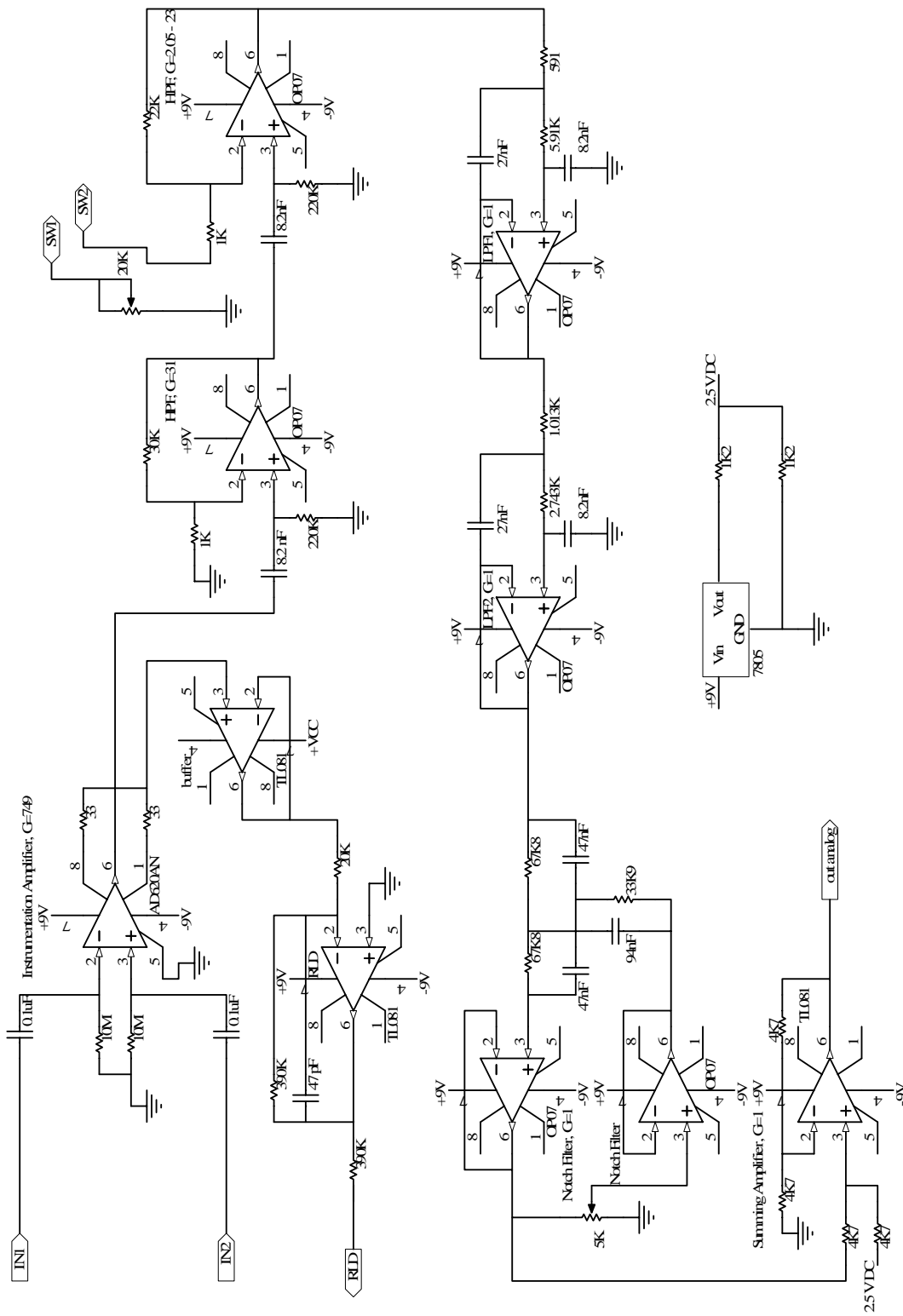


## **LAMPIRAN A**

### **RANGKAIAN SKEMATIK PERANGKAT KERAS**



**Rangkaian Penguat dan Filter**



**LAMPIRAN B**

**LIST PROGRAM MIKROKONTROLER**

```
.INCLUDE "C:\Program Files\Atmel\AVR Studio\Appnotes\m8535def.inc"
```

```
.def    tmp = R16
.def    tmp1 = r17
.def    tmp2 = r18
.def    tmp3 = r21
.def    tmp4 = r22
.def    txbyte = R19
.def    rxbyte = r20
.equ    fclock = 11059200
.equ    baud_rate = 230400
.equ    ubbr_value = (fclock/(16*baud_rate))-1
```

```
.org 0x0000
```

```
    rjmp    main
```

```
.org 0x000B
```

```
    rjmp    usart_rxc
```

```
main:
```

```
ldi    tmp,low(ramend)
out    spl,tmp
ldi    tmp,high(ramend)
out    sph,tmp
rcall  init_usart
```

```
ldi    tmp,0
out    ADMUX,tmp
ldi    tmp,0b11000100
out    ADCSRA,tmp
```

```
first_conv:
```

```
sbic   adcsra,adsc
rjmp   first_conv
in     tmp,adcl
in     tmp1,adch
com    tmp
out    portb,tmp
sei
```

```
ldi    txbyte,33
rcall  usart_tx
done:
rjmp   done
```

```
usart_rxc:
rcall  usart_rx
cpi    rxbyte,58
brne   gain
```

```
ldi    txbyte,33
rcall  usart_tx
rjmp   keluar
```

```
gain:
cpi    rxbyte,'?'
brne   kirimADC
```

```
rcall  convADC
add    tmp1,tmp1
add    tmp,tmp
brcc   nocarry1
inc    tmp1
```

```
nocarry1:
ldi    z1,low(2*msg)
ldi    zh,high(2*msg)
cpi    tmp1,0
breq   load4
add    zh,tmp1
```

```
load4:
add    z1,tmp
brcc   load5
inc    zh
load5:
lpm
mov    txbyte,r0
rcall  beda
```

```
rcall    usart_tx
```

```
inc      z1
```

```
brcc    load6
```

```
inc      zh
```

```
load6:
```

```
lpm
```

```
mov     txbyte,r0
```

```
rcall   beda
```

```
rcall   usart_tx
```

```
 kirimADC:
```

```
cpi     rxbyte,47
```

```
brne    keluar
```

```
ldi     tmp3,100
```

```
count1:
```

```
dec     tmp3
```

```
ldi     tmp4,200
```

```
count:
```

```
dec     tmp4
```

```
rcall   convADC
```

```
add     tmp1,tmp1
```

```
add     tmp,tmp
```

```
brcc    nocarry
```

```
inc     tmp1
```

```
nocarry:
```

```
ldi     z1,low(2*msg)
```

```
