VA, VAT, VAC,	Deklarasi variabel data

```
VATpref, DeltaVA, PA, PAT, PAC, PATpref, DeltaPA, iref, Ierr_new, INT_new, INT_old, Ierr_old, Diref =0.1;

double IA=0.00;

int mod,n;

int count1,count2;

int car1,car2;

const int setOverflow = 3360;
```

Deklarasi variabel data

```
double kp          = 20;

double ki          = 100;

double p,i,pi;
```

Deklarasai konstanta kontroler PI

```
void setup()
```

Setup program

```
Serial.begin(9600);
```

Inisialisasi bitrate port serial

```
Timer8.init();
 Timer8.pause();
Timer8.setMasterMode(TIMER_MASTER_MODE_UPDATE);
 Timer8.setPeriod(500);
 Timer8.setMode(TIMER_CH2, TIMER_OUTPUT_COMPARE);
 Timer8.setCompare(TIMER_CH2, 1);
 Timer8.attachInterrupt(TIMER_CH2,PnO);
 Timer8.refresh();
 Timer8.resume();
```

Setelan timer 8 sebagai interrupt

```
Timer1.init();
 Timer1.pause();
 imer1.setMasterMode(TIMER_MASTER_MODE_UPDATE);
 Timer1.setPeriod(2000);
 Timer1.setMode(TIMER_CH2, TIMER_OUTPUT_COMPARE);
 Timer1.setCompare(TIMER_CH2, 1);
 Timer1.attachInterrupt(TIMER_CH2,Cont);
 Timer1.refresh();
 Timer1.resume();
```

Setelan timer 1 sebagai interrupt

```
Timer2.init();
 Timer2.pause();
 Timer2.setMasterMode(TIMER_MASTER_MODE_UPDATE);
 Timer2.setPeriod(1000);
 Timer2.setMode(TIMER_CH2, TIMER_OUTPUT_COMPARE);
 Timer2.setCompare(TIMER_CH2, 1);
 Timer2.attachInterrupt(TIMER_CH2,INT1);
 Timer2.refresh();
 Timer2.resume();
```

Setelan timer 2 sebagai interrupt

```
Timer3.init(); //PWM timer (PB0)
 Timer3.setPeriod(20);
 Timer3.refresh();
 Timer4.init(); //(PB6)
 Timer4.setPeriod(20);
 Timer4.refresh();
 Timer4.setCount(1680);
 Timer3.resume();
 Timer4.resume();
```

Setelan PWM pada timer 3 dan timer 4

```
pinMode(PB6,PWM);

pinMode(PA0, INPUT_ANALOG);

pinMode(PA1, INPUT_ANALOG);
```

Inisialisasi port PWM dan sensor

```
void INT1()

{

 Serial.print(VA);

 Serial.print("  ");

 Serial.print(IA);

 Serial.print("  ");

 Serial.println(iref);

 //Serial.print("  ");

 //Serial.println(mod);

 }
```

Setelan menampilkan variabel melalui komunikasi serial

```
void loop()

{

 count1  = Timer3.getCount();

 count2  = Timer4.getCount();

 car1    = count1;

 car2    = count2;

 pwmWrite(PB6, mod);
```

Fungsi program Loop agar proses akan selalu terulang

```
void PnO ()

{

VA    = analogRead(PA0)/53.783;

IA    = analogRead(PA1)/780.012;

PA =VA*IA;

n++;

PAC=PAC+PA;

VAC=VAC+VA;

if(n>=400)

  {

  n=0;

  PAT=PAC;

  PAC=0;

  VAT=VAC;

  VAC=0;
```

Program MPPT P&O

Membaca data dari sensor dan perhitungan daya

```
DeltaVA=VAT-VATpref;
DeltaPA=PAT-PATpref;
if (DeltaPA>0){
  if (DeltaVA>0){
  D=D-Diref;}
  else if(DeltaVA<0){
  D=D+Diref;}
}
else if(DeltaPA<0){
  if (DeltaVA<0){
  D=D-Diref;   }
  else if(DeltaVA>0){
  D=D+Diref;  }
```

Program MPPT P&O

Perbandingan daya dari data lama dengan data terbaru

```
if(D>1.2){
D=1.2; }
else if(D<0.1){
D=0.1;//0.1 }
VATpref = VAT;
PATpref= PAT;
```

Program MPPT P&O

Hasil dari perhitungan dan menggantikan data lama ke data baru yang natinya akan dibandingkan dengan data terbaru

```
void Cont ()

{

iref = D;

Ierr_new= iref-IA;//error

//Proportional Integral control

INT_new =(Ierr_new + Ierr_old)+ INT_old;

mod = ki*INT_new + kp*Ierr_new;

Ierr_old = Ierr_new;

if ( mod >0 && mod<3360)

{

  INT_old = INT_new;

}
```

Program MPPT P&O

Perhitungan akhir dengan kontrol PI dan menghasilkan sinyal PWM