

BAB III

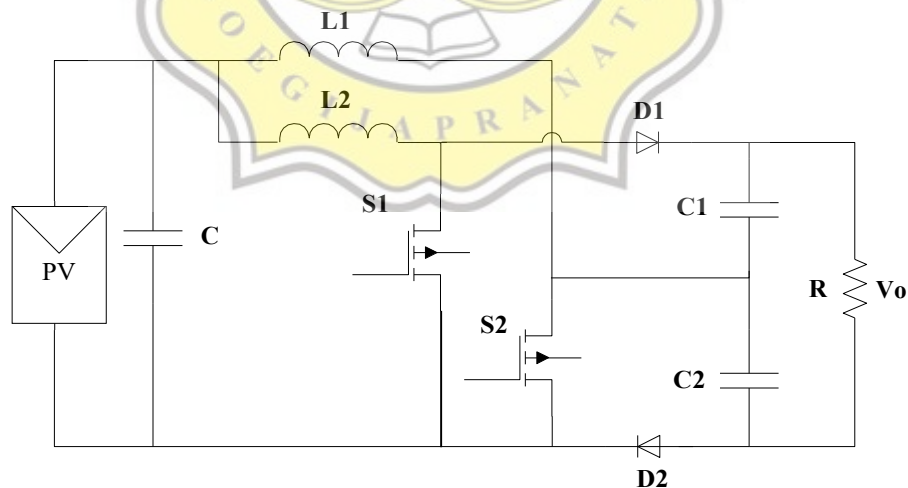
RANCANGAN DAN MODE OPERASI

3.1. Pendahuluan

Pada bab ini dibahas mengenai topologi, mode operasi, algoritma program dari *MPPT Boost Voltage Doubler*.

3.2. Topologi *MPPT Boost Voltage Doubler*

Topologi rangkaian *MPPT Boost Voltage Doubler* yang terlihat pada Gambar 3.1. merupakan penggabungan dua buah *DC-DC boost converter* konvensional. Rangkaian tersebut terdiri dari dua buah induktor, dua buah kapasitor, dua buah saklar aktif, dua buah dioda, satu buah beban resistor, dan satu buah sumber tegangan berupa PV.



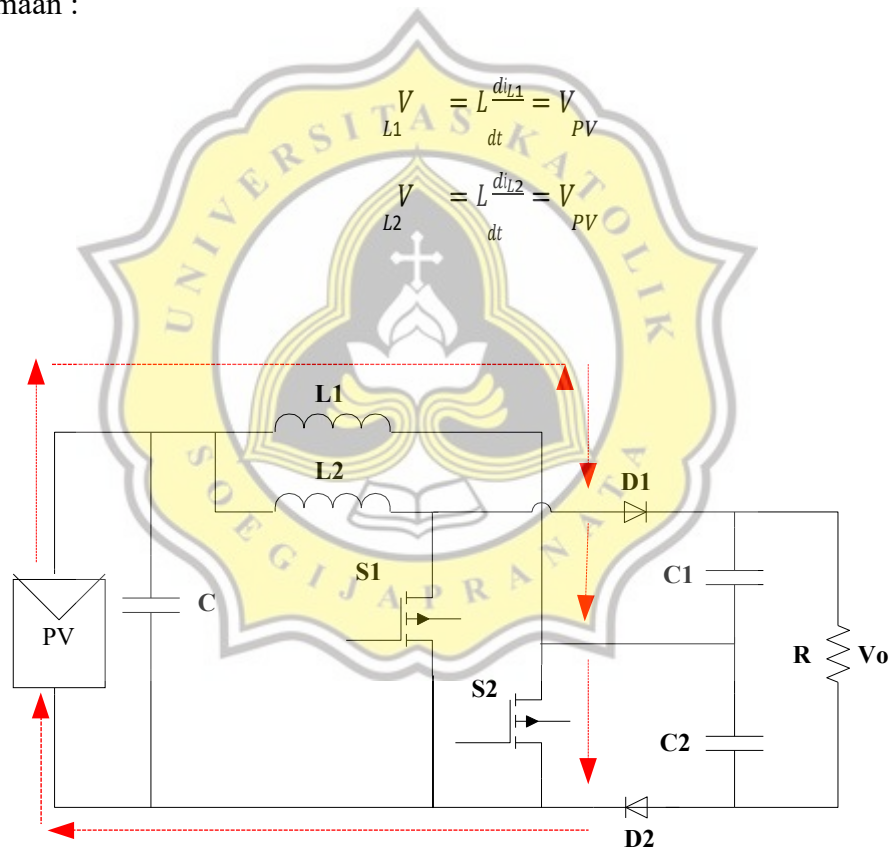
Gambar 3.1. Topologi Rangkaian MPPT Boost Voltage Doubler