ldi     zh,high(2*msg)
```

```
cpi     tmp1,0
```

```
breq    load1
```

```
add     zh,tmp1
```

```
load1:
```

```
add     z1,tmp
```

```
brcc    load2
```

```
inc    zh
```

```
load2:
```

```
lpm
```

```
mov    txbyte,r0
```

```
rcall  beda
```

```
rcall  usart_tx
```

```
inc    zl
```

```
brcc   load3
```

```
inc    zh
```

```
load3:
```

```
lpm
```

```
mov    txbyte,r0
```

```
rcall  beda
```

```
rcall  usart_tx
```

```
cpi    tmp4,0
```

```
brne   count
```

```
cpi    tmp3,0
```

```
brne   count1
```

```
keluar:
```

```
reti
```

```
=====
```

```
; USART init
```

```
=====
```

```
init_usart:
```

```
ldi    r21,0b10000000
```

```
out    ucusra,r21
```

```
ldi    tmp,high(ubbr_value)
```

```
out    ubrrh,tmp
```

```
ldi    tmp,low(ubbr_value)
```

```
out    ubrrl,tmp
```

```
ldi    tmp,(1<<rxen)|(1<<txen)|(1<<rxcie)
```

```
out    ucsrc,tmp
```

```
ldi    tmp,(1<<ursel)|(2<<ucsz0)
```

```
out    ucsrc,tmp
```

```
ret
```



```

;=====
; USART transmit data
;=====
usart_tx:
sbis    uc_sra,udre
rjmp    usart_tx
out     udr,txbyte
ret

;=====
; USART receive data
;=====
usart_rx:
sbis    uc_sra,rx_c
rjmp    usart_rx
in      rx_byte,udr
ret

;=====
; KONVERSI HEXA KE ASCII
;=====
HEX_ASCII2:
    push    tmp
    rcall   HEX_ASCII1
    mov     tmp1,tmp
    pop     tmp
    swap   tmp
    rcall   HEX_ASCII1
    ret

HEX_ASCII1:
    andi   tmp,0b00001111
    cpi    tmp,0x0A
    brsh   bukan_angka
    ldi    tmp2,0x30
    add    tmp,tmp2
    ret

```

```
bukan_angka:
    ldi    tmp2,0x37
    add    tmp,tmp2
    ret
```

```
;=====
```

```
; KONVERSI ADC
```

```
;=====
```

```
convADC:
```

```
sbi     adcsra,adsc
```

```
conv:
```

```
sbic    adcsra,adsc
```

```
rjmp    conv
```

```
in      tmp,adcl
```

```
in      tmp1,adch
```

```
ret
```

```
;=====
```

```
; BEDAKAN ANGKA & HURUF
```

```
;=====
```

```
beda:
```

```
cpi     txbyte,65
```

```
brge    huruf
```

```
subi    txbyte,13
```

```
ret
```

```
huruf:
```

```
subi    txbyte,20
```

```
ret
```

msg:

.db  
 "000102030405060708090A0B0C0D0E0F0G0H0I0J0K0L0M0N0O0P0Q0R0S0T0U0V0W0X0Y0Z"  
 .db  
 "101112131415161718191A1B1C1D1E1F1G1H1I1J1K1L1M1N1O1P1Q1R1S1T1U1V1W1X1Y1Z"  
 .db  
 "202122232425262728292A2B2C2D2E2F2G2H2I2J2K2L2M2N2O2P2Q2R2S2T2U2V2W2X2Y2Z"  
 .db  
 "303132333435363738393A3B3C3D3E3F3G3H3I3J3K3L3M3N3O3P3Q3R3S3T3U3V3W3X3Y3Z"  
 .db  
 "404142434445464748494A4B4C4D4E4F4G4H4I4J4K4L4M4N4O4P4Q4R4S4T4U4V4W4X4Y4Z"  
 .db  
 "505152535455565758595A5B5C5D5E5F5G5H5I5J5K5L5M5N5O5P5Q5R5S5T5U5V5W5X5Y5Z"  
 .db  
 "606162636465666768696A6B6C6D6E6F6G6H6I6J6K6L6M6N6O6P6Q6R6S6T6U6V6W6X6Y6Z"  
 .db  
 "707172737475767778797A7B7C7D7E7F7G7H7I7J7K7L7M7N7O7P7Q7R7S7T7U7V7W7X7Y7Z"  
 .db  
 "808182838485868788898A8B8C8D8E8F8G8H8I8J8K8L8M8N8O8P8Q8R8S8T8U8V8W8X8Y8Z"  
 .db  
 "909192939495969798999A9B9C9D9E9F9G9H9I9J9K9L9M9N9O9P9Q9R9S9T9U9V9W9X9Y9Z"  
 .db  
 "A0A1A2A3A4A5A6A7A8A9AAABACADAEAFAGAHAIAJAKALAMANAOAPAQARASATAUAVAW  
 AXAYAZ"  
 .db  
 "B0B1B2B3B4B5B6B7B8B9BABBBBCDBEBFBGBHBIJBKBLBMBNBOBPBQBRBSBTBUBVBWBX  
 YBZ"  
 .db  
 "C0C1C2C3C4C5C6C7C8C9CACBCCCDCECFCGCHCICJCKCLCMCNCOCPQCRCRCCTCUCVCWCXC  
 YCZ"  
 .db  
 "D0D1D2D3D4D5D6D7D8D9DADBDCDDDEDFDGDHDIJDKDLMDNDODPDQDRDSDTDUDVDW  
 DXDYDZ"  
 .db  
 "E0E1E2E3E4E5E6E7E8E9EAEBECEDEEEFEGEHEIEJEKELEMENEOEPEQERESETEUEVEWEXEYEZ  
 "  
 .db  
 "F0F1F2F3F4F5F6F7F8F9FABFBCFDFEFFFGFHFIFJFKFLFMFNFOFPFQFRFSFTFUFVFWFXFYFZ"  
 .db  
 "G0G1G2G3G4G5G6G7G8G9GAGBGC GDGEGFGGGHGIGJGKGLGMGN GOGPGQGRGSGTGUGV  
 GXGYGZ"  
 .db  
 "H0H1H2H3H4H5H6H7H8H9HAHBHCHDHEHFHGHGHHIHHJKHLHMHNHOHPHQHRHSHTHUHVHW  
 HXHYHZ"  
 .db "I0I1I2I3I4I5I6I7I8I9IAIBICIDIEIFIGIHIIIIJKILIMINIOIPIQIRISITIUUVIWXIYZ"  
 .db "J0J1J2J3J4J5J6J7J8J9AJBJCJDJEJFJGJHJJJKJLJMJNJOJPJQJRSJTJUJVJWJXJYJZ"

.db  
"K0K1K2K3K4K5K6K7K8K9KAKBKCKDKEKFKGKHKIKJKKCLKMKNKOKPKQKRKSKTKUKVKW  
KXKYKZ"  
.db  
"L0L1L2L3L4L5L6L7L8L9LALBLCLDLELFLGLHLILJLKLLMLNLLOLPLQLRLSLTLULVLWLXLYLZ  
"  
.db  
"M0M1M2M3M4M5M6M7M8M9MAMBMCMDMEMFMGMHMHMIMJMKMLMMMNMMOMPMQMRMSMT  
MUMVMWVMXMYMZ"  
.db  
"N0N1N2N3N4N5N6N7N8N9NANBNCNDNENFNNGHNHNINJKNLNMNNNONPNQNRNSNTNUNVNW  
NXNYNZ"  
.db  
"O0O1O2O3O4O5O6O7O8O9OAOBOCODOEFOGOGOHOIOJOKOLOMONOOPOQOROSOTOUOVOW  
OXOYOZ"  
.db  
"P0P1P2P3P4P5P6P7P8P9PAPBPCPDPEPFPGPHPIPIPKPLPMPNPOPPQPRPSPTPUPVPWPXPYPZ"  
.db  
"Q0Q1Q2Q3Q4Q5Q6Q7Q8Q9QAQBQCQDQEQFQGQHQIQJKQLQMKNQOQPQQQRQSQTQUQVQW  
QXQYQZ"  
.db  
"R0R1R2R3R4R5R6R7R8R9RARBRCDREFRGRHRIRJRKRLMRNRORPRQRRRSRTRURVRWRXR  
YRZ"  
.db  
"S0S1S2S3S4S5S6S7S8S9SASBSCSDSESEFSGSHSISJSKSLSMSNSOSPSQSRSSSTSUSVSWXSXSYSZ"

