

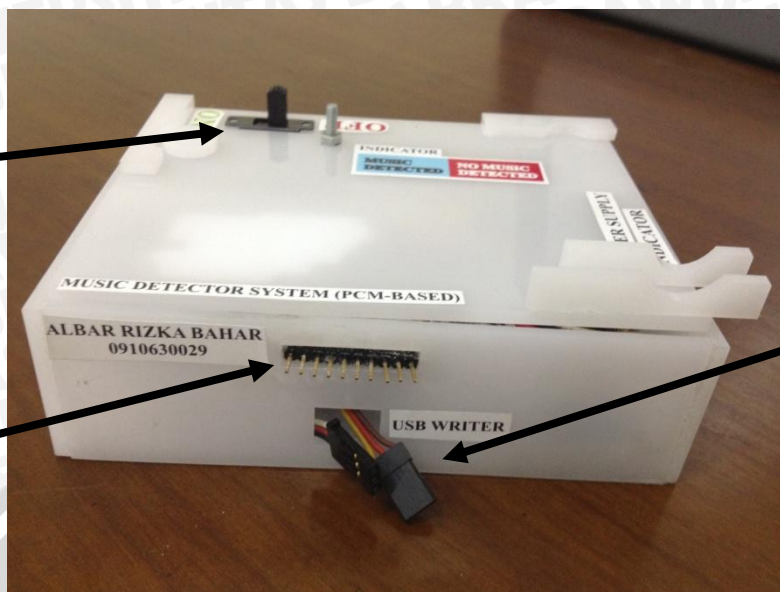


LAMPIRAN I

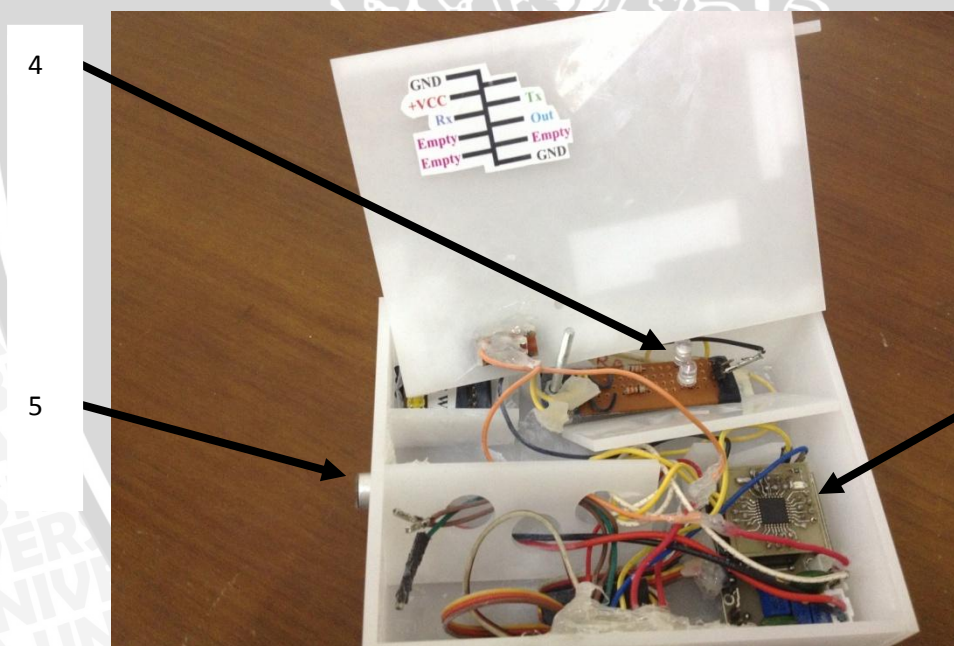
---

FOTO ALAT





**Sistem Pendeteksi Musik yang mengimplementasikan prinsip kerja PCM (pulse code modulation)**



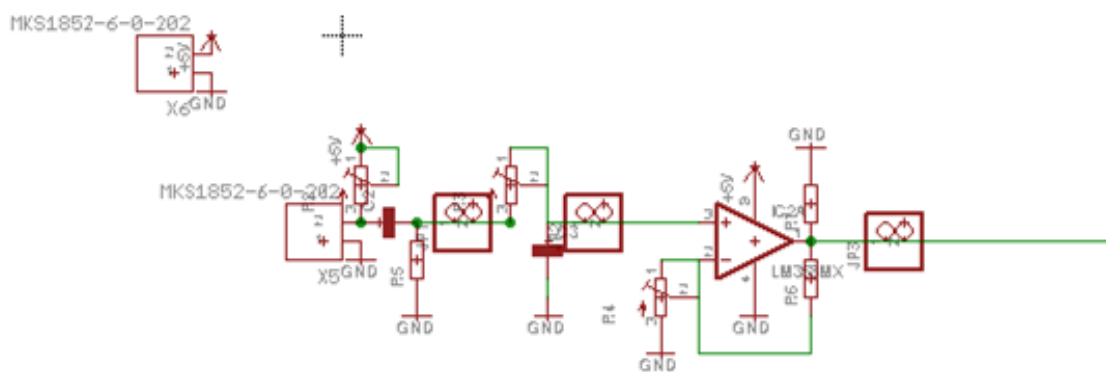
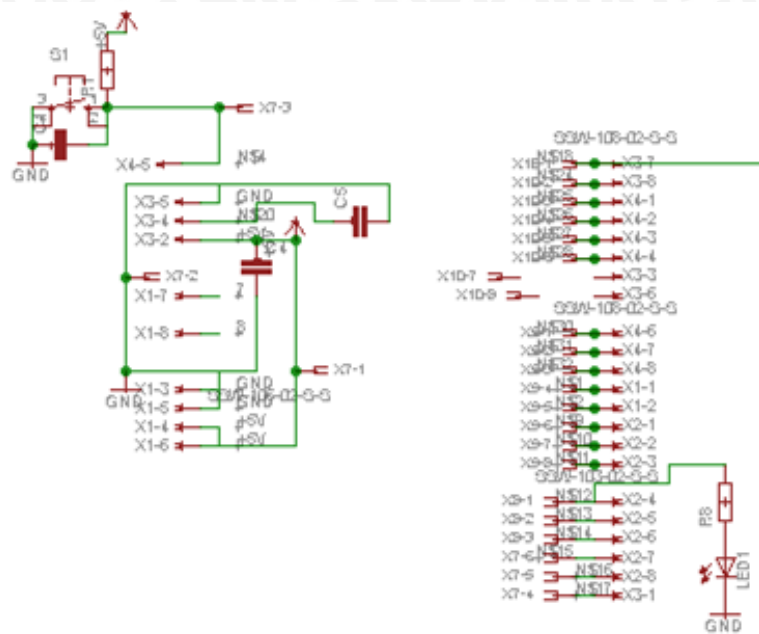
**Keterangan :**

- |                                |                     |
|--------------------------------|---------------------|
| 1. Saklar                      | 5. Microphone       |
| 2. Pin Interface               | 6. PCM system board |
| 3. Kabel konektor k USB writer |                     |
| 4. LED indikator               |                     |

