

BAB IV

HASIL PENGUJIAN DAN ANALISA

4.1 Pengujian Rangkaian dengan Simulator

Suatu *software power simulator* menyediakan fitur – fitur tentang teknik elektronika. Sebelum implementasi dilakukan, simulasi digunakan untuk mencari nilai beban dan perhitungan. Program *power simulator* dapat memberikan data – data yang digunakan untuk membuat program SPWM dalam mikrokontroler.

Data pointer yang termuat dalam file *.Txt* perlu diproses dan diakuisisi dengan menggunakan *Microsoft excel* untuk dijadikan data digital (*biner*) sebelum digunakan dalam pemrograman.

Untuk pengujian rangkaian, dilakukan dengan menggunakan parameter sebagai berikut:

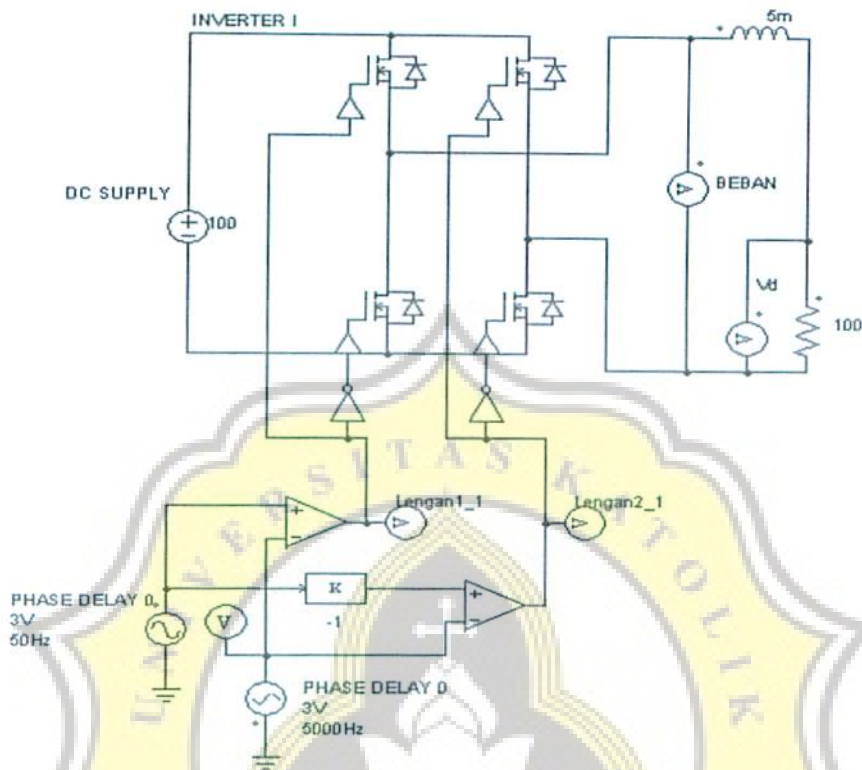
Tegangan DC = 100 Volt

Induktansi = 5 mH

Beban resistif = 100 Ohm

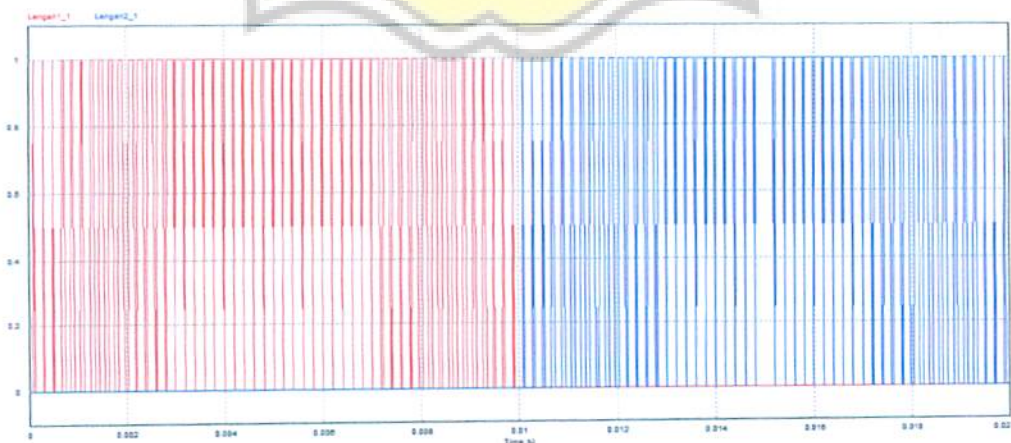
Frekuensi *switching* = 5000 Hz

4.1.1 Simulasi inverter mandiri



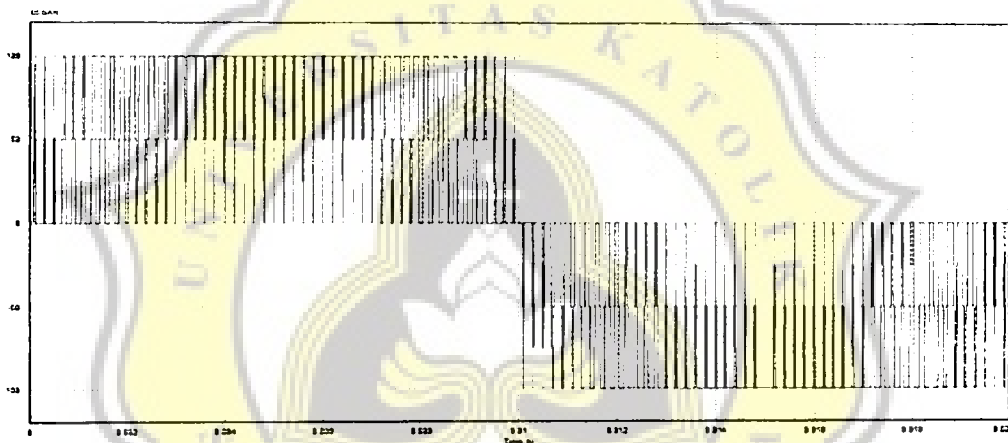
Gambar 4.1 Inverter satu fasa

Inverter I menggunakan gelombang informasi sinusoida 50 Hz, 3 Vpp; gelombang segitiga 5000 Hz, 3 Vpp yang tidak tergeser fasanya sebagai sinyal pembawa menghasilkan sinyal Modulasi Lebar Pulsa Sinusoida (SPWM).