**LAMPIRAN C**

**LIST PROGRAM DELPHI**

unit TA;

interface

uses

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,  
Dialogs, Menus, StdCtrls, ExtCtrls, CPort, ComCtrls, ActnMan, options, ActnColorMaps,  
xyun, xygraph, xycopy, xygraph3d, math, LMDAboutDlg, ToneGen,  
LMDCustomComponent, LMDContainerComponent, LMDBaseDialog;

type

TMainForm = class(TForm)  
Label1: TLabel;  
Label2: TLabel;  
StatusBar1: TStatusBar;  
Button1: TButton;  
ComPort1: TComPort;  
Button2: TButton;  
Button4: TButton;  
ComboBox1: TComboBox;  
Button3: TButton;  
PaintBox4: TPaintBox;  
CheckBox1: TCheckBox;  
LMDAboutDlg1: TLMDAboutDlg;  
ToneGen1: TToneGen;  
Button6: TButton;  
Edit4: TEdit;  
ScrollBar1: TScrollBar;  
TrackBar1: TTrackBar;  
Label3: TLabel;  
ComDataPacket1: TComDataPacket;  
Button7: TButton;  
Button8: TButton;  
Button9: TButton;  
Button10: TButton;  
Label4: TLabel;  
Label5: TLabel;  
Bevel1: TBevel;

```

Bevel2: TBevel;
Bevel3: TBevel;
Button5: TButton;
ComboBox2: TComboBox;
Label6: TLabel;
Button11: TButton;
Button12: TButton;
RadioGroup1: TRadioGroup;
procedure FormCreate(Sender: TObject);
procedure Activate(Sender: TObject);
procedure Button2Click(Sender: TObject);
procedure ComPortOpen(Sender: TObject);
procedure ComPortClose(Sender: TObject);
procedure Button1Click(Sender: TObject);
procedure Button4Click(Sender: TObject);
procedure Button3Click(Sender: TObject);
procedure formclose(Sender: TObject; var Action: TCloseAction);
procedure Paint(Sender: TObject);
procedure paintboxmousedown(Sender: TObject; Button: TMouseButton;
  Shift: TShiftState; X, Y: Integer);
procedure paintboxmouseup(Sender: TObject; Button: TMouseButton;
  Shift: TShiftState; X, Y: Integer);
procedure paintboxmousemove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
procedure Button6Click(Sender: TObject);
procedure Scrollchange(Sender: TObject);
procedure Freqchange(Sender: TObject);
procedure Changevolume(Sender: TObject);
procedure datapaket(Sender: TObject; const Str: String);
procedure Button7Click(Sender: TObject);
procedure Button8Click(Sender: TObject);
procedure Button9click(Sender: TObject);
procedure Button10Click(Sender: TObject);
procedure ComboBox2Change(Sender: TObject);
procedure Button5Click(Sender: TObject);
procedure Button12Click(Sender: TObject);
procedure Button11Click(Sender: TObject);

```

```

private
  { Private declarations }
public

end;

var
  MainForm: TMainForm; status,status1:string;
  flag,flag2,flag3,k,l,m,n,buffer,buffer1,setpoint,count2,signal1,signal2:integer;
  gain,signal:real;
  records : array [1..1000,1..20000] of real;
  data, data2 : Tdatatype;

implementation

{$R *.dfm}

procedure TMainForm.FormCreate(Sender: TObject);
var i : integer;
begin
  mainform.Height:= 864;
  mainform.Width:= 1152;
  enumcomports(comboBox1.Items);
  comport1.Port := comboBox1.Items[1];
  comport1.BaudRate := brcustom;
  comport1.CustomBaudRate:=230400;
  comport1.DataBits := dbseven;
  comport1.StopBits := sbOneStopBit;
  comport1.Connected := false;
  port1:=comport1.port;
  baudrate1:='230400';
  data00:='7';
  stop:='1-bit';
  status:='Disconnected';
  tonegen1.Frequency:=strtoint(edit4.Text);
  statusbar1.Panels.Items[6].Text:='Stimulation Freq : '+edit4.Text+' Hz';
  statusbar1.Panels.Items[5].Text:='T Sampling : 1 ms';
  statusbar1.Panels.Items[7].Text:='Stimulation Volume : '+inttostr(5*trackbar1.Position)+' %';

```



```

flag:=0;
flag3:=100;
k:=0;l:=1;
xysetdataarray(data,20000,1);
xysetdataarray(data2,20000,1);
end;

procedure TMainForm.Activate(Sender: TObject);
begin
comport1.Port := port1;
statusbar1.Panels.Items[0].Text:='Port : '+port1;

if baudrate1='9600' then
begin
comport1.BaudRate:=br9600;
end;
if baudrate1='19200' then
begin
comport1.BaudRate:=br19200;
end;
if baudrate1='56000' then
begin
comport1.BaudRate:=br56000;
end;
if baudrate1='115200' then
begin
comport1.BaudRate:=br115200;
end;
if baudrate1='230400' then
begin
comport1.BaudRate := brcustom;
comport1.CustomBaudRate:=230400;
end;
statusbar1.Panels.Items[1].Text:='Baudrate : '+baudrate1+' bps';

if data00='5' then
begin
comport1.DataBits:=dbfive;

