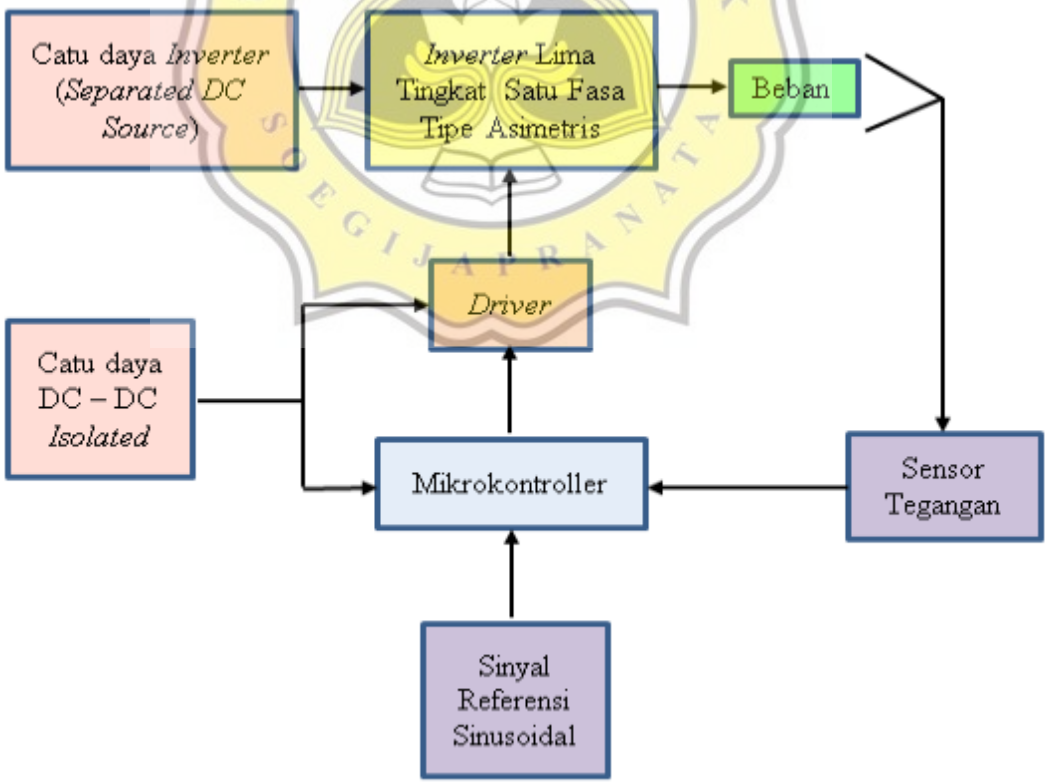
BAB III

PERANCANGAN ALAT

3.1 Pendahuluan

Bab ini menjabarkan langkah – langkah dalam merancang pengendalian tegangan pada *inverter* lima tingkat satu fasa tipe asimetris menggunakan mikrokontroler Arduino Due. Pertama *inverter* lima tingkat terlebih dahulu disimulasikan secara rangkaian terbuka (*open loop*) dengan menggunakan *Power Simulator*. Hasil simulasi dianalisa agar mendapatkan lima tingkatan tegangan sesuai yang diinginkan. Untuk menghasilkan lima tingkatan tegangan, dilakukan modulasi lebar pulsa antara sinyal pembawa segitiga dengan sinyal referensi *sinusoidal*. Metode modulasi lebar pulsa tersebut dikenal dengan sebutan SPWM. Setelah mensimulasikan secara rangkaian terbuka (*open loop*), lalu diterapkan sistem rangkaian tertutup (*closed loop*) pada *inverter* satu fasa lima tingkat tipe asimetris. Tujuan simulasi secara *closed loop* agar tegangan keluaran dari *inverter* satu fasa lima tingkat tipe asimetris dapat dikendalikan sesuai keinginan. Sistem *closed loop* akan membuat tegangan aktual dari *inverter* satu fasa lima tingkat tipe asimetris akan dapat mengikuti tegangan referensi yang diinjeksi pada mikrokontroler sebagai syarat dari catu daya mandiri. Syarat lain dari catu daya mandiri adalah tegangan keluaran dari *inverter* satu fasa lima tingkat tipe asimetris harus memenuhi standar THD IEEE 519. Hasil simulasi secara *open loop* dan *closed loop* pada *inverter* satu fasa lima tingkat tipe asimetris akan ditampilkan dan dianalisa pada BAB IV.

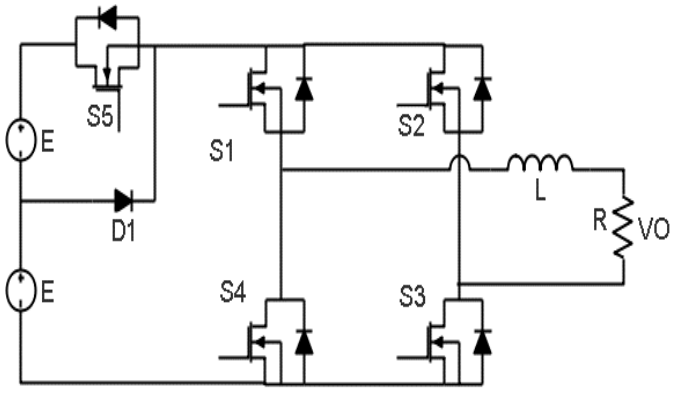
Proses selanjutnya yaitu merancang catu daya mandiri berdasarkan simulasi *inverter* satu fasa lima tingkat ripe asimetris yang telah dilaksanakan. Proses perancangan catu daya mandiri meliputi beberapa bagian. Pada bagian pertama membahas rancangan rangkaian daya dari *inverter* satu fasa lima tingkat tipe asimetris beserta mode operasi yang diterapkan. Bagian kedua yaitu merancang rangkaian kontrol serta strategi kontrol pada catu daya mandiri. Bagian ketiga yaitu rangkaian sensor tegangan untuk membaca tegangan aktual dari keluaran *inverter* lima tingkat satu fasa tipe asimetris. Perancangan alat secara garis besar disajikan dalam diagram blok yang ditunjukan pada Gambar-3.1.