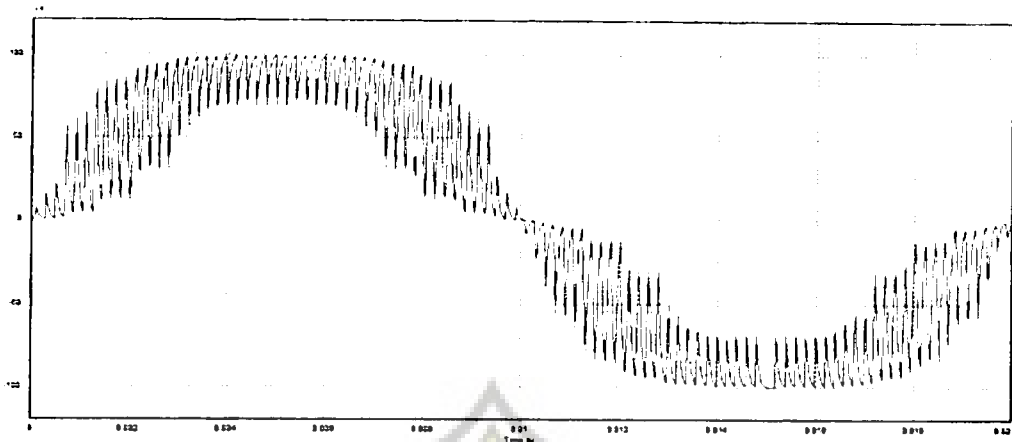
Gambar 4.2 Sinyal SPWM Inverter I

SPWM ini digunakan sebagai sinyal picu saklar daya (S1, S2, S3, dan S4). Sinyal SPWM keluaran dari komparator akan menuju *driver* saklar daya sebelum mengonduksi MOSFET. Hal ini bertujuan supaya kontrol tidak mengalami kerusakan jika terjadi lonjakan arus yang besar dari saklar daya. Sumber tegangan DC 100 Volt akan dicacah oleh saklar daya mengikuti pola SPWM, sehingga keluaran inverter sebelum melalui induktor (L) mempunyai pola yang sama dengan sinyal picu SPWM.



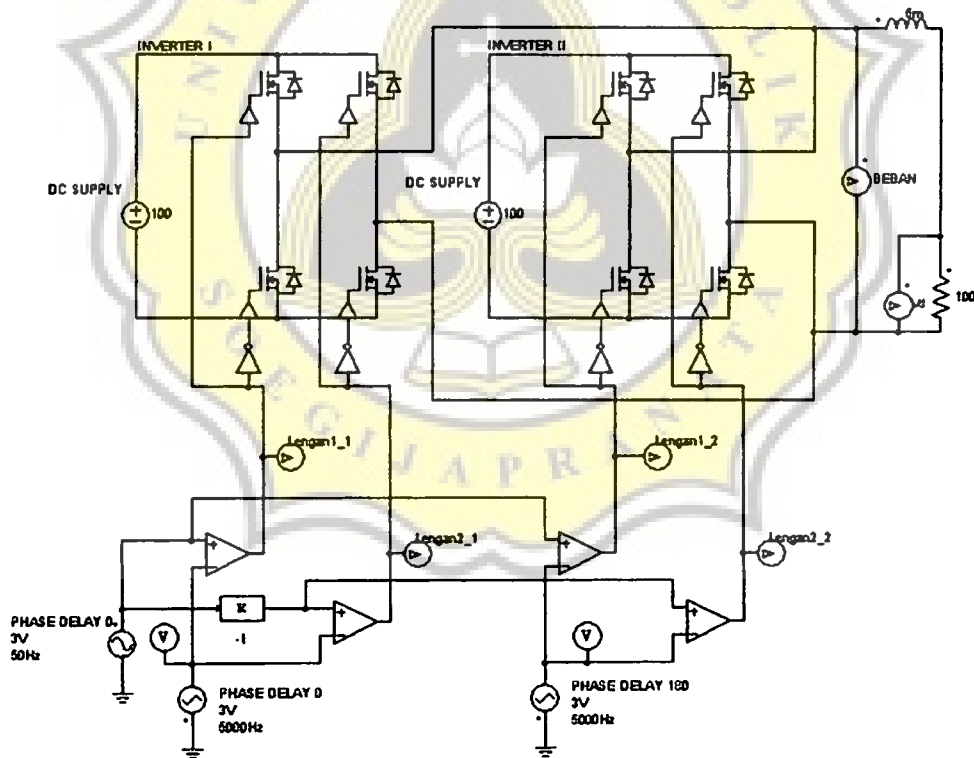
Gambar 4.3 Sinyal keluaran inverter sebelum melewati *inductor* (L)

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



Gambar 4.4 Sinyal keluaran inverter satu fasa setelah melewati *inductor* (L)

4.1.2 Simulasi inverter paralel dua tergeser 180°



Gambar 4.5 simulasi paralel dua inverter.

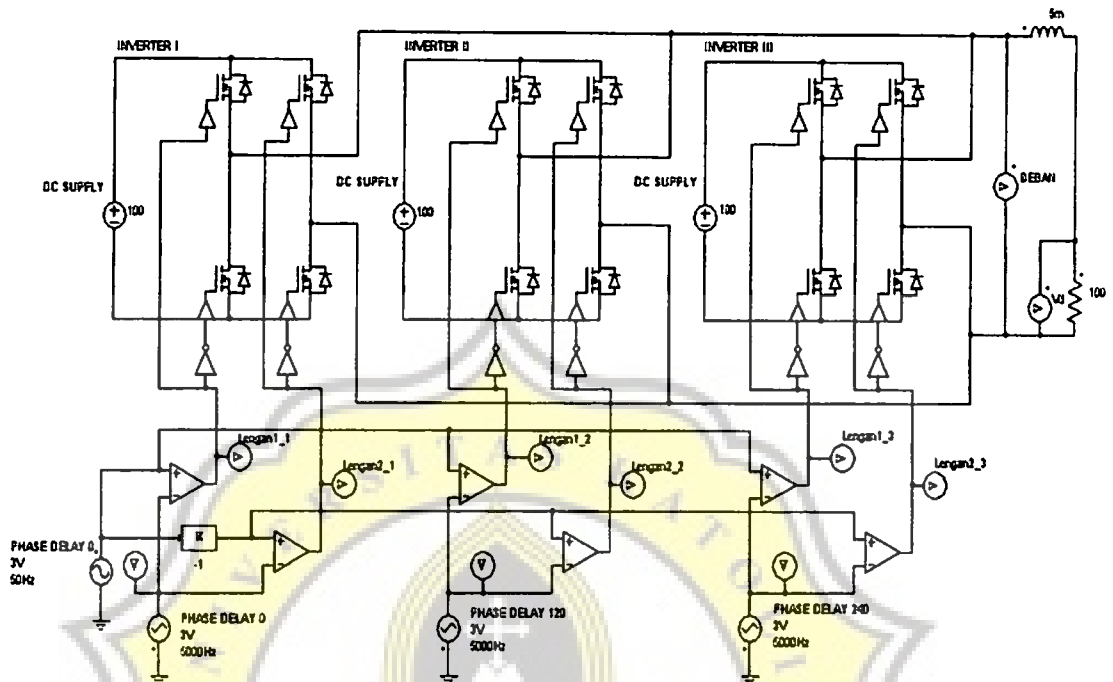
Inverter I menggunakan gelombang informasi sinusoida 50 Hz, 3 Vpp; gelombang segitiga 5000 Hz, 3 Vpp yang tidak tergeser fasanya sebagai sinyal