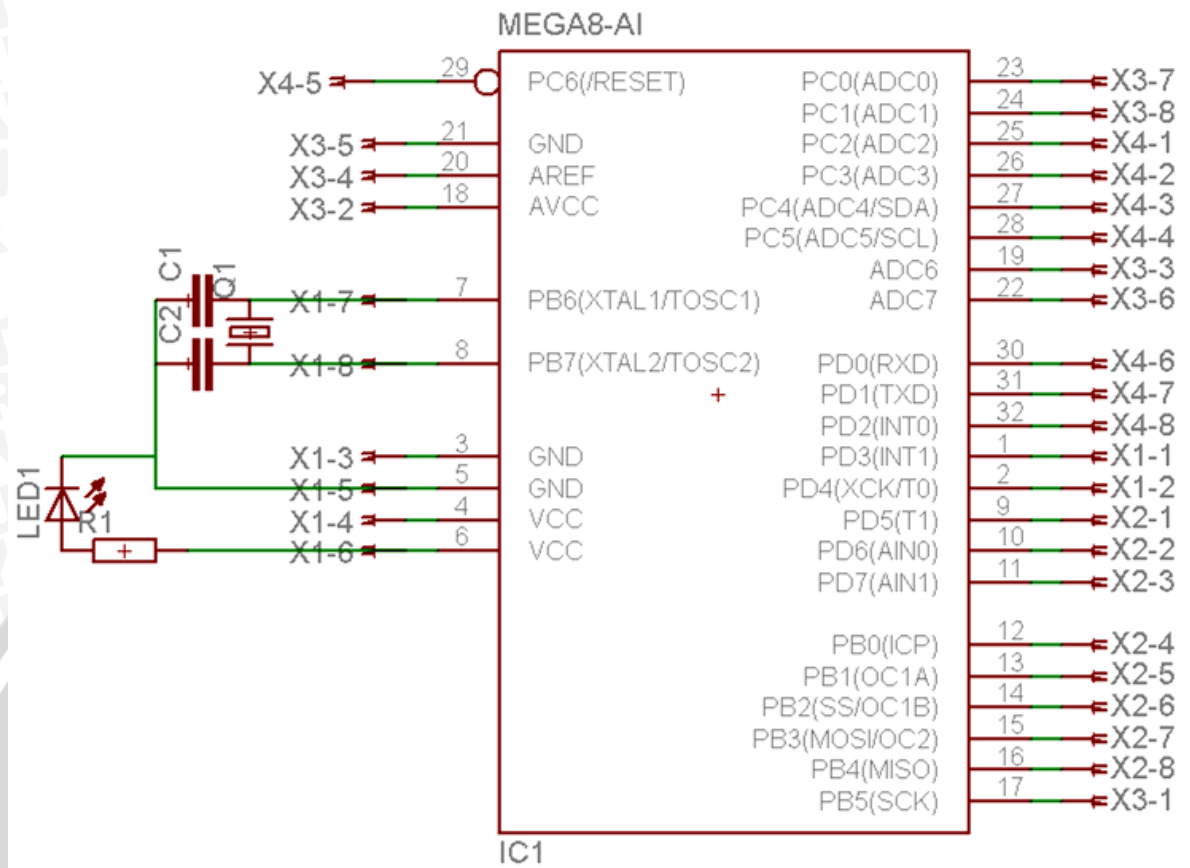
LAMPIRAN II

GAMBAR RANGKAIAN





Rangkaian *Board* Sistem Pendeteksi Musik



**Rangkaian Board Mikrokontroler**



LAMPIRAN III

LISTING PROGRAM  
MIKROKONTROLER ATMEGA8



```
//library
```

```
#include <mega8.h>
```

```
#include <stdio.h>
```

```
#include <delay.h>
```

```
//definisi variabel
```

```
#define ledb PORTB.0
```

```
#define ledm PORTB.1
```

```
#define sinyal PORTB.2
```

```
//variabel global
```

```
int mulai=0,ds[100],musik,suara,nrz;
```

```
signed int X[5],Y[5],i,j,tanda,batas;
```

```
int kuantisasi,b; //keluaran ADC
```

```
void pengkodean1(signed int data);//filter digital
```

```
void pengkodean2();
```

```
void pengkodean3();
```

```
void start();
```

```
// Timer2 output compare interrupt service routine
```

```
interrupt [TIM2_COMP] void timer2_comp_isr(void)
```

```
{
```

```
TCNT2 = 0x00; //membuat nilai inisial 0
```



```
TCCR2 = 0x00; //membuat OCR2 tidak aktif skala 0
```

```
ADCSRA.7 = 1; //ADC enable aktif saat 1
```

```
ADCSRA.6 = 1; //memulai pengambilan data
```

```
}
```

```
// ADC interrupt service routine
```

```
interrupt [ADC_INT] void adc_isr(void)
```

```
{
```

```
ADCSRA.6 = 0; //mengakhiri pengamban data
```

```
ADCSRA.7 = 0; //ADC enable tidak aktif saat 0
```

```
//memberikan nilai kuantisasi sesuai output kuantisasi ADC
```

```
kuantisasi = ADCW;
```

```
pengkodean1(kuantisasi); //pengkodean 1 >>> filterisasi
```

```
mulai = 1;
```

```
TCCR2 = 0x0A; //membuat frekuensi timer 8 kHz
```

```
}
```

```
void main(void)
```

```
{
```

```
// Input/Output Ports initialization
```

```
// Port B initialization
```

```
PORTB=0x00;
```

```
DDRB=0xFF;
```

```
// Port C initialization
```

```
PORTC=0x7F;
```

```
DDRC=0x00;
```

```
// Port D initialization
```

```
PORTD=0x00;
```

```
DDRD=0x00;
```

```
TCCR0=0x00;
```

```
TCNT0=0x00;
```

```
TCCR1A=0x00;
```

```
TCCR1B=0x00;
```

```
TCNT1H=0x00;
```

```
TCNT1L=0x00;
```

```
ICR1H=0x00;
```

```
ICR1L=0x00;
```

```
OCR1AH=0x00;
```

```
OCR1AL=0x00;
```

```
OCR1BH=0x00;
```