Gambar-3.1 Diagram Blok Perancangan Alat

3.2 Perancangan rangkaian daya dan mode operasi

Pada bagian pertama ini akan membahas perancangan topologi *inverter* satu fasa lima tingkat tipe asimetris beserta mode operasinya. Topologi *inverter* satu fasa lima tingkat tipe asimetris memerlukan dua buah sumber DC (E) secara terpisah atau yang dikenal dengan sebutan *separated DC source*. Topologi asimetris ini memiliki keunggulan, hanya menggunakan lima buah saklar daya yaitu (S1), (S2), (S3), (S4), dan (S5) ditunjukkan dalam Gambar-3.2 [23]. Dioda (D1) pada *inverter* lima tingkat tipe asimetris termasuk dalam kategori diode cepat dan dioperasikan dalam bias maju. Fungsi diode cepat pada (DI) dapat dibuat dari saklar daya MOSFET IRFP260 yang difungsikan sebagai dioda dengan cara menggabungkan kaki *gate* dengan kaki *source*. Pada akhir keluaran dari *inverter* satu fasa lima tingkat tipe asimetris diberikan tapis induktor sebelum menuju ke beban. Tapis induktor yang berupa lilitan kawat *email* yang dililitkan pada inti besi (*ferrite*) ini bertujuan mengurangi *ripple* pada tegangan keluaran sehingga tegangan keluaran dapat memenuhi standar THD IEEE 519 sebagai syarat catu daya mandiri.



Gambar-3.2 Topologi *inverter* satu fasa lima tingkat tipe asimetris

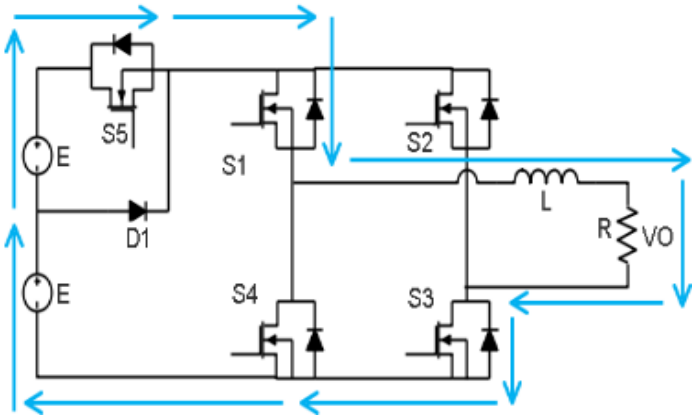
Inverter satu fasa lima tingkat tipe asimetris memiliki lima buah mode operasi [24]. Mode operasi pertama, saklar yang terkonduksi arus adalah (S5), (S1), dan (S3). Mode operasi kedua, saklar yang terinduksi arus adalah (S1) dan (S3). Mode operasi ketiga merupakan kondisi freewheeling. Mode operasi keempat, saklar yang terkonduksi arus adalah (S2) dan (S4). Mode operasi kelima, saklar yang terkonduksi arus adalah (S5), (S2), dan (S4).

3.2.1 Mode Operasi 1

Ketika saklar (S5) dan (S1) terkonduksi. Arus akan mengalir dari sumber tegangan DC (-E-E) menuju ke beban. Sedangkan, saat saklar (S3) terkonduksi, arus akan mengalir kembali menuju ke sumber tegangan DC (-E-E). Ilustrasi pada mode operasi 1 ditunjukkan pada Gambar-3.3. Hal ini diwakili dalam persamaan 3.1.

$$\mathbf{2E = V_{(L)} + V_{(O)}}$$

$$L\Delta i_{(L)} = \left(2E - V_{(O)}\right)\Delta t = \left(2E - V_{(O)}\right)t_{(on)} \qquad (3.1)$$



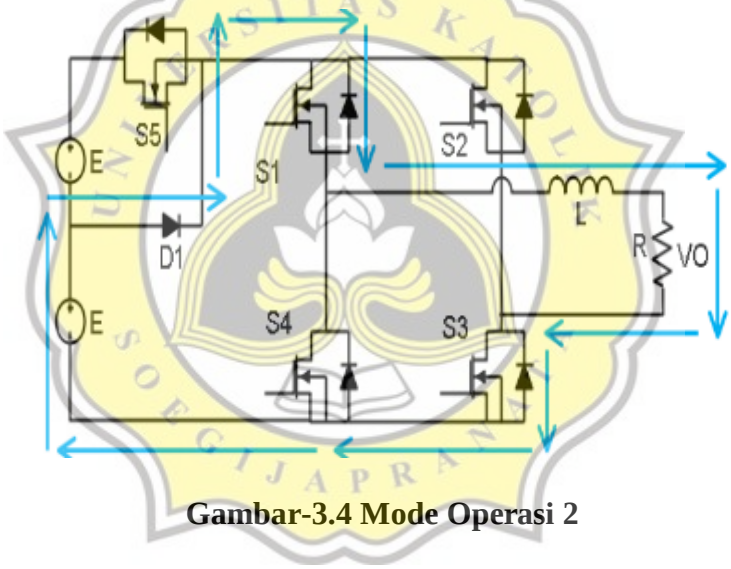
Gambar-3.3 Mode Operasi 1

3.2.2 Mode Operasi 2

Ketika saklar (S1) terkonduksi dan dioda (D1) telah dialiri arus. Arus akan mengalir dari sumber tegangan DC (E) menuju ke beban. Melalui saklar (S3), arus akan kembali mengalir, kembali ke sumber tegangan DC (E). Ilustrasi pada mode operasi 2 ditunjukkan pada Gambar-3.4. Hal Ini akan diwakili dalam persamaan 3.2.

