

BAB IV HASIL PENGUJIAN DAN ANALISA

4.1 Pengujian Rangkaian dengan simulator

Sebelum melakukan pengujian dalam skala nyata, rangkaian inverter diuji pada skala komputasi. Dengan menggunakan *software power simulator* rangkaian inverter dibuat dan diuji untuk mendapatkan parameter alat yang digunakan dan kepastian hasil tegangan keluaran. Setelah mendapatkan hasil yang diinginkan dalam simulasi ini, data yang dibutuhkan untuk pensaklaran digunakan dan diakuisisi dengan menggunakan *Microsoft excel* untuk dijadikan data digital (biner) sebelum digunakan dalam pemrograman.

Untuk pengujian rangkaian, dilakukan dengan menggunakan parameter sebagai berikut:

Tegangan DC 1 = 40 Volt

Tegangan DC 2 = 40 Volt

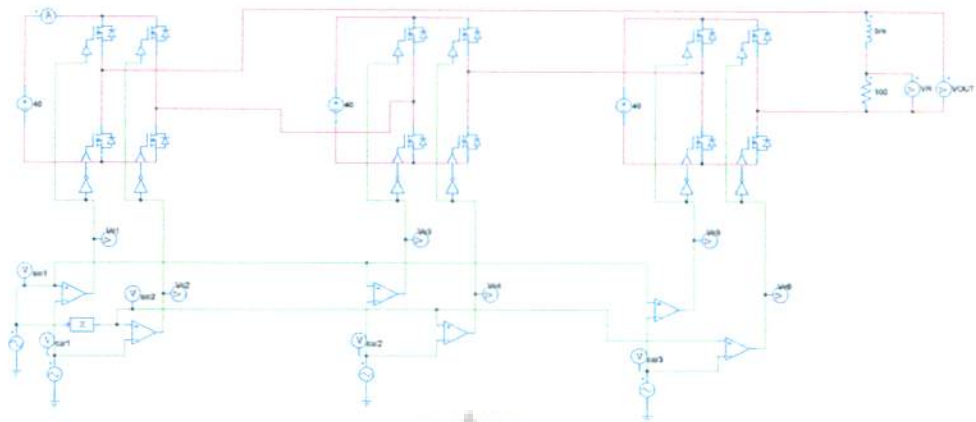
Tegangan DC 3 = 40 Volt

Induktansi = 5 mH

Beban resistif = 100 Ohm

Frekuensi switching = 97.65625 hz

Indeks Modulasi = 0.8 V



Gambar 4.1. simulasi Multilevel inverter satu fasa jembatan penuh

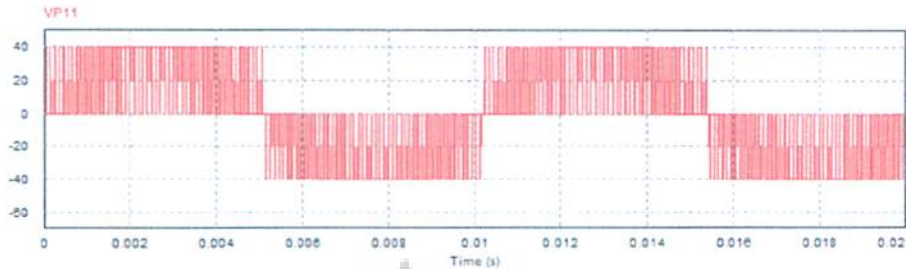
Gelombang sinusoida 50 Hz, 0.8 Vp sebagai sinyal pemodulasi (referensi) dikomparasikan dengan 3 gelombang segitiga 97.65625 Hz. Hasil dari setiap perbandingan tersebut akan menghasilkan sinyal Modulasi Lebar Pulsa Sinusoida (MLPS) dengan indeks modulasi 0,8. (Gambar 4.2)



Gambar 4.2. Salah satu Sinyal MLPS satu fasa simulasi Indeks modulasi 0.8

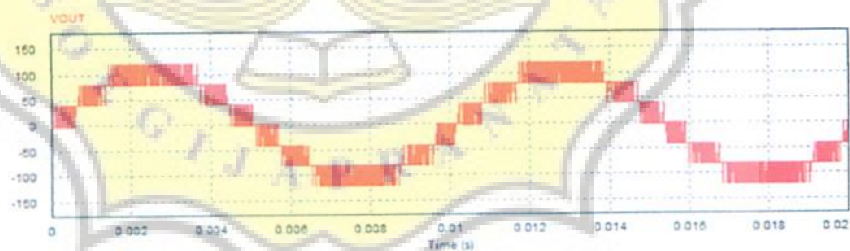
Sinyal ini kemudian digunakan untuk mensaklar tiap mosfet pada rangkaian inverter tersebut. Sumber tegangan DC 40 Volt akan dicacah oleh saklar daya mengikuti pola MLPS, sehingga keluaran inverter sebelum melalui induktor (L) mempunyai pola yang sama dengan sinyal picu MLPS. Hal ini menunjukkan bahwa simulasi multilevel inverter satu fasa jembatan penuh metode

