LAMPIRAN

```
{Listing Program Pengendali JST}
uses WinDos, crt,graph;

const
     pa          = $300; {alamat PPI port A}
     pb          = $301; {alamat PPI port B}
     pc          = $302; {alamat PPI port C}
     pcw_ppi     = $303; {alamat control word PPI}
     cw_ppi      = $80;  {inisialisasi PPI}

     pcw_pit     = $307; {alamat control word dari PIT}

     adc         = $308; {alamat pengaktifan ADC}
     dac0        = $30A; {alamat pengaktifan DAC0}
     dac1        = $30C; {alamat pengaktifan DAC1}

var
    pos_vec,tet_vec         : array        [1..3] of integer;
    dpos_vec,dtet_vec       : array        [1..3] of real;
    W1c                     : array [1..20,1..12] of real;
    W2c                     : array  [1..1,1..20] of real;
    phic                    : array       [1..12] of real;
    nethidc,outhidc         : array       [1..20] of real;
    netuc,outuc             : array        [1..1] of real;
    o                       : char;
    databobot               : text;
    temp                    : string;
    npwm,n_pwm              : longint;
    lsb_pwm,msb_pwm         : word;
    Driver,Mode,code        : integer;
    n,sud,pos,pancing,i,j   : integer;
    rev                     : pointer;
    sudut                   : boolean;
    yref,u3,u0,u1           : real;
    ratet,rapos,totet,topos,dtotet,dtopos,dsud,dpos                : real;
    teta1,teta0,posisi1,posisi0,dteta0,dposisi0,dteta1,dposisi1 : real;


procedure PWM;
begin
npwm        := round( (u0*2700) + (5*dposisi0) );
n_pwm       := abs(npwm);
if (((posisi0>=0.08) and (teta0>=0)) or ((posisi0<=-0.08) and
(teta0<=0))) then begin n_pwm:=1;npwm:=1;end;
lsb_pwm     := n_pwm mod 256;
msb_pwm     := n_pwm div 256;
Port[$306]  := lsb_pwm;
Port[$306]  := lsb_pwm;
if npwm>0 then port[pa] := $01 else
if npwm<0 then port[pa] := $02 else port[pa] := $00;
end;
```

```
function tansig(net:real):real;
begin
     tansig := 2/(1+exp(-2*net))-1;
end;

Procedure nncon;
begin
          phic[ 1] := yref;
          phic[ 2] := posisi0;
          phic[ 3] := dposisi0;
          phic[ 4] := teta0;
          phic[ 5] := dteta0;
          phic[ 6] := posisi1;
          phic[ 7] := dposisi1;
          phic[ 8] := teta1;
          phic[ 9] := dteta1;
          phic[10] := u0;
          phic[11] := u1;
          phic[12] := 1;         {Bias}

     for i := 1 to 20 do
      begin
          nethidc[i]:=0;
          for j := 1 to 12 do
           begin
             nethidc[i]:=nethidc[i]+phic[j]*W1c[i,j];
           end;
          outhidc[i]:=tansig(nethidc[i]);
      end;

     for i := 1 to 1 do
      begin
          netuc[i]:=0;
          for j:=1 to 20 do
           begin
             netuc[i]:=netuc[i]+ outhidc[j]*W2c[i,j];
           end;
          outuc[i]:=netuc[i];
      end;
    u1 := u0;
    u0 := outuc[1];
end;

procedure Baca_adc; interrupt;
begin

      if sudut then
          begin
             sud      := ((portw[adc] and $0FFF) div 2)-1024;
             sudut    := false;
             port[pc] := $07;
             if (sud>ratet+50) then sud := round(ratet)+50;
```

```
         if (sud<ratet-50) then sud := round(ratet)-50;
      end
  else
      begin
         pos      := ((portw[adc] and $0FFF) div 2)-1024;
         sudut    := true;
         port[pc] := $02;
         if (pos>rapos+50) then pos := round(rapos)+50;
         if (pos<rapos-50) then pos := round(rapos)-50;
      end;

  if sudut then
    begin
     inc(n);
       tet_vec[3] := tet_vec[2];
       pos_vec[3] := pos_vec[2];
       tet_vec[2] := tet_vec[1];
       pos_vec[2] := pos_vec[1];
       tet_vec[1] := sud;
       pos_vec[1] := pos;