3.3. Mode Operasi

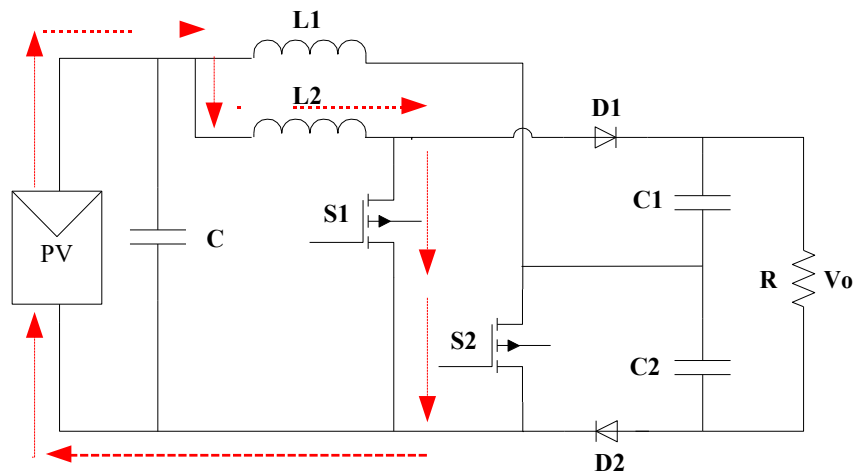
Berdasarkan topologi rangkaian yang sudah dibahas sebelumnya, rangkaian MPPT *Boost Voltage Doubler* ini mempunyai empat mode operasi. Pada mode operasi 1 yang terlihat pada Gambar 3.2. ini terjadi saat kedua saklar terkonduksi, sehingga arus mengalir dari sumber PV menuju induktor L1 dan L2, dan kembali ke sumber. Pada tegangan induktor V_{L1} dan V_{L2} dapat dinyatakan dengan persamaan :

