

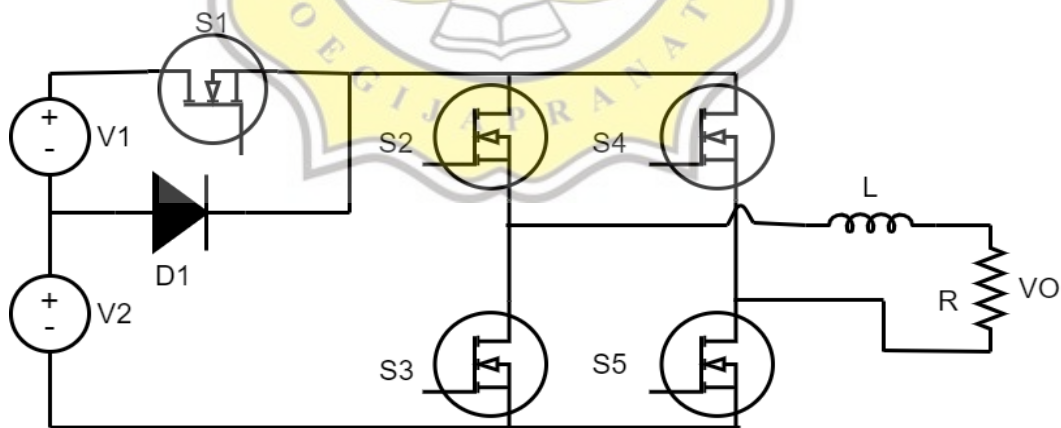
BAB III

RANCANGAN DAN MODE OPERASI

3.1 Pendahuluan

Dalam Bab ini dibahas mengenai perancangan dan implementasi *inverter* lima tingkat dengan kendali PI beserta mode operasinya. *Inverter* ini menggunakan lima buah saklar daya dan satu saklar pasif dengan dua sumber tegangan DC. Gerbang logika *OR* digunakan sebagai pembentuk tegangan lima tingkat. Modul pendukung dalam implementasi perangkat keras akan dibahas dalam bab ini. Referensi arus berasal dari AFG dan arus aktual berasal dari sensor arus LEM HX 10-P.

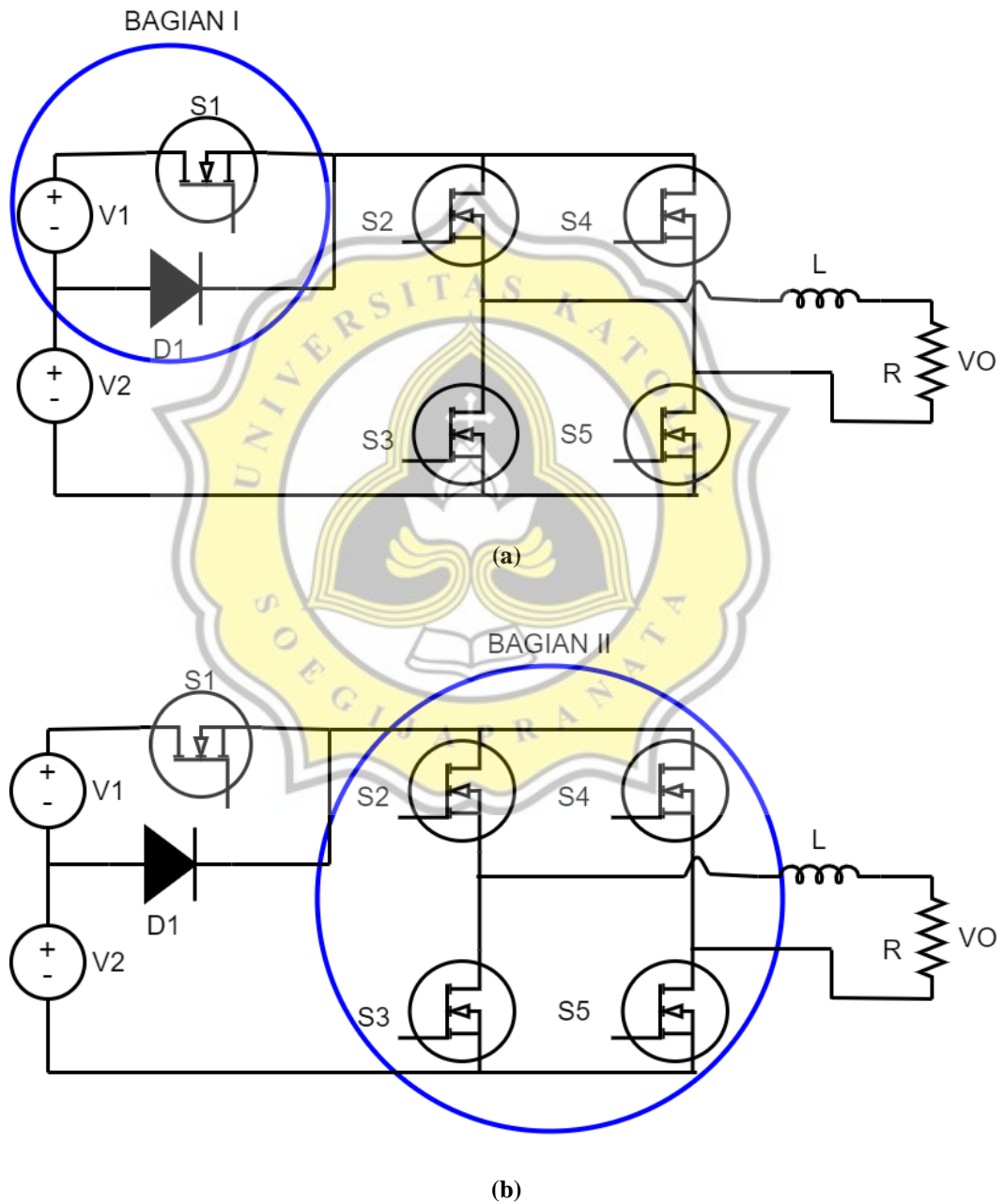
3.2 Topologi *Inverter* Lima Tingkat Satu Fasa



Gambar 3.1 Topologi *Inverter* Lima Tingkat Satu Fasa

Dapat dilihat topologi rangkaian *Inverter* Lima Tingkat Satu Fasa Pada Gambar 3.1 [24]. Pada topologi rangkaian *inverter* ini membutuhkan lima buah saklar daya yang terdiri dari S1, S2, S3, S4, dan S5. Satu buah dioda dan dua buah

sumber tegangan DC yang digunakan untuk pembentukan tingkatan tegangan. Tapis induktor digunakan untuk *filter* keluaran *inverter*. *Inverter* ini disebut dengan *Current Source Inverter (CSI)*. *Inverter* sumber arus (CSI) memiliki tapis induktor pada keluarannya.



