

LAMPIRAN

# LAMPIRAN

Listing program Delphi 6.0 dengan judul "Perancangan dan Pembuatan Perangkat Lunak Pengenalan Suara untuk Pembelajaran Anak Kecil dengan Gambar Buah Menggunakan JST"

## A. Unit Data

```
unit data;
Interface
implementation

Procedure TForm2.Button1Click(Sender: TObject);
begin
  if not buka.Execute then exit
  else
  begin
    dbedit1.Text:=buka.FileName;
    image1.Picture.LoadFromFile(buka.FileName);
  end;
end;

Procedure tform2.entriaktif(ya:boolean);
begin
  label2.Enabled:=ya;
  label3.Enabled:=ya;
  dbcombobox1.Enabled:=ya;
  dbedit1.Enabled:=ya;
  button3.Enabled:=not ya;
  button2.Enabled:=ya;
  button1.Enabled:=ya;
end;

Procedure TForm2.Button3Click(Sender: TObject);
begin
  table1.Insert;
  entriaktif(true);
end;

Procedure TForm2.Button2Click(Sender: TObject);
begin
  table1.Post;
  entriaktif(false);
end;

Procedure TForm2.DBNavigator1Click(Sender: TObject; Button: TNavigateBtn);
begin
  if dbedit2.Text<>' ' then
    image2.Picture.LoadFromFile(dbedit2.Text)
  else
    image2.Picture:=nil;
end;

Procedure TForm2.Button4Click(Sender: TObject);
begin
  if not buka.Execute then exit
  else
  begin
    dbedit2.Text:=buka.FileName;
    image2.Picture.LoadFromFile(buka.FileName);
  end;
end;
```

```

end;

Procedure TForm2.TabSheet2Show(Sender: TObject);
begin
  if dbedit2.Text<>' ' then
    image2.Picture.LoadFromFile(dbedit2.Text)
  else
    image2.Picture:=nil;
  end;

Procedure TForm2.ExitBtnClick(Sender: TObject);
begin
  Close
end;
end.

```

## B. Unit Parameter

```

unit parameter;
Interface

var
  Parameter: TParameter;
  htemp      : array of integer;
  jum        : integer;
implementation

Procedure TParameter.FormCreate(Sender: TObject);
var a:integer;
begin
  for a:=1 to 50 do
    begin
      combobox1.Items.Add(inttostr(a));
    end;
  for a:=1 to 35 do
    begin
      combobox2.Items.Add(inttostr(a));
      combobox3.Items.Add(inttostr(a));
    end;
  end;

Procedure TParameter.OKButtonClick(Sender: TObject);
var
  a,i,j,pan,datsbl:integer;
  temp  :TdataBobot;
  temps :string;
begin
  jdata:=strtoint(combobox1.Text);
  row:=strtoint(combobox2.Text);
  col:=strtoint(combobox3.Text);
  iterasi:=5000;
  alpha:=0.001;
  setlength(identitas,jdata);
  setlength(temp,jdata);
  pan:=0;
  for a:=0 to jdata-1 do
    begin
      form1.ReadWaveFile;
      temps:=copy(extractfilename(form1.opendata1.FileName),1,length
(extractfilename(form1.opendata1.FileName))-length(extractfileext
(form1.opendata1.FileName)));
      if not inputquery('Data Name','Nama Untuk Data :',temps) then
        application.MessageBox(pchar('Anda tidak menekan tombol OK'+#13+'Character
identified as '+temps),'Confirmation',mb_ok or mb_iconexclamation);
    end;
  end;

```

```

    identitas[a]:=temps;
    pan:=max(pan,length(realdata));
    setlength(temp[a],length(realdata));
    for i:=0 to high(realdata) do
    temp[a,i]:=realdata[i];
end;
datses:=pan;
for a:=0 to jdata-1 do
begin
    datsbl:=length(temp[a]);
    if datsbl<pan then
        form1.pemampatan(temp[a]);
end;
setlength(masuk,jdata);
for a:=0 to jdata-1 do
begin
    setlength(cep,0);
    form1.OlahSinyal(temp[a]);
    setlength(masuk[a],length(cep)*length(cep[0]));
    for i:=0 to high(cep) do
    for j:=0 to high(cep[i]) do
    masuk[a,i*length(cep[i])+j]:=cep[i,j];
end;
end;
end.