pembawa menghasilkan sinyal Modulasi Lebar Pulsa Sinusoida (SPWM). Pada inverter II fasa gelombang segitiga bergeser sebesar 180° .



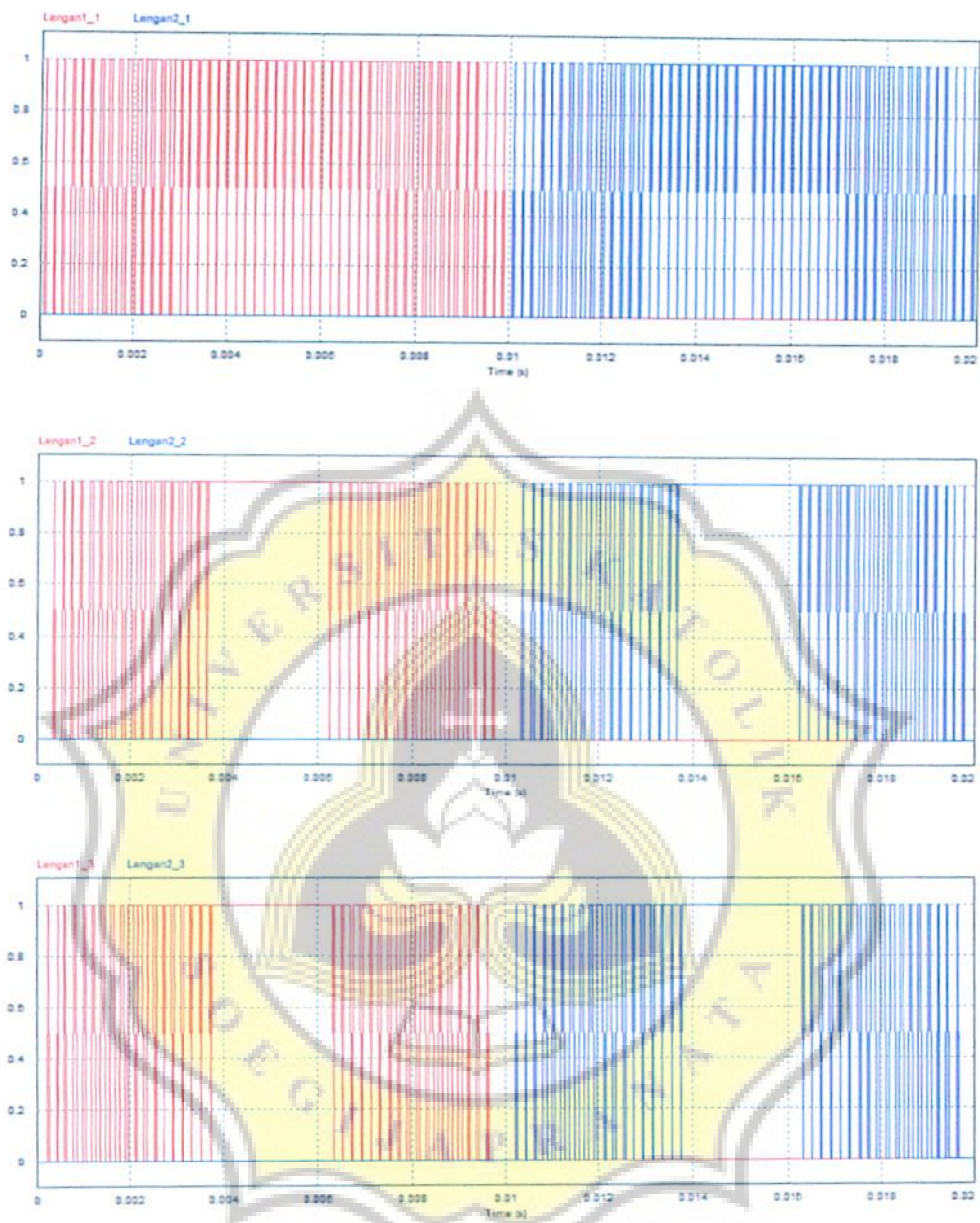
Gambar 4.6 Sinyal SPWM Inverter I & Inverter II

4.1.3 Simulasi Inverter Paralel tiga saling tergeser 120°



Gambar 4.7 simulasi paralel tiga inverter

Inverter I menggunakan gelombang informasi sinusoida 50 Hz, 3 Vpp; gelombang segitiga 5000 Hz, 3 Vpp yang tidak tergeser fasanya sebagai sinyal pembawa menghasilkan sinyal Modulasi Lebar Pulsa Sinusoida (SPWM). Pada inverter II fasa gelombang segitiga bergeser sebesar 120° dan 240° pada inverter III.