unipolar telah berjalan dengan baik. Gambar 4.3 menunjukkan salah satu sinyal keluaran inverter sebelum filter (L)



Gambar 4.3. Sinyal keluaran multilevel inverter sebelum melewati inductor (L)

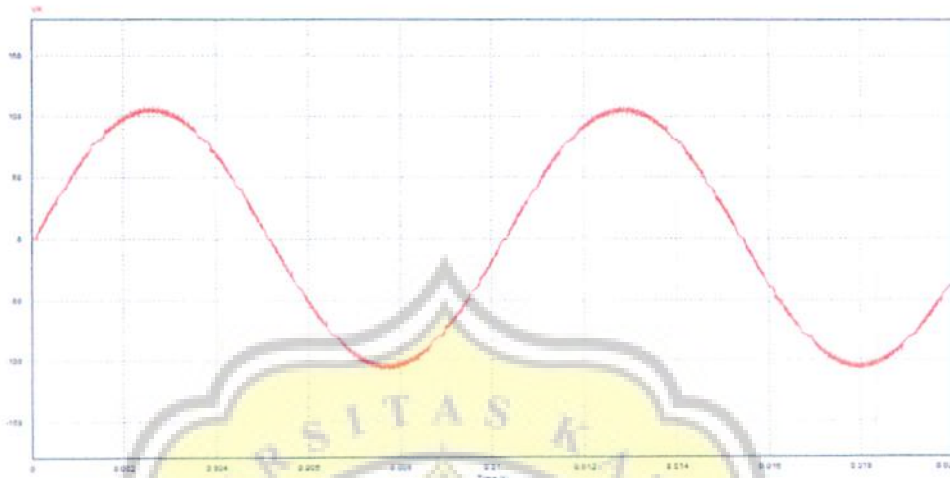
Keluaran dari tiap inverter tersebut saling berbeda pada tinggi tegangan yang dihasilkan. Sesuai persamaan multilevel inverter (1.1), maka keluaran pada inverter 1 akan menjadi pondasi awal untuk membentuk multilevel inverter, kemudian inverter 2 akan ditambahkan diatas tegangan keluaran inverter 1 dan inverter 3 menjadi penutup pembentukan multilevel tersebut. Gambar 4.4 menunjukkan hasil dari penggabungan sinyal keluaran.



Gambar 4.4 Hasil penggabungan sinyal keluaran inverter

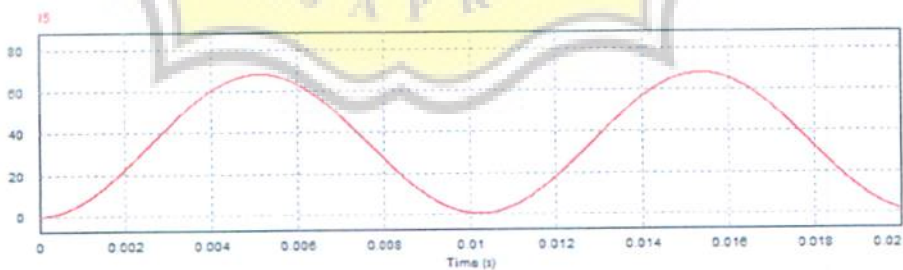
Setelah terbentuk sinyal multilevel tersebut maka sinyal diteruskan melewati Inductor (L) yang dipasang pada sisi beban (gambar 4.1). Inductor berfungsi sebagai filter (*Low Pass Filter*) sehingga sinyal keluaran dari inverter akan disaring dan akan membentuk pola sinusoida mendekati sinyal pemodulasi

(sinyal referensi). Sehingga sinyal keluaran inverter setelah melewati *inductor* disajikan dalam gambar 4.4



Gambar 4.5. Sinyal keluaran inverter setelah melewati *inductor* (L)

Beban yang ada adalah beban resistif, sehingga arus yang dirasakan oleh resistor mempunyai bentuk yang sama dengan bentuk gelombang tegangannya (Gambar 4.5). Tinggi rendahnya arus beban tergantung pada besar kecilnya beban. Semakin besar beban, arus yang mengalir akan semakin kecil, dan sebaliknya semakin kecil beban yang digunakan, arus yang mengalir akan semakin besar.