       totet      := tet_vec[1] + 0.8*tet_vec[2] + 0.2*tet_vec[3];
       topos      := pos_vec[1] + 0.8*pos_vec[2] + 0.2*pos_vec[3];
       ratet      := totet/2;
       rapos      := topos/2;

teta1   := teta0;
posisi1 := posisi0;

teta0   := (ratet*0.50)/1024;
posisi0 := (rapos*0.16)/1024;

dposisi1:=dposisi0;
dteta1  :=dteta0;

       dsud        := (teta0-teta1)*10;
       dpos        := (posisi0-posisi1)*10;
       dtet_vec[3] := dtet_vec[2];
       dpos_vec[3] := dpos_vec[2];
       dtet_vec[2] := dtet_vec[1];
       dpos_vec[2] := dpos_vec[1];
       dtet_vec[1] := dsud;
       dpos_vec[1] := dpos;
       dtotet      := dtet_vec[1]+0.8*dtet_vec[2]+0.2*dtet_vec[3];
       dtopos      := dpos_vec[1]+0.8*dpos_vec[2]+0.2*dpos_vec[3];

dteta0   := dtotet/2;
dposisi0 := dtopos/2;

if o=#13 then
  begin
    nncon;
    pwm;
```

```
      end
     else
       begin
              port[$300] := $00;
              port[$307] := $BA;
              Port[$306] := $01;
              Port[$306] := $00;
       end;
 end;
   port[$20]:= $20; {end of interrupt}
end;

procedure tampilan;
begin
  n:=35;
  cleardevice;
  SetColor(white);
  OutTextXY(10,0,'Exit=(ESC)   Start=(Enter)  Stop=(Space)
  Waktu sampling = 200 Hz ');
  OutTextXY(10, 18,' 25');
  OutTextXY(10, 38,' 20');
  OutTextXY(10, 58,' 15');
  OutTextXY(10, 78,' 10');
  OutTextXY(10, 98,'  5');
  OutTextXY(10,118,'  0');
  OutTextXY(10,138,' -5');
  OutTextXY(10,158,'-10');
  OutTextXY(10,178,'-15');
  OutTextXY(10,198,'-20');
  OutTextXY(10,218,'-25');
  SetTextStyle(DefaultFont, VertDir, 1);
  OutTextXY(8,90,'Sudut (deg)');
  SetTextStyle(DefaultFont, HorizDir, 1);
  OutTextXY( 40,225,'0');
  OutTextXY( 80,225,'0.25');
  OutTextXY(130,225,'0.50');
  OutTextXY(180,225,'0.75');
  OutTextXY(230,225,'1.25');
  OutTextXY(280,225,'1.00');
  OutTextXY(330,225,'1.50');
  OutTextXY(380,225,'1.75');
  OutTextXY(430,225,'2.00');
  OutTextXY(480,225,'2.25');
  OutTextXY(530,225,'2.50');
  OutTextXY(580,225,'2.75');
  OutTextXY(10,245,' 12');
  OutTextXY(10,270,'  9');
  OutTextXY(10,295,'  6');
  OutTextXY(10,320,'  3');
  OutTextXY(10,345,'  0');
  OutTextXY(10,370,' -3');
  OutTextXY(10,395,' -6');
OutTextXY(10,420,' -9');
OutTextXY(10,445,'-12');
SetTextStyle(DefaultFont, VertDir, 1);
OutTextXY(8,305,'Posisi  (cm)');
SetTextStyle(DefaultFont, HorizDir, 1);
OutTextXY( 40,455,'0');
OutTextXY( 80,455,'0.25');
OutTextXY(130,455,'0.50');
OutTextXY(180,455,'0.75');
OutTextXY(230,455,'1.00');
OutTextXY(280,455,'1.25');
OutTextXY(330,455,'1.50');
OutTextXY(380,455,'1.75');
OutTextXY(430,455,'2.00');
OutTextXY(480,455,'2.25');
OutTextXY(530,455,'2.50');
OutTextXY(580,455,'2.75');
OutTextXY(285,470,'Detik');
SetColor(green);
line( 40,275,600,275);
line( 40,300,600,300);
line( 40,325,600,325);
line( 40,350,600,350);
line( 40,375,600,375);
line( 40,400,600,400);
line( 40,425,600,425);
line( 90,450, 90,250);
line(140,450,140,250);
line(190,450,190,250);
line(240,450,240,250);
line(290,450,290,250);
line(340,450,340,250);
line(390,450,390,250);
line(440,450,440,250);
line(490,450,490,250);
line(540,450,540,250);
line(590,450,590,250);
line( 40, 40,600, 40);
line( 40, 60,600, 60);
line( 40, 80,600, 80);
line( 40,100,600,100);
line( 40,120,600,120);
line( 40,140,600,140);
line( 40,160,600,160);
line( 40,180,600,180);
line( 40,200,600,200);
line( 90,220, 90, 20);
line(140,220,140, 20);
line(190,220,190, 20);
line(240,220,240, 20);
line(290,220,290, 20);
line(340,220,340, 20);
```

```
  line(390,220,390, 20);
  line(440,220,440, 20);
  line(490,220,490, 20);
  line(540,220,540, 20);
  line(590,220,590, 20);
 end;

begin
     clrscr;
     assign(databobot,'bobot.m');
