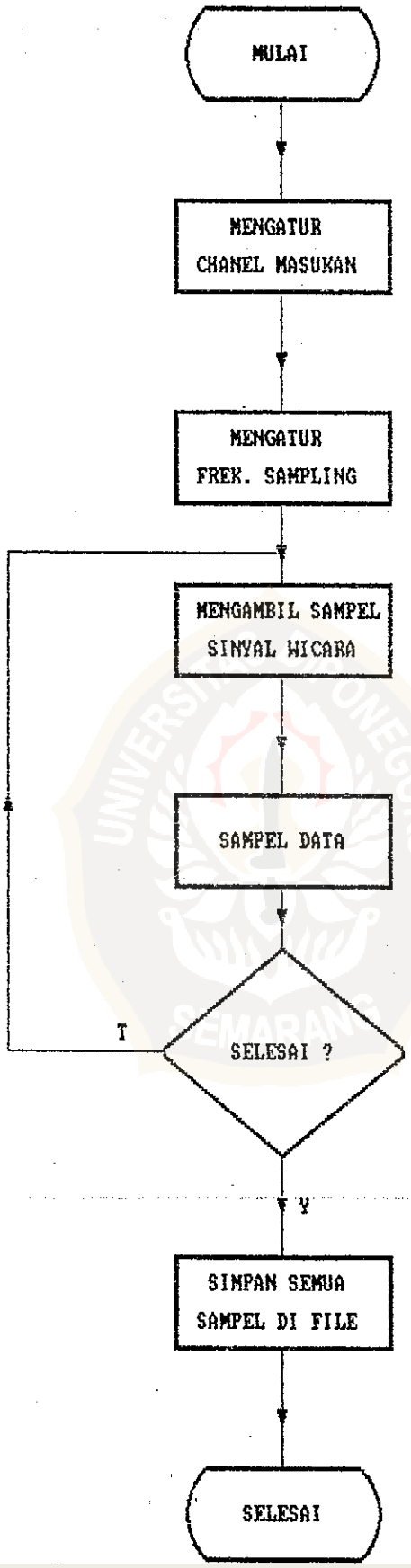
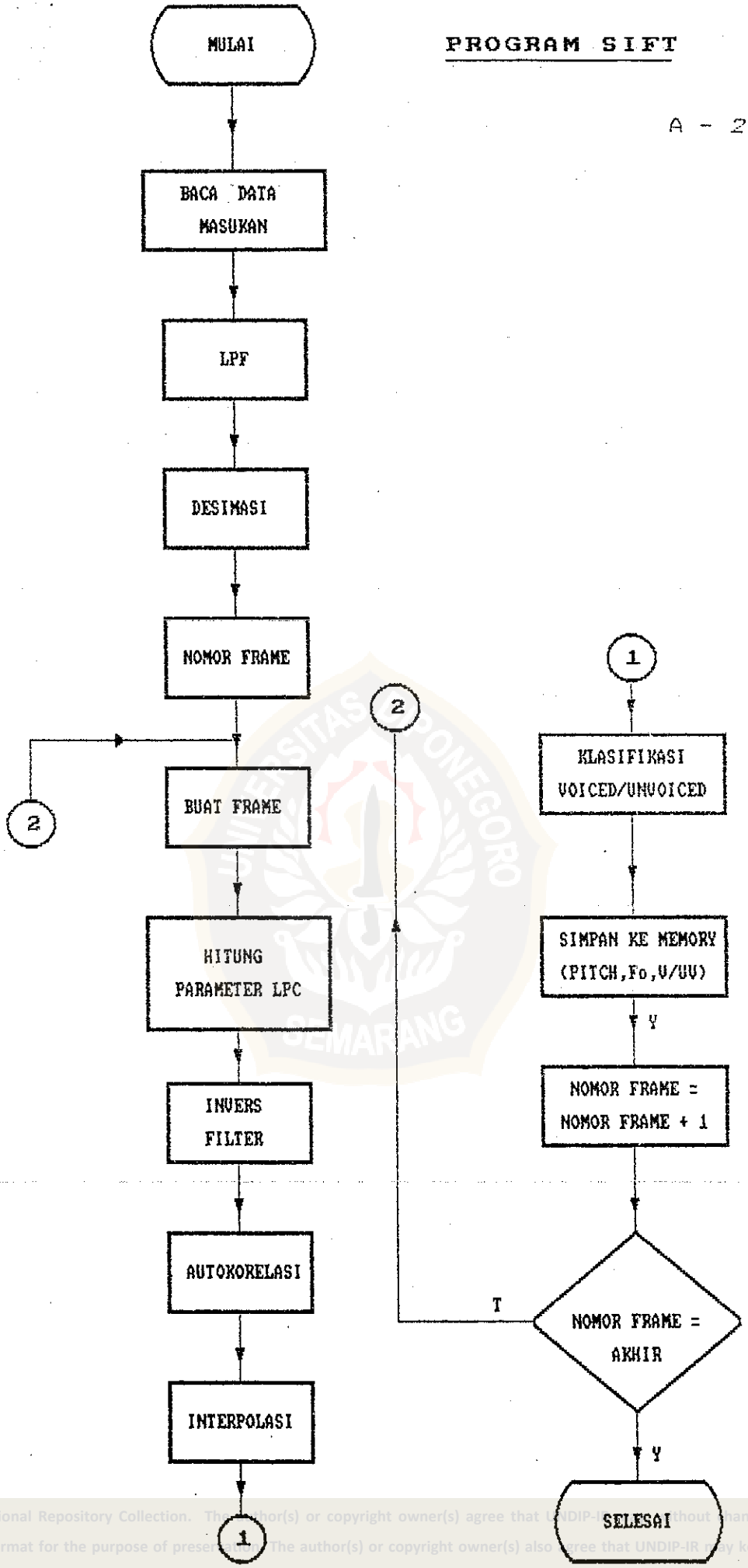
AMBIL DATA