$$V_{L1} = L \frac{di_{L1}}{dt} = V_{PV} \quad (3.1)$$

$$V_{L2} = L \frac{di_{L2}}{dt} = V_{PV} \quad (3.2)$$



(a) Saat S2 Konduksi

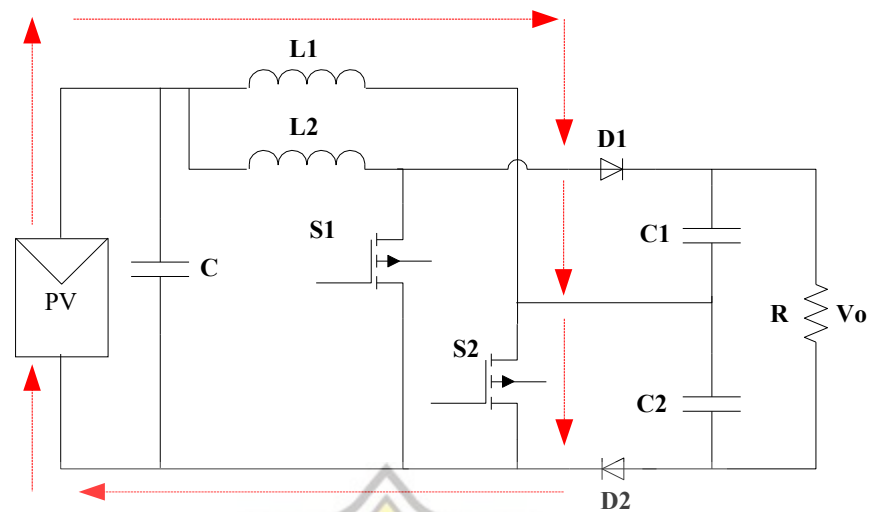


(b) Saat S1 Konduksi
Gambar 3.2. Mode Operasi 1

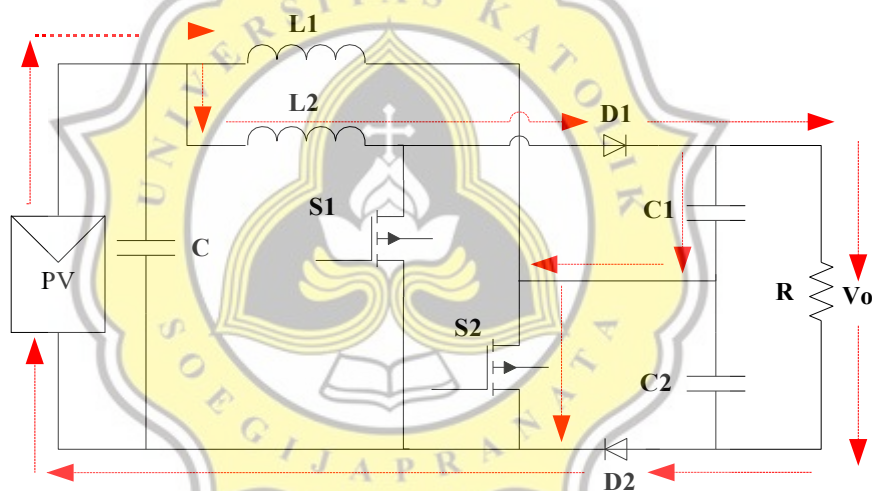
Pada mode operasi 2 yang dapat dilihat pada Gambar 3.3. hanya S2 yang terkonduksi, sehingga mengakibatkan arus mengalir menuju induktor L1 dan kembali ke sumber. Sedangkan S1 tidak terkonduksi sehingga arus mengalir dari sumber PV ke induktor L2 melalui D1 menuju ke beban R, lalu kembali ke sumber melalui D2. Pada tegangan induktor V_{L1} dan V_{L2} dapat dinyatakan dengan persamaan :

$$V_{L1} = L \frac{di_{L1}}{dt} = V_{PV} \quad (3.3)$$

$$V_{L2} = L \frac{di_{L2}}{dt} = V_{PV} - V_{C1} \quad (3.4)$$



(a) Saat S2 Konduksi

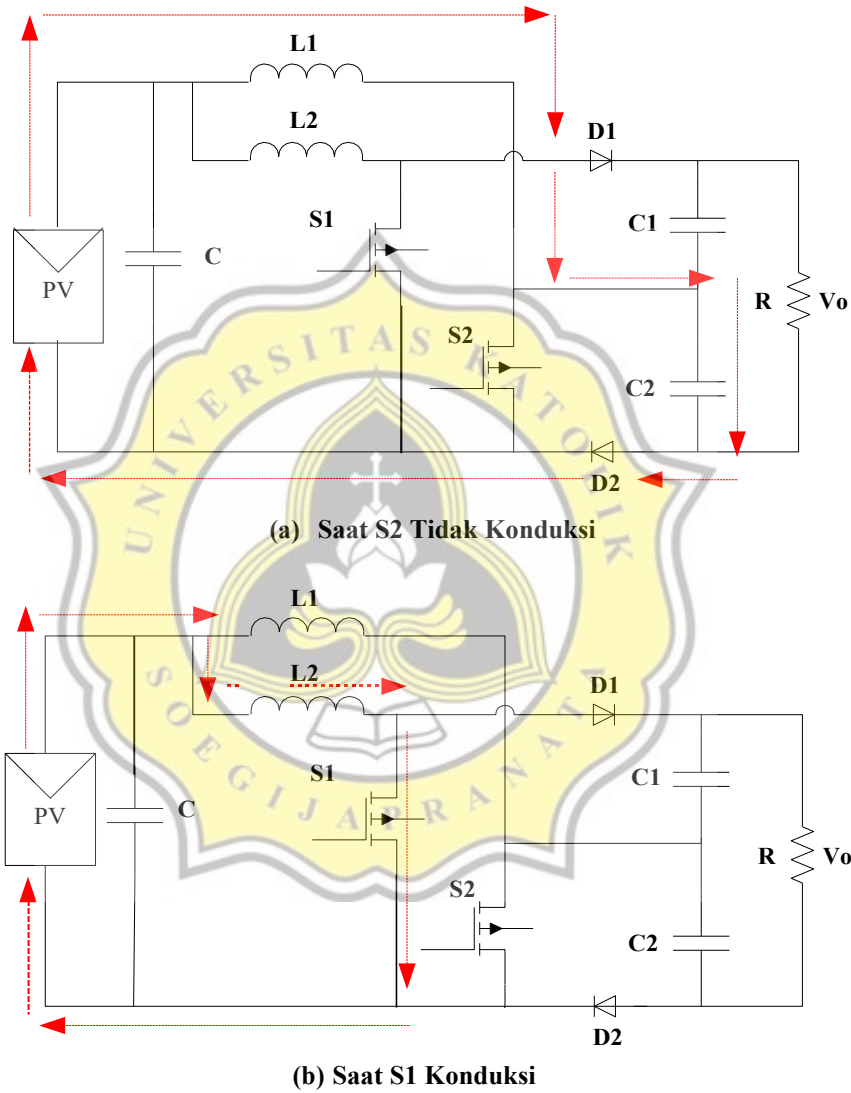


(b) Saat S1 Tidak Konduksi
Gambar 3.3. Mode Operasi 2

Pada mode operasi 3 yang terlihat pada Gambar 3.4. merupakan kebalikan dari mode operasi 2. Kondisi saat S1 terkonduksi menyebabkan arus mengalir melalui induktor L2 lalu kembali ke sumber. Sedangkan saat saklar S2 tidak terkonduksi, arus mengalir dari sumber ke induktor L1 menuju ke beban R lalu kembali ke sumber melalui dioda D2. Pada tegangan induktor V_{L1} dan V_{L2} dapat dinyatakan dengan persamaan :