```

### C. Unit Fast Fourier Transform

Berdasarkan file download dari <http://www.intersrv.com/~dcross/fft.html>  
[Don Croos, Fast Fourier Transforms]

```

unit Fourier;
Interface
implementation

function IsPowerOfTwo ( x: word ): boolean;
var
    i, y: word;
begin
    y := 2;
    for i := 1 to 15 do
    begin
        if x = y then
        begin
            IsPowerOfTwo := TRUE;
            exit;
        end;
        y := y SHL 1;
    end;
    IsPowerOfTwo := FALSE;
end;

function NumberOfBitsNeeded ( PowerOfTwo: word ): word;
var
    i: word;
begin
    for i := 0 to 16 do
    begin
        if (PowerOfTwo AND (1 SHL i)) <> 0 then
        begin
            NumberOfBitsNeeded := i;
            exit;
        end;
    end;
end;

```

```

end;

function ReverseBits ( index, NumBits: word ): word;
var
  i, rev: word;
begin
  rev := 0;
  for i := 0 to NumBits-1 do
  begin
    rev := (rev SHL 1) OR (index AND 1);
    index := index SHR 1;
  end;
ReverseBits := rev;
end;

function MakePowerOfTwo(nilai:integer ):integer;
var
  val,a:integer;
begin
  if val<=2 then
    result:=2;
    val:=2;
  repeat
    val:= val SHL 1;
  until val>=nilai;
  result:=val;
end;

Procedure FourierTransform (
AngleNumerator: double;
NumSamples: word;
var RealIn: array of double;
var ImagIn: array of double;
var RealOut: array of double;
var ImagOut: array of double );
var
  NumBits, i, j, k, n, BlockSize, BlockEnd: word;
  delta_angle, delta_ar: double;
  alpha, beta: double;
  tr, ti, ar, ai: double;
begin
  if not IsPowerOfTwo(NumSamples) or (NumSamples<2) then
  begin
    write ( 'Error in Procedure Fourier: NumSamples=', NumSamples );
    writeln ( ' is not a positive integer power of 2.' );
    halt;
  end;
  NumBits := NumberOfBitsNeeded (NumSamples);
  for i := 0 to NumSamples-1 do
  begin
    j := ReverseBits ( i, NumBits );
    RealOut[j] := RealIn[i];
    ImagOut[j] := ImagIn[i];
  end;
  BlockEnd := 1;
  BlockSize := 2;
  while BlockSize <= NumSamples do
  begin
    delta_angle := AngleNumerator / BlockSize;
    alpha := sin ( 0.5 * delta_angle );
    alpha := 2.0 * alpha * alpha;
    beta := sin ( delta_angle );
    i := 0;
    while i < NumSamples do
    begin

```

```

ar := 1.0;      (* cos(0) *)
ai := 0.0;      (* sin(0) *)
j := i;
for n := 0 to BlockEnd-1 do
begin
  k := j + BlockEnd;
  tr := ar*RealOut[k] - ai*ImagOut[k];
  ti := ar*ImagOut[k] + ai*RealOut[k];
  RealOut[k] := RealOut[j] - tr;
  ImagOut[k] := ImagOut[j] - ti;
  RealOut[j] := RealOut[j] + tr;
  ImagOut[j] := ImagOut[j] + ti;
  delta_ar := alpha*ar + beta*ai;
  ai := ai - (alpha*ai - beta*ar);
  ar := ar - delta_ar;
  INC(j);
end;
  i := i + BlockSize;
end;
BlockEnd := BlockSize;
BlockSize := BlockSize SHL 1;
end;
end;

Procedure fft (
NumSamples: word;
var RealIn: array of double;
var ImagIn: array of double;
var RealOut: array of double;
var ImagOut: array of double );
begin
  FourierTransform ( 2*PI, NumSamples, RealIn, ImagIn, RealOut, ImagOut );
end;
end.

```

#### D. Unit Kohonen

```

unit kohonen;
Interface
implementation

Procedure GenerateNetwork(var net:Tsom);
var temp:integer;
begin
  temp:=length(masuk[0]);
  net.alpha:=alpha;
  net.cols:=col;
  net.rows:=row;
  net.l_input.units:=temp;
  setlength(net.l_input.output,temp);
  setlength(net.indexwinner,jdata,2);
  net.l_kohonen.units:=net.cols*net.rows;
  setlength(net.l_kohonen.output,net.l_kohonen.units);
  setlength(net.l_kohonen.bobot,net.l_kohonen.units,temp);
end;

Procedure Randombobot(var net:Tsom);
var a,b,c:integer;
begin
  randomize;
  for a:=0 to net.l_kohonen.units-1 do
  begin
    c:=random(jdata);
    for b:=0 to net.l_input.units-1 do

```

```

    net.l_kohonen.bobot[a,b]:=masuk[c,b]+random;
end;
end;

```

```

Procedure SetInput(var net:Tsom;i:integer);
var a:integer;
begin
    for a:=0 to net.l_input.units-1 do
        net.l_input.output[a]:=masuk[i,a];
    end;