A - 1

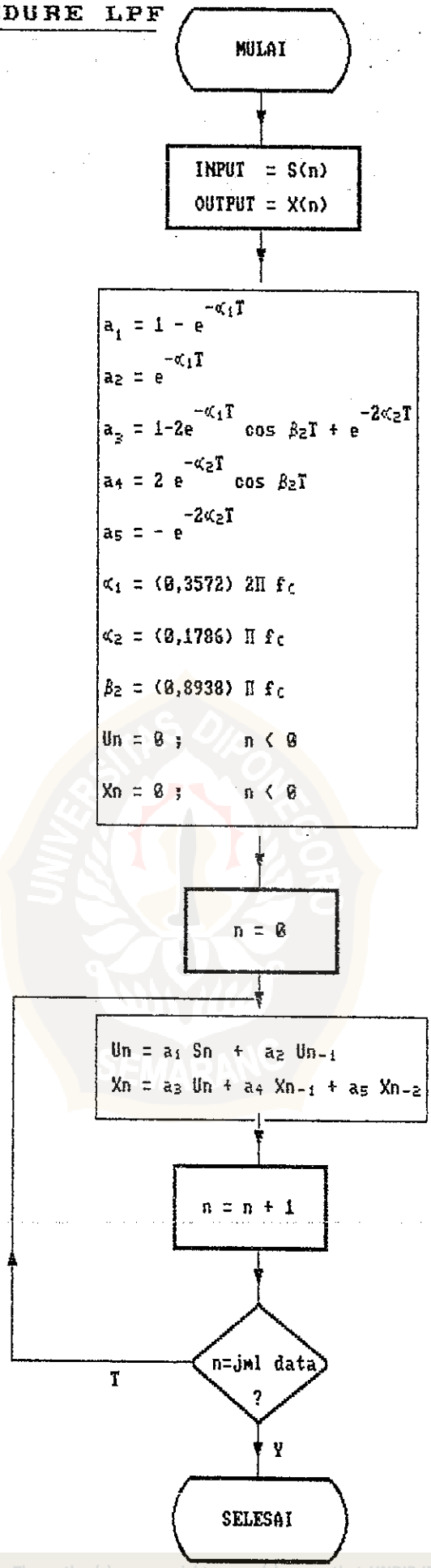


PROGRAM SIFT

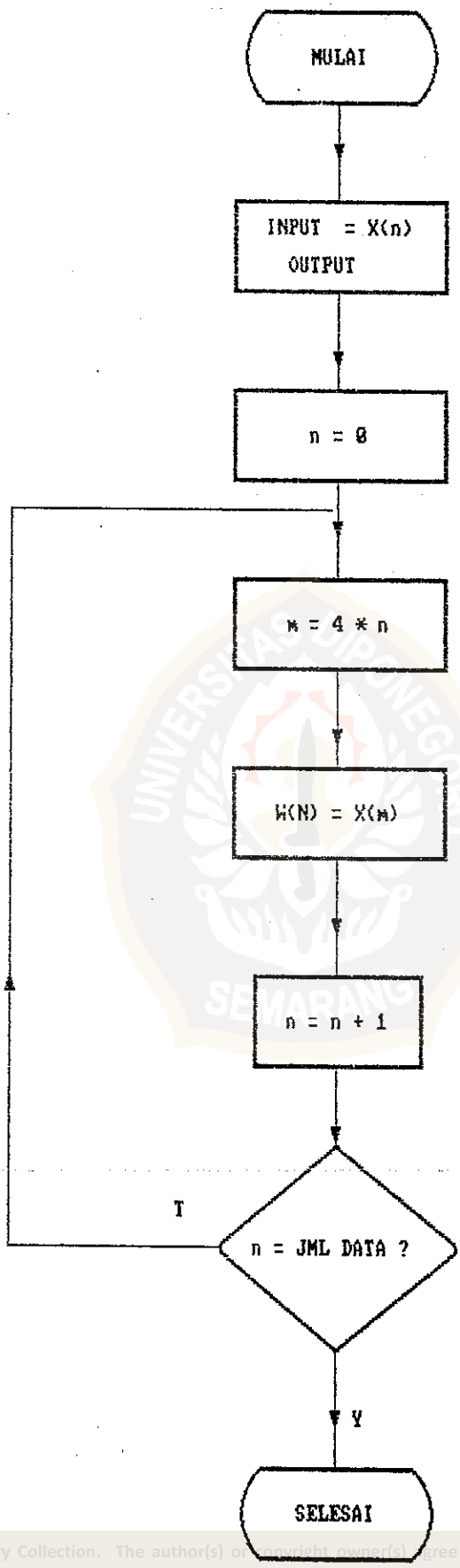
A - 2



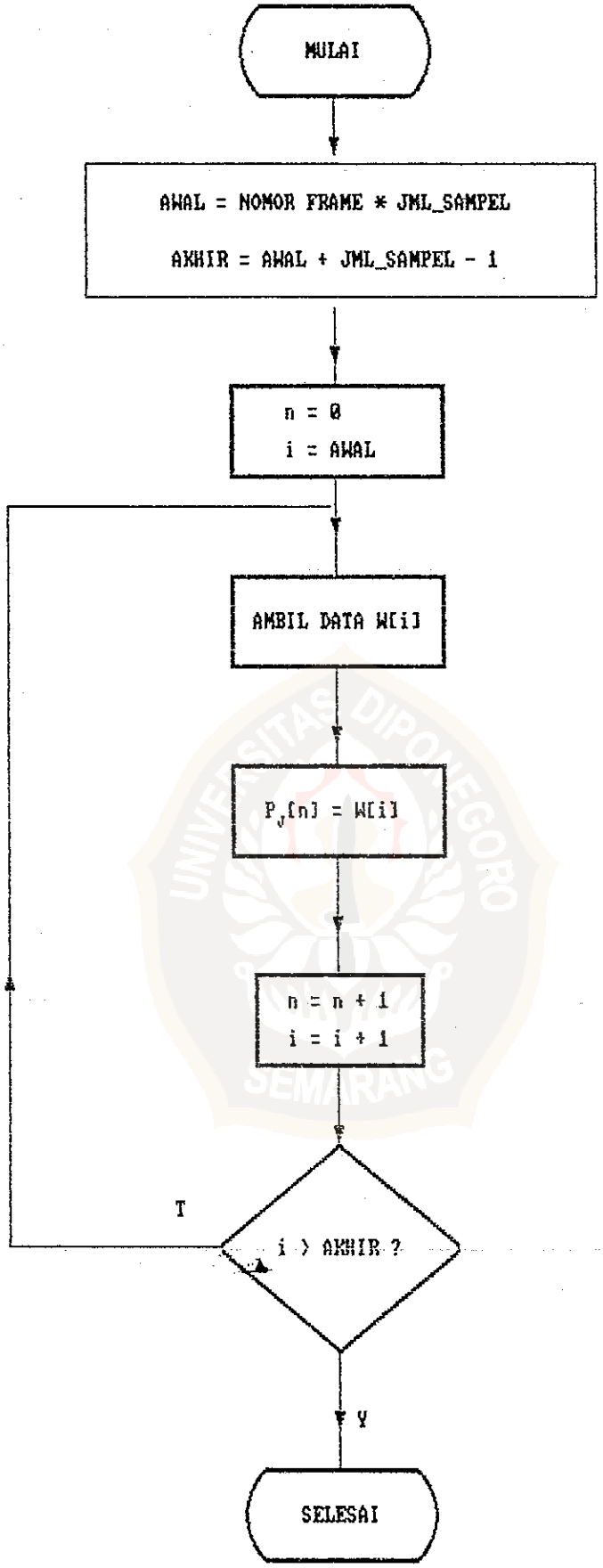
PROSEDURE LPF



PROSEDURE DESIMASI



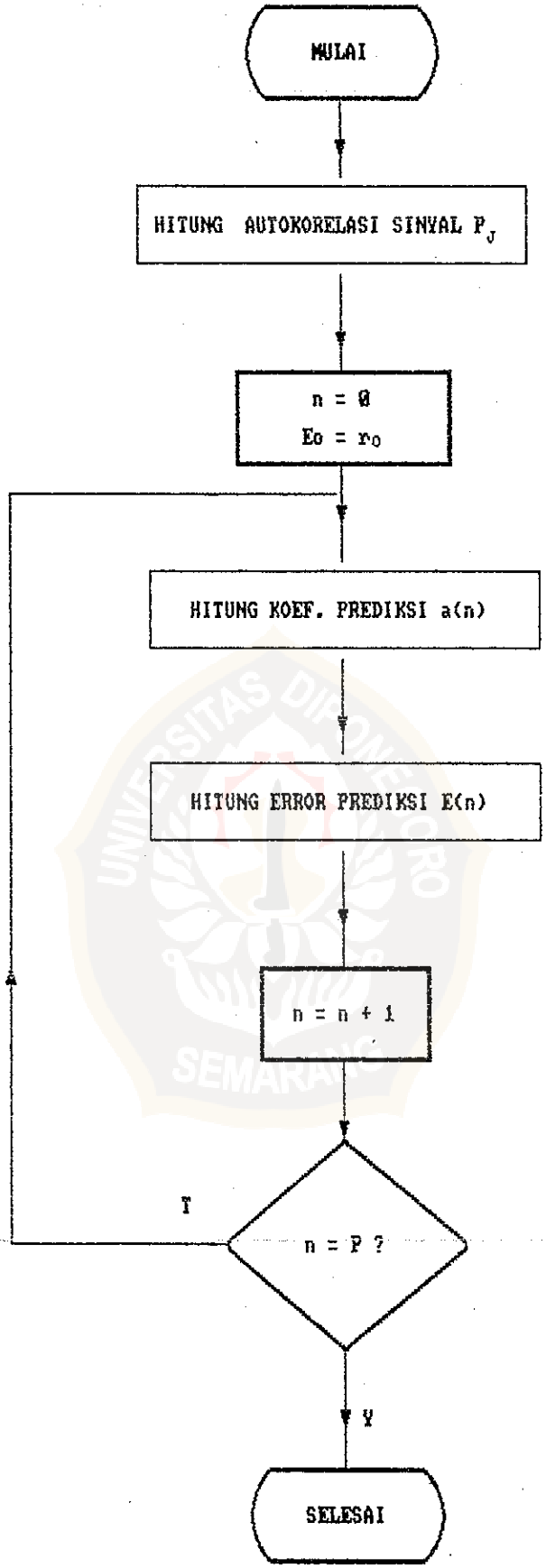
PROSEDURE BUAT FRAME



PROSEDURE

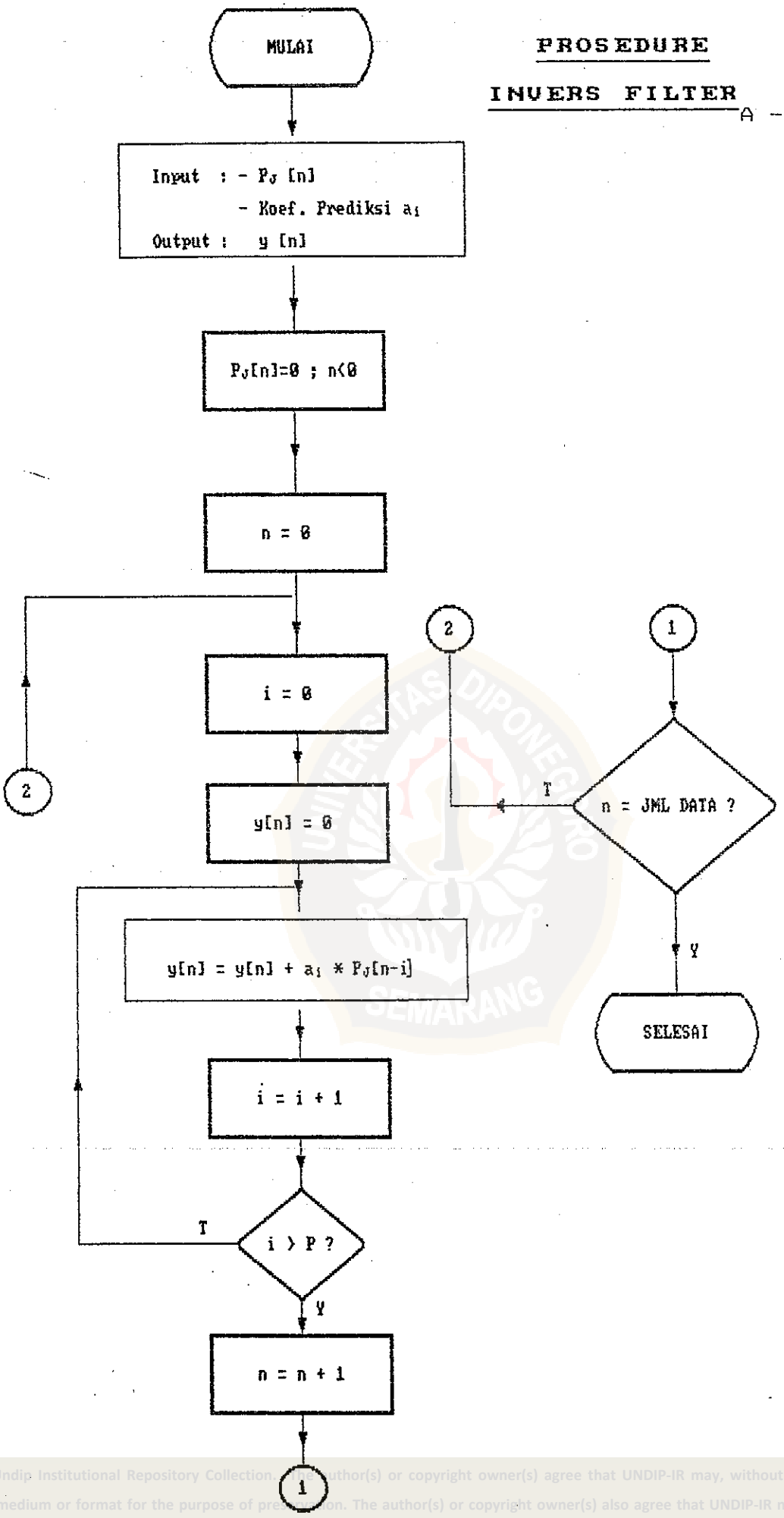
HITUNG PARAMETER LPC

A - 6

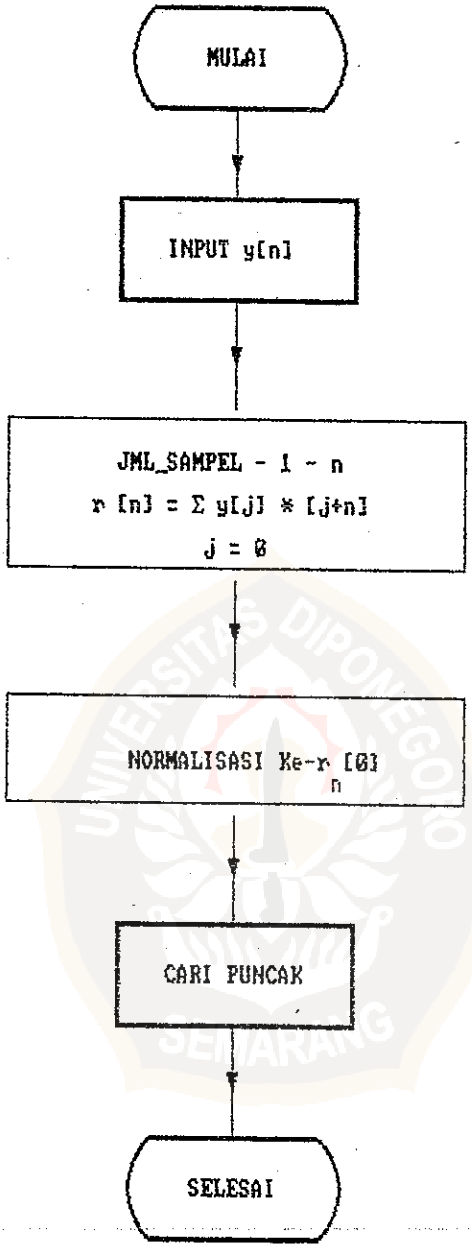


PROSEDURE INVERS FILTER

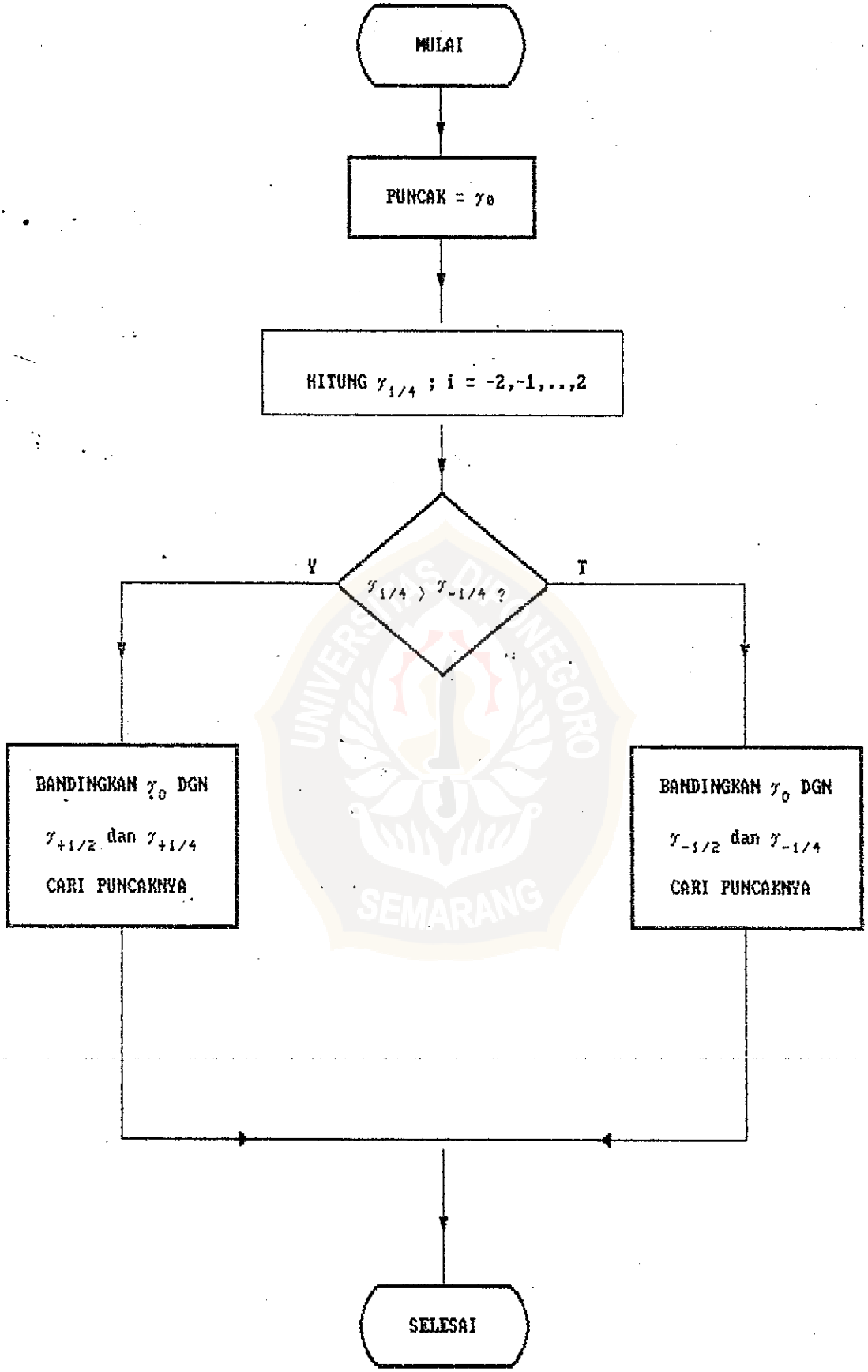
A - 7



PROSEDURE AUTOKORELASI

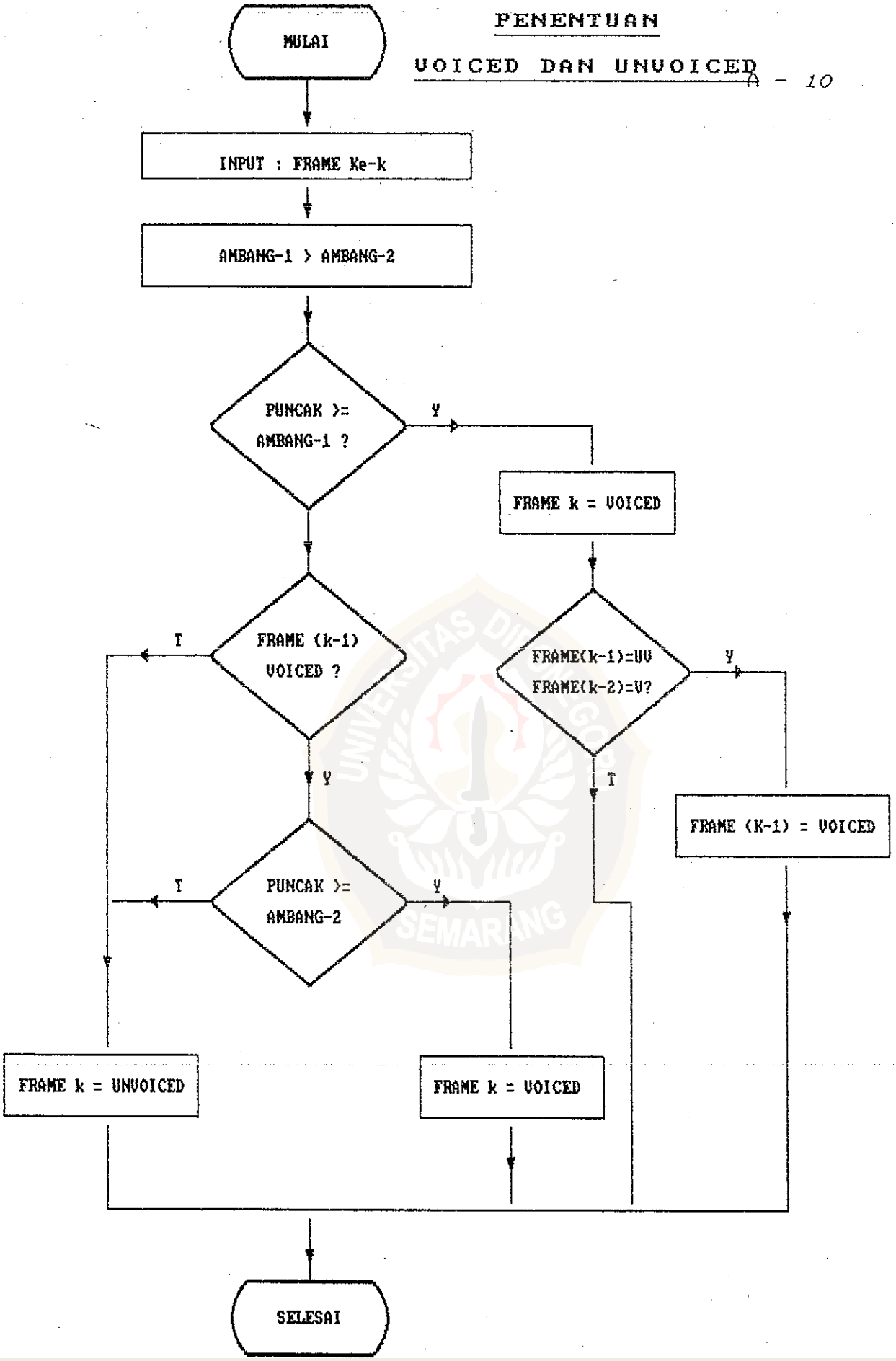


PROSEDURE INTERPOLASI



PENENTUAN VOICED DAN UNVOICED

A - 10



```
program Voice_Recognizer;


uses crt, dos;

const
  data_max                      = 10000;

type
  soundrec      = record
                    idlebyte    : byte;
                    databyte    : array[0..(data_max-1)] of integer;
                    datalength  : integer;
                    CRCValue    : longint;
                  end;

var
  FileData                      : file of soundrec;
  Larik                         : soundrec;
  IdleCount, NotInidle          : integer;
  Nama_File                     : string[14];


  {******* procedure untuk mereset DSP dari Sound Blaster *******}
procedure DSPinit;
begin
   port[$226]                   := 1;
   delay(3);
   port[$226]                   := 0;
   delay(3);
end;


  {****** function untuk mendektesi keberadaan Sound Blaster *****}
Function SBExist                :  boolean;
var
   loop                         :  byte;

   begin
     for loop                   := 1 to 100 do
       begin
         if port[$22c]           = $AA then
            SBExist             := true;
         Exit;
       end;
     SBExist                    := false;
   end;


   {*** procedure utk memberikan perintah pd DSP Sound Blaster ***}
Procedure DSPWrite(data : byte);
begin
  repeat;
  until port[$22c] and 128 <> 128;
  port[$22c]                      := data;
end;
```

```
   { Function untuk mendapat input dari Sound Blaster }
Function GetADC                    :  byte;
begin
  DSPWrite($20);
  repeat;
  until port[$22E] and 128=128;
  getADC                          := port[$22A];
end;


    {procedure untuk mendeteksi nilai byte saat Mic tidak ada}
procedure SetIdle( var Recs     :  SoundRec);
begin
  repeat;
    DSPWrite($20);
  until port[$22E] and 128=128;
  Recs.IdleByte                   := port[$22E];
end;


    {procedure untuk mengubah seting recording input
       (menjadi input from MIC)}
procedure SetMic;
begin
  port[$224]                      := $0;
  port[$225]                      := $0;
  port[$224]                      := $0C;
  port[$225]                      := $0C;
end;


    {procedure untuk memberikan waktu delay}
procedure Wait(delays : integer);
var
  index                           :  integer;
begin
  for index                       := 0 to delays do
end;


    {procedure untuk kalkulasi suara melalui Mikrophone}
procedure ReadVoice;
var
  get, index                      :  integer;
begin
  setMic;
  setIdle(Larik);
  Larik.IdleByte              := GetADC;
  repeat;
     Get                          := GetADC;
  Until (Get < 20) or (Get > 230);

  writeln(' Begin Recording .............');
  Index                           := 0;
  Larik.CRCValue              := 0;
  IdleCount                       := 0;
```

```
  repeat;
    Larik.DataByte[Index]    := GetADC;
    if (Larik.DataByte[Index] < 70) or
       (Larik.DataByte[Index] > 190) then
          IdleCount             := 0;
    inc(IdleCount);
    inc(index);
    inc(Larik.CRCValue,Larik.DataByte[Index]);
    wait(30);
  until (index = 20000) or (IdleCount = 1000);
  Larik.DataLength           := Index;
end;


   {procedure penghandle perekaman }
procedure MainHandle;
begin
  writeln(' Say something ..........!!');
  ReadVoice;
  writeln(Larik.DataLength,'bytes Recorder');
  write('Data akan disimpan diFile apa ? ');
  read(Nama_File);
  writeln('Proses perekaman sedang dimulai ');
  delay(3000);
  assign(FileData,Nama_File);
  rewrite(FileData);
  write(FileData,Larik);
  close(FileData);
  write(' Recorded CRC        : ',Larik.CRCValue);
end;




   { program utama }
begin
  dspinit;
  if not SBExist then
  begin
    writeln('Sound Blaster not found ');
    writeln('Please make sure that your Sound Card is set on port 0x220');
    halt;
  end;
  MainHandle;
end.
```

```
PROGRAM SIFT;

USES CRT, GRAPH;

CONST
    data_max        = 10000;
    jml_samp_asli   =   240;
    jml_samp        =    60;
    desi            =  2500;
    lpc_length      =     4;
    array_max       =    41;
    frame_max       =    40;
    data_kursor     : array[0..103] of byte = (0,0,100,0,64,64,64,64,64,
                      64,64,64,64,64,64,64,64,64,64,64,64,64,64,64,64,64,64,
                      64,64,64,64,64,64,64,64,64,64,64,64,64,64,64,64,64,64,
                      64,64,64,64,64,64,64,64,64,64,64,64,64,64,64,64,64,64,
                      64,64,64,64,64,64,64,64,64,64,64,64,64,64,64,64,64,64,
                      64,64,64,64,64,64,64,64,64,64,64,64,64,64,64,64,64,64,
                      64,64,64,64,64,64,64,64,64,64);
TYPE
    keytype         = (F1, F2, UpArrow, DownArrow, RightArrow, LeftArrow,
                      EndKey, HomeKey,  SpaceKey, PgUp, PgDn, EscapeKey);
    larik_real      = array [ 0..(data_max - 1)] of    real;
    larik_2         = ^larik_real;
    larik           = array [ 0..(data_max - 1)] of integer;
    larik_1         = ^larik;
    satu_frm        = array [ 0..(jml_samp - 1)] of integer;
    satu_frm_asli   = array [ 0..(jml_samp_asli - 1 )] of integer;
    array_1         = array [ 0 ..lpc_length ] of real;
    array_2         = array [ 0..jml_samp ] of real;
    array_5         = array [ 0..(jml_samp - 1 )] of real;
    file_array      = file of larik;