Gambar 4.8 Sinyal SPWM Inverter I, Inverter II & Inverter III

```

Chip type           : ATmega8535
Program type        : Application
Clock frequency     : 11.059200 MHz
Memory model        : Small
External SRAM size  : 0
Data Stack size     : 128

```

```

*****/

```

```

#include <mega8535.h>

```

```

#include <delay.h>

```

```

#define TOMBOL1 PIND.0

```

```

#define TOMBOL2 PIND.2

```

```

#define TOMBOL3 PIND.4

```

```

#define TOMBOL4 PIND.6

```

```

//define led1 PORTA.0

```

```

//define led2 PORTA.1

```

```

//define led3 PORTA.2

```

```

//define led4 PORTA.3

```

```

unsigned int a;

```

```

//data 0 derajat

```

```

flash unsigned char data1[]=

```

```

{
1 ;
1 ;
1 ;
1 ;
1 ;
1 ;
0 ;
.
.
.

```

```

// 256 data

```

```

};
flash unsigned char data2[]=

```

```

{
1 ;
1 ;
1 ;
1 ;
0 ;
0 ;
.
.

```



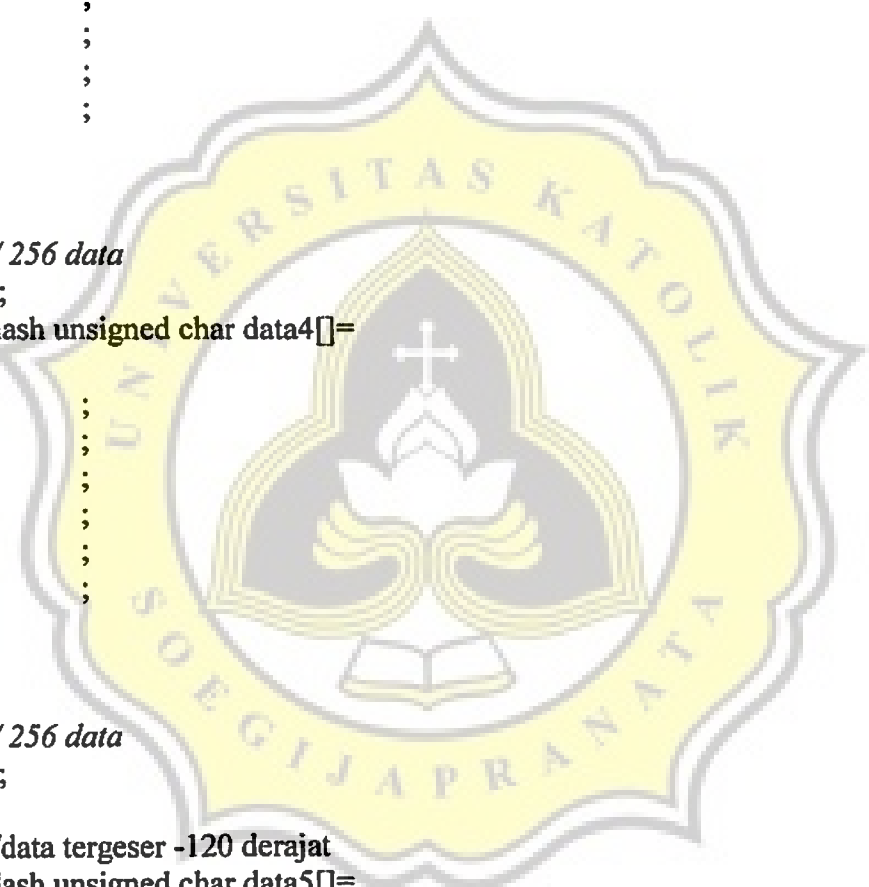
```

.
// 256 data
};

//data tergeser 120 derajat
flash unsigned char data3[]=
{
0      ;
1      ;
1      ;
1      ;
1      ;
1      ;
1      ;
.
.
.
// 256 data
};
flash unsigned char data4[]=
{
0      ;
1      ;
1      ;
1      ;
1      ;
1      ;
1      ;
.
.
.
// 256 data
};

//data tergeser -120 derajat
flash unsigned char data5[]=
{
0      ;
0      ;
0      ;
0      ;
0      ;
0      ;
.
.
.
// 256 data
};

```



```

flash unsigned char data6[]=
{
0      ;
0      ;
0      ;
0      ;
0      ;
0      ;
.
.
.
// 256 data
};

//data tergeser 180 derajat
flash unsigned char data7[]=
{
0      ;
0      ;
0      ;
0      ;
1      ;
1      ;
.
.
.
// 256 data
};

flash unsigned char data8[]=
{
0      ;
0      ;
0      ;
0      ;
0      ;
1      ;
.
.
.
// 256 data
};

```

void mode1()

```

{
  for (a=0;a<255;a++)
  {
    PORTA.0=data1[a];
    PORTA.1=data2[a];
    delay_us(19);
  };

  for (a=256;a>0;a--)
  {
    PORTA.0=(data1[a]);
    PORTA.1=(data2[a]);
    delay_us(19);
  };

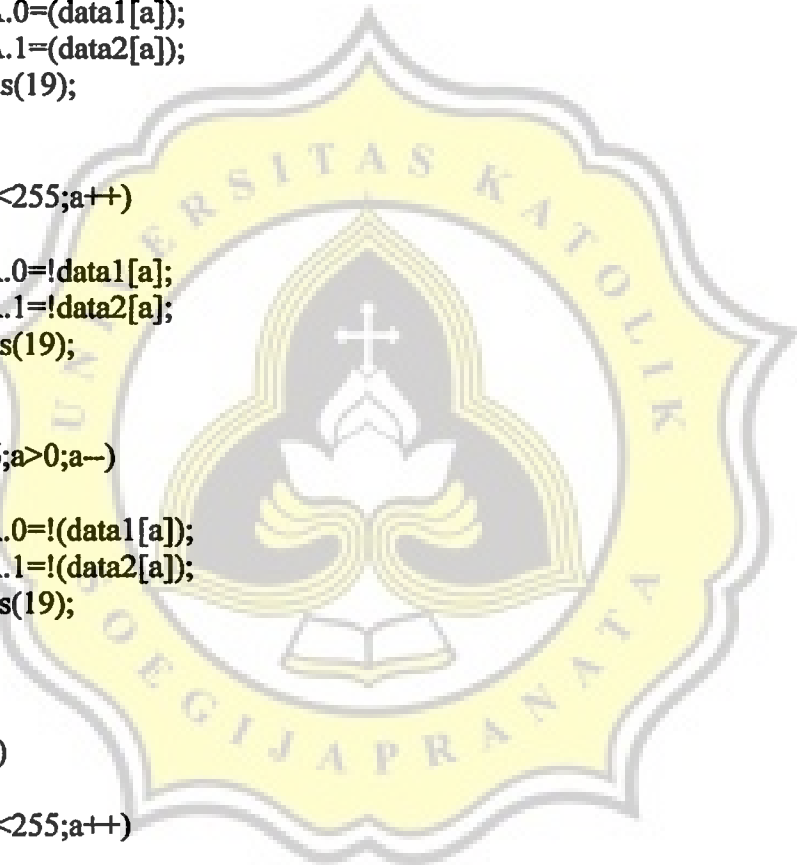
  for (a=0;a<255;a++)
  {
    PORTA.0=!data1[a];
    PORTA.1=!data2[a];
    delay_us(19);
  };

  for (a=256;a>0;a--)
  {
    PORTA.0!=(data1[a]);
    PORTA.1!=(data2[a]);
    delay_us(19);
  };
}

void mode2()
{
  for (a=0;a<255;a++)
  {
    PORTA.0=data7[a];
    PORTA.1=data8[a];
    delay_us(19);
  };

  for (a=256;a>0;a--)
  {
    PORTA.0=(data7[a]);
    PORTA.1=(data8[a]);
    delay_us(19);
  };
}