```

```

function Prop(net:Tsom;un:integer):double ;
var
    a:integer;
    sum,mag:double;
begin
    sum:=0;
    for a:=0 to net.l_input.units-1 do
        sum:=sum+sqr(net.l_input.output[a]-net.l_kohonen.bobot[un,a]);
        mag:=sqrt(sum);
    result:=mag;
end;

```

```

function Propagate(var net:Tsom):integer;
var
    a,b,winner:integer;
    mag,smallest:double;
begin
    winner:=0;
    smallest:=10000;
    for a:=0 to net.l_kohonen.units-1 do
        begin
            mag:=prop(net,a);
            net.l_kohonen.output[a]:=mag;
            if mag<smallest then
                begin
                    winner:=a;
                    smallest:=mag;
                end;
            end;
        end;
    net.winner:=winner;
    result:=winner;
end;

```

```

function Neighbor(net:Tsom;r,c,W:integer):boolean ;
var row,col,dR1,dR2,dC1,dC2:integer;
begin
    row:=(W+1) div net.cols+1;
    col:=(W+1) mod net.cols;
    dR1:=max(1,(row-net.deltaR));
    dR2:=min(net.rows,(row+net.deltaR));
    dC1:=max(1,(col-net.deltaC));
    dC2:=min(net.cols,(col+net.deltaC));
    result:=(((dR1<=r+1) and (r+1<=dR2)) and ((dC1<=c+1) and (c+1<=dC2)));
end;

```

```

function Update(var net:Tsom):integer;
var a,b,c,winner,un,upd:integer;
begin
    winner:=Propagate(net);
    un:=0;
    upd:=0;
    for a:=0 to net.rows-1 do
        for b:=0 to net.cols-1 do
            begin

```

```

    if neighbor(net,a,b,winner) then
    begin
        inc(upd);
        for c:=0 to net.l_input.units-1 do
            net.l_kohonen.bobot[un,c]:=net.l_kohonen.bobot[un,c]+net.alpha*
            (net.l_input.output[c]-net.l_kohonen.bobot[un,c]);
        end;
        inc(un);
    end;
    result:=upd;
end;
Procedure TrainNet(var net:Tsom;NP:integer);
var i,j,dum:integer;
begin
    GenerateNetwork(net);
    RandomBobot(net);
    for i:=0 to NP-1 do
    begin
        net.deltaR:=net.rows div 2;
        net.deltaC:=net.cols div 2;
        net.time:=0;
        setInput(net,i);
        while(update(net)>1) do
        begin
            inc(net.time);
            if net.time mod 10=0 then
            begin
                net.deltaR:=max(0,net.deltaR-1);
                net.deltaC:=max(0,net.deltaC-1);
            end;
        end;
    end;
    form1.ProgressBar.Position:=0;
    form1.ProgressBar.Visible:=true;
    for i:=1 to iterasi do
    begin
        net.alpha:=alpha*power(0.01,i/iterasi);
        form1.ProgressBar.Position:=trunc(form1.ProgressBar.Max*i/iterasi);
        for j:=0 to NP-1 do
        begin
            setinput(net,j);
            dum:=update(net);
            if i=iterasi then
            begin
                net.indexwinner[j,0]:=(net.winner+1) div net.cols+1;
                net.indexwinner[j,1]:=(net.winner+1) mod net.cols;
            end;
        end;
    end;
    form1.ProgressBar.Visible:=false;
end;

function Identify:integer;
var
    net:Tsom;
    a,b:integer;
begin
    GenerateNetwork(net);
    setInput(net,0);
    for a:=0 to high(bobot) do
        for b:=0 to high(bobot[a]) do
            net.l_kohonen.bobot[a,b]:=bobot[a,b];
            result:=propagate(net);
        end;
    end;
end.