    tulisan         = string   [14];
    hasil           = record
                        sn              : satu_frm_asli;
                        pitch           : real;
                        fo              : real;
                        v_uv            : boolean;
                      end;
    hasil_array     = array [0..(array_max)] of hasil;

VAR
    s_inp           : larik_1;
    x               : larik_2;
    w               : array[0..(desi - 1)] of real;
    pp              : array_1;
    file_input      : file_array;
    no_frm, peak    : integer;
    nama_file       : tulisan;
    periode, frekuensi  : array [0..(array_max - 1)] of real;
    vuv             : array [0..(array_max - 1)]  of  boolean;
    data_hasil      : hasil_array;
    gamma_0, gamma_min1, gamma_plus1, puncak, lokasi_puncak : real;
    y               : array_2;
    pj, ubuf        : array_5;
    pbuf, auto_norm: array [0..(jml_samp - 5)] of real;
    rn              : array [0..(jml_samp - 4)] of real;
```

```
    warna, frame_gambar, kursor_akhir  : integer;
    salib              : pointer;
    key                : keytype;
    ch                 : char;
    fkg, selesai       : boolean;
    p1, p2, p3, maks   : real;

 { ************************************************************************ }

Procedure Inisialisasi_1;
  Begin
    Fillchar (s_inp^, sizeof (s_inp^),0);
    Fillchar (x^,  sizeof (x^), 0);
    Fillchar (w, sizeof(w), 0);
    Fillchar (data_hasil, sizeof (data_hasil), 0);
    Fillchar (periode,  sizeof (periode), 0);
    Fillchar (frekuensi,sizeof (frekuensi),0);
    Fillchar (VUV,sizeof (VUV),0);
    Fillchar (PP,sizeof  (PP),0);
    Fillchar (y ,sizeof  (y), 0);
    Fillchar (Pj,sizeof  (Pj),0);
    Fillchar (ubuf,sizeof (ubuf),0);
  end;

  { ************************************************************************ }

Procedure Baca_data_input;
  Begin
    assign ( file_input, nama_file );
    reset  ( file_input );
    read   ( file_input, S_inp^ );
    close  ( file_input );
  end;

  { ************************************************************************ }

Procedure  LPF;
  const
    fc              = 800;
    T               = 0.125E-03;
  var
    S1, S2, B2, a1, a2, a3, a4, a5      : real ;
    u                                   : larik_1;
    n                                   : integer;

  begin
    new (u);
    fillchar (u^, sizeof(u^), 0);
    S1              := 0.3572 * 2 * PI * fc;
    S2              := 0.1786 * PI * fc;
    B2              := 0.8938 * PI * fc;
```

```
    a1              := 1 - exp ( -S1 * T );
    a2              := exp ( -S1 * T );
    a3              := 1 - (  2 * exp ( -S2 * T ) * cos (B2 * T )) +
                       exp ( -2 * S2 * T );
    a4              := 2 * exp ( -S2 * T ) * cos (B2 * T );
    a5              := - exp   ( -2  * S2  * T );
    u^[0]           := round   ( a1  * S_inp^[0] );
    x^[0]           := ( a3 * u^[0] );
    u^[1]           := round   ( a1  * s_inp^[1] + a2 * u^[0] );
    x^[1]           := ( a3 * u^[1]  + a4 * x^[0]);

    for n           := 2   to  ( data_max - 1 ) do
      begin
        u^[n]       := round   ( a1 * s_inp^[n] + a2 * u^[n-1] );
        x^[n]       := ( a3 * u^[n]  + a4 * x^[n-1] + a5 * x^[n-2]);
      end;
    dispose (u);
  end;


{  ************************************************************************ }


Procedure Desimasi;
  var
    n               : integer;

  begin
    for n           := 0 to ( desi - 1 ) do
      w [n]         := x^[4 * n];
  end;


 { ************************************************************************ }


Procedure buat_frame_1;
  var
    awal_f, akhir_f, n, i     : integer;
     begin
       awal_f                 := no_frm * jml_samp;
       akhir_f                := awal_f + jml_samp - 1;
       n                      := 0;
       i                      := awal_f;

       with data_hasil [no_frm]  do
         repeat
           Pj [n]             := w [i];
           n                  := n + 1;
           i                  := i + 1;
         until i > akhir_f;
     end;


 { ************************************************************************ }
```

```
Procedure Diff_Hamming;
  var
     uprev                    :  real;
     ja                       :  integer;

     begin
       uprev                  := 0;
       for ja                 := 0 to ( jml_samp - 1 ) do
         begin
           ubuf [ja]          := (Pj [ja] - uprev ) * (0.54 - 0.46 *
                                 cos ((ja - 1) * 6.28318 /
                                  (jml_samp - 2)));
           uprev              := Pj [ja];
       end;
     end;

 { ************************************************************************** }


Procedure Hitung_Parameter_LPC;
  Var
    q,l, n, i        :  Integer;
    buf              :  real;
    S                :  array_2;
    r                :  array_1;
    E                :  array [ 0 .. ((desi div jml_samp) - 1 ),
                        0.. LPC_length ] of real;
    a                :  array [ 0 .. LPC_length, 0 .. LPC_length] of real;

  Begin
    Fillchar (S, sizeof (S), 0);
    for  n           := 0 to (jml_samp - 1 ) do
      begin
        S[n]         := ubuf [n] / 25 ;
      end;

                                  { *..... hitung Autokrelasi ....... * }
    Fillchar (r, sizeof (r), 0 );
    for l            := 0 to LPC_length do
      for n          := 0 to (jml_samp - 1 - l ) do
        r[l]         := r[l] + S[n] * S[n+l];

                                  { * .... hitung Koef. Prediksi .... * }
    Fillchar (a, sizeof (a), 0 );
    Fillchar (E, sizeof (E), 0 );
    a[0,0]           := 1;
    E [no_frm, 0]    := r[0];
    For q            := 1 to LPC_length do
      Begin
        buf          := 0;