     reset(databobot);
     for i:=1 to 12 do
      for j:=1 to 20 do
       begin
         readln(databobot,temp);
         val(temp,W1c[j,i],code);
       end;
     for i:=1 to 20 do
      for j:=1 to 1 do
       begin
         readln(databobot,temp);
         val(temp,W2c[j,i],code);
       end;
     close(dataBOBOT);

     sudut      := true;
     yref       := 0;
     teta0      := 0;
     teta1      := 0;
     posisi0    := 0;
     posisi1    := 0;
     dposisi0   := 0;
     dposisi1   := 0;
     dteta0     := 0;
     dteta1     := 0;
     u0         := 0;
     u1         := 0;
     n_pwm      := 1;
     Driver     := Detect;
     InitGraph(Driver, Mode, 'e:\tp7\bgi');
     tampilan;

     port[$307] := $34; {Mode 2 untuk Counter0 PIT 8254                 }
     port[$304] := $10; {waktu sampling : 4 MHz/40000($9c40) = 100 Hz   }
     port[$304] := $27; {                       50000($c350) =  80 Hz   }
                        {                       20000($4e20) = 200 Hz   }
     port[$307] := $74; {                       10000($2710) = 400 Hz   }
     port[$305] := $88; {                        5000($1388) = 800 Hz   }
     port[$305] := $13; {                        4000($0FA0) =   1  KHz
                                                 2000($07D0) =   2  KHz
                                                 1000($03E8) =   4  KHz
                                                  500($01F4) =   8  KHz }
     port[$307] := $BA;
     Port[$306] := $01;
     Port[$306] := $00;

     port[pcw_ppi] := cw_ppi;
     port[pc]      := $02;    {<- pilih channel multiplexer }

     pancing   := (portw[adc] and $0FFF);

   Getintvec($0D,rev);          {menyimpan alamat IRQ5 ke rev}
   Setintvec($0D,@baca_adc);  {set alamat IRQ5}
   port[$21] := port[$21] and $DF;  {enable IRQ5}

    repeat
      if keypressed then o := readkey;
      putpixel(n,118 - round(  teta0 * 400),yellow);
      putpixel(n,349 - round(posisi0 * 700),lightred);
      SetColor(white);
      rectangle ( 40, 20,600,220);
      rectangle ( 40,250,600,450);
      if n>600 then Tampilan;
    until o=#27;
  closegraph;
  port[$306] := $01;
  port[$306] := $00;
  port[$300] := $00;
  port[$21]  := port[$21] or $20; {disable IRQ5 8259}
  setintvec($0D,rev);
end.
```

```
% Program untuk membuktikan bahwa state (0,0,0,0) merupakan titik
% minimum local/global serta menghitung nilai minimumnya.


%Mp =0.025;                %massa pendulum
%l  =0.5;                  %jarak antara pivot dan titik berat pendulum
%Jo =1/3*Mp*(l^2);         %momen inersia pendulum
%g  =9.8;                  %gravitasi bumi
%k  =20;                   %konstanta pegas

%y        = 0;   %posisi cart
%ydot     = 0;   %kecepatan cart
%theta    = 0;   %sudut pendulum terhadap vertikal
%thetadot = 0;   %kecepatan sudut pendulum
%syms Mp l Jo g k y ydot theta thetadot;

%L= 0.5*(Mp*(ydot^2 + 2*ydot*thetadot*l*cos(theta) + l^2*thetadot^2)...
    + Jo*thetadot^2) - Mp*g*l*cos(theta) + 0.5*k*y^2;

%Fy                 = diff(L,y);
%Fydot              = diff(L,ydot);
%Ftheta             = diff(L,theta);
%Fthetadot          = diff(L,thetadot);
%Fyy                = diff(Fy,y)
%Fydotydot          = diff(Fydot,ydot)
%Fthetatheta        = diff(Ftheta,theta)
%Fthetadotthetadot  = diff(Fthetadot,thetadot)
%Fyydot             = diff(Fy,ydot)
%Fytheta            = diff(Fy,theta)
%Fythetadot         = diff(Fy,thetadot)
%Fydottheta         = diff(Fydot,theta)
%Fydotthetadot      = diff(Fydot,thetadot)
%Fthetathetadot     = diff(Ftheta,thetadot)

%D = Fyy*Fydotydot*Fthetatheta*Fthetadotthetadot -(Fyydot^2)...
%    -(Fytheta^2) -(Fythetadot^2) -(Fydottheta^2) -(Fydotthetadot^2)...