```

```

end;
if data00='6' then
begin
    comport1.DataBits:=dbsix;
end;
if data00='7' then
begin
    comport1.DataBits:=dbseven;
end;
if data00='8' then
begin
    comport1.DataBits:=dbeight;
end;
statusbar1.Panels.Items[2].Text:='Data Bits : '+data00+'-bits';

if stop='1-bit' then
begin
    comport1.stopBits:=sbOneStopBit;
end;
if stop='2-bits' then
begin
    comport1.stopBits:=sbTwoStopBits;
end;
statusbar1.Panels.Items[3].Text:='Stop Bit(s) : '+stop;

if comport1.Connected=true then
begin
    status:='Connected';
end;
if comport1.Connected=false then
begin
    status:='Disconnected';
end;
statusbar1.Panels.Items[4].Text:='Status : '+status;
end;

procedure TMainForm.Button2Click(Sender: TObject);
begin

```

```

if ComPort1.Connected then
  begin
    ComPort1.Close;status:='Disconnected';
  end
else
  begin
    ComPort1.Open;status:='Connected';
  end;
statusbar1.Panels.Items[4].Text:='Status : '+status;
end;

procedure TMainForm.ComPortOpen(Sender: TObject);
begin
  Button2.Caption := 'Disconnect';
  button1.Enabled := true;
  statusbar1.Panels.Items[4].Text:='';
end;

procedure TMainForm.ComPortClose(Sender: TObject);
begin
  if Button2 <> nil then
    Button2.Caption := 'Connect';
    button1.Enabled := false;
    button3.Enabled := false;
    button4.Enabled := false;
    statusbar1.Panels.Items[4].Text:='';
end;

procedure TMainForm.Button1Click(Sender: TObject);
begin
  flag:=1;comdatapacket1.Size:=1;ComPort1.WriteStr(':');
end;

procedure TMainForm.Button4Click(Sender: TObject);
begin
  flag:=2;Comport1.WriteStr('?');
end;

```

```

procedure TMainForm.Button3Click(Sender: TObject);
var i:integer;
begin
  if button3.Caption='Get Signal' then
    begin
      button11.Enabled:=false;
      button12.Enabled:=false;
      flag3:=100;
      flag:=3;
      k:=0;l:=1;
      buffer1:=buffer;
      button3.Caption:='Stop Signal';
      comdatapacket1.Size:=40000;
      comport1.WriteStr('=');
    end
  else
    begin
      flag:=0;
      button3.Caption:='Get Signal';
    end;
end;

procedure TMainForm.formclose(Sender: TObject; var Action: TCloseAction);
begin
  comport1.Close;
end;

procedure TMainForm.Paint(Sender: TObject);
var tm : boolean;
begin
  xycleargraph(paintbox4,clwhite,clblack,1.0);
  xystartgraph(0,100,0,flag3,40,40,40,40,clipon);
  if checkbox1.checked then
    begin
      end
    else
      begin

```

```

    xyxaxis(clblack,0,20000,0,0,'Time (mS)',true,false,false);
    xyyaxis(clgreen,-5,5,0,0,'Voltage (uV)',1,false,false,false);
    xysymbol(2,6,2);
end;
xyplotarray(data,1,3);
xytitle(clmaroon,'Brainstem Voltage');

if flag3=50 then
begin
xystartgraph(0,100,flag3,100,40,40,40,40,clipon);
xyxaxis(clblack,0,20000,0,0,'Time (mS)',true,false,false);
xyyaxis(clblue,-5,5,0,0,'Voltage (uV)',1,false,false,false);
xysymbol(2,6,2);
xyplotarray(data2,1,3);
xytitle(clred,'Brainstem Voltage after Stimulation');
end;

if checkbox1.checked then
    xyinitruler(clred,100,round(paintbox4.height * 0.1)-20,1,3)
else
    xyinitruler(clmaroon,100,round(paintbox4.height * 0.05)-20,1,0)

end;

procedure TMainForm.paintboxmousedown(Sender: TObject; Button: TMouseButton;
    Shift: TShiftState; X, Y: Integer);
begin
    xymousedown(button,shift,x,y);
end;

procedure TMainForm.paintboxmouseup(Sender: TObject; Button: TMouseButton;
    Shift: TShiftState; X, Y: Integer);
begin
    xymouseup(button,shift,x,y);
end;

procedure TMainForm.paintboxmousemove(Sender: TObject; Shift: TShiftState;
    X, Y: Integer);