$$V_{L1} = L \frac{di_{L1}}{dt} = V_{PV} - V_{C2} \quad (3.5)$$

$$V_{L2} = L \frac{di_{L2}}{dt} = V_{PV} \quad (3.6)$$



Gambar 3.4. Mode Operasi 3

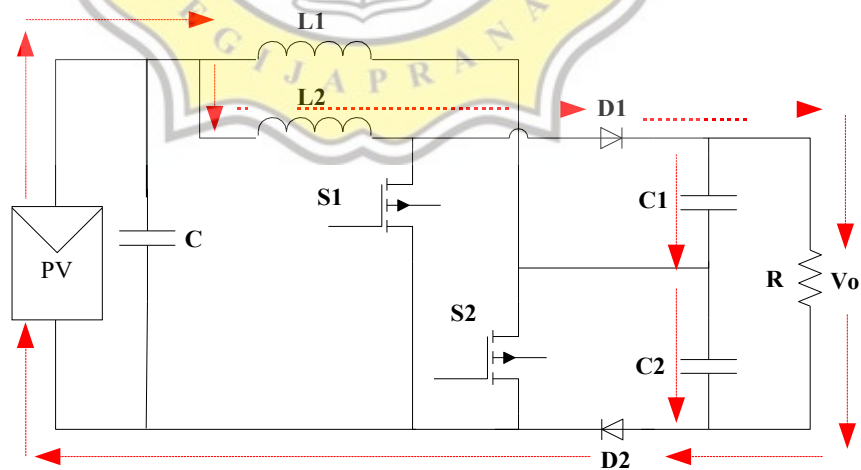
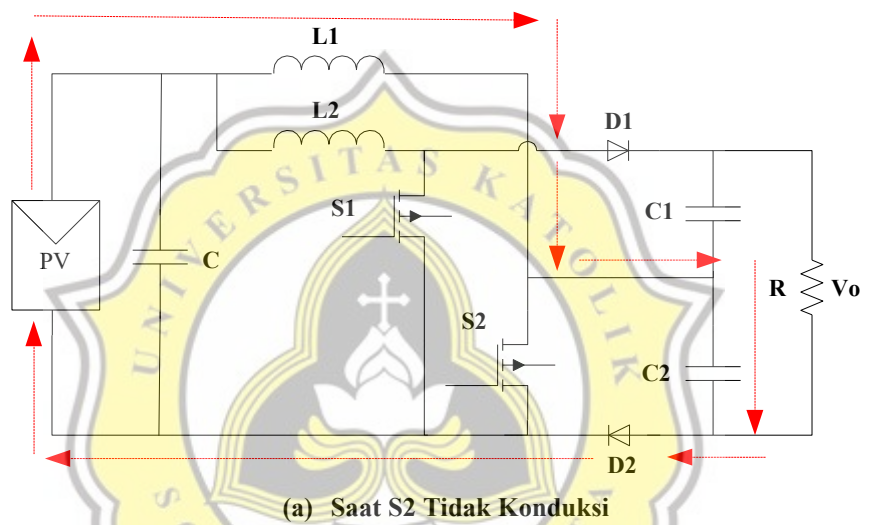
Pada mode operasi 4 yang terlihat pada Gambar 3.5. terjadi saat kedua saklar tidak terkonduksi. Hal ini menyebabkan arus mengalir dari sumber ke induktor L1 menuju beban R, dan kembali ke sumber melalui dioda D2. Arus juga mengalir dari sumber ke induktor L2 melalui dioda D1 dan menuju ke beban R, lalu kembali ke

sumber melalui dioda D2. Pada tegangan induktor V_{L1} dan V_{L2} dapat dinyatakan

dengan persamaan :