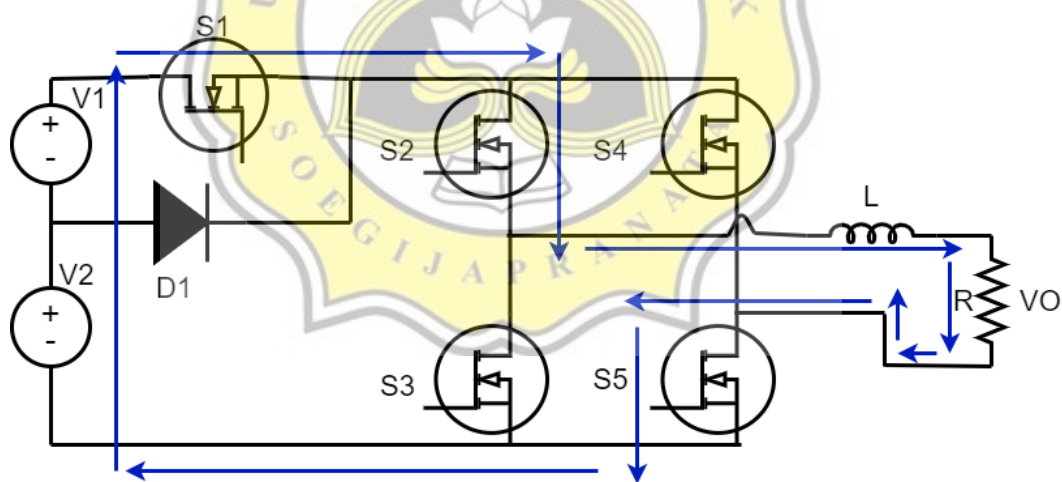
Gambar 3.2 Pembangkitan Tingkatan: (a). Bagian Pertama (b). Bagian Kedua.

Bagian pertama (S1) disebut sebagai pembangkitan tingkat $2E$ (V_2+V_1) pada polaritas positif dan negatif. Bagian kedua (S2-S5) sebagai pembentuk tingkat E (V_1). Bagian pertama dan kedua dapat dilihat pada Gambar 3.2.

3.3 Mode Operasi

Pensaklaran yang diimplementasikan pada topologi ini memiliki 5 mode operasi untuk mendapatkan keluaran tegangan 5 tingkat. Dimana mode operasi dimulai dari mode operasi 1, mode operasi 2, mode operasi 3, mode operasi 4, dan mode operasi 5.

3.3.1 Mode Operasi 1



Gambar 3.3 Mode Operasi 1.

Mode operasi 1 menghasilkan keluaran tegangan sebesar $2E$. S1, S2, S5 aktif dan S3, S4 mati. Gambar 3.3 menunjukkan jalur arus yang aktif pada tahap ini. Mode operasi 1 memiliki persamaan sebagai berikut:

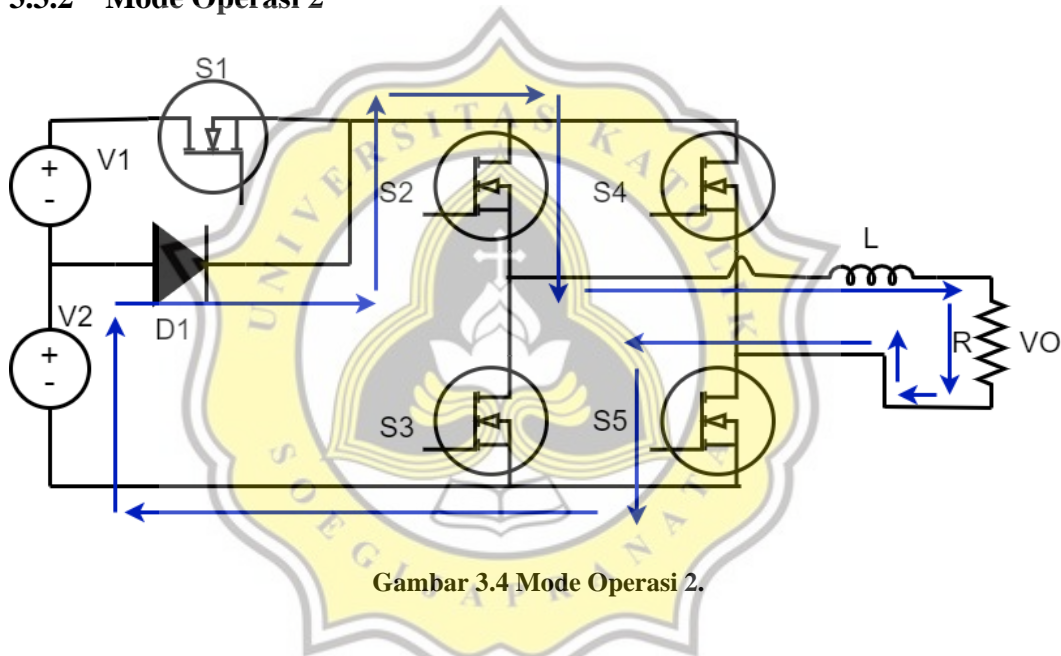
$$E - E = V_{(L)} + V_{(o)}$$

$$2E = L \frac{di_{(L)}}{dt} + V_{(o)}$$

$$L \frac{di_{(L)}}{dt} = 2E - V_{(o)}$$

$$L\Delta i_{(L)} = (2E - V_{(o)})\Delta t = (2E - V_{(o)})t_{(on)} \quad (3.1)$$

3.3.2 Mode Operasi 2



Gambar 3.4 Mode Operasi 2.

Mode operasi 2 menghasilkan keluaran tegangan sebesar E . S_2, S_5 aktif dan S_3, S_4, S_5 mati. Gambar 3.4 menunjukkan jalur arus yang aktif pada tahap ini.