$$E = V_{(L)} + V_{(O)}$$

$$L\Delta i_{(L)} = (E - V_{(O)})\Delta t = (E - V_{(O)})t_{(on)} \quad (3.2)$$



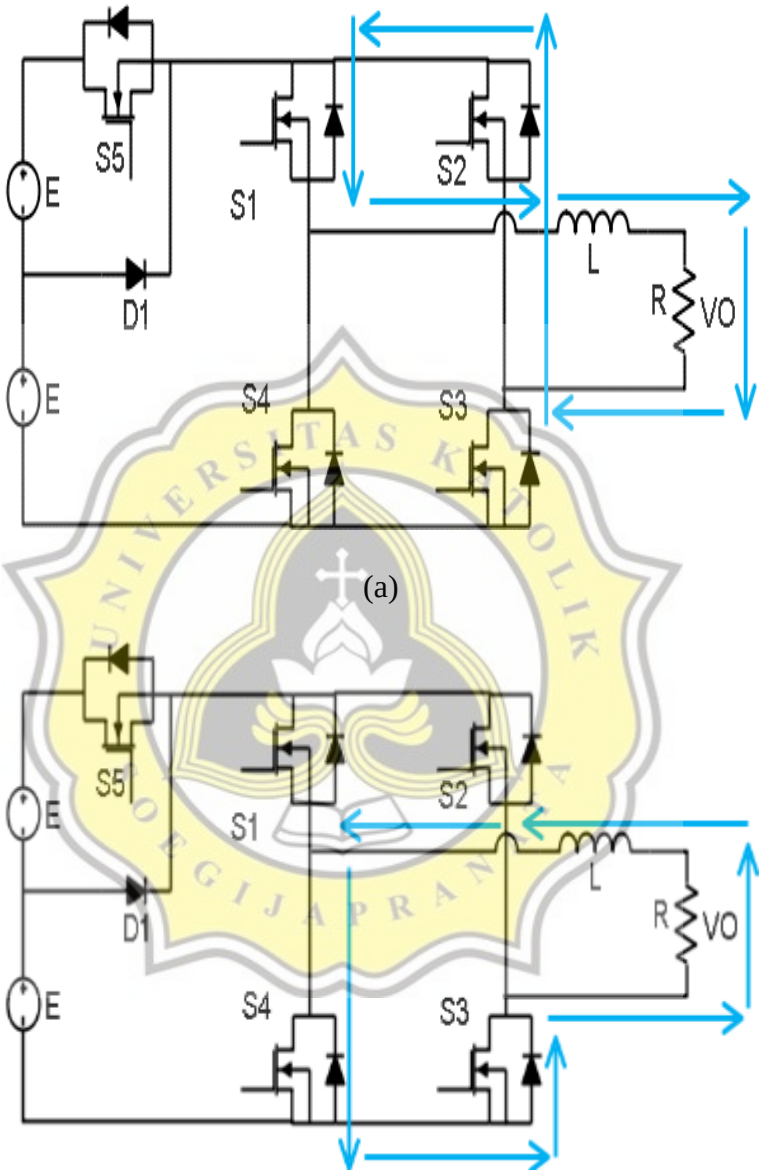
Gambar-3.4 Mode Operasi 2

3.2.3 Mode Operasi 3

Mode operasi 3 merupakan kondisi freewheeling. Kondisi freewheeling untuk nilai yang positif terjadi setelah saklar (S1) dan (S3) telah dialiri arus. Sementara kondisi freewheeling untuk nilai negatif terjadi saat saklar aktif (S2) dan (S4) dialiri arus. Ilustrasi mode operasi 3 ditunjukkan pada Gambar-3.5. Hal Ini diwakili dalam persamaan 3.3.

$$0 = V_{(L)} + V_{(O)}$$

$$L\Delta i_{(L)} = (V_{(O)})\Delta t = (V_{(O)})t_{(off)} \tag{3.3}$$



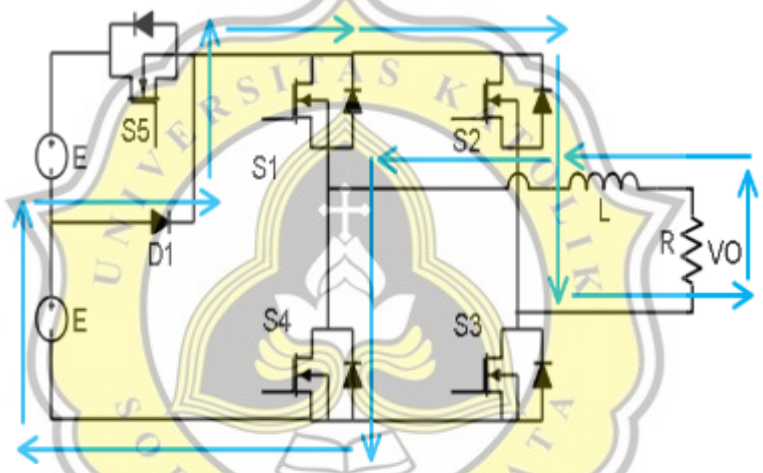
Gambar-3.5 Mode Operasi 3. (a) Siklus Positif. (b) Siklus Negatif

3.2.4 Mode Operasi 4

Ketika saklar (S2) dan dioda (D1) dialiri arus. Maka arus akan mengalir dari sumber tegangan DC (E) menuju ke beban. Melalui saklar (S4), arus mengalir kembali menuju sumber tegangan DC (E). Ilustrasi mode operasi 4 ditunjukkan pada Gambar 3.6. Ini diwakili dalam persamaan 3.4.

$$-\mathbf{E} = \boldsymbol{V}_{(\boldsymbol{L})} + \boldsymbol{V}_{(\boldsymbol{O})}$$

$$\mathrm{L}\Delta\mathrm{i}_{(\mathrm{L})} = \left(\mathrm{V}_{(\mathrm{O})} - \mathrm{E}\right)\Delta\mathrm{t} = \left(\mathrm{V}_{(\mathrm{O})} - \mathrm{E}\right)\mathrm{t}_{(\mathrm{on})} \qquad (3.4)$$



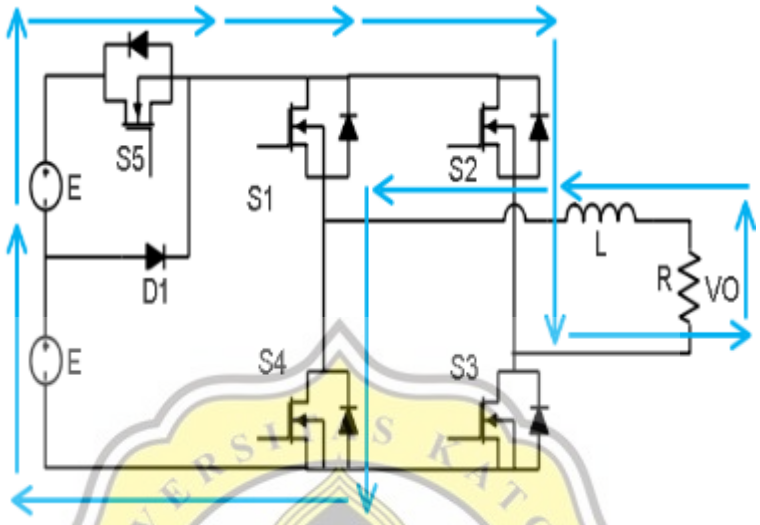
Gambar-3.6 Mode Operasi 4

3.2.5 Mode Operasi 5

Mode operasi 5 merupakan mode operasi terakhir dari *inverter* satu fasa lima tingkat tipe asimetris. Ketika saklar daya (S5) dan (S2) terkonduksi oleh arus. Arus akan mengalir dari sumber tegangan DC (E + E) menuju ke beban. Melalui saklar daya (S4), arus akan mengalir kembali menuju sumber tegangan DC (E + E). Ilustrasi mode operasi 5 ditunjukkan pada Gambar-3.7. Hal ini akan diwakili dalam persamaan 3.5.

$$-(\mathbf{E}-\mathbf{E}) = \boldsymbol{V}_{(\boldsymbol{L})} + \boldsymbol{V}_{(\boldsymbol{O})}$$

$$\mathrm{L}\Delta\mathrm{i}_{(\mathrm{L})} = \left(\mathrm{V}_{(\mathrm{O})} - 2\mathrm{E}\right)\Delta\mathrm{t} = \left(\mathrm{V}_{(\mathrm{O})} - 2\mathrm{E}\right)\mathrm{t}_{(\mathrm{on})} \qquad (3.5)$$