```



**E. Unit Window**

```
unit window;
```

```
Interface
```

```
implementation
```

```
function FrameCount(n,m,panjang:integer):integer;
```

```
var a,jum:integer;
```

```
begin
```

```
  a:=0;jum:=0;
```

```
  repeat
```

```
    inc(jum);
```

```
    inc(a,n-m);
```

```
  until a>panjang;
```

```
  result:=jum;
```

```
end;
```

```
Procedure framing(n,m:integer;sinyal:array of double;var hasil:Tdatabobot);
```

```
Var a,b,panjang, pos :integer;
```

```
begin
```

```
  panjang:=high(sinyal)+1;
```

```
  pos:=0;
```

```
  b:=0;
```

```
  repeat
```

```
  for a:=0 to n-1 do
```

```
  begin
```

```
    if pos+a>=panjang then
```

```
      hasil[b,a]:=hasil[b,a-1]
```

```
    else
```

```
      hasil[b,a]:=sinyal[pos+a];
```

```
    end;
```

```
    inc(pos,n-m);
```

```
    inc(b);
```

```
  until pos>panjang;
```

```
  end;
```

```
Procedure pre_emphasis(koefisien:double;var sinyal:array of double);
```

```
var
```

```
  temp:array of double;
```

```
  a :integer;
```

```
begin
```

```
  setlength(temp,high(sinyal));
```

```
  for a:=1 to high(sinyal) do
```

```
    temp[a]:=sinyal[a]-koefisien*sinyal[a-1];
```

```
  for a:=1 to high(sinyal) do
```

```
    sinyal[a]:=temp[a];
```

```
end;
```

```
Procedure hamming_win(var win:array of double);
```

```
var
```

```
  arg:double;
```

```
  a:integer;
```

```
  panjang:integer;
```

```
begin
```

```
  panjang:=high(win);
```

```
  arg:= M_2PI /panjang;
```

```
  for a:=0 to panjang do
```

```
    win[a]:= 0.54 - 0.46 * cos(a * arg);
```

```
end;
```

```
Procedure win_sinyal(nflg:integer;kode:twindow;var win:array of double);
```

```
var
```

```
  a, panjang:integer;
```

```
  g:double;
```

```
begin
```

```

g:=1;
panjang:=high(win);
for a:=0 to panjang do
  win[a]:=0;
  hamming:hamming_win(win);
for a:=0 to panjang do
  win[a]:=win[a]/g;
end;
end.

```

#### F. Unit LPC

```

unit lpc;
Interface
implementation

```

```

function MakeOrder(BandWith:integer):integer;
begin
  result:=2*(BandWith div 1000+1);
end;

```

```

Procedure auto_corellation(sinyal:array of double;frame_length,p:integer;
var r:array of double);

```

```

var
  a,b:integer;
  temp :double;
begin
  for a:=0 to p do
    begin
      temp:=0;
      for b:=0 to frame_length-1-a do
        temp:=temp+sinyal[b]*sinyal[b+a];
        r[a]:=temp;
      end;
    end;
end;

```

```

function CariKoeffisienPrediksi(r:array of double;p:integer;eps:double;
var kp:array of double):integer;

```

```

var
  rmd,mue :double;
  a,b,flag:integer;
  c :array of double;
begin
  flag:=0;
  setlength(c,p+1);
  if eps<0.0 then eps:=1.0e-6;
  rmd :=r[0];
  kp[0]:=0;
  for a:=1 to p do
    begin
      mue:= -r[a];
      for b:=1 to a-1 do
        mue:=mue - c[b] * r[a - b];
        mue:= mue / rmd;
      for b:=1 to a-1 do
        kp[b]:= c[b] + mue * c[a - b];
        kp[a]:=mue;
        rmd:=(1.0 - mue * mue) * rmd;
        if rmd<0 then
          rmd:=-rmd;
        if rmd<=eps then
          begin
            result:=1;
            exit;
          end;
      end;
    end;
end;

```

```

    end;
    if mue<0 then
        mue:=-mue;
    if mue>=1 then
        flag:=2;
    for b:= 0 to a do
        c[b]:=kp[b];
    end;
    kp[0]:=sqrt(rmd);
    result:=flag;
end;

```

```

function Gain(p:integer;a:array of double;r:array of double):double;
var
    b:integer;
    temp:double;
begin
    temp:=0;
    for b:=1 to p do
        temp:=temp+a[b]*r[b];
        temp:=r[0]-temp;
        result:=sqrt(temp);
    end;
end;

```

```

function LPCAnalisis(sinyal:array of double;framelength,p:integer;
var a:array of double):integer;
var
    r, prediksi:array of double;
    flag,b,c:integer;
    temp :double;
begin
    setlength(r,p+1);
    setlength(prediksi,framelength);
    auto_corellation(sinyal,framelength,p,r);
    flag:=CariKoefisienPrediksi(r,p,-1,a);
    for b:=1 to framelength-1 do
        begin
            temp:=0;
            for c:=1 to p do
                if b-c>=0 then
                    temp:=temp+sinyal[b-c]*a[c];
                    prediksi[b]:=temp;
            end;
        result:=flag;
    end;
end;

```

```

Procedure lpc2cepstral(p1,p2:integer;a:array of double;var c:array of double);
var
    i,j,k :integer;
    temp :double;
begin
    c[0]:=log10(a[0]);
    c[1]:=-a[1];
    for i:=2 to p2 do
        begin
            j:=i;
            if i>p1 then k:=i-p1
            else k:=1;
            temp:=0;
            repeat
                temp:=temp+k*c[k]*a[i-k];
                inc(k);
            until k>=j;
            c[i]:=-temp/i;
            if i<=p1 then c[i]:=c[i]-a[i];
        end;
    end;
end;

```

```

    end;
end;

Procedure cepstralBobot(p:integer;var c:array of double);
var
    a:integer;
    w:array of double;
    arg:double;
begin
    setlength(w,p+1);
    arg:=M_PI/p;
    for a:=1 to p do w[a]:=1+(p/2)*sin(a*arg);
    for a:=1 to p do c[a]:=c[a]*w[a];
end;
end.