Mode operasi 2 memiliki persamaan sebagai berikut:

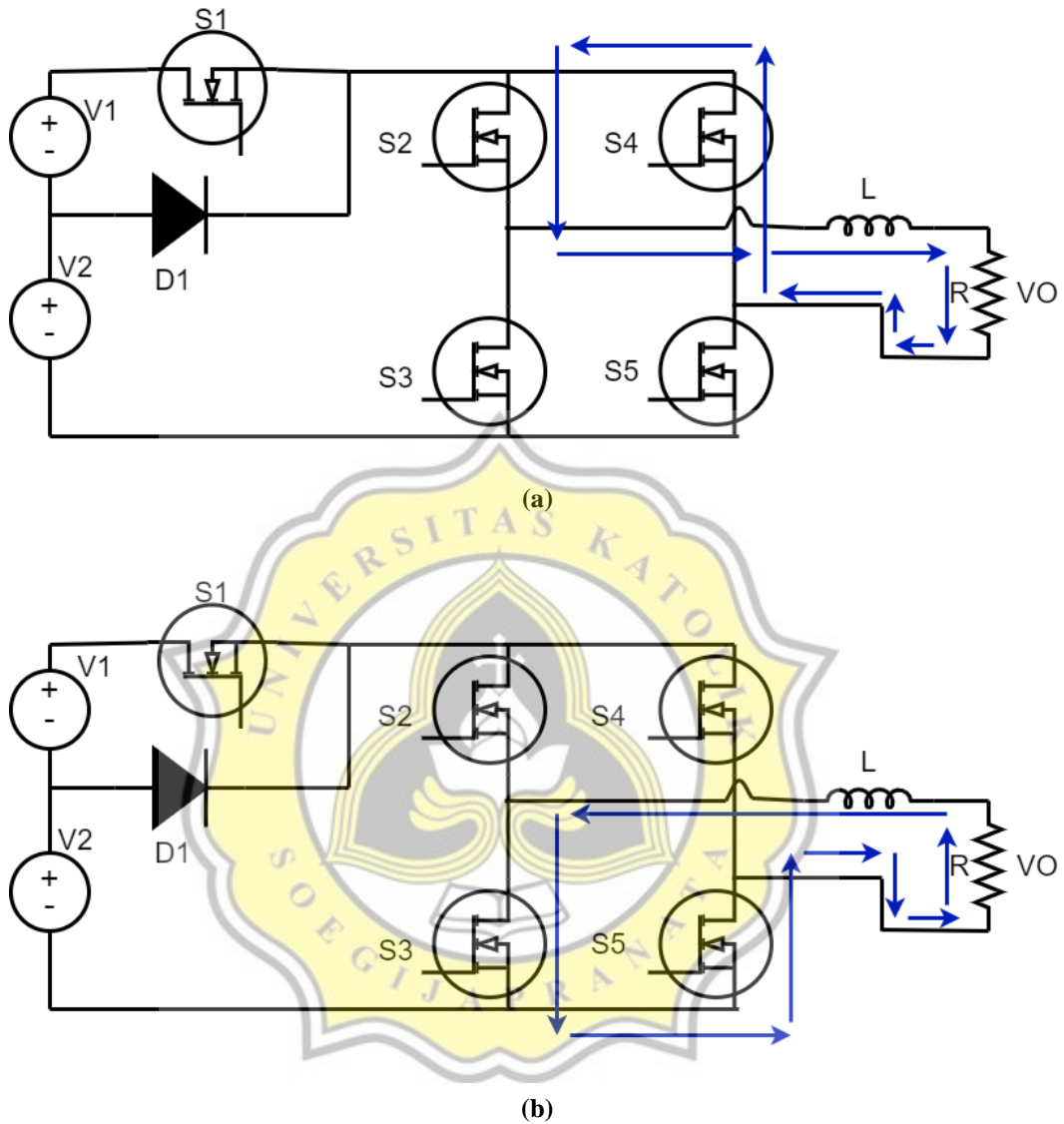
$$E = V_{(L)} + V_{(o)}$$

$$E = L \frac{di_{(L)}}{dt} + V_{(o)}$$

$$L \frac{di_{(L)}}{dt} = E - V_{(o)}$$

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

3.3.3 Mode Operasi 3



Gambar 3.5 Mode Operasi 3: (a). Siklus Positif. (b). Siklus negatif

Mode operasi 3 merupakan kondisi *freewheeling*, menghasilkan dua keluaran tegangan sebesar nol. Mode tersebut terdiri dari 2 siklus, yaitu siklus positif dan negatif. Pada siklus positif S2, S4 aktif dan S1, S3, S5 mati. Gambar 3.5 (a) menunjukkan jalur arus yang aktif pada tahap ini. Pada siklus negatif S3, S5

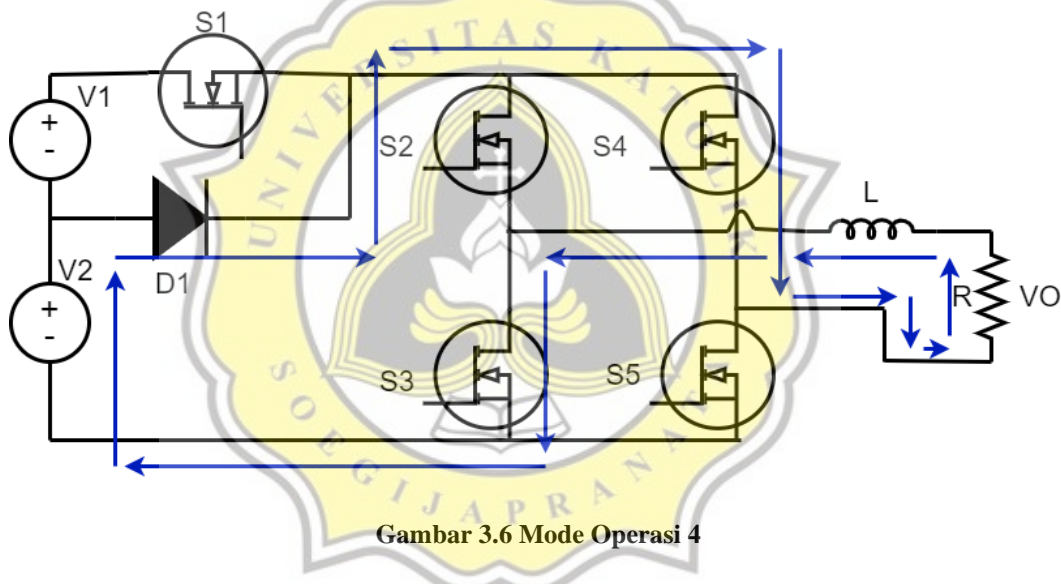
aktif dan S1, S2, S4 mati. Gambar 3.5 (b) menunjukkan jalur arus yang aktif pada tahap ini. Mode operasi 3 memiliki persamaan sebagai berikut:

$$V_{(L)} = V_{(O)} - V_{(d)}$$

$$L \frac{di_{(L)}}{dt} = V_{(O)} - 0$$

$$L\Delta i_{(L)} = (V_{(O)})\Delta t = (V_{(O)})t_{(off)} \quad (3-3)$$

3.3.4 Mode Operasi 4



Gambar 3.6 Mode Operasi 4

Mode operasi 4 menghasilkan keluaran tegangan sebesar $-E$. S3, S4 aktif dan S1, S2, S5 mati. Gambar 3.6 menunjukkan jalur arus yang aktif pada tahap ini.

Mode operasi 4 memiliki persamaan sebagai berikut:

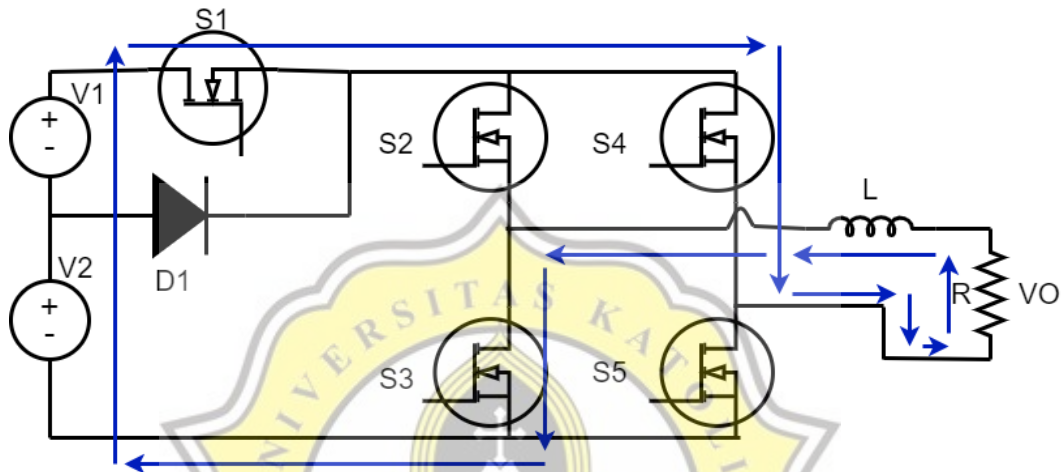
$$-E = V_{(L)} + V_{(O)}$$

$$-E = L \frac{di_{(L)}}{dt} + V_{(O)}$$

$$L \frac{di_{(L)}}{dt} = V_{(O)} - E$$

$$L\Delta i_{(L)} = (V_{(o)} - E)\Delta t = (V_{(o)} - E)t_{(on)} \quad (3.4)$$

3.3.5 Mode Operasi 5



Gambar 3.7 Mode Operasi 5

Mode operasi 5 menghasilkan keluaran tegangan sebesar $-2E$. S1, S3, S4 aktif dan S2, S5 mati. Gambar 3.7 menunjukkan jalur arus yang aktif pada tahap ini. Mode operasi 5 memiliki persamaan sebagai berikut:

$$-(E - E) = V_{(L)} + V_{(o)}$$

$$-2E = L \frac{di_{(L)}}{dt} + V_{(o)}$$

$$L \frac{di_{(L)}}{dt} = V_{(o)} - 2E$$

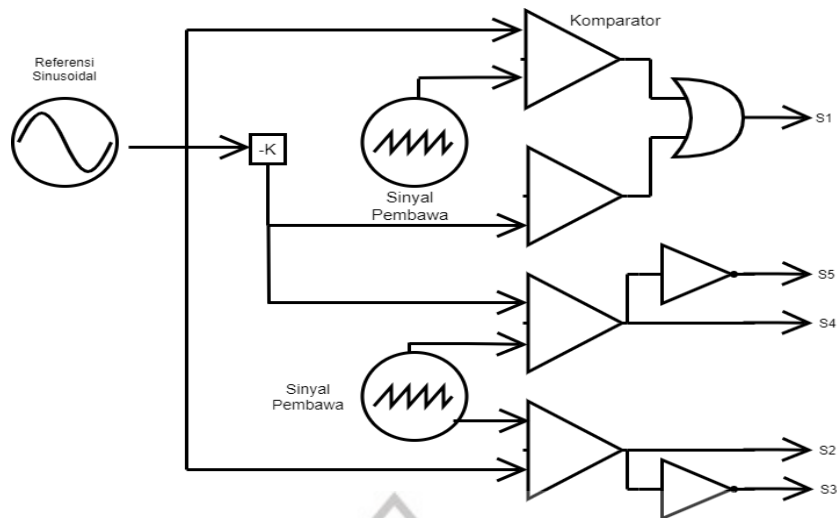
$$L\Delta i_{(L)} = (V_{(o)} - 2E)\Delta t = (V_{(o)} - 2E)t_{(on)} \quad (3.5)$$

Dari mode operasi 1 sampai mode operasi 5, dapat dibuat pola pensaklaran pada setiap saklar daya *inverter* lima tingkat satu fasa. Pola pensaklaran *Inverter* lima tingkat dapat dilihat pada Tabel 3.1.

Tabel 3.1. Pola Pensaklaran *Inverter* Lima Tingkat.