%    -(Fthetathetadot^2)

Mp =0.025;                %massa pendulum
l  =0.1;                  %jarak antara pivot dan titik berat pendulum
Jo =1/3*Mp*(l^2);         %momen inersia pendulum
g  =9.8;                  %gravitasi bumi
k  =20;                   %konstanta pegas

syms y ydot theta thetadot;

L = 0.5*(Mp*(ydot^2 + 2*ydot*thetadot*l*cos(theta) + l^2*thetadot^2)...
    + Jo*thetadot^2) - Mp*g*l*cos(theta) + 0.5*k*y^2;

Fy        = k.*y;
Fydot     = 1/2.*Mp.*(2.*ydot+2.*thetadot.*l.*cos(theta));
Ftheta    = -Mp.*ydot.*thetadot.*l.*sin(theta)+ Mp.*g.*l.*sin(theta);
Fthetadot = 1/2.*Mp.*(2.*ydot.*l.*cos(theta)+2.*l.^2.*thetadot) ...
            +Jo.*thetadot;
```

```
[y,ydot,theta,thetadot] = solve(Fy,Fydot,Ftheta,Fthetadot);

L = 0.5*(Mp*(ydot^2 + 2*ydot*thetadot*l*cos(theta) + l^2*thetadot^2)...
    + Jo*thetadot^2) - Mp*g*l*cos(theta) + 0.5*k*y^2;

Fyy           = k
Fydotydot     = Mp
Fthetatheta   = -Mp.*ydot.*thetadot.*l.*cos(theta)+Mp.*g.*l.*cos(theta)
Fthetadotthetadot = Mp.*l.^2+Jo

Fyydot         = 0;
Fytheta        = 0;
Fythetadot     = 0;
Fydottheta     = -Mp.*thetadot.*l.*sin(theta);
Fydotthetadot  = Mp.*l.*cos(theta);
Fthetathetadot = -Mp.*ydot.*l.*sin(theta);

D =-5/96.*ydot.*thetadot.*cos(theta)+49/96.*cos(theta)- ...
     Mp.^2.*thetadot.^2.*l.^2.*sin(theta).^2- ...
     Mp.^2.*l.^2.*cos(theta).^2- Mp.^2.*ydot.^2.*l.^2.*sin(theta).^2
```

```
% Program untuk melatih model plan.

load datamod;
netplan=newff([-10 10;-5 5;-3 3;-3 3;-10 10;-5 5;-3 3;-3 3;-5 5;-5 5],...
         [20 4],{'tansig' 'purelin'},'trainbfg')
netplan.biasConnect= [1; 0]
netplan.performFcn='mse'
netplan.trainParam.show = 1
netplan.trainParam.epochs = 300
netplan.trainParam.goal=1e-8
netplan.trainParam.min_grad=0
netplan = train(netplan,u2',y2')
Y = sim(netplan,u2');
E=Y-y2';
mse(E)
subplot(3,1,1),plot(y2);
subplot(3,1,2),plot(Y');
subplot(3,1,3),plot(u2);
```

```
clear;
clc;
% -----------------------------> OTRINIT.M  <-----------------------------
% Inisialisasi file untuk "otrain"

model    = 'simulink';
l  = .1;
m = 0.025;
M = 0.1;
g = 9.8;

% ------    Inisialisasi Umum  -------
Ts       = 0.01;             % waktu sampling (detik)
maxiter  = 1;
samples  = 200;             % jumlah sampel;
u_0      = 0;                % sinyal kontrol awal
y_0      = 0;                % state awal
ulim_min = -5;             % sinyal kontrol minimum
ulim_max = 5;              % sinyal kontrol maksimum
maxepoch = 200;
yref = 0;


% --  State awal --
stateku;

% --  Sistem yang akan dikontrol (SIMULINK) --
integrator = 'ode45';    % nama dif. eq. solver (contoh: ode45 atau ode15s)
sim_model  = 'pend';     % nama model SIMULINK


% ----- Neural Network ------
nnforw = 'forward';          % nama file jaringan model plan
nnctrl = 'bobotawal';        % nama file jaringan kontroler
```

```
%-----------------------------------------------------------------------
%-----------------            >>>  INISIALISASI  <<<        ---------------
%-----------------------------------------------------------------------

%>>>>>>>>>>>>>>>>>>      Membaca file inisialisasi      <<<<<<<<<<<<<<<<<<
clear plot_a plot_b
global ugl
otrinit;                                     % Run file inisisalisasi
eval(['load ' nnctrl]);                      % Membuka jaringan kontroler
eval(['load ' nnforw]);                      % Membuka jaringan model plan

%>>>>>>>>>>>>>>>>>        INISIALISASI VARIABEL      <<<<<<<<<<<<<<<<<<<<
% Inisialisasi vektor yang berisi sinyal lama

maxlength = 2;
y_old2     = repmat(y_0,maxlength,4);
u_old2     = repmat(u_0,maxlength,1);


% Inisisalisasi vektor yang digunakan untuk menyimpan data
   ref_data    = zeros(samples,1);
   u_data      = zeros(samples,1);
   y_data      = zeros(samples,4);

% Inisialisasi lain
t    = -Ts;
u    = u_0;
y    = y_0;
yhat = y_0;
i    =   0;                                    % Counter iterasi
[jumlahstateawal,n0]=size(stateawal_mat);

% Inisialisasi model simulink

simoptions = simset('Solver',integrator,'MaxRows',0); % menset pilihan int.