        for i        := 0 to (q-1) do
          buf        := buf + a[q-1, i] * r[q-i];

        buf          := -buf / E [no_frm, q - 1];
        a[q,q]       := buf;
        for i        := 1 to (q-1) do
```

```
        a[q,i]    := a[q-1,i] + (buf * a[q-1, q-i]);
      a[q,0]      := 1;
      E[no_frm,q]:= (1 - sqr(buf)) * E[no_frm, q - 1];
    end;
    For n          := 0 to LPC_length do
      PP[n]        := a[n, n];
  End;


  { ************************************************************** }


Procedure Invers_filter;
  var
    alpha          : array_1;
    n, i           : integer;
    Pj_w           : array [LPC_length .. (jml_samp -1)] of real;
    Pj             : array [ 0..(jml_samp - 1 )] of real;
  begin
    fillchar (alpha, sizeof(alpha), 0);
    fillchar (Pj_w,  sizeof (Pj_w), 0);
    fillchar (pbuf,  sizeof(pbuf),  0);
    fillchar (Pj,    sizeof(Pj),    0);
    alpha [0]      := 1e+00;
    for n          := 1 to LPC_length do
      alpha [n]    := pp[n];
    y[0]           := Pj [0] /25;
    for n          := 0 to (jml_samp - 1) do
      begin
        Pj_w [n]   := Pj [n] /25;
      end;
    for n          := 1 to (jml_samp - 1 ) do
      begin
        y [n]      := 0;
        for i      := 1 to LPC_length do
          begin
            y[n]   := y[n] + alpha [i] * Pj_w [n-i];
          end;
        y[n]       := y[n] + Pj_w [n];
      end;
    for n          := 4 to (jml_samp - 1 ) do
      begin
        pbuf [n-4] := y[n] * (0.54 - 0.46 * cos ((n-4) * 6.28318 /
                      (jml_samp - 6 )));
        pbuf [n-4] := pbuf [n-4] / 10;
      end;
    end;


  { ******************************************************************* }


Procedure autokorelasi;
  var
    maks           :  real;
    n, j           :  integer;
  begin
```

```
    fillchar (auto_norm, sizeof(auto_norm), 0);
    fillchar (rn, sizeof(rn), 0);
    rn        [jml_samp - 4] := 0;
    for n           := 0 to (jml_samp - 5) do
      begin
        rn[n]       := 0;
        for j       := 0 to (jml_samp - 5 - n) do
          begin
            rn[n]   := rn[n] + pbuf[j] * pbuf [j+n];
          end;
      end;
    for n           := 0 to (jml_samp - 5) do
      auto_norm [n]:= rn[n] / Abs ( rn[0]);
      P1            := periode [no_frm - 1];
      P2            := periode [no_frm - 2];
      P3            := periode [no_frm - 3];
    end;


  { ************************************************************** }


Procedure Interpolasi;
  const
    dua_1           =  0.637643;
    dua_2           =  0.636110;
    dua_3           =  0.212208;
    tiga_1          =  0.322745;
    tiga_2          =  0.878039;
    tiga_3          =  0.158147;
  var
    gamma_per2, gamma_per4, gamma_min_per2, gamma_min_per4  : real;
  begin
    gamma_per2      := (dua_1  * gamma_plus1) + (dua_2 * gamma_0) -
                       (dua_3  * gamma_min1 );
    gamma_min_per2 := (dua_1  * gamma_min1 ) + (dua_2 * gamma_0) -
                       (dua_3  * gamma_plus1);
    gamma_per4      := (tiga_1 * gamma_plus1) + (tiga_2 * gamma_0) -
                       (tiga_3 * gamma_min1);
    gamma_min_per4 := (tiga_1 * gamma_min1)  + (tiga_2 * gamma_0) -
                       (tiga_3 * gamma_plus1);
    puncak          := gamma_0;
    lokasi_puncak   := peak;
      if gamma_per4          >  gamma_min_per4 then
        begin
          if gamma_per4      >  gamma_per2 then
            begin
              puncak         := gamma_per4;
              lokasi_puncak  := peak  + 0.25;
            end
          else
            begin
              puncak         := gamma_per2;
              lokasi_puncak  := peak + 0.5;
            end;
        end;
      if gamma_per4          <  gamma_min_per4 then
```

```
        begin
          if gamma_min_per4 >  gamma_min_per2 then
            begin
              puncak        := gamma_min_per4;
              lokasi_puncak := peak  - 0.25;
            end
          else
            begin
              puncak        := gamma_min_per2;
              lokasi_puncak := peak - 0.5;
            end;
        end;
      if puncak              < gamma_0 then
        begin
          puncak             := gamma_0;
          lokasi_puncak      := peak;
        end;
    end;


  { ************************************************** }


Procedure Cari_pitch_V_UV;
  var
    amax, DD, PO    :  real;
    j               :  integer;

  begin
    peak            := 5;
    amax            := auto_norm [5];

    for j           := 5 to 31 do
      begin
        if auto_norm [j]  > amax then
          begin
            peak    := j;
            amax    := auto_norm [peak];
          end;
      end;

    gamma_0         := amax;
    gamma_min1      := auto_norm [peak - 1];
    gamma_plus1     := auto_norm[peak + 1];

    if amax         <> 0 then
      begin

        if auto_norm [peak]  > auto_norm [peak -1] then
          begin

            if auto_norm [peak]        >  auto_norm [peak -1] then
              begin
                Interpolasi;
                  if lokasi_puncak     >= 18 then
                    DD                 := -0.1 * ((lokasi_puncak - 18)/13)
                                          + 1
```

```
        else
          DD                    := - 1 * ((lokasi_puncak - 5) / 13)
                                   + 2;
      puncak                    := puncak / DD;
      if puncak                 <  0.35 then
        begin
          if P1                 <> 0 then
            begin
              if puncak         >= 0.3 then
                begin
                  PO                    := lokasi_puncak;
                  periode [no_frm]      := PO;
                  frekuensi[no_frm]     := 1 / PO;
                  VUV [no_frm]          := true;
                end;
            end