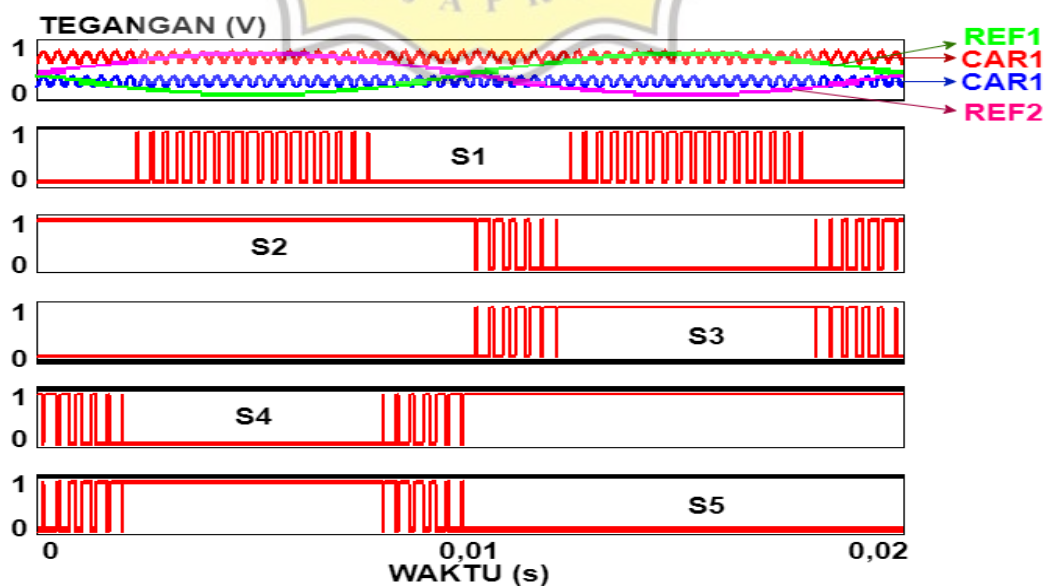
| S1 | S2 | S3 | S4 | S5 |
|----|----|----|----|----|
| 1 | 1 | 0 | 0 | 1 |
| 0 | 1 | 0 | 0 | 1 |
| 0 | 1 | 0 | 1 | 0 |
| 0 | 0 | 1 | 0 | 1 |
| 0 | 0 | 1 | 1 | 0 |
| 1 | 0 | 1 | 1 | 0 |

Dari pola pensaklaran pada Tabel 3.1 dibuatlah modulasi lebar pulsa sinusoidal. Untuk mendapatkan modulasi lebar pulsa sinusoidal dibutuhkan referensi satu buah gelombang sinusoidal. Gelombang ini diubah sebesar 120^0 sehingga mendapatkan dua buah gelombang sinusoidal yang saling bergeser sebesar 120^0 . Dua buah gelombang sinusoidal ini dibandingkan dengan dua sinyal pembawa. Hasil sinyal ini diteruskan ke setiap saklar daya. Pensaklaran Pada *Inverter* Lima Tingkat Gambar 3.8.



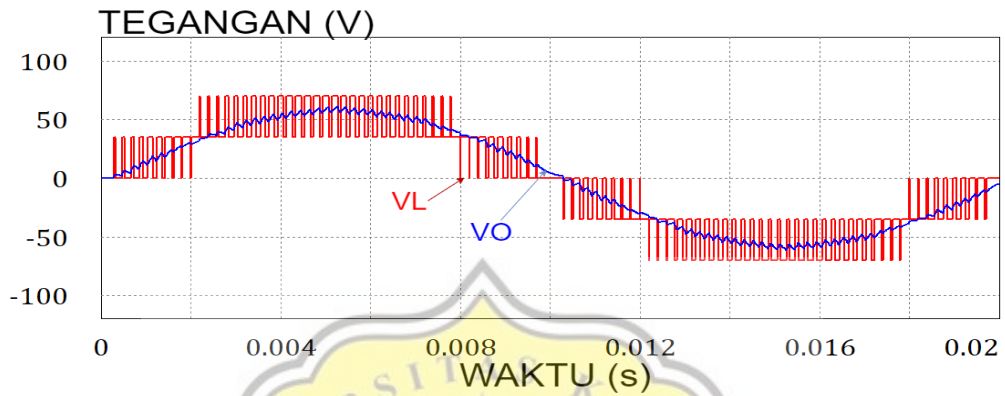
Gambar 3.8. Pensaklaran Pada *Inverter* Lima Tingkat

Teknik *unipolar* digunakan pada perangkat keras ini. Teknik ini membandingkan 2 gelombang sinyal sinusoidal yang saling bergeser sebesar 120° yang dibandingkan dengan 2 buah sinyal pembawa untuk menghasilkan modulasi lebar pulsa sinusoidal. Hasil dari modulasi ini diteruskan ke setiap saklar daya. CAR1 dan CAR2 merupakan sinyal pembawa. Modulasi lebar pulsa sinusoidal dapat dilihat pada Gambar 3.9. Teknik ini dinamakan dengan kontrol *unipolar*.



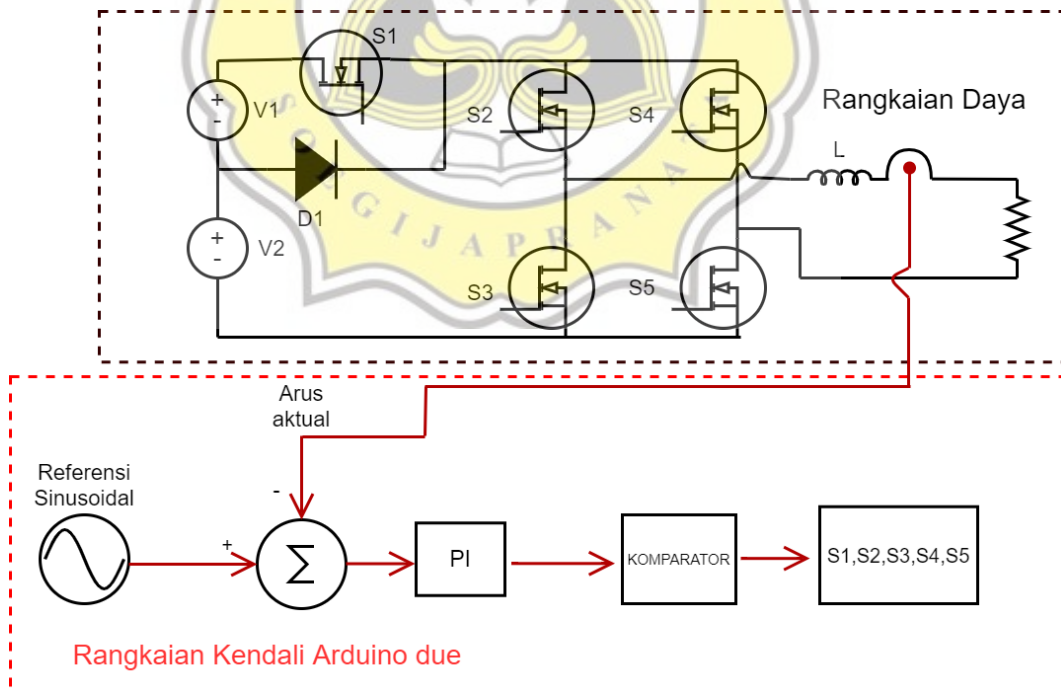
Gambar 3.9 Modulasi Lebar Pulsa Sinusoidal *Unipolar*