eval(['[sizes,x0]=pend([],[],[],0);']); %membaca state awal model simulink

y_old2(:,1)  = shift(y_old2(:,1),x0(1));
y_old2(:,2)  = shift(y_old2(:,2),x0(2));
y_old2(:,3)  = shift(y_old2(:,3),x0(3));
y_old2(:,4)  = shift(y_old2(:,4),x0(4));
u_old2       = shift(u_old2,u_0);

%-----------------------------------------------------------------------
%--------------------          >>>   MAIN LOOP   <<<          ------------
%-----------------------------------------------------------------------
 L_vec     = zeros(maxepoch,1);

for epoch=1:maxepoch,

for s=1:jumlahstateawal,
in0 = [stateawal_mat(s,1) stateawal_mat(s,2) stateawal_mat(s,3) ...
      stateawal_mat(s,4)];

  langkah = langkah_mat(s);
```

```
   % Inisisalisasi vektor yang digunakan untuk menyimpan data
   phic_vec    = zeros(langkah,12);
   outhidc_vec = zeros(langkah,20);
   nethidc_vec = zeros(langkah,20);
   netuc_vec   = zeros(langkah,1);
   u_vec       = zeros(langkah,1);
   phim_vec    = zeros(langkah,11);
   outhidm_vec = zeros(langkah,20);
   nethidm_vec = zeros(langkah,20);
   netum_vec   = zeros(langkah,4);
   yhat_vec    = zeros(langkah,4);
   dL_vec      = zeros(langkah,4);
   jumF_W1c    = zeros(size(W1c));
   jumF_W2c    = zeros(size(W2c));
   deltaW2=0;
   deltaW1=0;
   oldW1c=0;
   oldW2c=0;

  % Inisialisasi vektor yang berisi sinyal lama
  maxlength = 2;
  y_old    = repmat(y_0,maxlength,4);
  u_old    = repmat(u_0,maxlength,1);

  y_old(:,1) = shift(y_old(:,1),in0(1));
  y_old(:,2) = shift(y_old(:,2),in0(2));
  y_old(:,3) = shift(y_old(:,3),in0(3));
  y_old(:,4) = shift(y_old(:,4),in0(4));
  u_old = shift(u_old,u_0);

  for N=1:langkah,

  %>>>>>>>>>>>>>>>>>>>    Menentukan sinyal kontrol     <<<<<<<<<<<<<<<<<<<
  phic = [yref;y_old(1,1); y_old(1,2); y_old(1,3); y_old(1,4);...
     y_old(2,1); y_old(2,2); y_old(2,3); y_old(2,4); u_old(1); u_old(2);1];
  nethidc = W1c*phic;
  outhidc = tansig(nethidc);
  netuc   = W2c*outhidc;
  outuc   = netuc;
  u_net   = outuc;

% Memastikan sinyal kontrol dalam batas yang ditentukan
  if u_net > ulim_max,
     u_net = ulim_max;
  elseif u_net < ulim_min
     u_net = ulim_min;
  end

%>>>>>>>>>>  Memprediksi output dari plan dengan model neural <<<<<<<<<<<<
  phim = [y_old(1,1); y_old(1,2); y_old(1,3); y_old(1,4);...