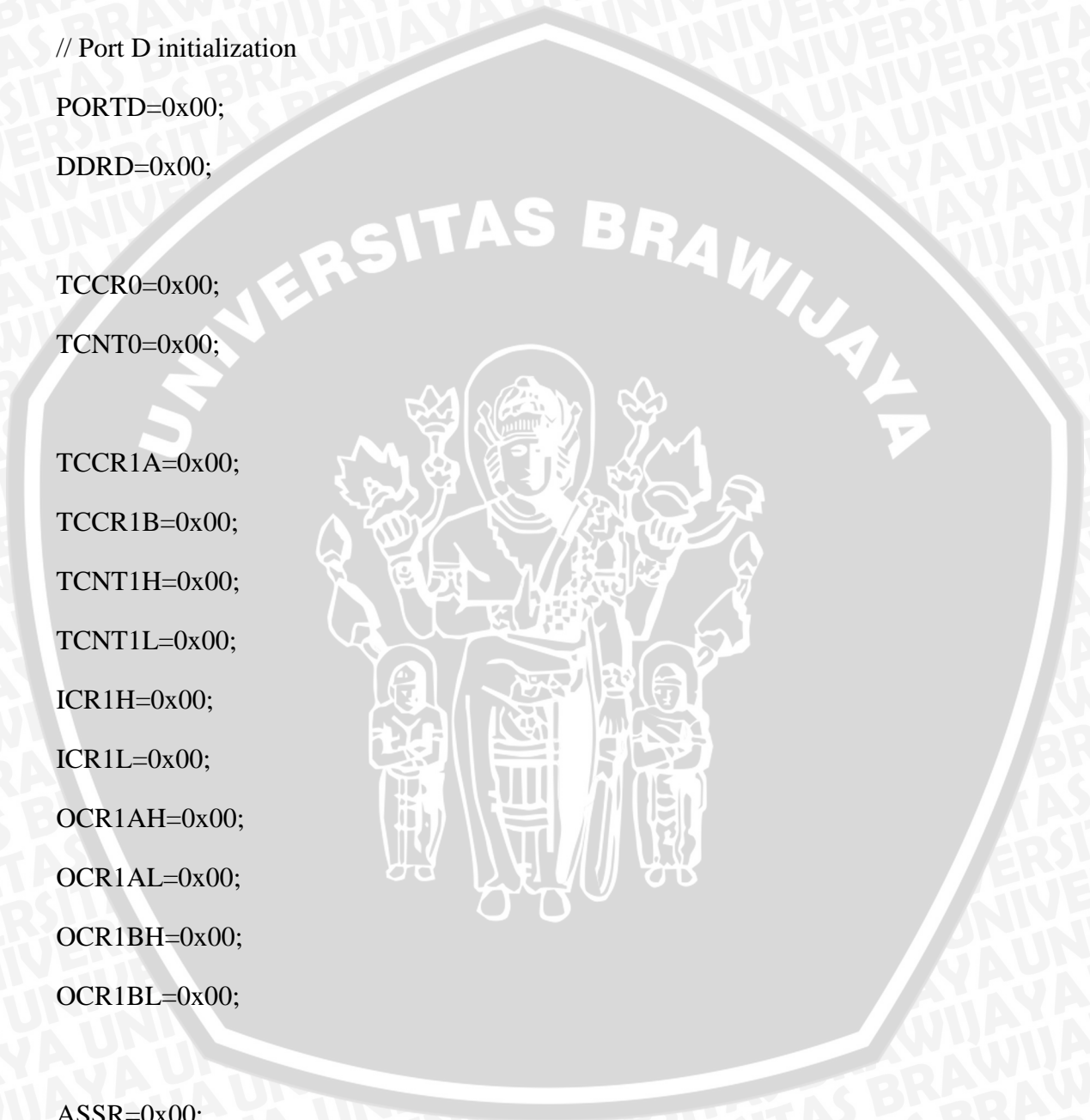
```
OCR1BL=0x00;
```

```
ASSR=0x00;
```

```
TCCR2=0x0A;
```

```
TCNT2=0x00;
```

```
OCR2=0x64;
```



```
// Timer(s)/Counter(s) Interrupt(s) initialization
```

```
TIMSK=0x80; //bit 7 aktif maksudnya timer2 yang digunakan
```

```
// USART initialization
```

```
UCSRA=0x00;
```

```
UCSRB=0x18;
```

```
UCSRC=0x86;
```

```
UBRRH=0x00;
```

```
UBRRL=0x67;
```

```
// Analog Comparator initialization
```

```
ACSR=0x80; // bit 7 bernilai 1 >>> mengaktifkan fungsi analog komparator
```

```
SFIOR=0x00;
```

```
// ADC initialization
```

```
ADMUX=0b01000000; // bit 7 dan 6 bernilai 01 berarti menggunakan AVCC
```

```
ADCSRA=0b11001100; // bit 7 dan 6 untuk memulai dan mengaktifkan ADC
```

```
//,faktor pembagi 16, bit 3 mengaktifkan interrupt ADC
```

```
#asm("sei")
```

```
PORTB = 0;
```

```
nrz = 0;
```

```
tanda = 0;
```

```
while (1)
{
start();
while(1)
{
if(mulai == 1)
{
pengkodean2(); //memanggil fungsi pengkodean 2 filter IIR butterworth
pengkodean3(); //memberikan logika 1 setiap memenuhi frekuensi jiyuu e
no shotai
mulai = 0;
} } } }

//subfungsi

//BPF butterwrth fc1 = 300Hz fc2 = 2500Hz orde 4 fs=8000Hz
void pengkodean1(signed int data)