Dengan sinyal modulasi lebar pulsa sinusoidal yang telah dibuat pada Gambar 3.10. Hasil keluaran tegangan akan membentuk tingkatan tegangan E dan $2E$.



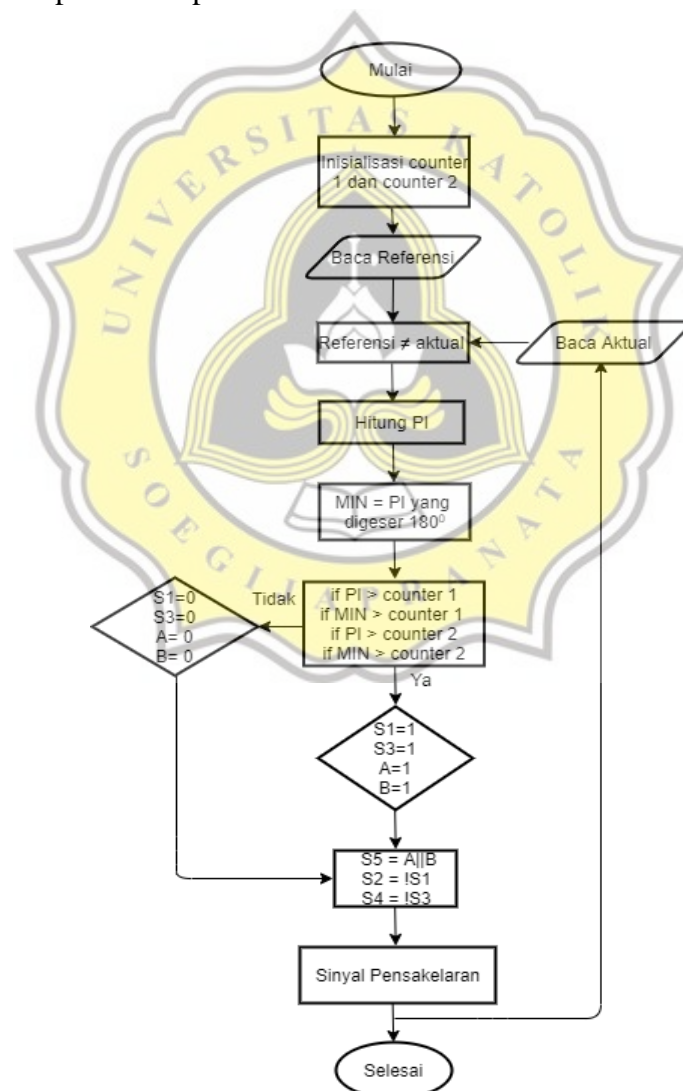
Gambar 3.10. Hasil Keluaran Tegangan Sesudah Dan Sebelum Induktor

3.4 Desain Kendali *Inverter*



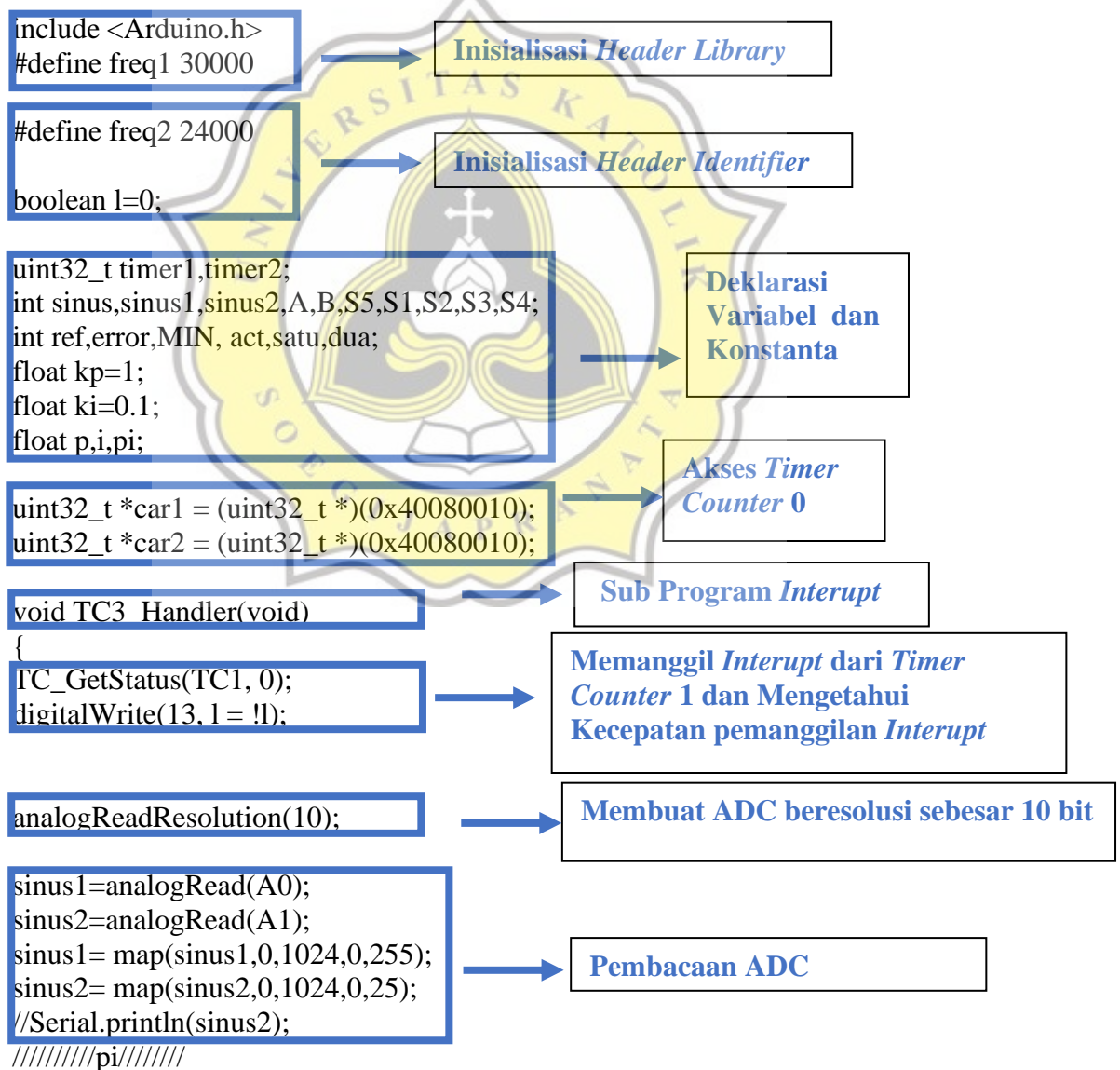
Gambar 3.11 Blok Diagram Sistem Kendali *Inverter* Lima Tingkat