            ELSE                { P1 <> 0 }
              begin
                PO                      := 0;
                periode [no_frm]        := 0;
                frekuensi[no_frm]       := 0;
                VUV [no_frm]            := false;
              end;
        end
        ELSE                    { Puncak < 0.35 }
          begin
            PO                          := lokasi_puncak;
            periode [no_frm]            := PO;
            frekuensi[no_frm]           := 1 / PO;
            VUV [no_frm]                := true;
          end;
      end ELSE                    { auto_normal[peak] }
        begin
          PO                            := 0;
          periode [no_frm]              := 0;
          frekuensi[no_frm]             := 0;
          VUV [no_frm]                  := false;
        end;
    end ELSE                    { auto_normal[peak] }
        begin
          PO                            := 0;
          periode [no_frm]              := 0;
          frekuensi[no_frm]             := 0;
          VUV [no_frm]                  := false;
        end;

if no_frm  > 2 then
  begin
    if Abs(P1-P3)                       <= 0.375 * P3 then
      begin
        P2                              := (P1 + P3 ) /2;
        periode [no_frm - 2]            := P2;

        if P2                           =  0 then
          begin
            frekuensi [no_frm - 2]      := 0;
            VUV       [no_frm - 2]      := false;
```

```
            end
          else
            begin
              frekuensi [no_frm - 2]      := 1 / P2;
              VUV       [no_frm - 2]      := true;
            end;
        end;

      if P3                               =  0 then
        begin

          if P2                           <> 0 then
            begin
              if Abs ( PO -P1 )           <= 0.2 * P1 then
                begin
                  P2                      := 2 * P1 - PO;
                  periode     [no_frm - 2] := P2;
                  if P2                   =  0 then
                    frekuensi [no_frm -2]  := 0
                  else
                    frekuensi [no_frm - 2] := 1 /P2;
                    VUV       [no_frm - 2] := true;
                end;
              end;
        end;

      if P1                               =  0 then
        begin
          if Abs (P2-P3 )                 >  0.375 * P3 then
          begin
            P2                            := 0;
            periode   [no_frm - 2]        := 0;
            frekuensi [no_frm - 2]        := 0;
            VUV       [no_frm - 2]        := false;
          end;
        end;

    end;        { of if no_frm > 2 }
  end;          { of procedure cari_Pitch_V_UV }
  end;

  { ************************************************************************ }

Procedure Simpan_hasil;
  const
    frek_samp      =  2;
  var
    n, k           :  integer;
  begin
    with data_hasil [no_frm] do
      begin
        pitch      := (periode   [no_frm] / frek_samp);
        Fo         := (frekuensi [no_frm] * frek_samp);
        V_UV       := VUV [no_frm];
      end;
  end;
```

```
{ ************************************************************** }

Procedure Inisialisasi_grafik;
  Var
    Graphdriver, graphmode   :  integer;
    kodeerror                :  integer;
  begin
    graphdriver              := vga;
    graphmode                := VGAHi;
    initgraph   (graphdriver, graphmode, ' ');
    kodeerror                := graphresult;
    if kodeerror             <> grOK then
      Begin
        writeln ('ERROR GRAFIK : ', grapherrormsg(kodeerror));
        readln;
        halt(1);
      end;
  end;


{ ************************************************************** }


Procedure Inkey ( var Kar     : char;
                  var FK      : Boolean;
                  var Key     : Keytype);
  Begin
    Kar                      := Readkey;
    FK                       := False;
    if Kar                   =  #0 then
      begin
        FK                   := true;
        Kar                  := Readkey;
      end;
    if FK then
      case Kar of
        #72          : key     := UpArrow;
        #80          : key     := DownArrow;
        #75          : key     := LeftArrow;
        #77          : key     := RightArrow;
        #73          : key     := PgUp;
        #81          : key     := PgDn;
        #71          : key     := HomeKey;
        #79          : key     := EndKey;
        #59          : key     := F1;
        #60          : key     := F2;
      end
    else
      case Kar of
        #27          : key     := EscapeKey;
        #32          : key     := SpaceKey;
      end;
  end;
```

```
{ ***************************************************************** 14

Procedure Buat_frame_2;
  Var
    awal_frame, akhir_frame, ii, nn          :  integer;
  begin
    for no_frm     := 0 to frame_max do
      begin
       awal_frame  := no_frm * jml_samp_asli;
       akhir_frame := awal_frame + jml_samp_asli - 1;
       ii          := awal_frame;
       nn          := 0;
       with data_hasil [no_frm] do
         begin
           repeat
             sn[nn]:= s_inp^[ii];
             ii    := ii + 1;
             nn    := nn + 1;
           until ii > akhir_frame;
         end;
      end;
  end;


  { ***************************************************************** }


Procedure Buat_gambar ( nomor : integer );
  var
    titikX, titikY, n, n1, titikn1, my, fo, pitch,
    fo_int, pitch_int                                  : integer;
    teks,   p_pitch,   f_fo, tulis_pitch               : string;
    V_UV                                               : boolean;
  begin
    str (nomor, teks);
    setviewport (0, 0, 239, getmaxY, clipon);
    cleardevice;