{
X[0] = data;
Y[0] = (8*X[0]-16*X[2])+(8*X[4])+(17*Y[1])+(6*Y[2])-(3*Y[3])-(
4*Y[4])/16;
b=4;
for(b=4;b>0;b--) //mengisi nilai y{n)
{
X[b]=X[b-1];
Y[b]=Y[b-1]; } }
```

```
//pemisahan lagu jiyuu e no shotai
void pengkodean2()
{
//if(Y[0]>=54||Y[0]<=-33) >> lagu normal
if(Y[0]>=413||Y[0]<=-43) //lagu jiyuu e no shotai >=413 <=-40
{
ds[0] = 1;
}
else ds[0] = 0;
j++;

//100 kali sampling mengisi variabel ds
for(i=99;i>0;i--)
{
ds[i] = ds[i-1];
}

//setelah 100 kali sampling
//variabel di jumlah
if(j>=100)
{
tanda=0;
for(i=0;i<100;i++)
{
tanda = tanda + ds[i];
}
}
```



```
//apabila lebih dari 0 batas akan ditambah 1
```

```
if(tanda>0)
```

```
{
```

```
    batas++;
```

```
//untuk menghemat memori di reset menjadi 200 kembali
```

```
if(batas==1300)batas=100;//200
```

```
}
```

```
else batas = 0;
```

```
//semua variabel dikembalikan untuk proses berikutnya
```

```
j = 0;
```

```
tanda=0;
```

```
for(i=0;i<100;i++)
```

```
{
```

```
    ds[i] = 0;
```

```
    }
```

```
    }
```

```
    }
```

```
//memberikan nilai suara 1 apabila memenuhi  $120 \cdot 100 = 12000$  kali sampling
```

```
void pengkodean3()
```

```
{
```

```
    if(batas >= 120) suara=1; //120
```