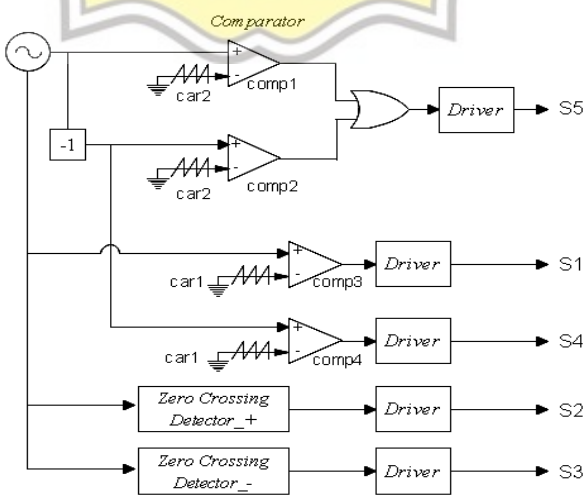
Gambar-3.7 Mode Operasi 5

Berdasarkan mode operasi satu hingga mode operasi lima dari *inverter* satu fasa lima tingkat tipe asimetris, maka diperoleh pensaklaran yang diperlihatkan pada Tabel-3.1. Tabel tersebut menyajikan pergantian *on* dan *off* untuk setiap saklar daya.

Tabel-3.1 Tabel Pensaklaran

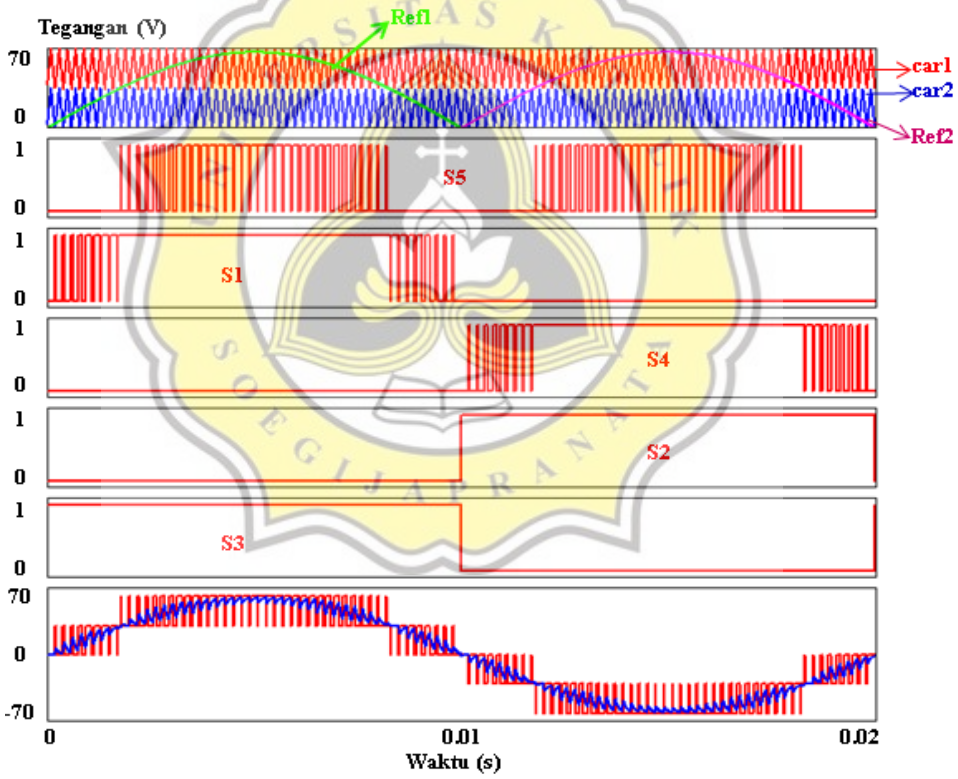
S1	S2	S3	S4	S5	VO
1	0	1	0	1	*2E*
1	0	1	0	0	*E*
1	0	0	0	0	*0*
0	0	0	1	0	*0*
0	1	0	1	0	*-E*
0	1	0	1	1	*-2E*

Gambar-3.8 menampilkan kendali proses pensaklaran untuk membangun *inverter* lima tingkat satu fasa tipe asimetris. Untuk membangun *inverter* lima tingkat satu fasa tipe asimetris memerlukan 2 sinyal pembawa dan sinyal referensi. Sinyal pembawa pertama diinisialisasikan car1 dan sinyal pembawa kedua diinisialisasikan car2. Pensaklaran pada S1 diperoleh dengan mengkomparasikan car1 dengan sinyal referensi, hal itu terjadi pada komparator keempat (comp4). Pensaklaran pada S4 diperoleh dengan mengkomparasikan car1 dengan sinyal referensi yang telah dikali dengan -1, hal itu terjadi pada komparator keempat (comp4). Pada komparator (comp1), mengkomparasikan car2 dengan sinyal referensi. Pada komparator (comp2), mengkomparasikan car2 dengan sinyal referensi yang telah dikali dengan -1. Hasil comp1 dan comp2 dimasukan ke gerbang logika *or* untuk mendapatkan pensaklaran pada S5. Sebelum pensaklaran dimasukan ke rangkaian daya, terlebih dahulu dimasukan ke *driver*.



Gambar-3.8 Kendali proses pensaklaran

Gambar-3.9 menampilkan proses modulasi lebar pulsa SPWM. Metode SPWM yang diterapkan menggunakan dua buah sinyal referensi. Sinyal referensi dari SPWM menggunakan sinyal *sinusoidal* dengan frekuensi 50 Hz. Sinyal referensi yang pertama diinisialisasikan sebagai ref1. Dengan menggeser fasa ref1 sebesar 180^{O} didapatkan sinyal referensi yang kedua dan diinisialisasikan sebagai ref2. Metode pergeseran fasa pada dua buah sinyal referensi merupakan metode unipolar. Metode tersebut diterapkan pada *inverter* satu fasa lima tingkat tipe asimetris karena mengadopsi dari *inverter* unipolar.

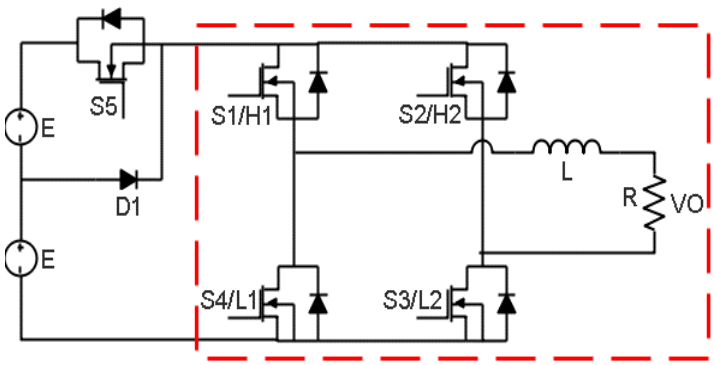


Gambar-3.9 Modulasi SPWM

Modulasi lebar pulsa SPWM juga membutuhkan dua buah sinyal pembawa atau yang dikenal dengan sebutan *carrier*. Sinyal pembawa yang digunakan merupakan sinyal gelombang segitiga dan memiliki frekuensi 5 Khz.