        y_old(2,1); y_old(2,2); y_old(2,3); y_old(2,4); u_net; u_old(1);1];
  nethidm = W1m*phim;
  outhidm = tansig(nethidm);
  netum   = W2m*outhidm;
  outum   = netum;
  yhat    = outum;
```

```
%>>>>>>>>>>>>>>  Costfun <<<<<<<<<<<<<
  [L,dL_dy,dL_dydot,dL_dtheta,dL_dthetadot] = costfun(yhat,yref);
  R=L;
%>>>>>>>>>>>>>>  simpan data <<<<<<<<<<<<<
  phic_vec(N,:)     = phic';
  outhidc_vec(N,:) = outhidc';
  nethidc_vec(N,:) = nethidc';
  netuc_vec(N,:)   = netuc';
  u_vec(N,:)       = u_net';
  phim_vec(N,:)    = phim';
  outhidm_vec(N,:) = outhidm';
  nethidm_vec(N,:) = nethidm';
  netum_vec(N,:)   = netum';
  yhat_vec(N,:)    = yhat';
  dL_vec(N,:)      = [dL_dy dL_dydot dL_dtheta dL_dthetadot];

 %>>>>>>>>>>>>>>>>>>>>     Menyimpan sinyal lama      <<<<<<<<<<<<<<<<<<<<
 y_old(:,1) = shift(y_old(:,1),outum(1));
 y_old(:,2) = shift(y_old(:,2),outum(2));
 y_old(:,3) = shift(y_old(:,3),outum(3));
 y_old(:,4) = shift(y_old(:,4),outum(4));
 u_old      = shift(u_old,u_net);

end
  %>>>>>>>>>>>>>>  backprop error <<<<<<<<<<<<<
  du_fut      = [0];
  dy_fut      = [0 0 0 0] + dL_vec(langkah,:);
  F_inmod_fut = zeros(11,1);
  F_incon_fut = zeros(12,1);
  F_W2c       = zeros(size(W2c));
  F_W1c       = zeros(size(W1c));
  for N=langkah:-1:1,
     F_Y          = dy_fut;% + dL_vec(N,:);
     F_netum      = F_Y*1;
     F_outhidm    = F_netum*W2m;
     F_nethidm    = F_outhidm.*dtansig(nethidm_vec(N,:),outhidm_vec(N,:));
     F_inmod      = F_nethidm*W1m;
     F_U          = du_fut + F_inmod(9);
     F_netuc      = F_U*1;
     F_outhidc    = F_netuc*W2c;
     F_nethidc    = F_outhidc.*dtansig(nethidc_vec(N,:),outhidc_vec(N,:));
     F_incon      = F_nethidc*W1c;
     F_W2c        = (F_netuc*outhidc)';
     F_W1c        = (phic*F_nethidc)';
     jumF_W1c     = jumF_W1c + F_W1c;
     jumF_W2c     = jumF_W2c + F_W2c;
     F_inmod_fut = F_inmod;
     F_incon_fut = F_incon;
     du_fut       = F_incon(10) + F_inmod(10)+ F_incon_fut(11);
     dy_fut       = F_incon(2:5) + F_inmod(1:4) + F_incon_fut(6:9) +
F_inmod_fut(5:8);
  end

  F_W1c = jumF_W1c/jumlahstateawal;
  F_W2c = jumF_W2c/jumlahstateawal;
```

```
    W2c = W2c - 0.001*F_W2c; %+ 0.01*deltaW2;
    W1c = W1c - 0.001*F_W1c; %+ 0.01*deltaW1;
%    deltaW2=W2c-oldW2c;
 %   deltaW1=W1c-oldW1c;
  %  oldW2c=W2c;
   % oldW1c=W1c;

 % subplot(3,1,1);stairs(0:langkah,[in0(1); yhat_vec(:,1)]);title(['posisi
epoch= ' num2str(epoch) ' ']);grid;
%  axis([0 langkah -2 2]);
%  subplot(3,2,2);stairs(0:langkah,[in0(2);
yhat_vec(:,2)]);title('kecepatan');grid;
 % axis([0 langkah -3 3]);
 % subplot(3,1,2);stairs(0:langkah,[in0(3);
yhat_vec(:,3)]);title('sudut');grid;
%  axis([0 langkah -1 1]);
%  subplot(3,2,4);stairs(0:langkah,[in0(4);
yhat_vec(:,4)]);title('kec_sudut');grid;
%  axis([0 langkah -2 2]);
% subplot(3,1,3);stairs(0:langkah,[u_0; u_vec(:,1)]);title('sinyal
kontrol');grid;
%  axis([0 langkah -20 20]);
%R
L_vec(epoch)= L_vec(epoch)+R;

%  pause(0.1);

end
end;

 t    = -Ts;
 for iter=1:samples*maxiter,

  i = i+1;
  t = t + Ts;

%>>>>>>>>>>>>>>>>>>>>>>>    Menentukan sinyal kontrol
<<<<<<<<<<<<<<<<<<<<<<<
phic = [yref;y_old2(1,1); y_old2(1,2); y_old2(1,3); y_old2(1,4);...