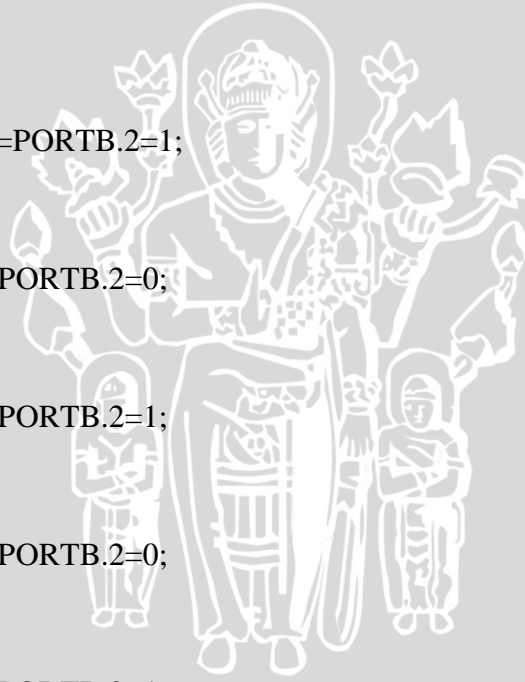
```
        else suara=0;
```

```
    if(nrz == 1)
```

```
        { sinyal=1; //sebagai sinyal kontrol ke board master
```

```
    ledm=0; //Led merah diberikan tegangan 0 volt
    ledb=1; //Led biru diberikan tegangan +Vcc volt
}
else
{
    sinyal=0; //sebagai sinyal kontrol ke board master
    ledm=1; //Led merah diberikan tegangan +Vcc volt
    ledb=0; //Led biru diberikan tegangan 0 volt
} }
```

```
void start()
{PORTB.0=PORTB.1=PORTB.2=1;
delay_ms(1000);
PORTB.0=PORTB.1=PORTB.2=0;
delay_ms(300);
PORTB.0=PORTB.1=PORTB.2=1;
delay_ms(100);
PORTB.0=PORTB.1=PORTB.2=0;
delay_ms(100);
PORTB.0=PORTB.1=PORTB.2=1;
delay_ms(100);
PORTB.0=PORTB.1=PORTB.2=0;
delay_ms(100);
PORTB.0=PORTB.1=PORTB.2=1;
delay_ms(100);
PORTB.0=PORTB.1=PORTB.2=0;}
```



```
delay_ms(100);  
PORTB.0=PORTB.1=PORTB.2=1;  
delay_ms(100);  
PORTB.0=PORTB.1=PORTB.2=0;  
delay_ms(500);  
}
```





LAMPIRAN IV

DATASHEET