    with data_hasil[nomor] do
      begin
        moveto  (0, 75);
        if nomor   >  0 then
          begin
            my    := 75 - round(data_hasil[nomor-1]
                       .sn[jml_samp_asli - 1] div 25);
            moveto (0, my);
          end;
        setcolor ( warna + 2 );
        line     (0,75,319,75);
        for n                 := 0 to (jml_samp_asli - 1) do
          begin
            titikX            := n;
            titikY            := 75 - round (sn[n] div 25);
            setcolor(warna+1);
            lineto(titikX, titikY);
          end;
        if nomor              < frame_max then
```

```
    setviewport (240, 0, getmaxX, getmaxY, clipoff);
setcolor  (warna+2);
line      (0, 75, getmaxX-240, 75);
moveto    (0, 75 - round (data_hasil[nomor].sn[jml_samp_asli-1]
           div 25));
n1                  := 0;
  for n1            := 0 to (getmaxX - 240) do
    begin
      titikY          := 75 - round (data_hasil[nomor + 1]
                         .sn[n1] div 25);
     setcolor  (warna + 1 );
     lineto   (n1, titikY);
    end;
  end;
fo_int                := round (100 * fo);
str (fo_int, f_fo);
pitch_int             := round (1000 * pitch);
str (pitch_int, p_pitch);

if V_UV then
  begin
    tulis_pitch       := copy (p_pitch, 1, (length (p_pitch) - 3)) +
                       ' . ' +copy (p_pitch, (length(p_pitch) - 2 ), 3);
  end ELSE
    begin
      tulis_pitch       := '0';
    end;
setviewport    (0, 0, 239, getmaxY, clipon);
settextstyle   (smallfont, horizdir, 4);
settextjustify (lefttext, centertext);
setcolor       (warna);
outtextXY      (1, 150, ' <----------');
settextjustify (righttext, centertext);
outtextXY      (240, 150, ' ---------->');
settextjustify (centertext,  centertext);
outtextXY      (120, 150, 'FRAME' + teks);
setcolor       (warna+2);
line           (0, 135, 0, 165);
line           (239, 135, 239, 165);
setcolor       (warna);
settextstyle   (smallfont, horizdir, 4);
settextjustify (lefttext, centertext);
outtextXY      (5, 170, 'PITCH : ' + tulis_pitch + 'ms');
outtextXY      (5, 180, 'Fo    : ' + copy(f_fo, 1, 3) + 'Hz');
setviewport    (0, 188,  getmaxX, getmaxY, clipon);
setcolor       (warna+2);
bar            (0, 0, 320, 10);
setcolor       (0);
settextstyle   (smallfont, horizdir, 4);
outtextXY      (5, 3, 'F1 - cursor <--> move cursor'+
               ' PgUp-Prev PgDn - Next');
settextjustify (righttext, centertext);
if V_UV then outtextXY (239, 180, 'Voiced')
  else outtextXY(239, 180, 'Unvoiced');
if nomor       <  frame_max then
  begin
    setviewport  (240, 0, getmaxX, getmaxY, clipoff);
```

```
            settextjustify(lefttext, centertext);
            setcolor     (warna);
            outtextXY    (0, 150, '<------');
          end;
      end;

  { ************************************************************** }


Procedure HalPrev;
  begin
    dec (frame_gambar);
      if frame_gambar        <  0 then
        frame_gambar         := 0;
        buat_gambar (frame_gambar);
    end;


  { ************************************************************** }


Procedure HalNext;
   begin
     inc(frame_gambar);
       if frame_gambar       >  frame_max then
       frame_gambar          := frame_max;
       buat_gambar (frame_gambar);
    end;


  { ************************************************************** }


Procedure Akhir;
  begin
    buat_gambar (frame_max);
      frame_gambar           := frame_max;
  end;


  { ************************************************************** }


Procedure Awal;
  begin
    buat_gambar (0);
      frame_gambar           := 0;
    end;


  { ************************************************************** }


Procedure Hitung_Pitch ( var tekan, posisi2  : integer );
  var
    periode_pitch            :  string;
    posisi1, posisiy         :  integer;
```

```
  period                    :  real;
  kar                       :  char;
  Fk                        :  boolean;
begin
    tekan                   := tekan + 1;
    if tekan                <  2 then
      begin
        selesai             := false;
        setcolor (0);
        settextjustify (righttext, centertext);
        outtextXY      (239, 5, ' Mark 1st point - Press F2. ');
        posisi1             := getX;

        settextstyle   (smallfont, horizdir, 4);
        settextjustify (righttext, centertext);
        setcolor (warna);
        outtextXY      (239, 5, ' Mark 2nd point.  Press F2. ');
      end else
      begin
        tekan               := 0;
        posisi2             := getX;
        posisiy             := getY;
        putimage (posisi2, posisiy, salib^, xorput);
        setcolor (0);
        settextjustify (righttext, centertext);

        outtextXY (239, 5, ' Mark 2nd point.  Press F2. ');
        period              := Abs(posisi2 - posisi1);
        period              := period / 8;
        str (period, periode_pitch);
        setcolor (warna);
        settextstyle (smallfont, horizdir, 4);
        settextjustify (centertext, centertext);
        outtextXY (160, 5, 'PITCH  : ' + copy(periode_pitch, 1, 7) + 'ms');
        delay     (3000);
        setcolor  (0);
        settextjustify(centertext, centertext);
        outtextXY (160, 5, 'PITCH  : ' + copy(periode_pitch, 1, 7) + 'ms');
        setcolor  (warna);
        settextjustify(centertext, centertext);
        outtextXY (160, 5, 'Press Space Bar ! ');
        repeat
          inkey (Kar, Fk, Key);
        until Key  = SpaceKey;
        setcolor(0);
           outtextXY(160, 5, 'Press Space Bar  !.');
        selesai             := true;
      end;
  end;


{ ************************************************************************** }


Procedure Kursor_kanan;
  var
    oldX, oldY, aX, aY        :  integer;
```