Gambar 4.6. Simulasi Arus beban

Ada beberapa komponen yang harus diperhatikan untuk mendapatkan sinyal MLPS tersebut. Dalam *software simulator* terdapat komponen pewaktuan yaitu *time step* yang berfungsi mengatur rentang pengambilan data dan kerja rangkaian simulasi. Untuk mendapatkan MLPS $\frac{1}{4}$ panjang gelombang maka *time step* dirubah sehingga mendapatkan data sebanyak 256 dalam file dengan format .txt. Setelah mendapatkan 256 data dalam file .txt, data ini kemudian diolah dengan menggunakan *Microsoft excel* untuk dijadikan data digital (biner). Setelah terbentuk data biner dari *Microsoft excel* kemudian data data tersebut dimasukkan kedalam program mikrokonroller ATMEGA8535. Program dibuat dengan menggunakan bahasa C dengan *software Code Vision AVR (CVAVR)*. Pemrograman ini dilakukan untuk mendapatkan sinyal SPWM dari mikrokontroller ATMEGA8535.

Pemrograman dilakukan dengan metode *lookup table* dengan memasukkan data data simulasi sebanyak 256 data dari $\frac{1}{4}$ panjang gelombang sinyal SPWM satu periode. Berikut adalah program yang digunakan

/******

Project : Multilevel Inverter

Version : 1.0

Date : 5/2/2013

Author : Grace Lordi

Chip type : ATmega8535

Program type : Application

AVR Core Clock frequency: 11.059200 MHz

Memory model : Small

External RAM size : 0

Data Stack size : 128

*****/

#include <mega8535.h>

#include <delay.h>

unsigned int a;

flash unsigned char data1[]={

1,

1,

1,

1,

1,

0,

0,

0,

0,

0,

0,

0,

0,

0,


```

flash unsigned char data4[]={
};
flash unsigned char data5[]={
};
flash unsigned char data6[]={
};
void main(void)
{
PORTB=0x00;
DDRB=0xFF;

while (1)
{
for(a=0;a<256;a++)
{
PORTB.0=data1[a];
PORTB.1=data2[a];
PORTB.2=data3[a];
PORTB.3=data4[a];
PORTB.6=data5[a];
PORTB.5=data6[a];
}
for(a=256;a>0;a--)
{
PORTB.0=data1[a];
PORTB.1=data2[a];
PORTB.2=data3[a];
PORTB.3=data4[a];
PORTB.6=data5[a];
PORTB.5=data6[a];
}
}

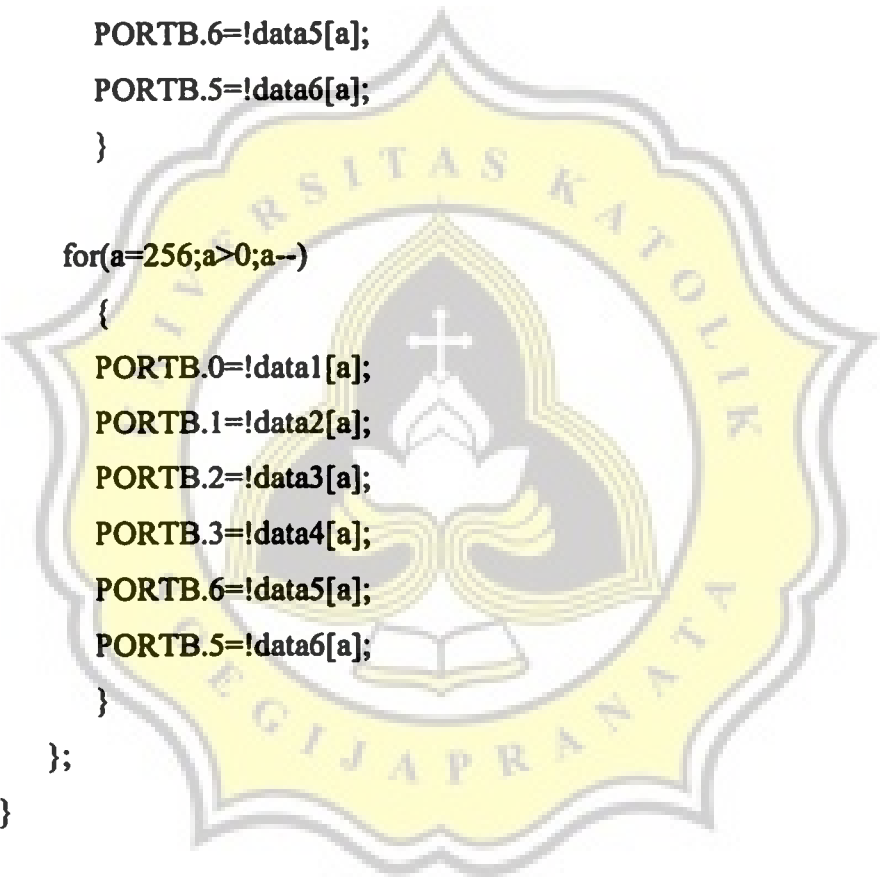
```

```

for(a=0;a<256;a++)
{
PORTB.0=!data1[a];
PORTB.1=!data2[a];
PORTB.2=!data3[a];
PORTB.3=!data4[a];
PORTB.6=!data5[a];
PORTB.5=!data6[a];
}

for(a=256;a>0;a--)
{
PORTB.0=!data1[a];
PORTB.1=!data2[a];
PORTB.2=!data3[a];
PORTB.3=!data4[a];
PORTB.6=!data5[a];
PORTB.5=!data6[a];
}
};
}