Berdasarkan Gambar 3.8 dilakukan suatu pemrograman perangkat lunak dengan menggunakan *Arduino IDE* yang diimplementasikan pada mikrokontroler *Arduino Due*. Dimana referensi sinusoidal merupakan referensi arus dan arus aktual hasil dari keluaran sensor arus. Kendali yang digunakan adalah kendali PI. Sinyal keluaran kontrol dimodulasi dengan sinyal pembawa sehingga menghasilkan modulasi lebar pulsa sinusoidal. Implementasi blok diagram sistem kendali *inverter* lima tingkat dapat dilihat pada Gambar 3.8.



Gambar 3.12 Diagram Alur Program

Diagram alur program pada Gambar 3.9 menunjukkan diagram alur program yang digunakan. Menggunakan referensi sinusoidal yang berasal dari *Audio Function Generator* (AFG) dan arus aktual yang berasal dari sensor arus. Apabila Referensi tidak sesuai dengan aktual maka sistem kendali PI aktif. Hasil sinyal keluaran PI digeser 180^0 sehingga menghasilkan dua buah sinyal kontrol PI yang tergeser 180^0 . Dua sinyal kontrol ini akan dimodulasi dengan *counter* 1 dan *counter* 2. Algoritma program yang telah dibuat seperti berikut:



```

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 Proporsional dan Integral

```

void TC0_Handler(void)
{
    TC_GetStatus(TC0, 0);
    //digitalWrite(13, 1 = !1);
}

```

Sub Program Interupt

Memanggil Interupt dari Timer Counter 0 dan Mengetahui Kecepatan pemanggilan 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_R
    C | 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_R
    C | 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/O Port

→ Inisialisasi I/O Port

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

→ Inisialisasi Timer Counter 1 dan 0

```
void loop() {
```

→ Fungsi Perulangan

```
timer1=*car1;  
timer1=map(timer1,0,109,128,192);  
timer2=*car2;  
timer2=map(timer2,0,109,255,192);
```

→ Pemanggilan Timer Counter 0

```
/////sinus/////
```

```
////////switching////////
```

```
if(pi>timer1)//s1  
{  
  S1=1;  
}  
if(pi<timer1)  
{  
  S1=0;  
}
```

→ Pensaklaran Pada Saklar Daya S1

```
if(MIN>timer1)//s2  
{  
  S3=1;  
}  
if(MIN<timer1)  
{  
  S3=0;  
}
```

→ Pensaklaran Pada Saklar Daya S3

```
if(pi>timer2)
{
A=1;
}
if(pi<timer2)
{
A=0;
}
```

Data Untuk Proses Logika OR

```
if (MIN>timer2)
{
B=1;
}
if(MIN<timer2)
{
B=0;
}
```

Data Untuk Proses Logika OR

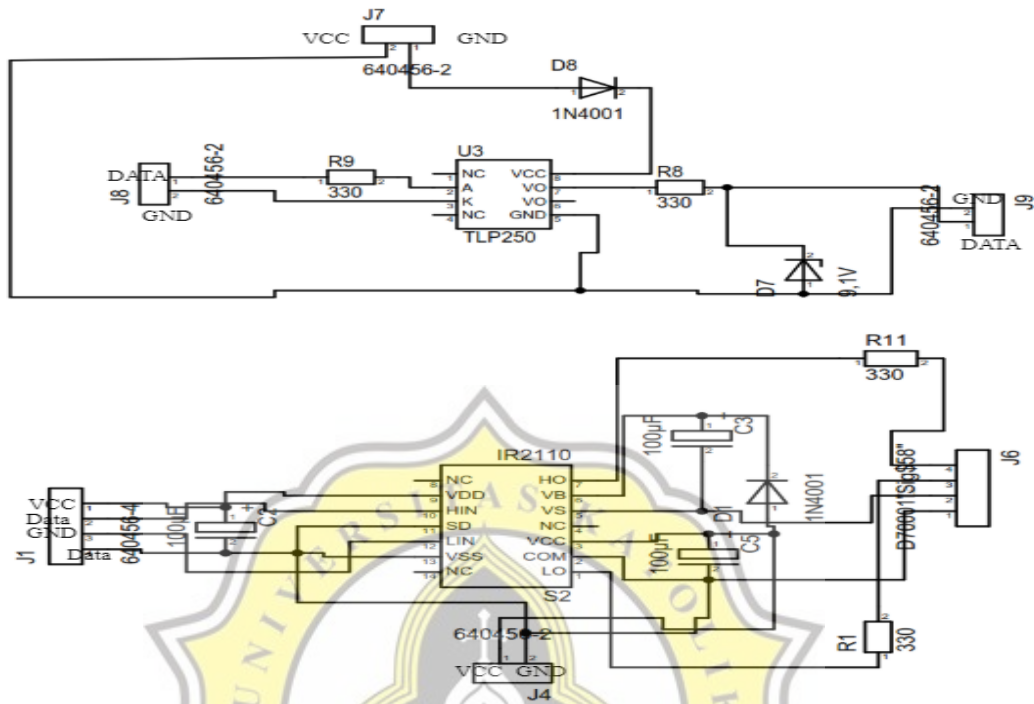
```
S5=A||B;
```

Pensaklaran Pada Saklar Daya S5