```



```

for (a=0;a<255;a++)
{
    PORTA.0=!data7[a];
    PORTA.1=!data8[a];
    delay_us(19);
};

for (a=256;a>0;a--)
{
    PORTA.0!=(data7[a]);
    PORTA.1!=(data8[a]);
    delay_us(19);
};
}

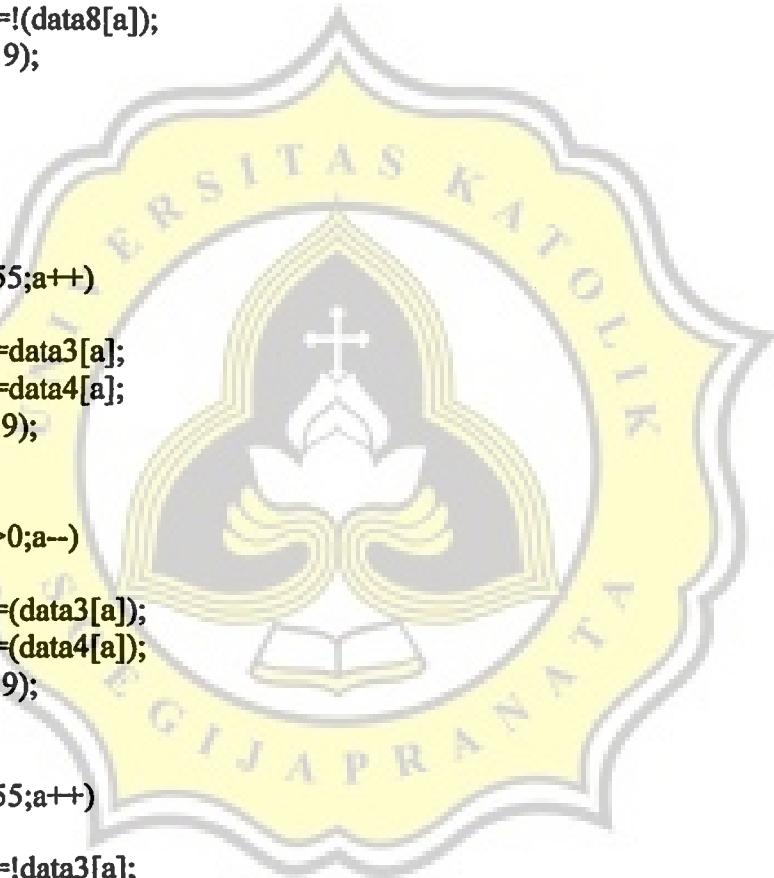
void mode3()
{
    for (a=0;a<255;a++)
    {
        PORTA.0=data3[a];
        PORTA.1=data4[a];
        delay_us(19);
    };

    for (a=256;a>0;a--)
    {
        PORTA.0=(data3[a]);
        PORTA.1=(data4[a]);
        delay_us(19);
    };

    for (a=0;a<255;a++)
    {
        PORTA.0=!data3[a];
        PORTA.1=!data4[a];
        delay_us(19);
    };

    for (a=256;a>0;a--)
    {
        PORTA.0!=(data3[a]);
        PORTA.1!=(data4[a]);
        delay_us(19);
    };
}

```



```

void mode4()
{
    for (a=0;a<255;a++)
    {
        PORTA.0=data5[a];
        PORTA.1=data6[a];
        delay_us(19);
    };

    for (a=256;a>0;a--)
    {
        PORTA.0=(data5[a]);
        PORTA.1=(data6[a]);
        delay_us(19);
    };

    for (a=0;a<255;a++)
    {
        PORTA.0=!data5[a];
        PORTA.1=!data6[a];
        delay_us(19);
    };

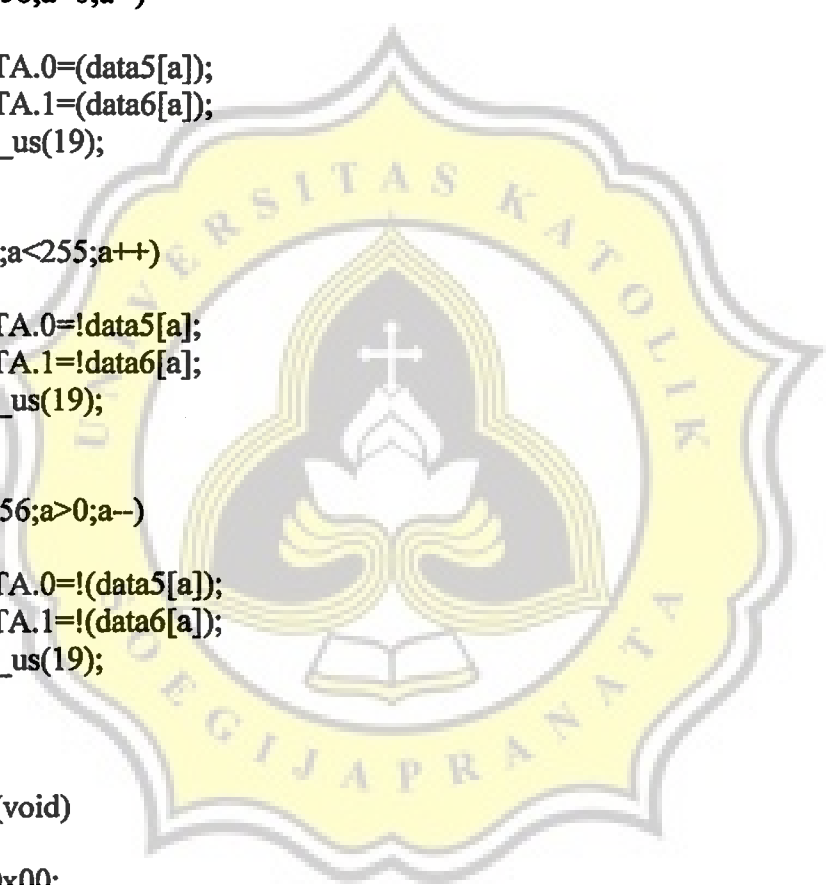
    for (a=256;a>0;a--)
    {
        PORTA.0!=(data5[a]);
        PORTA.1!=(data6[a]);
        delay_us(19);
    };
}

void main(void)
{
    PORTA=0x00;
    DDRA=0xFF;

    PORTD=0xFF;
    DDRD=0x00;

    while(1)
    {
        if(!TOMBOL1)
        {
            mode1();
        }
    }
}