```

### G. Unit Utama

```

unit main;
Interface

const
WAVE_BUFSIZE = 6144;
FOURCC_WAVE = $45564157; { 'WAVE' }
FOURCC_FMT = $20746d66; { 'fmt ' }
FOURCC_FACT = $74636166; { 'fact' }
FOURCC_DATA = $61746164; { 'data' }
WindowCaption = 'Pengenalan Suara dengan Gambar Buah Menggunakan Metode
    Kohonen SOM';
UkuranFrame = 512;

var
Form1: TForm1;
panjang,sequen:integer;
ulang,jumbit,modechan:byte;
confe : double;
databyte : array of byte;
frek,maksval,posisi : integer;
realdata : array of double; //data dalam domain waktu
cep : Tdatabobot;
kunci : string;

//data untuk kohonen
iterasi,row,col,jdata,datses,iunit : integer;
alpha : double;
masuk,bobot : Tdatabobot;
identitas : array of string;
theWinner : array of array of integer;

implementation
Uses paramtra, window, lpc, kohonen, data;

Procedure TForm1.FormCreate(Sender: TObject);
begin
    FRecorderMode:= recModeOff;
    FFilename:= 'baru.wav';
    if InitWaveRecorder <> 0 then Application.Terminate;
    confe:=strtoint(edit.Text)/100;
end;

function TForm1.StartWaveRecord : Integer;
var Status : MMRESULT;
begin

```

```

FTotalWaveSize := 0;
FByteDataSize := 0;
FBufIndex := 0;

Status := waveInOpen(@FWaveIn, WAVE_MAPPER, FWaveFormat,
                    Handle, 0, CALLBACK_WINDOW);
if Status <> MMSYSERR_NOERROR then begin
  ierrormsg('Could not open the input device for recording.');
```

Result := -1;

```
Exit;
end;
```

```
Procedure TForm1.Timer1Timer(Sender: TObject);
begin
  if FRecorderMode = recModeRecord then UpdateRecordDisplay
  else
    if FRecorderMode = recModePlay then UpdatePlayDisplay;
end;
```

```
Procedure TForm1.RecordButtonClick(Sender: TObject);
begin
  rekam:=true;
  if FRecorderMode <> recModeOff then StopWaveRecord
  else // start recording...
    begin
      statusbarsignal.Panels[0].Text:='Recording Sound';
      posisi:=0;
      setlength(databyte,posisi);
      StartWaveRecord;
    end;
end;
```

```
Procedure TForm1.TampilanGrafik(isidata:array of double);
var a:integer;
begin
  form1.Chart1.SeriesList[0].Clear;
  for a:=0 to high(isidata) do
    form1.Chart1.SeriesList[0].AddXY(1000*a/frek,isidata[a],',',clgray);
end;
```

```
function TForm1.QueryUserSave(const name : String) : Integer;
begin
  Result := Application.MessageBox(PChar('File ' + name + ' exists,
  overwrite?'),'Save File', MB_YESNOCANCEL);
end;
```

```
Procedure TForm1.BacaFileTemp;
var
  fromf : file;
  buf : array[1..512000] of byte;
  a,numread : longint;
begin
  assignfile(fromF,FTmpFileName);
  {$I-} reset(fromF,1);
  {$I+};
  BlockRead(FromF, Buf, SizeOf(Buf), NumRead);
  setlength(databyte,numread);
  for a:=0 to numread-1 do databyte[a]:=buf[a+1];
  closefile(fromF);
end;
```

```
function TForm1.StartWavePlay : Integer;
begin
  Result := -1;
```

```

if FDeviceOpened then Exit;
FByteDataSize := 0;
FBufIndex      := 0;

// open the device for recording...
if waveOutOpen(@FWaveOut, WAVE_MAPPER, FWaveFormat,
  Handle, 0, CALLBACK_WINDOW or WAVE_ALLOWSYNC) <> 0 then
begin
  ierrormsg('Error opening wave out device. ');
  Result := -1;
  Exit;
end;
FDeviceOpened := TRUE;

// prepare the headers...
InitWaveHeaders;
if (waveOutPrepareHeader(FWaveOut, FWaveHdr[0], sizeof(TWAVEHDR)) <> 0) or
  (waveOutPrepareHeader(FWaveOut, FWaveHdr[1], sizeof(TWAVEHDR)) <> 0) then
begin
  CloseWaveDevicePlay;
  ierrormsg('Error preparing header for playing. ');
  Result := -2;
  Exit;
end;
end.

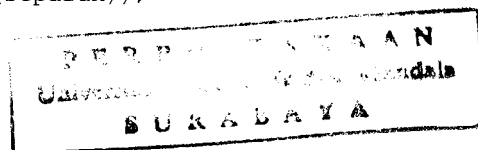
Procedure TForm1.SizeButtonClick(Sender: TObject);
var olah,separuh:string;
begin
  GetWaveFormat;
  StatusbarSignal.Panels[1].Text:=' '+FFormatTag+' '+FFormatDesc;
  separuh:=FFormatDesc;
  olah:=copy(separuh,0,pos(',',',',separuh)-5);
  frek:=round(strtoint(olah)*1000);
  separuh:=copy(separuh,pos(',',',',separuh)+1,length(separuh));
  olah:=copy(separuh,1,pos(',',',',separuh)-5);
  jumbit:=strtoint(olah);
  if jumbit=8 then maksval:=127;
  if jumbit=16 then maksval:=32768;
  separuh:=copy(separuh,pos(',',',',separuh)+2,length(separuh));
  if separuh='Mono' then modechan:=1
  else
    modechan:=2;
end;

Procedure TForm1.SimpanSuaraClick(Sender: TObject);
begin
  SaveWaveFile(smSave);
end;
Procedure TForm1.Replace1Click(Sender: TObject);
begin
  SaveWaveFile(smSaveAs);
end;

Procedure TForm1.BukaSuaraClick(Sender: TObject);
begin
  ReadWaveFile;
  Rekam:=false;
  Playbutton.Enabled:=true;
  SignalButton.Enabled:=true;
  InputButton.Enabled:=true;
end;

Procedure TForm1.Parameter1Click(Sender: TObject);
begin