```
digitalWrite(8,S5);
digitalWrite(12,S1);
digitalWrite(11,!S1);
digitalWrite(10,S3);
digitalWrite(9,!S3);
}
```

Modulasi Lebar Pulsa Sinusoidal Pada Setiap Saklar Daya

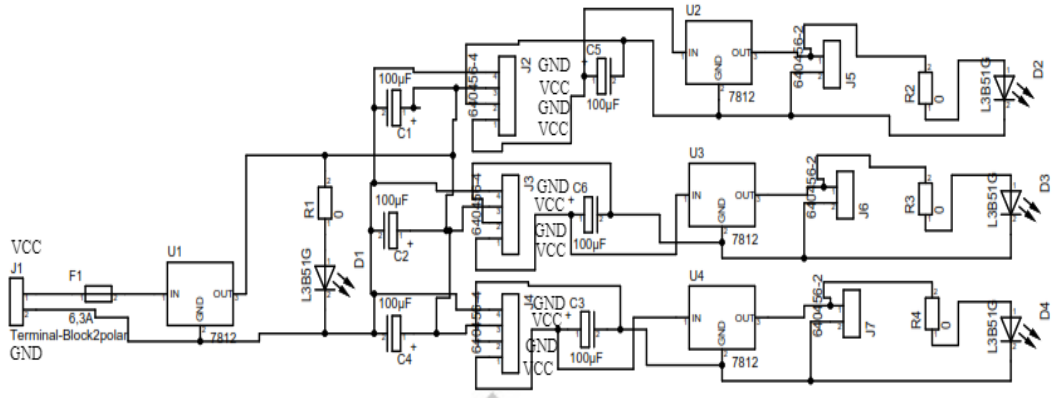
3.5 Rangkaian Driver



Gambar 3.13 Rangkaian Driver

Rangkaian *driver* terdiri dari 2 komponen utama yaitu TLP250 dan IR2110 yang digunakan untuk pensaklaran pada setiap saklar daya. TLP250 berfungsi sebagai isolasi sinyal keluaran dari mikrokontroler Arduino Due dan sebagai pensaklaran untuk satu saklar daya. IR2110 berfungsi sebagai pensaklaran pada dua saklar daya dan meminimalisir jumlah penggunaan TLP250. Susunan rangkaian *driver inverter* lima tingkat satu fasa ditunjukkan pada Gambar 3.10. Sinyal keluaran mikrokontroler Arduino Due berupa modulasi lebar pulsa sinusoidal dilanjutkan menuju TLP250 dan IR2110. Kemudian TLP250 mengisolasi sinyal yang berasal dari mikrokontroler Arduino Due. Sinyal tersebut dilanjutkan menuju *driver* IR2110 yang digunakan sebagai kendali pada saklar daya *inverter*.

3.6 Rangkaian Catu Daya



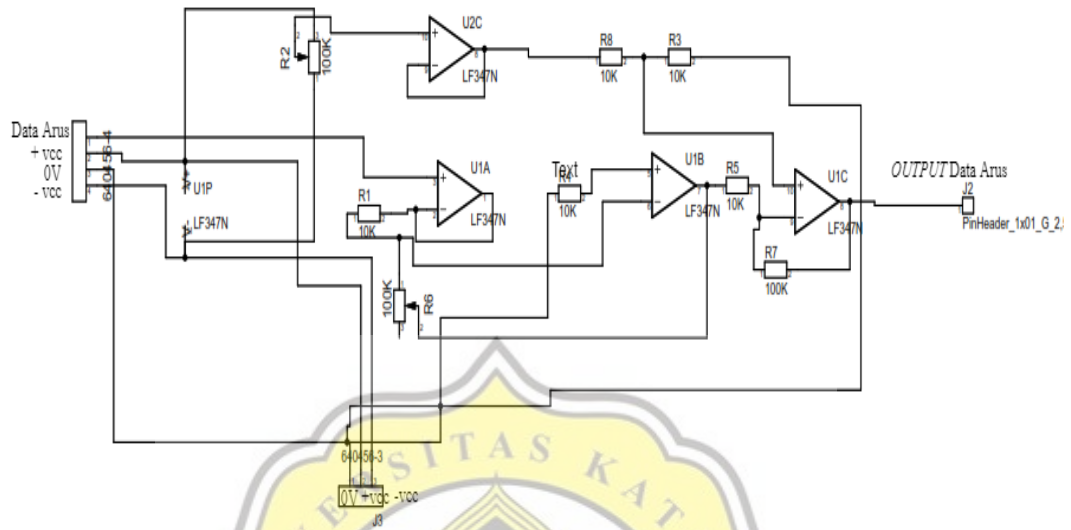
Gambar 3.14 Rangkaian Catu Daya

Catu daya digunakan sebagai suplai tegangan pada *driver* saklar daya dan mikrokontroler Arduino Due. *Driver* MOSFET membutuhkan sumber DC 12V dan Arduino Due membutuhkan sumber DC 3.3V untuk dapat bekerja. Untuk mendapatkan suplai tegangan sebesar nilai tersebut, dibutuhkan sumber daya yang terisolasi pada setiap rangkaian. Komponen catu daya DC-DC *isolated* B1212S-1W digunakan untuk memenuhi suplai tegangan pada rangkaian *driver*. Dapat dilihat pada Gambar 3.11.

3.7 Rangkaian Sensor Arus LEM HX 10-P

Sensor arus yang digunakan pada *inverter* lima tingkat satu fasa adalah sensor arus LEM HX 10-P. Prinsip kerja dari sensor arus ini dengan mendeteksi arus dan selanjutnya diubah dalam bentuk tegangan yang nantinya akan masuk ke mikrokontroler Arduino Due. Arus maksimal yang dapat dilewati pada sensor arus LEM HX 10-P ini adalah sebesar 10A. Rangkaian sensor arus terdapat pada

Gambar 3.12 Rangkaian sensor arus terdiri dari pengatur *offset* dan penguatan untuk selanjutnya diproses oleh mikrokontroler Arduino Due.



Gambar 3.15 Skema Rangkaian Sensor Arus LEM HX 10-P