```



```

if(!TOMBOL2)
{
mode2();
}

if(!TOMBOL3)
{
mode3();
}

if(!TOMBOL4)
{
mode4();
}

else
{
PORTA=0x00;
}
};
}

```

4.3 Analisa program

Sinyal SPWM dari mikrokontroler diperoleh dengan mengeksekusi program dalam memori. Proses pembentukan sinyal SPWM dilakukan dengan menggunakan menggunakan metode *look up table* dari rangkaian simulasi PWM inverter dari *software power simulator*. Metode *look up table* adalah adalah suatu metode pengisian data ke dalam mikrokontroler dengan mengambil data-data sampling yang membentuk suatu gelombang. Gelombang yang digunakan dalam pengisian data tersebut adalah sinyal SPWM $\frac{1}{4} \lambda$.

Sebelum membentuk program utama, terlebih dahulu dibuat delapan buah *data flash* yang berisi 256 data dari hasil simulasi. Dalam pemrograman ditulis:


```

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

```

Perlu diketahui delapan data *flash* tersebut adalah data dari masing - masing fasa yang tergeser, yaitu sebagai berikut:

- Data fasa tidak tergeser; data1 & data2
- Data tergeser 120° ; data3 & data4
- Data tergeser -120° ; data5 & data6
- Data tergeser 180° ; data7 & data8

Program utama yang digunakan dalam sistem terdiri dari empat mode yang nantinya akan dipanggil menggunakan tombol saklar eksternal.

- **Inverter fasa tidak tergeser**

```

void mode1()
{
for (a=0;a<255;a++)
{
PORTA.0=data1[a];
PORTA.1=data2[a];
delay_us(19);
};

for (a=256;a>0;a--)
{
PORTA.0=(data1[a]);
PORTA.1=(data2[a]);
delay_us(19);
};
}

```

```

for (a=0;a<255;a++)
{
    PORTA.0=!data1[a];
    PORTA.1=!data2[a];
    delay_us(19);
};

for (a=256;a>0;a--)
{
    PORTA.0!=(data1[a]);
    PORTA.1!=(data2[a]);
    delay_us(19);
};
}

```

Mikrokontroler memanggil data yang tersimpan dalam memori dan menggeser data ke kanan sejumlah 256. Dari memori tersebut telah terbentuk sinyal SPWM $\frac{1}{4}$ periode. Data kemudian dimanipulasi seperti program diatas untuk membuat sinus yang tidak tergeser.

- **Inverter fasa tergeser 180^0**

```

void mode2()
{
    for (a=0;a<255;a++)
    {
        PORTA.0=data7[a];
        PORTA.1=data8[a];
        delay_us(19);
    };

    for (a=256;a>0;a--)
    {
        PORTA.0=(data7[a]);
        PORTA.1=(data8[a]);
        delay_us(19);
    };

    for (a=0;a<255;a++)
    {
        PORTA.0=!data7[a];
    }
}

```

```

        PORTA.1=!data8[a];
        delay_us(19);
    };

    for (a=256;a>0;a--)
    {
        PORTA.0!=(data7[a]);
        PORTA.1!=(data8[a]);
        delay_us(19);
    };
}

```

Mikrokontroler memanggil data yang tersimpan dalam memori dan menggeser data ke kanan sejumlah 256. Dari memori tersebut telah terbentuk sinyal SPWM $\frac{1}{4}$ periode. Data kemudian dimanipulasi seperti program diatas untuk membuat sinus yang tergeser 180° .

- **Inverter tergeser 120°**

```

void mode3()
{
    for (a=0;a<255;a++)
    {
        PORTA.0=data3[a];
        PORTA.1=data4[a];
        delay_us(19);
    };

    for (a=256;a>0;a--)
    {
        PORTA.0=(data3[a]);
        PORTA.1=(data4[a]);
        delay_us(19);
    };

    for (a=0;a<255;a++)
    {
        PORTA.0=!data3[a];
        PORTA.1=!data4[a];
        delay_us(19);
    };

    for (a=256;a>0;a--)
    {

```

```

        PORTA.0!=(data3[a]);
        PORTA.1!=(data4[a]);
        delay_us(19);
    };
}

```

Mikrokontroler memanggil data yang tersimpan dalam memori dan menggeser data ke kanan sejumlah 256. Dari memori tersebut telah terbentuk sinyal SPWM $\frac{1}{4}$ periode. Data kemudian dimanipulasi seperti program di atas untuk membuat sinus yang tergeser 120° .

- **Inverter tergeser -120°**

```

void mode4()
{
    for (a=0;a<255;a++)
    {
        PORTA.0=data5[a];
        PORTA.1=data6[a];
        delay_us(19);
    };

    for (a=256;a>0;a--)
    {
        PORTA.0=(data5[a]);
        PORTA.1=(data6[a]);
        delay_us(19);
    };

    for (a=0;a<255;a++)
    {
        PORTA.0=!data5[a];
        PORTA.1=!data6[a];
        delay_us(19);
    };

    for (a=256;a>0;a--)
    {
        PORTA.0!=(data5[a]);
        PORTA.1!=(data6[a]);
        delay_us(19);
    };
}

```

Mikrokontroler memanggil data yang tersimpan dalam memori dan menggeser data ke kanan sejumlah 256. Dari memori tersebut telah terbentuk sinyal SPWM $\frac{1}{4}$ periode. Data kemudian dimanipulasi seperti program di atas untuk membuat sinus yang tergeser -120° .