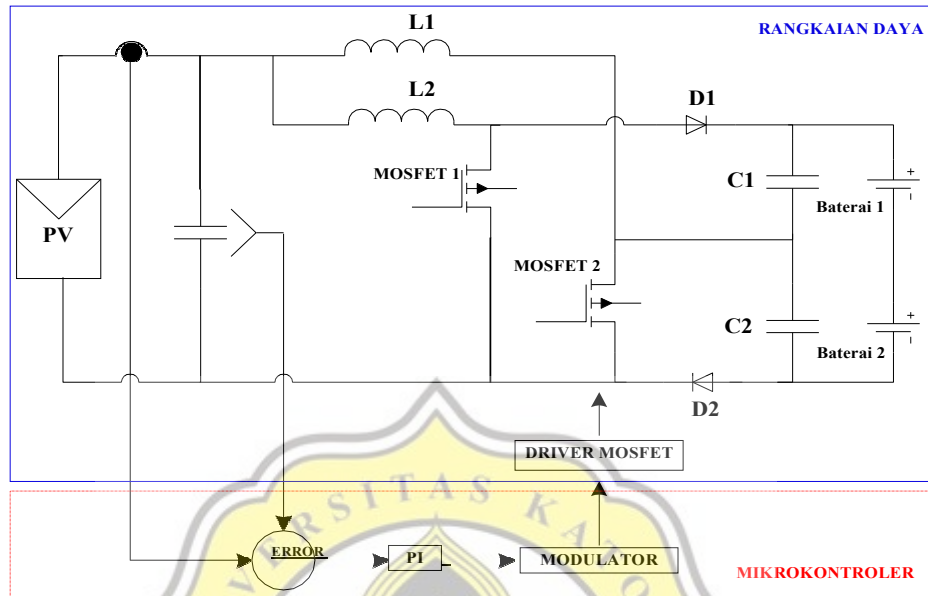
$$V_{L1} = L \frac{di_{L1}}{dt} = V_{PV} - V_{C2} \quad (3.7)$$