y_old2(2,1); y_old2(2,2); y_old2(2,3); y_old2(2,4); u_old2(1);u_old2(2);1];
  nethidc = W1c*phic;
  outhidc = tansig(nethidc);
  netuc   = W2c*outhidc;
  outuc   = netuc;
  u_net   = outuc;

% Memastikan sinyal kontrol dalam batas yang ditentukan
  if u_net > ulim_max,
     u_net = ulim_max;
  elseif u_net < ulim_min
     u_net = ulim_min;
  end

%>>>>>>>>>       Membaca output dari model simulink        <<<<<<<<<<<<<
if strcmp(model,'simulink'),
  utmp=[t-Ts,u_net;t,u_net];
  simoptions.InitialState=x0;
  [time,x0,y] = sim('pend',[t-Ts t],simoptions,utmp);
  x0 = x0(size(x0,1),:)';
  y  = y(size(y,1),:)';
else
%>>>>>>>>>>>>      Membaca output dari model NN        <<<<<<<<<<<<<<<
phim = [y_old2(1,1); y_old2(1,2); y_old2(1,3); y_old2(1,4);...
  y_old2(2,1); y_old2(2,2); y_old2(2,3); y_old2(2,4); u_net; u_old2(1);1];
  nethidm = W1m*phim;
  outhidm = tansig(nethidm);
  netum   = W2m*outhidm;
  outum   = netum;
  yhat    = outum;
  y=yhat;
end;

%>>>>>>>>>>>>>>      Meyimpan data pada vektor data        <<<<<<<<<<<<<<
u_data(i)      = u_net;
y_data(i,:)    = y';

%>>>>>>>>>>>>>>>>>>>>     Menyimpan sinyal lama      <<<<<<<<<<<<<<<<<<<<
 y_old2(:,1)  = shift(y_old2(:,1),y(1));
 y_old2(:,2)  = shift(y_old2(:,2),y(2));
 y_old2(:,3)  = shift(y_old2(:,3),y(3));
 y_old2(:,4)  = shift(y_old2(:,4),y(4));
 u_old2       = shift(u_old2,u_net);



%>>>>>>>>>>>>>>>>>>>>>>>>       DRAW PLOTS       <<<<<<<<<<<<<<<<<<<<<<<<<
if i==samples,
figure(gcf)

subplot(2,1,1);
stairs([0:Ts:(Ts*(samples-1))],y_data);          % Plot plmat
legend('posisi(m)','kecepatan(m/s)','sudut(rad)','kecepatan sudut(rad/s)');
xlabel(' Waktu (detik)');
set(gca,'Xlim',[0 Ts*(samples-1)]);       % Set x-axis
grid on

subplot(2,1,2);
stairs([0:Ts:(Ts*(samples-1))],u_data);          % Plot plmat
legend('sinyal kontrol');
xlabel(' Waktu (detik)');
set(gca,'Xlim',[0 Ts*(samples-1)]);       % Set x-axis
grid on

figure(gcf); drawnow
i   = 0;
end
end

%-----------------------------------------------------------------------
%-----------------          >>>   END OF MAIN LOOP   <<<         ---------
%-----------------------------------------------------------------------
subplot(111)
```

```
% Fungsi untuk menghitung hasil turunan pertama terhadap masing-masing
% variable state

function [L,dL_dy,dL_dydot,dL_dtheta,dL_dthetadot]=costfun(state,yref)

Mp =0.025;              %massa pendulum
l  =0.1;                %jarak antara pivot dan titik berat pendulum
Jo =1/3*Mp*(l^2);       %momen inersia pendulum
g  =9.8;                %gravitasi bumi
k  =2;                  %konstanta pegas

y        = (state(1)-yref);   %posisi cart
ydot     = state(2);          %kecepatan cart
theta    = state(3);          %sudut pendulum terhadap vertikal
thetadot = state(4);          %kecepatan sudut pendulum

L = 0.5*(Mp*(ydot^2 + 2*ydot*thetadot*l*cos(theta) + l^2*thetadot^2)...