```



```

parameter:=Tparameter.Create(self);
parameter.ShowModal;
end;

```

```

Procedure TForm1.OlahSinyal(sinyal:array of double);

```

```

var p,i,jfrm,j           : integer;
    win,aut              : array of double;
    frekdata            : array of double;
    realtime,imgtime: Tdatabobot;
    realfrek,imgfrek: Tdatabobot;
    : integer;

```

```

begin

```

```

    jfrm:=FrameCount(UkuranFrame,UkuranFrame div 2,high(sinyal)+1);

```

```

    setlength(realtime,jfrm);

```

```

    setlength(imgtime,jfrm);

```

```

    setlength(realfrek,jfrm);

```

```

    setlength(imgfrek,jfrm);

```

```

    setlength(cep,jfrm);

```

```

    pre_emphasis(0.94,sinyal);

```

```

    setlength(realtime,jfrm);

```

```

    for i:=0 to jfrm-1 do

```

```

        begin

```

```

            setlength(realtime[i],UkuranFrame);

```

```

            setlength(imgtime[i],UkuranFrame);

```

```

            setlength(realfrek[i],UkuranFrame);

```

```

            setlength(imgfrek[i],UkuranFrame);

```

```

        end;

```

```

        framing(UkuranFrame,UkuranFrame div 2,sinyal,realtime);

```

```

        setlength(win,UkuranFrame);

```

```

        win_sinyal(0,hamming,win);

```

```

    for i:=0 to jfrm-1 do

```

```

        for j:=0 to UkuranFrame-1 do

```

```

            realtime[i,j]:=realtime[i,j]*win[j];

```

```

    for i:=0 to jfrm-1 do

```

```

        fft(UkuranFrame,realtime[i],imgtime[i],realfrek[i],imgfrek[i]);

```

```

    for i:=0 to jfrm-1 do

```

```

        for j:=0 to UkuranFrame-1 do

```

```

            realfrek[i,j]:=log10(sqrt(sqrt(realfrek[i,j])+sqrt(imgfrek[i,j])));

```

```

    for i:=0 to jfrm-1 do

```

```

        setlength(realfrek[i],length(realfrek[i]) div 2);

```

```

p:=16;

```

```

setlength(aut,p+1);

```

```

for i:=0 to jfrm-1 do

```

```

    begin

```

```

        setlength(cep[i],p+1);

```

```

        LPCAnalisis(realfrek[i],length(realfrek[i]),p,aut);