```



4.2 Analisa program

Inverter dikendalikan dengan metode unipolar sehingga membutuhkan 6 data MLPS. Semua data dimasukkan kedalam *lookup table* mikrokontroller. Pembentukan MLPS dilakukan dengan memanggil semua data yang telah tersimpan dalam *look up table*. Metode *look up table* adalah adalah suatu metode pengisian data kedalam mikrokontroller dengan mengambil data-data sampling

yang membentuk suatu gelombang. Gelombang yang digunakan dalam pengisian data tersebut adalah sinyal MLPS $\frac{1}{4}$ periode dari simulasi.

Sebelum membentuk program utama, terlebih dahulu dibuat suatu konstanta global yang berisi 256 data dari hasil simulasi. Dalam pemrograman ditulis:

```
Flash unsigned char data[]=
{
0      ;
0      ;
1      ;
1      ;
.      ;
.      ;
.      ;
// 256 data
};

unsigned int a;
```

Memori yang dimiliki ATMEGA 8535 sangatlah terbatas sehingga panjang instruksi yang dapat dituliskan kedalam mikrokontroler juga terbatas. Dalam penulisan konstanta diatas terdapat 2 variabel yaitu *flash* dan *unsigned char*. Variabel *flash* berfungsi sebagai instruksi memanggil data yang selamanya hanya bersifat sementara pada memori sehingga tidak memenuhi memori mikro, sedangkan *unsigned char* berfungsi sebagai variabel penyimpanan data. Konstanta tersebut ditulis sebanyak 6x sesuai jumlah sumber data MLPS yang digunakan.

Program utama yang digunakan dalam membentuk sinyal MLPS terdiri dari empat bagian yaitu masing masing akan membentuk sinyal SPWM $\frac{1}{4}$ periode.

- $\frac{1}{4}$ periode pertama

```

for(a=0;a<256;a++)
{
PORTB.0=data1[a];
PORTB.1=data2[a];
PORTB.2=data3[a];
PORTB.3=data4[a];
PORTB.6=data5[a];
PORTB.5=data6[a];
}

```

Mikrokontroler memanggil data yang tersimpan dalam memori dan menggeser data ke kanan sejumlah 256. Dari memori tersebut telah terbentuk sinyal MLPS $\frac{1}{4}$ periode. Jika diilustrasikan dalam bilangan adalah 00 – FF.

- $\frac{1}{4}$ periode ke dua

```

for(a=256;a>0;a--)
{
PORTB.0=data1[a];
PORTB.1=data2[a];
PORTB.2=data3[a];
PORTB.3=data4[a];
PORTB.6=data5[a];
PORTB.5=data6[a];
}

```

Mikrokontroler memanggil data yang tersimpan dalam memori dan menggeser data ke kiri sejumlah 256. Dari memori tersebut telah terbentuk sinyal MLPS $\frac{1}{4}$ periode. Jika diilustrasikan dalam bilangan adalah FF - 00.