Sinyal pembawa pertama diinisialisasikan sebagai car1 dan sinyal pembawa kedua diinisialisasikan sebagai car2. Kedua buah sinyal pembawa tersebut bertingkat karena memiliki DC *offset* yang berbeda. Car1 berada di tingkat pertama dan car2 berada di tingkat kedua.

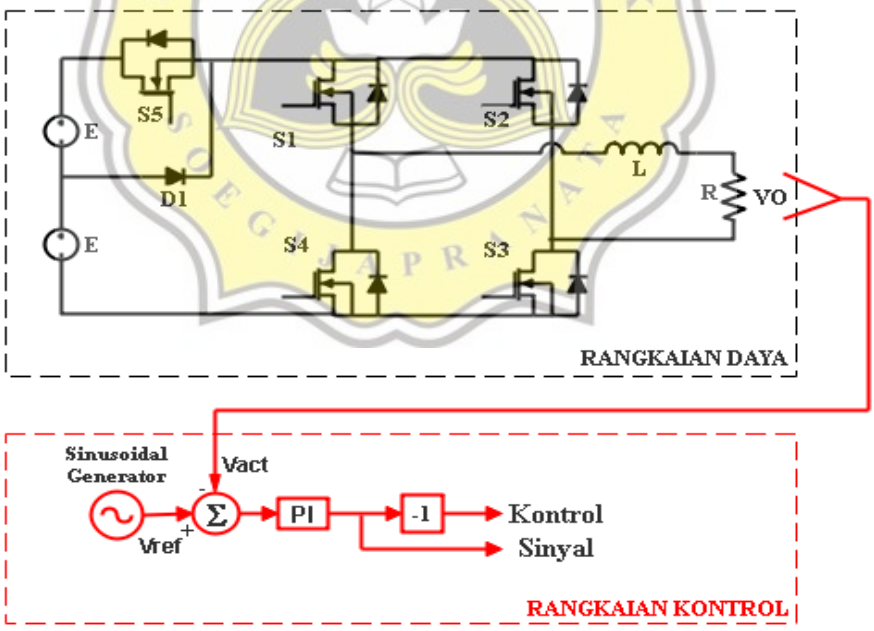
Inverter satu fasa lima tingkat tipe asimetris ini merupakan adopsi dari *inverter* konvensional jenis unipolar bertipe *H-bridge* hal itu ditunjukan pada Gambar-3.10. *Inverter* unipolar memerlukan proses pensaklaran dua buah lengan. Setiap lengan *inverter* unipolar dikendalikan secara independen dari lengan lainnya. Oleh sebab itu, merujuk dari tabel pensaklaran diatas ada beberapa ketentuan yang harus dipenuhi sesuai kaidah *inverter* unipolar. Saklar daya bagian atas atau yang dikenal dengan sebutan *high* yaitu S1 sebagai H1 dan S2 sebagai H2 tidak ada yang menyala bersamaan. Saklar daya bagian bawah atau yang dikenal dengan sebutan *low* yaitu S4 sebagai L1 dan S3 sebagai L2 juga tidak ada yang menyala bersamaan. Sedangkan untuk antar lengan H1 dengan L1 tidak ada yang menyala bersamaan begitu pula dengan H2 dengan L2 juga tidak ada yang menyala bersamaan dalam setiap siklus, 2E, E, 0, 0, -E, -2E.