$$V_{L2} = L \frac{di_{L2}}{dt} = V_{PV} - V_{C1} \quad (3.8)$$



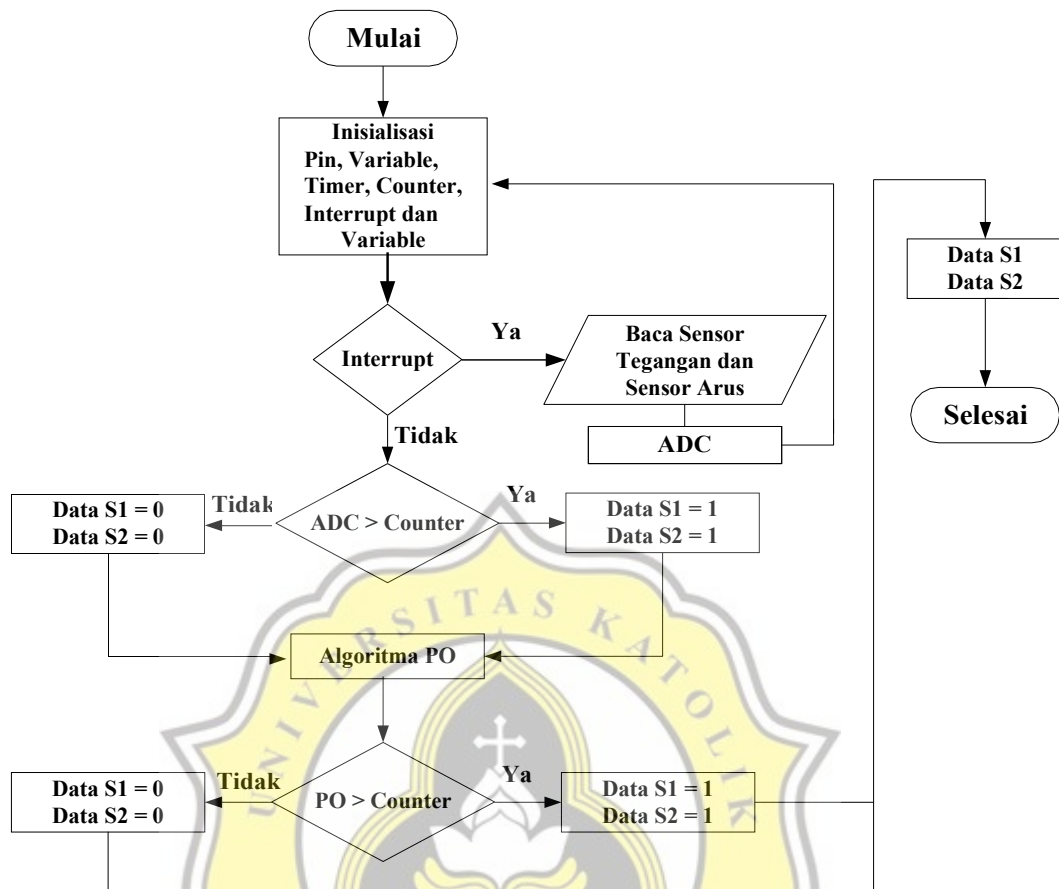
Gambar 3.5. Mode Operasi 4

3.4. Mikrokontroler dan Algoritma Program



Gambar 3.6. Blok Diagram Sistem Kendali MPPT Boost Voltage Doubler

Berdasarkan Gambar 3.6. yang merupakan diagram alir pemrograman, dapat dilakukan suatu pemrograman menggunakan perangkat lunak *Arduino IDE* untuk aplikasi *MPPT Boost Voltage Doubler*. *STM32F1308CT* yang terlihat pada Gambar 3.7. digunakan sebagai mikrokontroler untuk mengimplementasi program tersebut.



Gambar 3.7. Diagram Alir Program MPPT Boost Doubler Voltage

Di dalam algoritma MPPT, keluaran sensor arus dikalikan dengan keluaran sensor tegangan untuk mendapatkan hasil keluaran daya. Kemudian hasil keluaran daya tersebut harus dibandingkan dengan *counter*, dimana *counter* yang dimaksud diasumsikan sebagai operator. Implementasi dari diagram alir yang terlihat pada Gambar 3.6. dituliskan sebagai program seperti berikut :

```

#define Teg PA0
#define Arus PA1
#define PWM1 PB12
#define PWM2 PB13
#define LED PC13
  
```

} Inialisasi Port


```
const int PS = 512;
const int counter = 4000;
int car, car2;
const int kp = 30;
const float ki = 0.01;
const float kd = 1000;
const float cteg = 0.016;
const float cars = 0.019;
```

Inisialisasi *Variable*

```
float p, i, pi, da, da1, db, db1, d1, d2, A;
double daya, NilaiArus, NilaiTeg;
```

```
void setup() {
  Timer1.pause();
  Timer1.setMode(TIMER_CH1, TIMER_OUTPUTCOMPARE);
  Timer1.setCount(0);
  Timer1.setOverflow(counter);
  Timer1.setCompare(TIMER_CH1, 4095);
  Timer4.setPrescaleFactor(PS);
  Timer1.attachInterrupt(TIMER_CH1, LALAT);
  Timer1.refresh();
  Timer1.resume();
```

Inisialisasi
Timer

```
pinMode(Teg, INPUT_ANALOG);
pinMode(Arus, INPUT_ANALOG);
pinMode(PWM1, OUTPUT);
pinMode(PWM2, OUTPUT);
pinMode(LED, OUTPUT);
Serial.begin(9600);
```

Inisialisasi ADC

```
if (Timer1.getCount() >= 2000) {
  car = 4000 - Timer1.getCount();
}
if (Timer1.getCount() <= 2000) {
  car = Timer1.getCount();
}
car2 = map(car, 0, 2000, 2000, 0);
```

Komparasi ADC
dan *Counter*

```
if (car2 < 200) {
  digitalWrite(PWM1, 1);
}
else {
  digitalWrite(PWM1, 0);
}
```

```

if (car < 200) {
    digitalWrite(PWM2, 1);
}
else {
    digitalWrite(PWM2, 0);
}
}

```

Komparasi ADC
dan Counter

```

void loop() {
    if (Timer1.getCount() >= 2000) {
        car = 4000 - Timer1.getCount();
    }
    if (Timer1.getCount() <= 2000) {
        car = Timer1.getCount();
    }
    car2 = map(car, 0, 2000, 2000, 0);

```

HitungMPPT(NilaiTeg, NilaiArus);

```

if (car < pi) {
    digitalWrite(PWM1, 1);
}
else {
    digitalWrite(PWM1, 0);
}

```

Komparasi PO
dan Counter

```

if (car2 < pi) {
    digitalWrite(PWM2, 1);
}
else {
    digitalWrite(PWM2, 0);
}
}

```

```

void LALAT(void) {
    digitalWrite(PC13, !digitalRead(PC13));
    NilaiTeg = map(analogRead(Teg), 0, 4095, 0, 2000);
    NilaiTeg = NilaiTeg * cteg;
    NilaiArus = map(analogRead(Arus), 0, 4095, 0, 2000);
    NilaiArus = NilaiArus * cars;
}