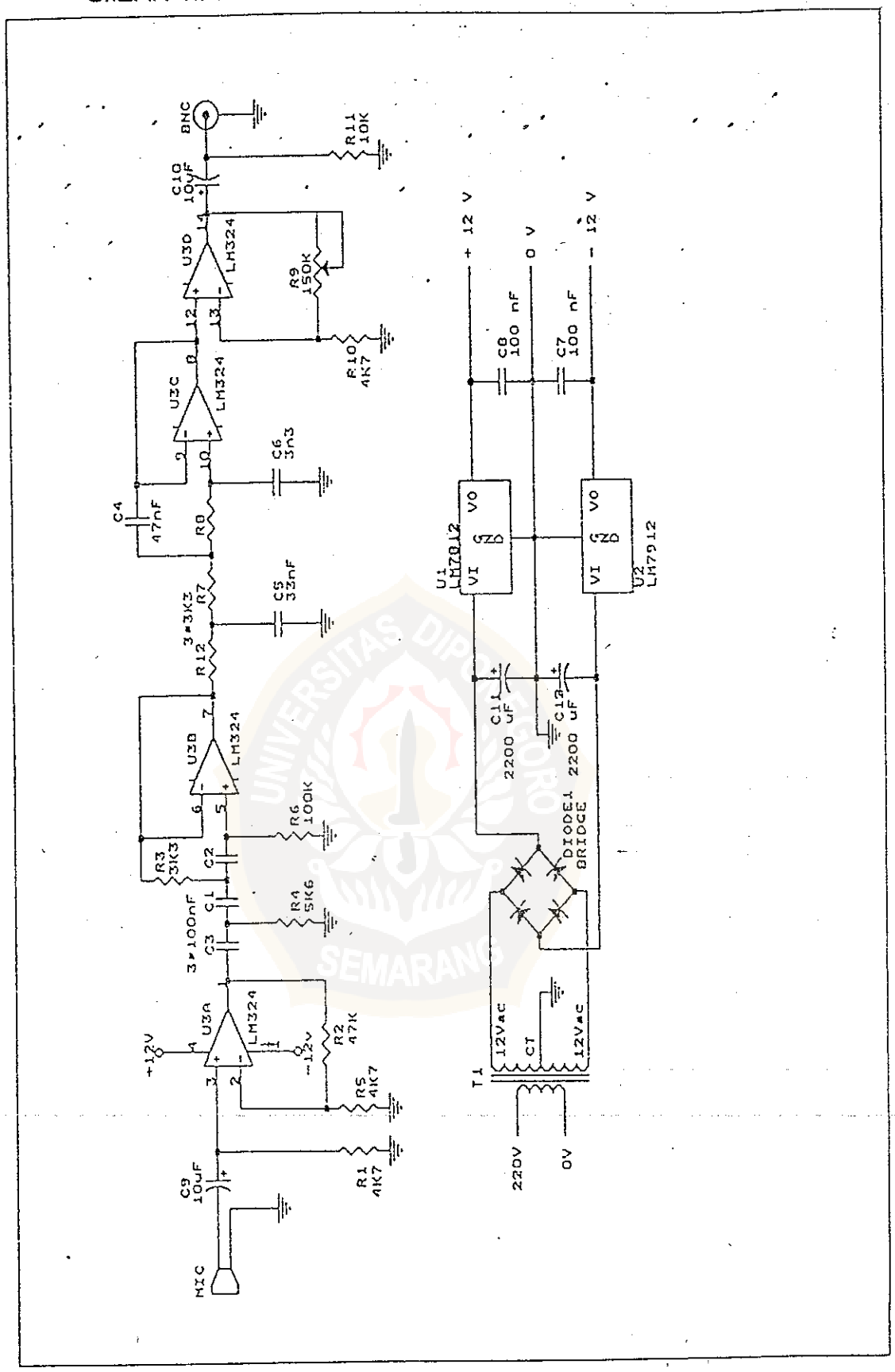
```
  begin
    oldX                    := getX;
    oldY                    := getY;
    putimage (oldX, oldY, salib^, Xorput);
    aX                      := oldX + 1;
    ay                      := oldY;
    putimage (aX, aY, salib^, xorput);
    moveto(aX, aY);
  end;

{ ************************************************************************ }

Procedure Kursor_kiri;
  var
    oldX, oldY, bX, bY        :  integer;
  begin
    oldX                      := getX;
    oldY                      := getY;
    putimage(oldX, oldY, salib^, Xorput);
    bX                        := oldX - 1;
    bY                        := oldY;
    putimage (bX, bY, salib^, xorput);
    moveto(bX, bY);
  end;

{ ************************************************************************ }

Procedure Kursor_kanan_cepat;
  var
    oldX, oldY, cX, cY        :  integer;
  begin
    oldX                      := getX;
    oldY                      := getY;
    putimage (oldX, oldY, salib^, Xorput);
    cX                        := oldX + 10;
    cY                        := oldY;
    putimage (cX, cY, salib^, xorput);
    moveto   (cX, cY);
  end;

{ ************************************************************************ }

Procedure kursor_kiri_cepat;
  var
    oldX, oldY, dX, dY        :  integer;
  begin
    oldX                      := getX;
    oldY                      := getY;
    putimage (oldX, oldY, salib^, Xorput);
    dX                        := oldX - 10;
```

```
    dY                          := oldY;
    putimage (dX, dY, salib^, xorput);
    moveto   (dX, dY);
  end;


{ ************************************************************************ }


Procedure Kursor_awal( var posisi_akhir  : integer);
  var
    tekan_f2, awalX, awalY   :  integer;
    tombol                   :  char;
    Fk                       :  boolean;
  begin
    tekan_f2                 := 0;
    salib                    := @data_kursor;
    awalX                    := posisi_akhir;
    awalY                    := 15;
    setviewport    (0, 0, getmaxX, 20, clipon);
    settextstyle   (smallfont, horizdir, 4);
    settextjustify (righttext, centertext);
    setcolor       (warna);
    outtextXY      (239, 5, ' Mark 1st point.  Press F2. ');
    setviewport    (0, 0, getmaxX, 240, clipon);
    putimage       (awalX, awalY, salib^, xorput);
    moveto         (awalX, awalY);
    selesai                  := false;
    repeat
      inkey (tombol, Fk, key);
      case key of
        RightArrow           :  kursor_kanan;
        LeftArrow            :  kursor_kiri;
        UpArrow              :  kursor_kanan_cepat;
        DownArrow            :  kursor_kiri_cepat;
        F2                   :  hitung_pitch(tekan_f2, posisi_akhir);
      end;
    until  selesai
  end;


{ ************************************************************************ }


{     ------------------  MAIN  PROGRAM  --------------------------- }

begin
   clrscr;
   new    (s_inp); new(x);
   inisialisasi_1;
   write  ('Nama File          : ');
   readln (nama_file);
   baca_data_input;
   writeln('tunggu proses SIFT sedang dikerjakan  .... ');
   LPF;
```

```
  desimasi;
  for no_frm                  := 0 to (array_max - 1) do
    begin
      buat_frame_1;
      Diff_Hamming;
      Hitung_parameter_LPC;
      invers_filter;
      autokorelasi;
      cari_pitch_V_UV;
    end;
  for no_FRM                  := 0 to (array_max - 1) do
    simpan_hasil;

  buat_frame_2;
  inisialisasi_grafik;
  warna                     := 1;
  buat_gambar (0);
  frame_gambar              := 0;
  kursor_akhir              := 10;

  repeat
    inkey (ch, Fkg, key);
    case key of
      PgUp                  :  halprev;
      PgDn                  :  halnext;
      HomeKey               :  awal;
      EndKey                :  akhir;
      F1                    :  kursor_awal(kursor_akhir);
    end;
  until key                 =  EscapeKey;

 closegraph;
 dispose (s_inp);
 dispose (x);
end.
```