Gambar-3.10 Pengadobsian dari *inverter* unipolar tipe *H-bridge*

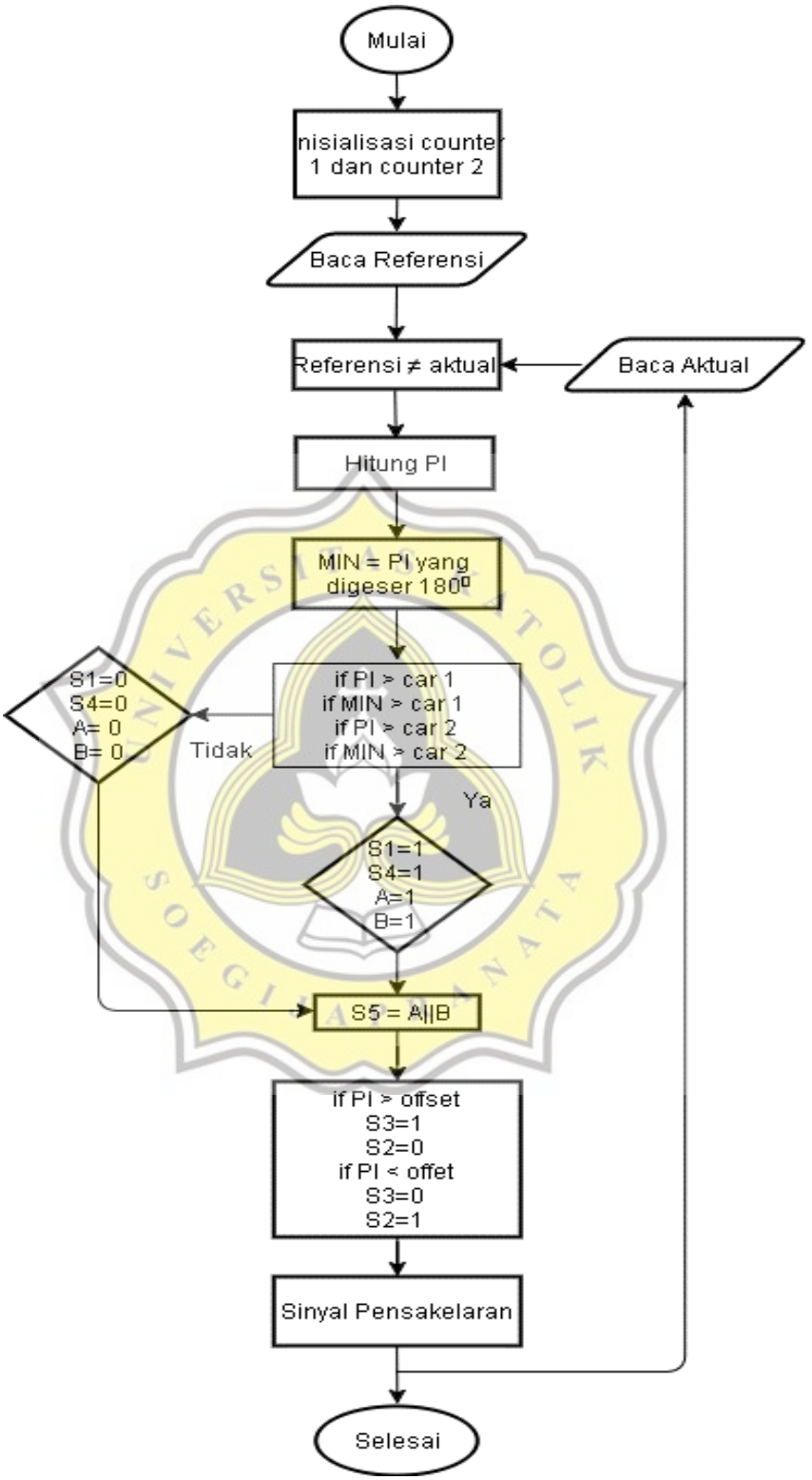
3.3 Perancangan Kendali

Inverter satu fasa lima tingkat tipe asimetris terkendali tegangan terbagi menjadi dua bagian yaitu rangkaian daya dan rangkaian kendali. Rangkaian daya terdiri dari lima buah saklar daya yaitu (S1), (S2), (S3), (S4), (S5), sebuah dioda (D1), dua buah sumber DC terpisah (E), tapis induktor (L), dan beban. Rangkaian kendali menggunakan sinyal referensi yang berupa sinyal *sinusoidal* dan sinyal aktual dari pembacaan sensor tegangan. Selisih dari sinyal referensi dan sinyal aktual menghasilkan sinyal error yang dikalikan dengan nilai ke PI. Hasil PI dikalikan dengan -1 agar tergeser fasanya sesuai dengan metode kendali unipolar. Skema rangkaian daya dan kendali ditunjukan pada Gambar-3.11.



Gambar-3.11 Rangkaian Daya Beserta Rangkaian Kendali

Pemrograman *inverter* lima tingkat satu fasa menggunakan mikrokontroler Arduino Due. Algoritma pemrograman dimulai dari pembacaan sinyal referensi dan pembacaan sinyal aktual dari tegangan keluaran *inverter*. Kemudian sinyal referensi dikurangkan dengan sinyal aktual dan dikalikan dengan nilai PI. PI yang digeser sebesar 180^0 diinisialisasikan dengan MIN. Jika nilai PI lebih besar daripada car1 maka S1 berlogika 1. Jika nilai MIN lebih besar dari car1 maka S4 berlogika 1. Jika nilai PI lebih besar daripada car2 maka A akan berlogika 1. Jika nilai MIN lebih besar daripada car2 maka B akan berlogika 1. Jika tidak maka S1, S4, A, dan B akan berlogika 0. Hasil dari A *or* B adalah S5. Nilai lebih dari batas 0 diinisialisasikan *offset*. Jika nilai PI lebih besar dari nilai *offset* maka S3 berlogika 1 dan S2 akan berlogika 0. Jika nilai PI lebih kecil dari nilai *offset* maka S3 berlogika 0 dan S2 akan berlogika 1. Algoritma pemrograman ditunjukan pada Gambar-3.12.



Gambar-3.12 Algoritma Program

Pemrograman pada arduino due adalah sebagai berikut:

```
#include<Arduino.h>
```
→ ***Header Library***

```
#define freq1 35000
#define freq2 24000
```
→ ***Header Identifier***

```
uint32_t *car1 = (uint32_t *)(0x40080010);
uint32_t *car2 = (uint32_t *)(0x40080010);
```
→ **Akses *Timer Counter* 0**

```
uint32_t t1,t2;
int off;
int x;
int y;
boolean l=0;
int sinus,sinus1,sinus2,A,B,S5,S1,S2,S3,S4,act,satu,dua;
int ref,error,MIN;
float kp=1;
float ki=0.1;
float p,i,pi;
```
→ **Deklarasi Variabel dan Konstanta**

```
void TC3_Handler(void)
{
```
→ **Sub Program *Interupt***

```
TC_GetStatus(TC1, 0);
digitalWrite(13, l = !l);
```
→ **Memanggil *interupt* dari *timer* 1 dan mengetahui kecepatan *interupt***

```
analogReadResolution(10);
```
→ **Membuat resolusi ADC menjadi 10 bit**

```
sinus1=analogRead(A0);
sinus2=analogRead(A1);
sinus1= map(sinus1,0,1024,0,255);
sinus2= map(sinus2,0,1024,0,25);
```
→ **Pembacaan ADC**

```
//Serial.println(sinus2);
//////////pi///////
error=sinus1-sinus2;
p=kp*error;
i=ki*error;
satu = p*((1+i)/i);
satu = map(satu, 0, 255, 0, 255);
dua  = map(satu, 0, 255, 255, 0);
pi=satu;
MIN=dua;
}
```
→ **Kendali *Proportional* dan *Integral***

```
void TC0_Handler(void)
{
      TC_GetStatus(TC0, 0);
```
→ **Sub Program *Interupt***

```
}
```

```
void startTimer1(Tc *tc, uint32_t channel, IRQn_Type irq, uint32_t frequency) {
        pmc_set_writeprotect(false);
        pmc_enable_periph_clk((uint32_t)irq);
        TC_Configure(tc,channel,TC_CMR_WAVE|
TC_CMR_WAVSEL_UPDOWN_RC | TC_CMR_TCCLKS_TIMER_CLOCK3);
        uint32_t rc = VARIANT_MCK/32/frequency;
        TC_SetRA(tc, channel, rc/2);
        TC_SetRC(tc, channel, rc);
        TC_Start(tc, channel);
        tc->TC_CHANNEL[channel].TC_IER=TC_IER_CPCS;
        tc->TC_CHANNEL[channel].TC_IDR=~TC_IER_CPCS;
        NVIC_EnableIRQ(irq);
}
```

Sub Program *Timer Counter* 1

```
void startTimer2(Tc *tc, uint32_t channel, IRQn_Type irq, uint32_t frequency) {
        pmc_set_writeprotect(false);
        pmc_enable_periph_clk((uint32_t)irq);
        TC_Configure(tc,channel,TC_CMR_WAVE|
TC_CMR_WAVSEL_UPDOWN_RC | TC_CMR_TCCLKS_TIMER_CLOCK3);
        uint32_t rc = VARIANT_MCK/32/frequency;
        TC_SetRA(tc, channel, rc/2);
        TC_SetRC(tc, channel, rc);
        TC_Start(tc, channel);
        tc->TC_CHANNEL[channel].TC_IER=TC_IER_CPCS;
        tc->TC_CHANNEL[channel].TC_IDR=~TC_IER_CPCS;
        NVIC_EnableIRQ(irq);
}
```