Ke empat sinyal tersebut kemudian dikombinasikan dengan program mode untuk mengatur keluaran tidak tergeser, tergeser 180° , tergeser 120° , dan tergeser 120°

```
void main(void)
{
  PORTA=0x00;
  DDRA=0xFF;
  PORTD=0xFF;
  DDRD=0x00;

  while(1)
  {
    if(!TOMBOL1)
    {
      mode1();
    }

    if(!TOMBOL2)
    {
      mode2();
    }

    if(!TOMBOL3)
    {
      mode3();
    }

    if(!TOMBOL4)
    {
      mode4();
    }
  }
}
```

```
else
{
PORTA=0x00;
}
};
}
```

Selanjutnya, sinyal SPWM keluaran dari PORT mikrokontroler ATMEGA8535 masuk ke rangkaian *driver*. Dengan memilih mode sesuai dengan aturan paralel saklar akan daya bekerja menghasilkan tegangan keluaran.

4.4 Hasil Pengukuran Menggunakan Osiloskop

Sinyal kendali dibangkitkan dengan menggunakan metode *look up table* dari simulasi pembentukan sinyal SPWM menggunakan *software power simulator*. Data yang diperoleh diolah menggunakan CVAVR yang kemudian di *download* ke dalam memori mikrokontroler ATMEGA8535. Disediakan tiga buah mikrokontroler karena keterbatasan kemampuan memori untuk mengendalikan tiga buah inverter secara bersamaan.

4.4.1 Pengujian Rangkaian Kontrol

Pengujian *driver* dilakukan dengan tujuan untuk melihat pola *switching* yang telah diprogram ke dalam mikrokontroler apakah sudah sesuai dengan simulasi yang dilakukan.



Gambar 4.9 Sinyal SPWM satu fasa keluaran mikrokontroler

Sinyal keluaran mikrokontroler mempunyai pola pensaklaran (SPWM) yang sama dengan sinyal SPWM simulasi. Setelah mempunyai bentuk yang sama dengan SPWM simulasi, sinyal SPWM tersebut akan melewati opto coupler TLP 250 sebelum masuk ke dalam driver IR 2111 dan mengonduksi saklar MOSFET (saklar daya).

4.4.2 Pengujian Inverter

Berikut adalah hasil keluaran tegangan inverter diukur pada frekuensi osiloskop 100 Hz, volt/div diatur pada *range* 2-10 V, parameter yang digunakan dalam rangkaian daya adalah sebagai berikut :

- Beban yang digunakan adalah resistor 50 Ω
- Induktor (L) berupa tiga buah trafo identik sebesar 2 mH.

- Tiga buah catu daya DC pada masing-masing inverter diatur pada nilai tegangan 40 volt.

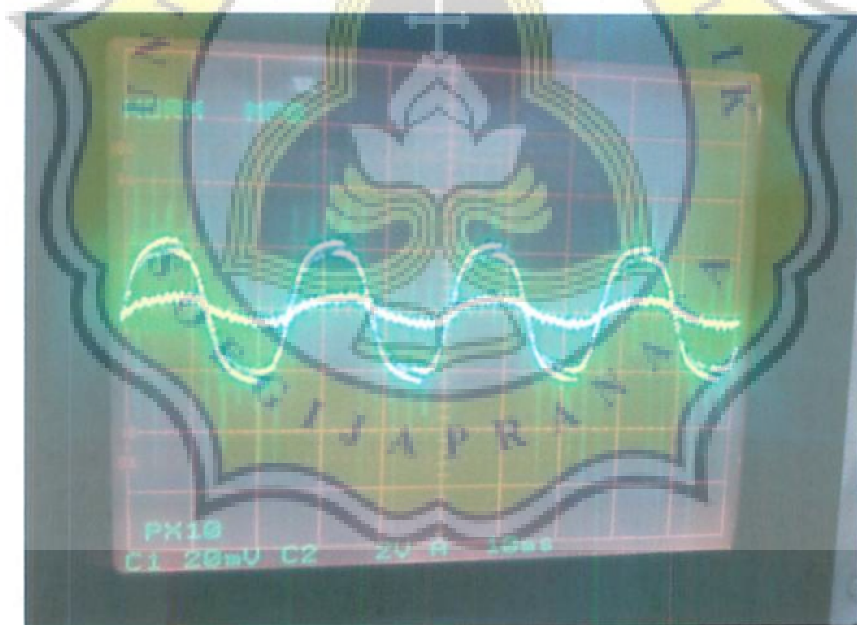
Pengujian pertama, masing-masing inverter berkerja secara mandiri. Yaitu inverter belum dihubungkan secara paralel dengan pembebanan seri R dan L. Inverter dapat bekerja dengan baik.



Gambar 4.10 Arus keluaran inverter 1

Pada pengujian pertama arus hanya mengalir pada l_1 sehingga dari gambar (4.10) diperoleh persamaan seperti berikut :