```

```

begin
  xmousemove(shift,x,y);
end;

procedure TMainForm.Button6Click(Sender: TObject);
begin
  flag:=1;
end;

procedure TMainForm.Scrollchange(Sender: TObject);
begin
  edit4.Text:=inttostr(20+scrollbar1.Position*scrollbar1.Position);
  tonegen1.Frequency:=strtoint(edit4.Text);
  statusbar1.Panels.Items[6].Text:='Stimulation Freq : '+edit4.text+' Hz';
end;

procedure TMainForm.Freqchange(Sender: TObject);
begin
  scrollbar1.Position:=trunc(sqrt(strtfloat(edit4.text)-20));
  tonegen1.Frequency:=strtoint(edit4.Text);
  statusbar1.Panels.Items[6].Text:='Stimulation Freq : '+edit4.Text+' Hz';
end;

procedure TMainForm.Changevolume(Sender: TObject);
begin
  tonegen1.LeftVolume:=5*trackbar1.Position;
  tonegen1.RightVolume:=5*trackbar1.Position;
  statusbar1.Panels.Items[7].Text:='Stimulation Volume : '+inttostr(tonegen1.leftvolume)+' %';
end;

procedure TMainForm.datapaket(Sender: TObject; const Str: String);
var i:integer;
begin
  if str='!' then
    begin
      statusbar1.Panels.Items[4].Text:='PC-based BERA detected';
      button3.enabled:=true;
      button4.Enabled:=true;
    end;
end;

```

```

end
else
begin
statusbar1.Panels.Items[4].Text:='PC-based BERA not detected';
end;
if str<>'!' then
begin
if flag3=100 then
begin
for i:=1 to 20000 do
begin
signal:=0.005*((36*(ord(str[(2*i)-1])-35))+(ord(str[(2*i)]))-35))-2.5;
k:=k+1;
data[k,1]:=signal;
if radiogroup1.ItemIndex=1 then
records[1,k]:=signal;
end;
if radiogroup1.ItemIndex=1 then
begin
m:=1;l:=l+1;
if flag=0 then
begin
n:=m;
if m>1 then
button12.Enabled:=true;
end;
end;
end;
if flag3=50 then
begin
for i:=1 to 20000 do
begin
signal:=0.005*((36*(ord(str[(2*i)-1])-35))+(ord(str[(2*i)]))-35))-2.5;
k:=k+1;
data2[k,1]:=signal;
if radiogroup1.ItemIndex=1 then
records[1,k]:=signal;
end;

```

```

if radiogroup1.ItemIndex=1 then
begin
m:=1;l:=1+1;
n:=m;
if m>1 then
button12.Enabled:=true;
end;
paintbox4.Refresh;
end;
if flag=3 then
begin
paintbox4.Refresh;
k:=0;comport1.WriteStr('/');
end;
if flag=0 then
begin
k:=0;
button2.Enabled:=true;button3.Enabled:=true;button4.Enabled:=true;
button3.Caption:= 'Get Signal';
paintbox4.Refresh;
end;
if flag=1 then
begin
k:=0;
flag3:=50;
tonegen1.Frequency:=strtoint(edit4.Text);
tonegen1.Play;
comport1.WriteStr('/');
flag:=0;
end;
statusbar1.Panels.Items[4].Text:='PC-based BERA detected';
end;
end;

procedure TMainForm.Button7Click(Sender: TObject);
begin
form2.Show;
end;