Sub Program *Timer Counter* 0

```
void setup() {
//Serial.begin(9600);
pinMode(A0,INPUT);
pinMode(A1,INPUT);
pinMode(13,OUTPUT);
pinMode(12,OUTPUT);
pinMode(11,OUTPUT);
pinMode(10,OUTPUT);
pinMode(9,OUTPUT);
pinMode(8,OUTPUT);
```

Inisialisasi I/0 *Port*

```
startTimer1(TC1, 0, TC3_IRQn,freq1);
startTimer2(TC0, 0, TC0_IRQn,freq2);
}
```

Inisialisasi *Timer Counter* 1 dan 0

```
void loop() {
```

Fungsi Perulangan

```
x=1;
y=255;
off=(x+y)/2;
```

Fungsi *Zero Crossing*

```
t1=*car1;
t1=map(t1,0,109,128,192);
t2=*car2;
t2=map(t2,0,109,270,192);
```

Pemanggilan *Timer Counter* 0

```
///////sinus/////
//Serial.println(sinus2);
//////////switching//////
if(pi<=t1)
{
digitalWrite(12,LOW); //s1H1
}
if(pi>=t1)
{
digitalWrite(12,HIGH); //s1H1
}
```

Pensaklaran pada saklar daya S1

```
if(MIN<=t1)
{
digitalWrite(11,LOW); //s4L1
}
if(MIN>=t1)
{
digitalWrite(11,HIGH); //s4L1
}
```

Pensaklaran pada saklar daya S4

```
if(pi>t2)
{
A=1;
}
if(pi<t2)
{
A=0;
}
if(MIN>t2)
{
B=1;
}
if(MIN<t2)
{
B=0;
}
```

Data untuk proses logika *OR*

```
S5=A||B;
digitalWrite(8,S5);
```

Pensaklaran pada saklar daya S5

```
//zero
if(sinus1<=off)      //s3 L2
  {
  digitalWrite(9,LOW);}
  else if (sinus1>=off)
  {
  digitalWrite(9,HIGH);}
```

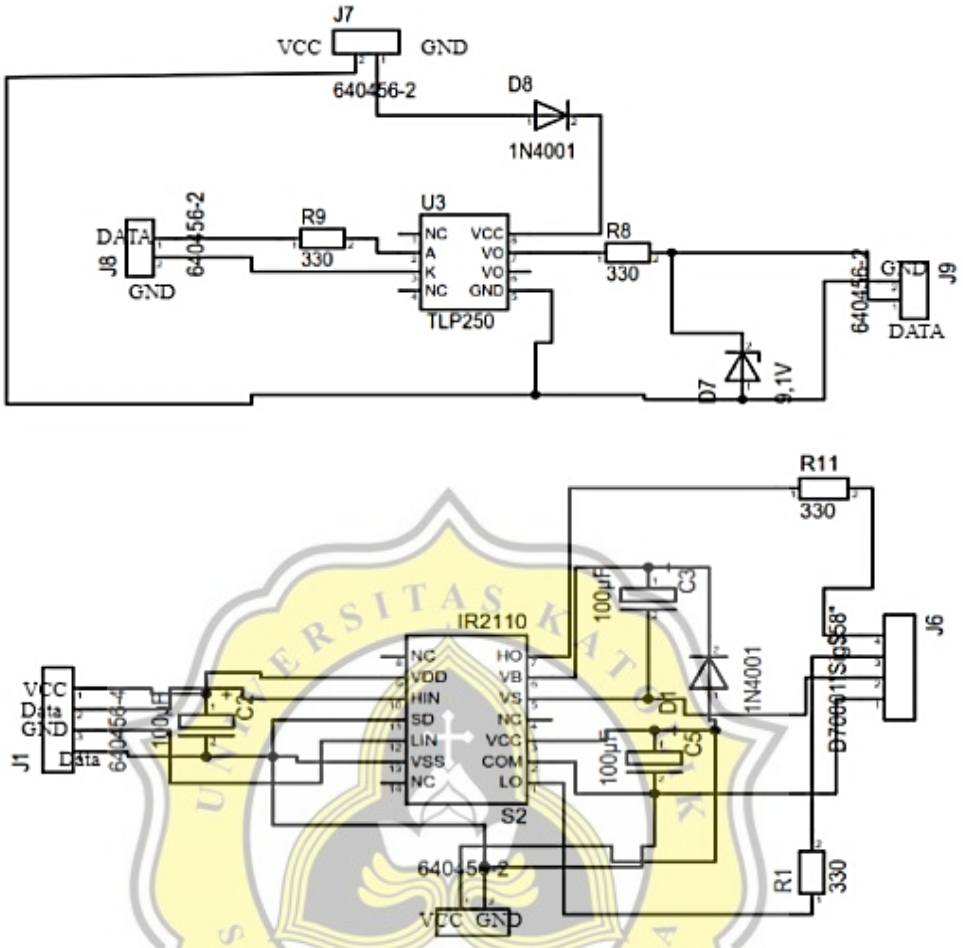
***Zero Crossing* untuk pensaklaran pada saklar daya S3**

```
if(sinus1<=off)      //s2 H2
  {
  digitalWrite(10,HIGH);}
  else if (sinus1>=off)
  {
  digitalWrite(10,LOW);}
}
```

***Zero Crossing* untuk pensaklaran pada saklar daya S2**

3.3.1 Rangkaian *Driver*

Rangkaian kontrol untuk *inverter* satu fasa lima tingkat ini diimplementasikan menggunakan 3 buah *driver* MOSFET. *Driver* MOSFET yang digunakan pada tugas akhir ini adalah 1 buah TLP250 dan 2 buah IR2110. Skema rangkaian *driver* MOSFET ditunjukan pada Gambar-3.13. Rangkaian *driver* ini digunakan sebagai pemisah / isolator antara rangkaian daya dengan mikrokontroler. Hal itu untuk meminimalisir jika terjadi masalah dalam rangkaian daya seperti terjadi *short*, tidak akan langsung berdampak pada mikrokontroler, demikian sebaliknya. Rangkaian *driver* MOSFET ini juga sebagai media untuk mengirim sinyal SPWM dari mikrokontroler menuju rangkaian daya. MOSFET pada rangkaian daya mempunyai spesifikasi sinyal *gate* sebesar 12 *Volt* untuk aktif. Oleh sebab itu, *driver* MOSFET mempunyai fungsi sebagai penguat sinyal dari mikrokontroler sebesar 5 *Volt* dikuatkan menjadi 12 *Volt*.