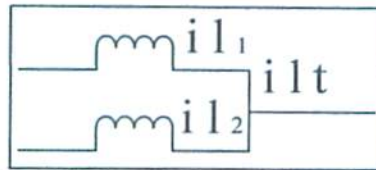
$$P = V \cdot i l_1 (\cos\theta)$$



Gambar 4.11 Hasil pengujian tegangan dan arus keluaran inverter mandiri

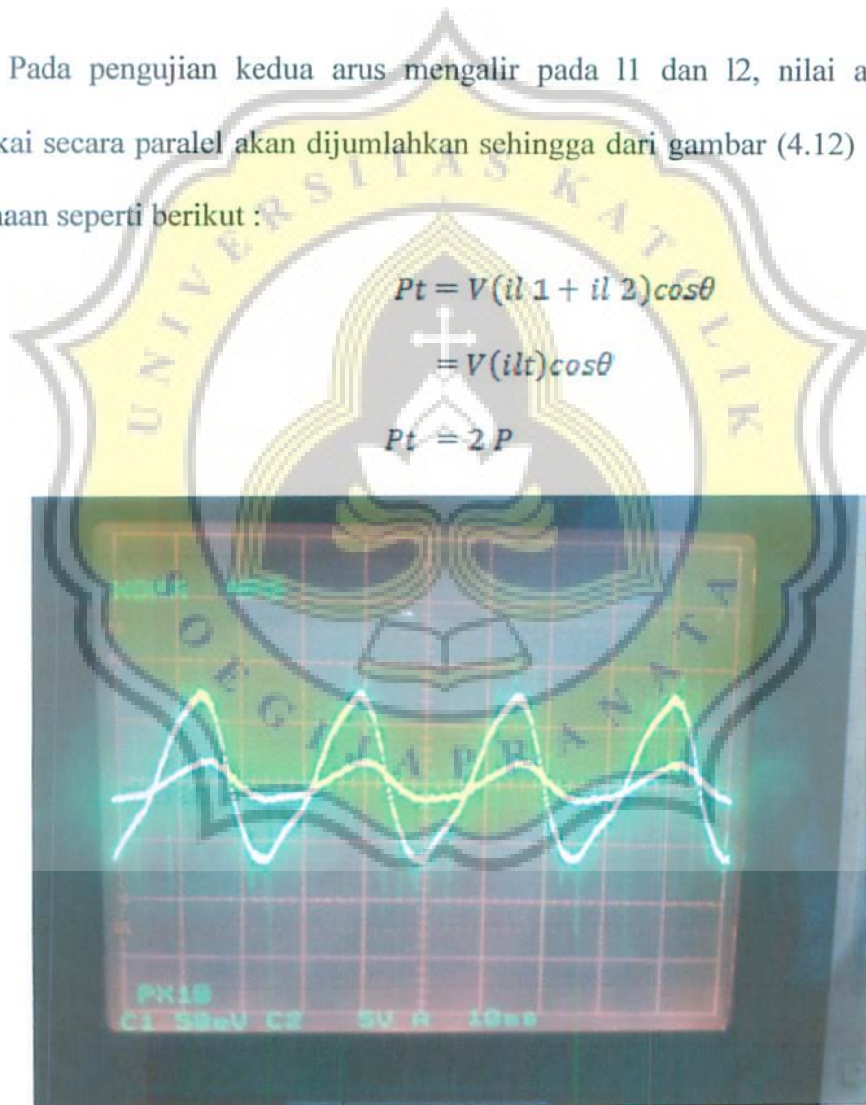
Arus keluaran sefasa dan tidak bergeser namun terlihat bentuk tegangan belum menyerupai sinus yang baik dan riak pada gelombang masih terlalu tebal.

Pengujian kedua, dua buah inverter dihubungkan secara paralel dengan pembebanan seri R dan L. Paralel dua inverter dapat bekerja dengan baik.



Gambar 4.12 Arus keluaran inverter paralel 2

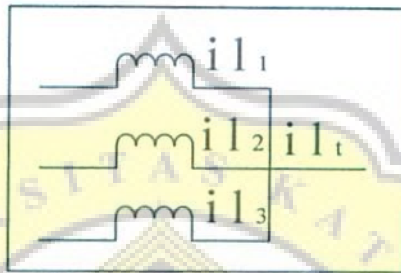
Pada pengujian kedua arus mengalir pada i_{l1} dan i_{l2} , nilai arus yang dirangkai secara paralel akan dijumlahkan sehingga dari gambar (4.12) diperoleh persamaan seperti berikut :



Gambar 4.13 Hasil pengujian tegangan dan arus keluaran paralel dua inverter

Arus keluaran gabungan sefasa dan tidak bergeser, terlihat bentuk tegangan mendekati sinus yang baik. Riak pada gelombang berkurang, nilai daya keluaran lebih baik dibandingkan dengan nilai tegangan inverter mandiri.

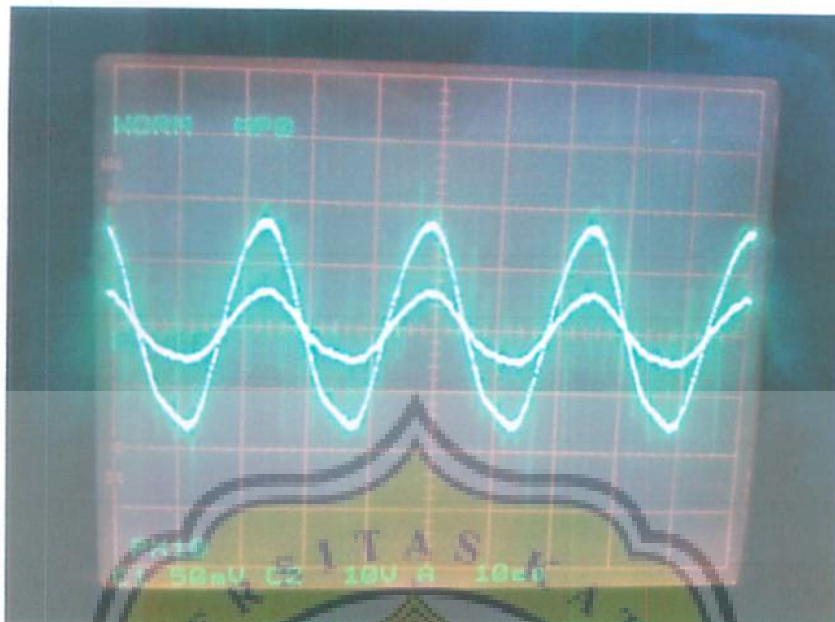
Pengujian ketiga, tiga buah inverter dihubungkan secara paralel dengan pembebanan seri R dan L. Paralel tiga inverter dapat bekerja dengan baik.



Gambar 4.14 Arus keluaran inverter paralel 3

Pada pengujian ketiga arus mengalir pada i_{l1} , i_{l2} , dan i_{l3} , nilai arus yang dirangkai secara paralel akan dijumlahkan sehingga dari gambar (4.16) diperoleh persamaan seperti berikut :

$$\begin{aligned}
 P_t &= V(i_{l1} + i_{l2} + i_{l3})\cos\theta \\
 &= V(3i_l)\cos\theta \\
 P_t &= 3P
 \end{aligned}$$



Gambar 4.15 Hasil pengujian tegangan dan arus keluaran paralel tiga inverter

Arus keluaran gabungan sefasa dan tidak bergeser, terlihat bentuk tegangan menyerupai sinus. Riak pada gelombang berkurang, nilai daya keluaran lebih baik dibandingkan dengan paralel dua inverter.