```

```

        lpc2cepstral(p,p,aut,cep[i]);

```

```

        cepstralBobot(p,cep[i]);

```

```

    end;

```

```

end;

```

```

Procedure TForm1.SignalButtonClick(Sender: TObject);

```

```

begin

```

```

    chart1.SeriesList[0].Clear;

```

```

    if rekam then bacafiletemp;

```

```

    TampilanGrafik(realdata);

```

```

end;

```

```

Procedure TForm1.Train1Click(Sender: TObject);

```

```

var a,b:integer;

```

```

    dum:string;

```

```

    net:Tsom;
begin
iunit:=length(masuk[0]);
for a:=0 to 1 do
    begin
        application.ProcessMessages;
        statusbarsignal.Panels[0].Text:=' Process Kohonen..';
    end;
TrainNet(net,jdata);
dum:='['+inttostr(net.indexwinner[0,0])+','+inttostr(net.indexwinner[0,1])+']'
;
for a:=1 to jdata-1 do
    begin
        dum:=dum+'
'+['+inttostr(net.indexwinner[a,0])+','+inttostr(net.indexwinner[a,1])+']';
        if a mod 6=0 then
            dum:=dum+#13;
        end;
    end;
pesan('Hasil Clustering'+#13+dum,'Clustering Information');
setlength(bobot,net.l_kohonen.units,net.l_input.units);
for a:=0 to net.l_kohonen.units-1 do
    for b:=0 to net.l_input.units-1 do
        bobot[a,b]:=net.l_kohonen.bobot[a,b];
    end;
setlength(theWinner,jdata,2);
for a:=0 to jdata-1 do
    for b:=0 to 1 do
        theWinner[a,b]:=net.indexwinner[a,b];
    end;
end;

function caripos(s:string):integer;
var a:integer;
    temp:string;
begin
    a:=1;
    repeat
        temp:=copy(s,a,length(s)-a+1);
        inc(a);
    until ansistrscan(pchar(temp),'=')=nil;
    result:=a;
end;

Procedure TForm1.pemampatan(var temp:T1dimensi);
var a,b,cur,num:integer;
    pembagi:double;
    sem :T1dimensi;
    val :double;
begin
    cur:=length(temp);
    if cur<datses then
        pembagi:=datses/cur
    else
        pembagi:=cur/datses;
    setlength(sem,datses);
    b:=1;
    if cur<datses then
        begin
            for a:=1 to datses do
                begin
                    sem[a-1]:=temp[b-1];
                    if (b*pembagi)-0.5<a then
                        inc(b);
                    end;
                end;
            else
                begin

```



```

val:=0;num:=0;
for a:=1 to cur do
  begin
    val:=val+temp[a-1];
    inc(num);
    if (b*pembagi)-0.5<a then
      begin
        sem[b-1]:=val/num;
        inc(b);
        val:=0;num:=0;
      end;
    end;
  end;
setlength(temp,datses);
for a:=0 to datses-1 do
  temp[a]:=sem[a];
end;

Procedure TForm1.InsDataClick(Sender: TObject);
begin
  form2:=tform2.create(self);
  form2.showmodal;
end;

Procedure TForm1.soalbuttonClick(Sender: TObject);
var jum,num,a:integer;
begin
  query1.Close;
  query1.SQL.Clear;
  query1.SQL.Add('select count(*) from databuah.db');
  query1.Open;
  jum:=query1.Fields[0].AsInteger;
  query1.Close;
  query1.SQL.Clear;
  query1.SQL.Add('select * from databuah.db');
  query1.Open;
  randomize;
  num:=1+random(jum);
  query1.First;
  for a:=1 to num-1 do
    query1.Next;
    Gambar1.Picture.LoadFromFile(query1.Fields[1].AsString);
    kunci:=query1.Fields[0].AsString;
end;

Procedure TForm1.SimpanBobotClick(Sender: TObject);
var a,b :integer;
    F :textfile;
begin
if not Simdatses.Execute then exit
else
  begin
    AssignFile(F,Simdatses.FileName); Rewrite(F);
    Writeln(F,'The Result of Mapping Speech Recognition Using Kohonen SOM');
    Writeln(F,'Made On --> '+ FormatDateTime('d mmmmm yyyy',now));
    Writeln(F,'Jumlah Data           = '+inttostr(jdata));
    Writeln(F,'Panjang Sample       = '+inttostr(datses));
    Writeln(F,'Jumlah unit input      = '+inttostr(iunit));
    Writeln(F,'Jumlah kolom kohonen   = '+inttostr(col));
    Writeln(F,'Jumlah baris kohonen    = '+inttostr(row));
    for a:=0 to jdata-1 do
      Writeln(F,'The Winner '+inttostr(a+1)+' is = '+inttostr(theWinner[a,0])+
'+inttostr(theWinner[a,1]));
    for a:=1 to jdata do
      Writeln(F,'Identitas data ke '+inttostr(a)+' = '+identitas[a-1]);
  end;