```

Pembacaan
ADC

```

void HitungMPPT(float a, float b) {
    daya = a * b;

    da = daya;
    d1 = (da - da1) * kd;
    da1 = da;

    db = a;
    d2 = (db - db1) * kd;
    db1 = db;

    A = d1 / d2;

    p = kp * A;
    i += ki * A;

    pi = (p * (1 + 1)) / i;

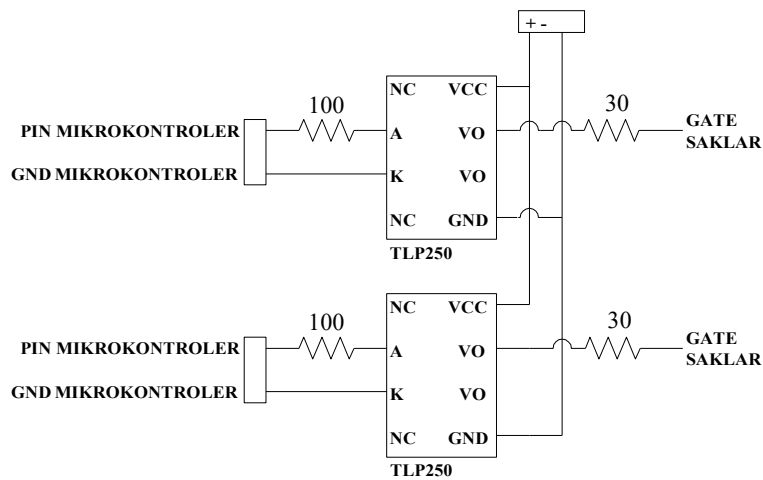
    Serial.print(NilaiArus);
    Serial.print(" , ");
    Serial.print(NilaiTeg);
    Serial.print(" , ");
    Serial.print(A);
    Serial.print(" , ");
    Serial.println(pi);
}

```

Algoritma PO

3.5. Rangkaian *Driver Mosfet Boost Voltage Doubler*

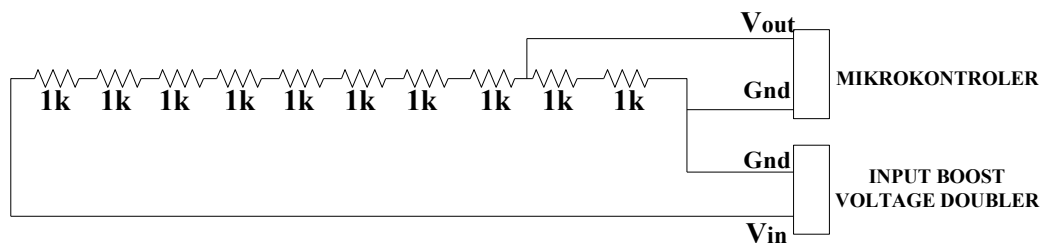
Pada MPPT *Boost Voltage Doubler* menggunakan saklar berupa mosfet IRFP250. Untuk mengendalikan saklar yang bekerja diperlukan rangkaian *driver*. Pada alat ini menggunakan TLP250 sebagai *driver* dengan skematik rangkaian yang diterapkan seperti yang terlihat pada Gambar 3.8.



Gambar 3.8. Rangkaian Driver

3.6. Rangkaian Sensor Tegangan

Pada Gambar 3.9. diperlihatkan gambar rangkaian *voltage divider* atau pembagi tegangan yang berfungsi sensor tegangan. Sensor tegangan berfungsi mengukur tegangan masuk dari PV. Tegangan yang terukur pada sensor akan digunakan sebagai tegangan masuk pada mikrokontroler. Tetapi tegangan maksimal pada mikrokontroler hanya 3,3V, sedangkan tegangan yang terukur lebih besar dari tegangan maksimal. Untuk mengatasi hal tersebut dibutuhkan rangkaian op-amp yang menggunakan IC LF347N.



Gambar 3.9. Rangkaian Sensor Tegangan

Penentuan perbandingan sensor tegangan dapat dihitung menggunakan persamaan (2.4) berdasarkan rangkaian pada Gambar 3.9.