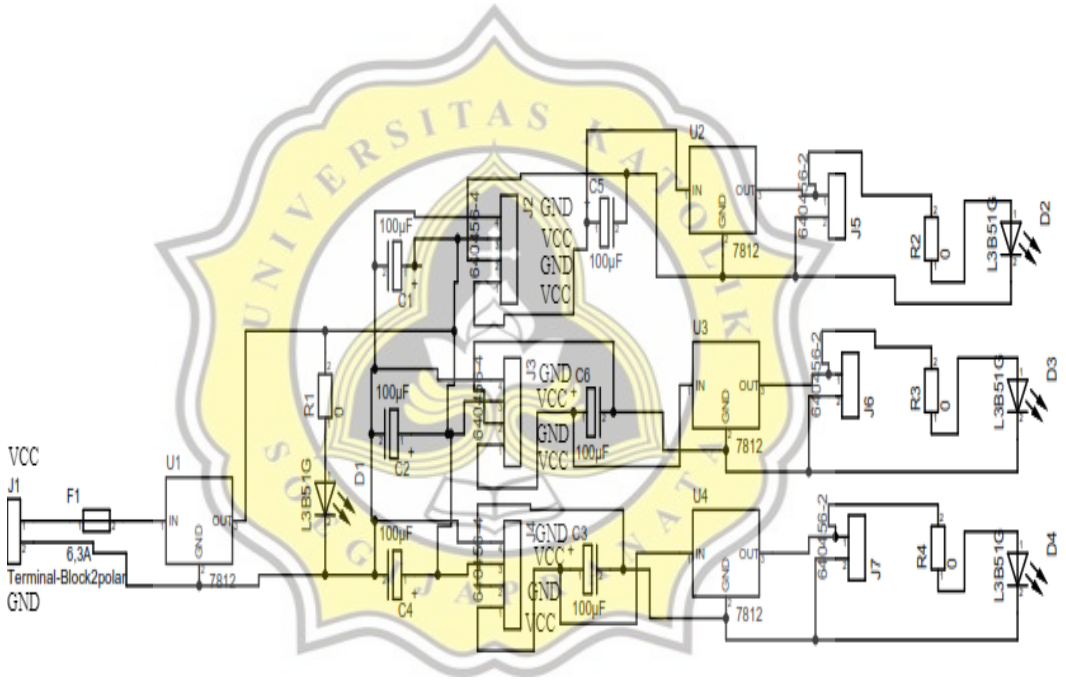


Gambar-3.13 Rangkaian *Driver*

3.3.2 Rangkaian Catu Daya *Driver*

Rangkaian *driver* MOSFET memerlukan catu daya dengan tegangan 12 *volt* DC. Catu daya yang dibutuhkan sebanyak 3 buah dan menggunakan catu daya DC – DC *isolated*. Untuk membuat catu daya DC – DC *isolated* menggunakan komponen B1212. Prinsip kerja dari B1212 yaitu mengubah 12 *volt* DC menjadi 12 *volt* DC yang terisolasi tegangan keluarannya dari tegangan sumbernya. B1212 memiliki fungsi lain yaitu, untuk membatasi tegangan keluaran agar tidak melebihi 12 volt DC. Pada tegangan input dari B1212 diberi

fuse sebagai pengaman / proteksi jika terjadi hubung singkat atau yang dikenal dengan sebutan *short*. *Fuse* mempunyai batasan arus, jika terjadi hubung singkat maka arus yang mengalir menjadi besar sehingga *fuse* akan terputus dan arus tidak mengalir ke B1212. Selain itu diberikan lampu LED sebagai indikator pada catu daya. Jika catu daya telah di injek oleh arus maka LED menyala. Jika lampu LED menyala redup maka terjadi hubung pendek. Rangkaian catu daya *driver* ditunjukan pada Gambar-3.14.



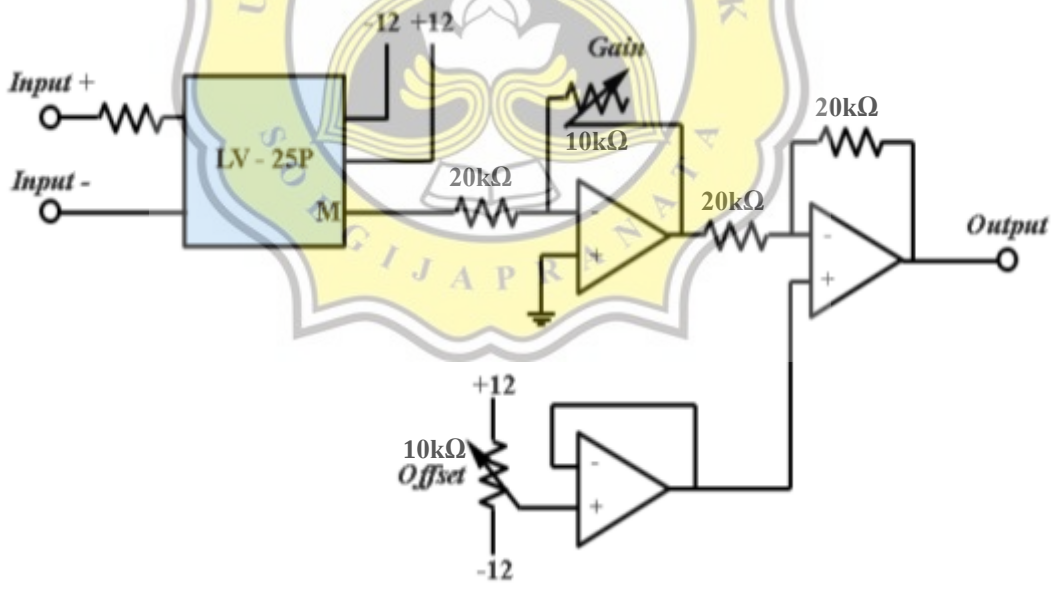
Gambar-3.14 Rangkaian Catu Daya *Driver*

3.4 Perancangan Rangkaian Sensor Tegangan

Catu daya mandiri memiliki dua buah syarat. Syarat pertama yaitu tegangan yang dihasilkan dapat teregulasi sesuai keinginan. Untuk mendapatkan tegangan keluaran yang teregulasi membutuhkan pengendalian tegangan pada sisi

output catu daya mandiri. Pengendalian tegangan keluaran membutuhkan sensor tegangan untuk membaca hasil tegangan keluaran dari catu daya mandiri.

Sensor tegangan LV25P dirangkai secara paralel pada beban yang terpasang pada catu daya mandiri dalam hal ini *inverter* satu fasa lima tingkat tipe asimetris. Sensor tegangan ini dibutuhkan untuk menerapkan rangkaian tertutup (*closed loop*) pada *inverter* satu fasa lima tingkat tipe asimetris. LV25P membutuhkan rangkaian penguat berupa *op – amp* agar *gain* dan *offset* dapat diatur. Tujuan mengatur gain dan *offset* agar mikrokontroler tidak mengalami kerusakan, akibat melebihi karakteristik tegangan masukan pada mikrokontroler. Skema rangkaian sensor tegangan ditunjukan pada Gambar-3.15.



Gambar-3.15 Skema Rangkaian Sensor Tegangan