LAMPIRAN C

SKEMA RANGKAIAN PENGUAT DAN FILTER MIKROFON



NILAI PERODE PITCH DAN Fo TIAP FRAME

Suara Vocal 'a' (Pria)

No.	Fo	Periode Pitch
1	125	8,000
2	125	8,000
3	127	7,891
4	126	7,938
5	129	7,814
6	127	7,891
7	126	7,938
8	129	7,814
9	125	8,000
10	127	7,891
11	126	7,938
12	129	7,814
13	127	7,891
14	126	7,938
15	129	7,814
16	127	7,891
17	126	7,938
18	129	7,814
19	127	7,891
20	126	7,938
21	129	7,814
22	125	8,000
23	127	7,891
24	126	7,938
25	129	7,814
26	127	7,891
27	126	7,938
28	129	7,814
29	125	8,000
30	125	8,000
31	127	7,891
32	126	7,938
33	129	7,814
34	127	7,891
35	126	7,938
36	129	7,814
37	125	8,000
38	127	7,891
39	126	7,938
40	129	7,814

Suara Vocal 'i' (Pria)

No	Fo	Periode Pitch
1	138	7,250
2	138	7,250
3	138	7,250
4	138	7,250
5	138	7,250
6	138	7,250
7	138	7,250
8	138	7,250
9	138	7,250
10	138	7,250
11	138	7,250
12	138	7,250
13	138	7,250
14	138	7,250
15	138	7,250
16	138	7,250
17	138	7,250
18	138	7,250
19	138	7,250
20	138	7,250
21	138	7,250
22	138	7,250
23	138	7,250
24	138	7,250
25	138	7,250
26	138	7,250
27	138	7,250
28	138	7,250
29	138	7,250
30	138	7,250
31	138	7,250
32	138	7,250
33	138	7,250
34	138	7,250
35	138	7,250
36	138	7,250
37	138	7,250
38	138	7,250
39	138	7,250
40	138	7,250

Suara Vocal 'e' (Pria)

No.	Fo	Periode Pitch
1	119	8,375
2	120	8,313
3	120	8,313
4	120	8,313
5	121	8,268
6	121	8,281
7	120	8,313
8	120	8,313
9	120	8,313
10	121	8,268
11	121	8,281
12	119	8,375
13	120	8,313
14	120	8,313
15	120	8,313
16	121	8,268
17	121	8,281
18	120	8,313
19	120	8,313
20	120	8,313
21	121	8,268
22	121	8,281
23	120	8,313
24	120	8,313
25	120	8,313
26	121	8,268
27	121	8,281
28	119	8,375
29	120	8,313
30	120	8,313
31	120	8,313
32	121	8,268
33	121	8,281
34	120	8,313
35	120	8,313
36	120	8,313
37	121	8,268
38	121	8,281
39	120	8,313
40	121	8,268

Suara Vocal 'o' (Pria)

No	Fo	Periode Pitch
1	123	8,125
2	123	8,125
3	123	8,125
4	125	8,016
5	124	8,063
6	124	8,063
7	122	8,171
8	123	8,125
9	123	8,125
10	125	8,016
11	124	8,063
12	124	8,063
13	122	8,171
14	123	8,125
15	123	8,125
16	123	8,125
17	125	8,016
18	124	8,063
19	124	8,063
20	122	8,171
21	123	8,125
22	123	8,125
23	125	8,016
24	124	8,063
25	124	8,063
26	122	8,171
27	125	8,016
28	124	8,063
29	124	8,063
30	122	8,171
31	123	8,125
32	123	8,125
33	125	8,016
34	124	8,063
35	124	8,063
36	122	8,171
37	123	8,125
38	123	8,125
39	123	8,125
40	125	8,016

Suara Vocal 'u' (Pria)

No.	Fo	Periode Pitch
1	140	7,125
2	142	7,063
3	144	6,953
4	144	6,953
5	144	6,953
6	143	6,988
7	138	7,250
8	139	7,219
9	144	6,953
10	144	6,953
11	144	6,953
12	143	6,988
13	138	7,250
14	139	7,219
15	142	7,063
16	144	6,953
17	144	6,953
18	144	6,953
19	143	6,988
20	138	7,250
21	139	7,219
22	144	6,953
23	144	6,953
24	144	6,953
25	143	6,988
26	138	7,250
27	139	7,219
28	144	6,953
29	144	6,953
30	143	6,988
31	138	7,250
32	139	7,219
33	144	6,953
34	144	6,953
35	144	6,953
36	143	6,988
37	138	7,250
38	139	7,219
39	142	7,063
40	144	6,953

M. Arsyad

NILAI PERODE PITCH DAN Fo TIAP FRAME

Suara Vocal 'a' (Wanita)

No.	Fo	Periode Pitch
1	216	4,625
2	229	4,376
3	217	4,625
4	222	4,500
5	227	4,406
6	222	4,500
7	217	4,625
8	216	4,625
9	227	4,406
10	227	4,406
11	228	4,391
12	222	4,500
13	216	4,625
14	217	4,625
15	225	4,438
16	225	4,438
17	225	4,438
18	228	4,391
19	222	4,500
20	216	4,625
21	222	4,500
22	227	4,406
23	222	4,500
24	217	4,625
25	216	4,625
26	225	4,438
27	228	4,391
28	222	4,500
29	216	4,625
30	222	4,500
31	216	4,625
32	229	4,376
33	217	4,625
34	222	4,500
35	227	4,406
36	222	4,500
37	217	4,625
38	216	4,625
39	227	4,406
40	227	4,406

Suara Vocal 'i' (Wanita)

No	Fo	Periode Pitch
1	242	4,125
2	246	4,063
3	246	4,063
4	248	4,031
5	249	4,020
6	250	4,000
7	249	4,020
8	250	4,000
9	249	4,020
10	249	4,020
11	249	4,020
12	250	4,000
13	246	4,063
14	248	4,031
15	249	4,020
16	242	4,125
17	246	4,063
18	249	4,020
19	246	4,063
20	248	4,031
21	242	4,125
22	246	4,063
23	250	4,000
24	249	4,020
25	249	4,020
26	249	4,020
27	250	4,000
28	246	4,063
29	248	4,031
30	249	4,020
31	242	4,125
32	246	4,063
33	249	4,020
34	249	4,020
35	250	4,000
36	246	4,063
37	248	4,031
38	249	4,020
39	242	4,125
40	248	4,031

Suara Vocal 'e' (Wanita)

No.	Fo	Periode Pitch
1	229	4,375
2	232	4,313
3	230	4,352
4	230	4,352
5	235	4,250
6	235	4,250
7	235	4,250
8	232	4,313
9	230	4,352
10	230	4,352
11	229	4,375
12	232	4,313
13	230	4,352
14	230	4,352
15	235	4,250
16	230	4,352
17	235	4,250
18	235	4,250
19	235	4,250
20	232	4,313
21	235	4,250
22	235	4,250
23	235	4,250
24	235	4,250
25	235	4,250
26	235	4,250
27	232	4,313
28	232	4,313
29	230	4,352
30	230	4,352
31	235	4,250
32	235	4,250
33	235	4,250
34	235	4,250
35	232	4,313
36	230	4,352
37	230	4,352
38	229	4,375
39	229	4,375
40	232	4,313

Suara Vocal 'o' (Wanita)

No	Fo	Periode Pitch
1	235	4,250
2	239	4,188
3	242	4,133
4	242	4,130
5	238	4,203
6	242	4,130
7	238	4,203
8	239	4,188
9	242	4,133
10	242	4,130
11	238	4,203
12	242	4,130
13	238	4,203
14	242	4,130
15	238	4,203
16	239	4,188
17	242	4,133
18	242	4,130
19	239	4,188
20	242	4,133
21	242	4,130
22	238	4,203
23	235	4,250
24	239	4,188
25	242	4,133
26	242	4,130
27	238	4,203
28	242	4,130
29	238	4,203
30	239	4,188
31	242	4,133
32	242	4,130
33	238	4,203
34	242	4,130
35	238	4,203
36	242	4,130
37	239	4,188
38	242	4,133
39	242	4,130
40	238	4,203

Suara Vocal 'u' (Wanita)

No.	Fo	Periode Pitch
1	242	4,125
2	246	4,063
3	250	4,000
4	249	4,011
5	245	4,078
6	246	4,063
7	250	4,000
8	249	4,011
9	242	4,125
10	246	4,063
11	250	4,000
12	249	4,011
13	245	4,078
14	245	4,078
15	246	4,063
16	250	4,000
17	249	4,011
18	246	4,063
19	250	4,000
20	249	4,011
21	245	4,078
22	246	4,063
23	250	4,000
24	249	4,011
25	250	4,000
26	249	4,011
27	245	4,078
28	246	4,063
29	250	4,000
30	245	4,078
31	245	4,078
32	246	4,063
33	250	4,000
34	249	4,011
35	246	4,063
36	250	4,000
37	249	4,011
38	245	4,078
39	246	4,063
40	250	4,000