```

```

for a:=0 to high(bobot) do
  for b:=0 to high(bobot[a]) do
    Writeln(F, floattostr(bobot[a,b]));
  Writeln(F, '');
  closefile(F);
  end;
end;

Procedure TForm1.BukaBobotClick(Sender: TObject);
var a,b,c,d,jhidden,sisa: integer;
    F : TextFile;
    s : string;
    hasil : array of string;
begin
  if not OpenData2.Execute then exit
  else
  begin
  AssignFile(F, OpenData2.FileName);
  Reset(F);
  Readln(F, s);
  if s<>'The Result of Mapping Speech Recognition Using Kohonen SOM' then
  begin
    application.MessageBox('File tak berisi data Kohonen'+#13+'Try open
    another file', 'Error on opening file', mb_ok or mb_iconstop);
    exit;
  end;
  Readln(F, s); jdata:=strtoint(copy(s, caripos(s), length(s)));
  Readln(F, s); datses:=strtoint(copy(s, caripos(s), length(s)));
  Readln(F, s); iunit:=strtoint(copy(s, caripos(s), length(s)));
  Readln(F, s); col:=strtoint(copy(s, caripos(s), length(s)));
  Readln(F, s); row:=strtoint(copy(s, caripos(s), length(s)));
  setlength(theWinner, jdata, 2);
  setlength(hasil, 2);
  for a:=1 to jdata do
  begin
    readln(F, s);
    pecah(copy(s, caripos(s), length(s)), hasil);
    theWinner[a-1, 0]:=strtoint(hasil[0]);
    theWinner[a-1, 1]:=strtoint(hasil[1]);
  end;
  setlength(identitas, jdata);
  for a:=1 to jdata do
  begin
    readln(F, s);
    identitas[a-1]:=copy(s, caripos(s), length(s));
  end;
  setlength(bobot, col*row, iunit);
  for a:=0 to high(bobot) do
  for b:=0 to high(bobot[a]) do
  begin
    Readln(F, s);
    bobot[a,b]:=strtofloat(s);
  end;
  closefile(F);
  end;
end;

function GetDecision(win:integer):string ;
var a,flag:integer;
    wcol,wrow:integer;
    dist,smallest:double;
begin
  wrow:=(win+1) div col;
  wcol:=(win+1) mod col;
  smallest:=1000;

```

```

flag:=0;
for a:=0 to high(thewinner) do
  begin
    dist:=sqrt(sqr(theWinner[a,0]-wrow)+sqr(theWinner[a,1]-wcol));
    if smallest>=dist then
      begin
        smallest:=dist;
        flag:=a;
      end;
    end;
  result:=identitas[flag];
end;

```

```

Procedure TForm1.InputButtonClick(Sender: TObject);
var i,j,datsbl,win      : integer;
      temp              : Tldimensi;
begin
  bacafiletemp;
  datsbl:=length(Realdata);
  setlength(temp,datsbl);
  for i:=0 to datsbl-1 do temp[i]:=realdata[i];
  if datsbl<>datses then
    begin
      pemampatan(temp);
    end;
  setlength(masuk,1);
  setlength(cep,0);
  OlahSinyal(temp);
  setlength(masuk[0],iunit);
  for i:=0 to high(cep) do
    for j:=0 to high(cep[i]) do
      masuk[0,i*length(cep[i])+j]:=cep[i,j];
  win:=identify;
  pesan('Input      '+GetDecision(win)+'#13+'Gambar      '+kunci,'Your      answer
Information');
end;

```

```

Procedure TForm1.About1Click(Sender: TObject);
begin
  Close;
end;
end.

```

## BIODATA

Nama : Hernawati  
Nrp : 5103097008  
Alamat : Petemon II / 94D Surabaya  
Tempat/tgl lahir : Pematang Siantar, 28-11-1978  
Agama : Budha  
Hobby : Membaca

### Riwayat pendidikan:

- Tahun 1985-1989 di SD Methodis, L.Pakam - Medan.
- Tahun 1989-1991 di S.D.K. Don Bosco, Surabaya.
- Tahun 1991-1994 di S.M.P.K. Vincentius, Surabaya.
- Tahun 1994-1997 di S.M.U.K. St.Maria, Surabaya.
- Tahun 1997-2002 di Universitas Katolik Widya Mandala Fakultas Teknik Jurusan Teknik Elektro, Surabaya.