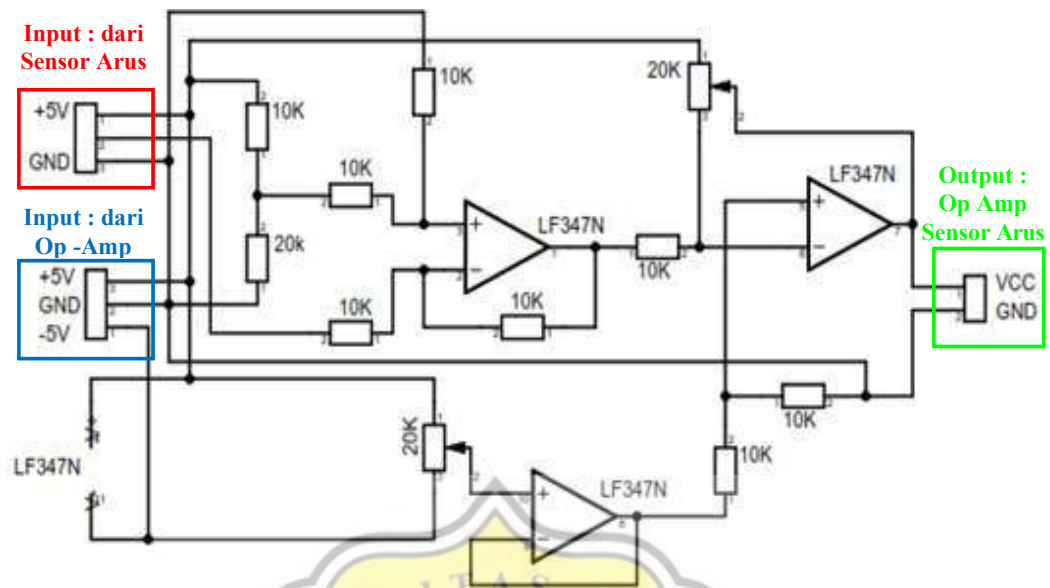
$$V_{out} = \frac{R1 + R2}{R1 + R2 + R3 + R4 + R5 + R6 + R7 + R8 + R9 + R10} \times V_{in} \times 2k$$

$$V_{out} = \frac{10k}{10k} \times 17.2$$

Berdasarkan perhitungan di atas, V_{out} yang diperlukan dapat dicari dengan hasil perbandingan dikalikan tegangan masukan yang merupakan tegangan dari PV.

3.7. Rangkaian Sensor Arus ACS712

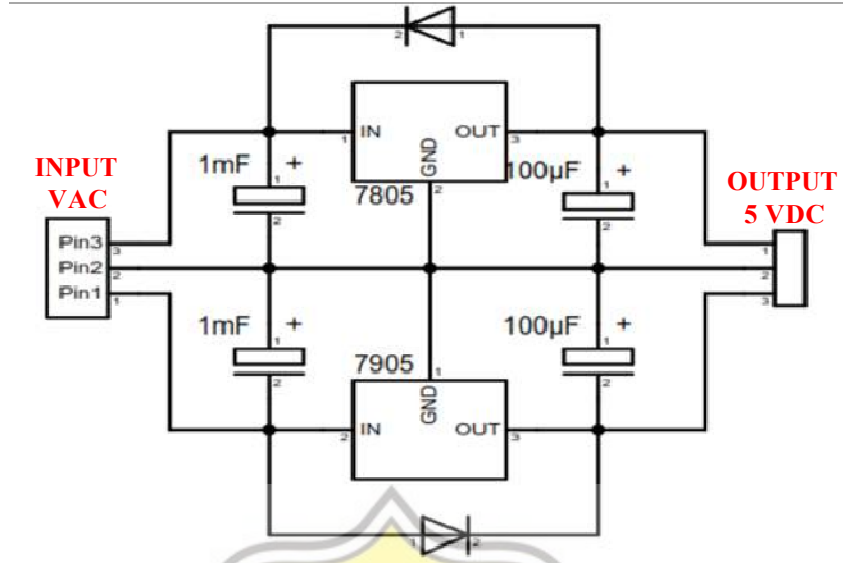
Sensor arus yang digunakan pada implementasi perangkat keras MPPT *Boost Voltage Doubler* adalah sensor arus ACS712. Prinsip kerja sensor ini berdasarkan prinsip *hall effect*, dimana arus yang mengalir menghasilkan medan magnet yang dideteksi oleh IC hall dan diubah menjadi tegangan proporsional. Pada implementasi perangkat keras ini, sensor ACS712 dirangkai dengan cara IP positif akan dihubungkan ke input positif dari *Boost Voltage Doubler* dan IP negatif dihubungkan dengan *ground*. Sedangkan keluarannya yang berupa tegangan VCC dihubungkan ke catu daya positif, VOUT dihubungkan ke catu daya negatif, terakhir filter dan ground dihubungkan ke *ground* catu daya seperti yang terlihat pada Gambar 3.10.



Gambar 3.10. Rangkaian Sensor Arus

3.8. Rangkaian Catu Daya

Pada Gambar 3.11. ditunjukkan rangkaian catu daya yang dipakai. Rangkaian catu daya ini berfungsi untuk mengaktifkan sensor tegangan dan arus, op – amp, dan mikrokontrolernya. Rangkaian ini merupakan titik integrasi, dimana memberikan sumber tegangan DC bernilai +12V, -12V, dan 5V



Gambar 3.11. Rangkaian Catu Daya