```



```
procedure TMainForm.Button8Click(Sender: TObject);
```

```
begin
```

```
xycopystart;
```

```
end;
```

```
procedure TMainForm.Button9click(Sender: TObject);
```

```
begin
```

```
if MessageDlg('Exit now?', mtConfirmation, [mbYes, mbNo], 0) = mrYes then
```

```
begin
```

```
MessageDlg('Exiting PC-based BERA application.', mtInformation, [mbOk], 0);
```

```
Close;
```

```
end;
```

```
end;
```

```
procedure TMainForm.Button10Click(Sender: TObject);
```

```
begin
```

```
lmdaboutdlg1.Execute;
```

```
end;
```

```
procedure TMainForm.ComboBox2Change(Sender: TObject);
```

```
begin
```

```
case combobox2.ItemIndex of
```

```
0 : tonegen1.Waveform := tgsine;
```

```
1 : tonegen1.Waveform := tgsquare;
```

```
2 : tonegen1.Waveform := tgtriangle;
```

```
3 : tonegen1.waveform := tgsawtooth;
```

```
4 : tonegen1.Waveform := tgnoise;
```

```
end;
```

```
end;
```

```
procedure TMainForm.Button5Click(Sender: TObject);
```

```
begin
```

```
tonegen1.Play;
```

```
end;
```

```
procedure TMainForm.Button12Click(Sender: TObject);
```

```
begin
```

```
button11.Enabled:=true;
m:=m-1;
for k:=1 to 20000 do
data[k,1]:=records[m,k];
paintbox4.Refresh;
if m=1 then
button12.Enabled:=false;

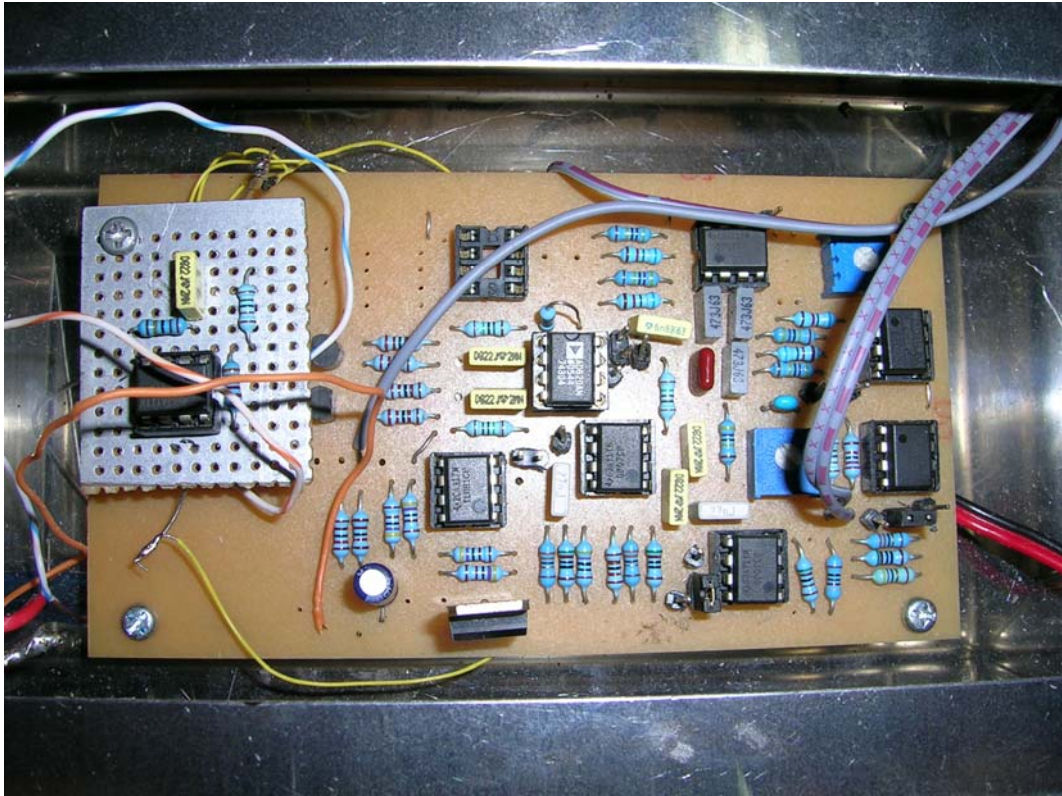
end;
```

```
procedure TMainForm.Button11Click(Sender: TObject);
begin
button12.Enabled:=true;
m:=m+1;
for k:=1 to 20000 do
data[k,1]:=records[m,k];
paintbox4.Refresh;
if m=n then
button11.Enabled:=false;
end;

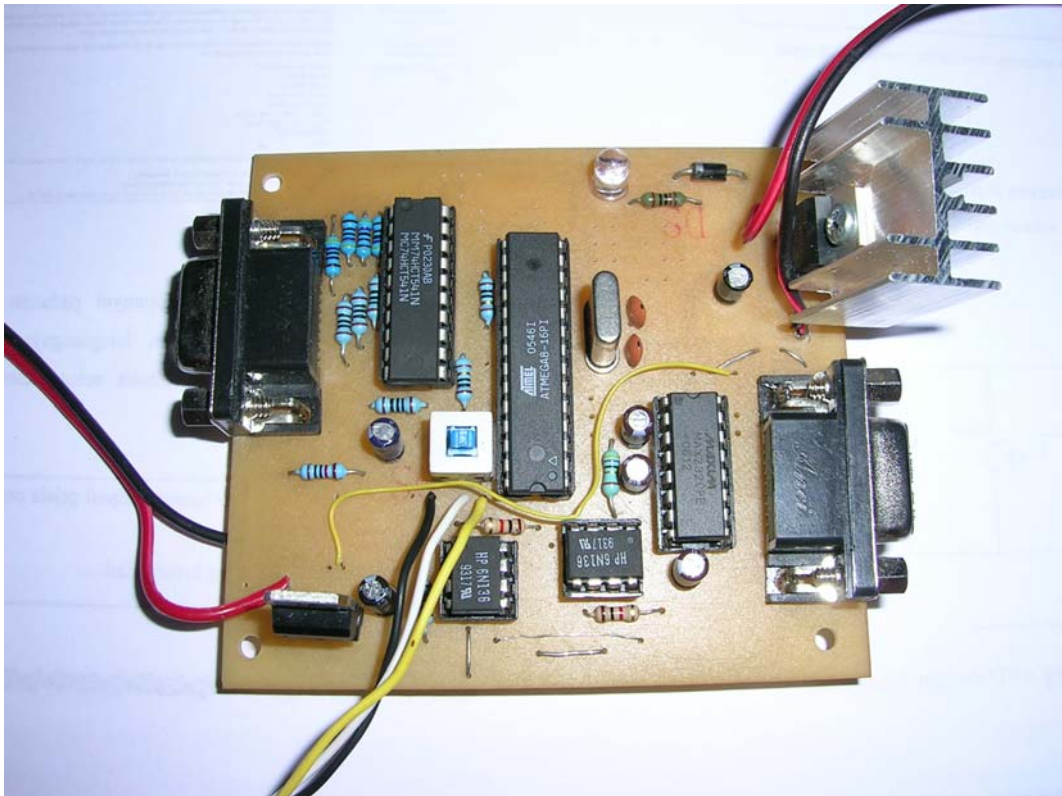
end.
```