- $\frac{1}{4}$ periode ke tiga

```
for(a=0;a<256;a++)  
{  
    PORTB.0=!data1[a];  
    PORTB.1=!data2[a];  
    PORTB.2=!data3[a];  
    PORTB.3=!data4[a];  
    PORTB.6=!data5[a];  
    PORTB.5=!data6[a];  
}
```

Mikrokontroler memanggil dan menginvers data yang tersimpan dalam memori kemudian menggeser data ke kanan sejumlah 256.

Dari memori tersebut telah terbentuk sinyal MLPS $\frac{1}{4}$ periode.

- $\frac{1}{4}$ periode ke empat

```
for(a=256;a>0;a--)  
{  
    PORTB.0=!data1[a];  
    PORTB.1=!data2[a];  
    PORTB.2=!data3[a];  
    PORTB.3=!data4[a];  
    PORTB.6=!data5[a];  
    PORTB.5=!data6[a];  
}
```

Mikrokontroler memanggil dan menginvers data yang tersimpan dalam memori kemudian menggeser data ke kiri sejumlah 256.

Setelah semua sinyal tersebut terbentuk, maka terbentuklah sebuah sinyal MLPS satu periode. Dengan demikian, sinyal MLPS satu periode dapat dibentuk dengan menggunakan sampling $\frac{1}{4}$ periode. Inilah yang kemudian disebut sebagai pemrograman $\frac{1}{4} \lambda$.

Selanjutnya sinyal MLPS dikeluarkan berdasarkan PORT yang telah diperintahkan untuk menuju rangkaian driver. Sinyal melewati TLP250 dimana sinyal hanya lewat saja tanpa ada perubahan apapun. Setelah melewati TLP250 dilanjutkan melewati IR2111 dan langsung bekerja memicu gebang GATE pada MOSFET.

Selain rangkain driver pada setiap aplikasi saklar daya antara gate dan source diberikan sebuah dioda Zener (dalam aplikasi ini menggunakan zener 18 volt). Dioda zener berfungsi untuk melindungi driver TLP 250 dan IR2111 dari umpan balik tegangan daya apabila terjadi hubung singkat pada rangkaian daya. Dipilih nilai 18 volt karena tegangan kerja maksimal driver sebesar 22 volt sehingga apabila terjadi umpan balik tegangan, zener bisa menahannya pada nilai aman terhadap kerusakan TLP 250.

4.3 Hasil Pengujian Menggunakan Osiloskop

Pengujian Alat multilevel inverter menggunakan parameter dibawah ini :

Tegangan Masukan	: 40Volt DC
Induktasi induktor	: 2 mH
Beban resistif	: 100Ohm
Frekuensi pensaklaran	: 5KHz

Gambar 4.7. Parameter Pengujian

Masing-masing inverter diberikan tegangan masukan sebesar 40 volt DC.

Tiap sisi keluaran inverter tersebut dihubungkan secara seri dan pada sisi keluaran akhir dihubungkan pada solenoid yang berfungsi sebagai induktor dengan nilai 2mH dan beban resistif sebesar 100 ohm. Setiap keluaran dari inverter menghasilkan sinyal yang sesuai dengan MLPS yang telah diberikan seperti pada gambar (4.3). hasil gabungan ketiga sinyal keluaran tersebut pada pengujian laboratorium ditunjukkan (gambar 4.6)



Gambar 4.8. Sinyal tegangan keluaran inverter

Gabungan tegangan keluaran inverter tersebut membentuk anak tangga yang berundak. Sinyal tersebut menjadi cetakan bagi sinyal AC yang terbentuk pada gambar 4.8



Gambar 4.9. Hasil pengujian Tegangan pada R dan Tegangan pada L

Dari gambar diatas terlihat sinyal sinus bolak-balik (AC) yang terbentuk sefasa dan sesuai dengan sinyal MLPS yang terbentuk. Sinyal tegangan keluaran pada sisi beban L dan sisi beban R terbentuk sama persis dengan tegangan keluaran pada simulasi komputer. Teknik pemrograman multilevel inverter menggunakan metode lookup table $\frac{1}{4}$ periode terbukti mampu mengendalikan inverter tersebut. Tegangan dan arus bolak-balik (AC) yang dihasilkan mempunyai kualitas yang baik. Pada gambar (5.0) ditunjukkan tegangan dan arus pada sisi beban.



Gambar 5.0. Hasil pengujian tegangan dan arus keluaran inverter

Dengan parameter diatas dihasilkan tegangan keluaran inverter sebesar 100v. Dalam Tugas Akhir ini tidak menggunakan sensor arus, sehingga karakteristik dari arus yang mengalir tidak dapat dianalisa.