**LAMPIRAN D**

**FOTO ALAT DAN PENGUJIAN ALAT**



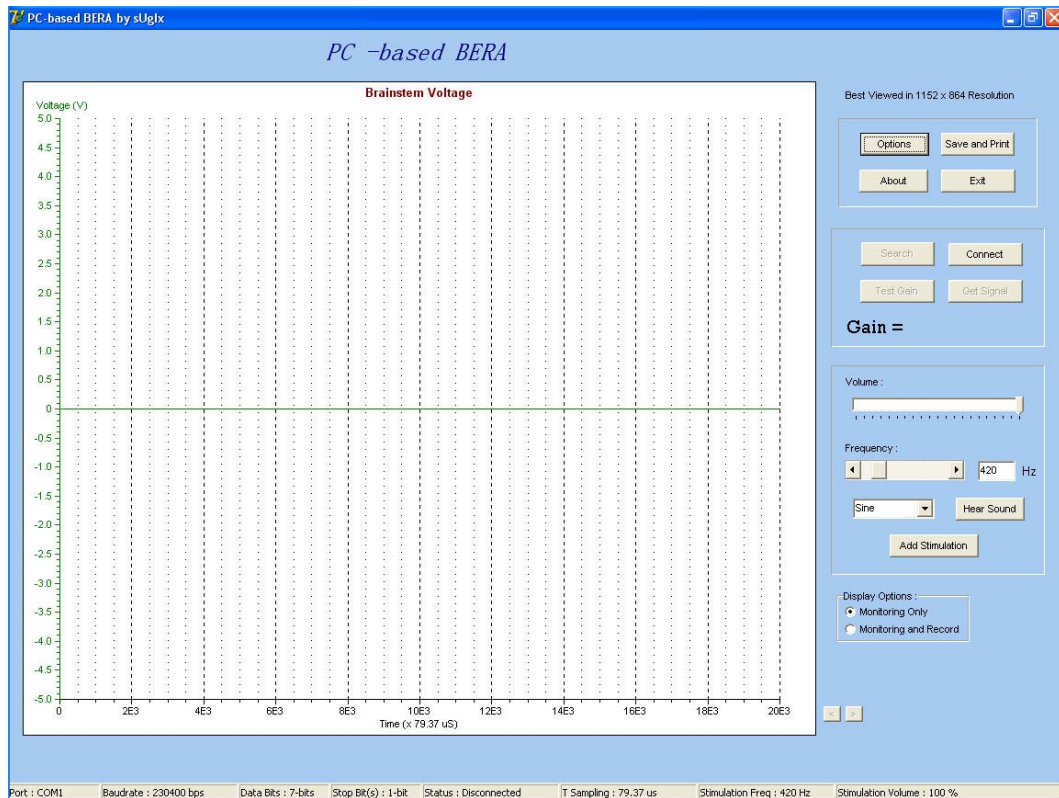
**Rangkaian Analog**



**Rangkaian Digital**



**Rangkaian Analog dan Digital**



**Tampilan Program Delphi**



**Pengujian Alat terhadap Manusia**



**Pengujian Alat terhadap Manusia**

**LAMPIRAN E**

**TABEL BESSEL**

<b>n</b>	<b>i</b>	<b>a<sub>i</sub></b>	<b>b<sub>i</sub></b>	<b>k<sub>i</sub> = f<sub>ci</sub> / f<sub>c</sub></b>	<b>Q<sub>i</sub></b>
1	1	1.0000	0.0000	1.000	—
2	1	1.3617	0.6180	1.000	0.58
3	1	0.7560	0.0000	1.323	—
	2	0.9996	0.4772	1.414	0.69
4	1	1.3397	0.4889	0.978	0.52
	2	0.7743	0.3890	1.797	0.81
5	1	0.6656	0.0000	1.502	—
	2	1.1402	0.4128	1.184	0.56
	3	0.6216	0.3245	2.138	0.92
6	1	1.2217	0.3887	1.063	0.51
	2	0.9686	0.3505	1.431	0.61
	3	0.5131	0.2756	2.447	1.02
7	1	0.5937	0.0000	1.648	—
	2	1.0944	0.3395	1.207	0.53
	3	0.8304	0.3011	1.695	0.66
	4	0.4332	0.2381	2.731	1.13
8	1	1.1112	0.3162	1.164	0.51
	2	0.9754	0.2979	1.381	0.56
	3	0.7202	0.2621	1.963	0.71
	4	0.3728	0.2087	2.992	1.23
9	1	0.5386	0.0000	1.857	—
	2	1.0244	0.2834	1.277	0.52
	3	0.8710	0.2636	1.574	0.59
	4	0.6320	0.2311	2.226	0.76
	5	0.3257	0.1854	3.237	1.32
10	1	1.0215	0.2650	1.264	0.50
	2	0.9393	0.2549	1.412	0.54
	3	0.7815	0.2351	1.780	0.62
	4	0.5604	0.2059	2.479	0.81
	5	0.2883	0.1665	3.466	1.42