    + Jo*thetadot^2) - Mp*g*l*cos(theta) + 0.5*k*y^2;

dL_dy        = k*y;

dL_dydot     = 1/2*Mp*(2*ydot+2*thetadot*l*cos(theta));

dL_dtheta    = (-Mp*ydot*thetadot*l*sin(theta)+Mp*g*l*sin(theta));

dL_dthetadot = 1/2*Mp*(2*ydot*l*cos(theta)+2*l^2*thetadot)+Jo*thetadot;
```

```
%anime.m
%Program Animasi Pendulum terbalik

%Spesifikasi model pendulum
maxlength = 2;
y_old2    = repmat(0,maxlength,4);
u_old2    = repmat(0,maxlength,1);
yref      = 0;
ulim_min  = -5;            % sinyal kontrol minimum
ulim_max  = 5;             % sinyal kontrol maksimum

l  = .1;
m = 0.025;
M = 0.1;
g = 9.8;

pmodel = [2*l m M ];
degrad = pi/180;
raddeg = 180/pi;

% INITIAL STATE
simoptions = simset('Solver','ode45','MaxRows',0);% menset integrator
eval(['[sizes,x0]=pend([],[],[],0);']);  % membaca state awal model simulink

y_old2(:,1) = shift(y_old2(:,1),x0(1));
y_old2(:,2) = shift(y_old2(:,2),x0(2));
y_old2(:,3) = shift(y_old2(:,3),x0(3));
y_old2(:,4) = shift(y_old2(:,4),x0(4));
u_old2      = shift(u_old2,0);

Cart_position         = x0(1);
Cart_velocity         = x0(2);
Pendulum_angle        = x0(3)*raddeg;
Pendulum_velocity     = x0(4)*raddeg;

u_net = 0;

x0=[Cart_position Cart_velocity Pendulum_angle*degrad Pendulum_velocity*degrad ];

% Spesifikasi waktu
final_time = 10;   % waktu simulasi
Ts = 0.01;         % waktu sampling

t = 0;
T = Ts:Ts:final_time;
steps = length(T);
plotstep = 10;      % update grafik tiap ... langkah

%untuk merekam state variabel dan sinyal kontrol
X = zeros(4,steps);
U = zeros(1,steps);

% inisialisasi demo
anime_init
```

```
% Simulasi pendulum
% tampilkan kondisi awal pendulum
anime_plot(x0,u_net,0,f1,1,pmodel);
drawnow;

fprintf('\n Tekan Enter untuk mulai simulasi \n')
pause

i = 1;
while (i<steps),
 i = i+1;
 t = t + Ts;
%>>>>>>>>>>>>>>>>>    Menentukan sinyal kontrol     <<<<<<<<<<<<<<<<<<
phic = [yref;y_old2(1,1); y_old2(1,2); y_old2(1,3); y_old2(1,4);...
y_old2(2,1); y_old2(2,2); y_old2(2,3); y_old2(2,4); u_old2(1); u_old2(2);1];
  nethidc = W1c*phic;
  outhidc = tansig(nethidc);
  netuc   = W2c*outhidc;
  outuc   = netuc;
  u_net   = outuc;

% Memastikan sinyal kontrol dalam batas yang ditentukan
  if u_net > ulim_max,
     u_net = ulim_max;
  elseif u_net < ulim_min
     u_net = ulim_min;
  end
%>>>>>>>>>>>>>       Membaca output dari model simulink          <<<<<<<<<<<
  utmp=[t-Ts,u_net;t,u_net];
  simoptions.InitialState=x0;
  [time,x0,y] = sim('pend',[t-Ts t],simoptions,utmp);
  x0 = x0(size(x0,1),:)';
  y  = y(size(y,1),:)';

y_old2(:,1)  = shift(y_old2(:,1),x0(1));
y_old2(:,2)  = shift(y_old2(:,2),x0(2));
y_old2(:,3)  = shift(y_old2(:,3),x0(3));
y_old2(:,4)  = shift(y_old2(:,4),x0(4));
u_old2       = shift(u_old2,u_net);

% Tampilkan tiap plotstep
 if rem(i,plotstep) <= eps
    anime_plot(y,u_net,t,f1,0,pmodel);
 end;
 % rekam state
 X(:,i) = y;
 U(:,i) = u_net;

end

% grafik respon
anime_view

%anime_figs.m
% PLOT RESPON untuk anime
% ==============

set([f1,info_win],'vis','off');
set(info_str,'string','');

% respon posisi
figure(f2);
subplot(211); plot([0 T],[x0(1) X(1,:)])
xlabel('Waktu (detik)')
ylabel('Posisi (meter)')
grid

subplot(212); plot([0 T],[x0(2) X(2,:)])
xlabel('Waktu (detik)')
ylabel('Kecepatan (meter/detik)')
grid

% respon sudut
figure(f3);
subplot(211); plot([0 T],[x0(3) X(3,:)]*raddeg)
xlabel('Waktu (detik)')
ylabel('Sudut (derajat)')
grid

subplot(212); plot([0 T],[x0(4) X(4,:)]*raddeg)
xlabel('Waktu (detik)')
ylabel('Kecepatan Sudut (derajat/detik)')
grid

% sinyal kontrol
figure(f5);
plot([0 T],[0 U])
xlabel('Waktu (detik)')
ylabel('Sinyal Kontrol (Newton)')
grid

set(info_str(1),'string','Simulasi Pendulum Selesai!');
set(b([1,3]),'vis','off');
set(b(2),'string','Selesai','callback',get(b(1),'callback'),'vis','on');
set(info_win,'vis','on');
```

```
% anime__init.m
% window pendulum
f1 = figure('name','Simulasi Pendulum Terbalik','numbertitle','off',...
            'units','norm','pos',[20,20,600,.66*430]./scrn,'menubar','none',...
            'vis','off');

% window respon posisi
f2 = figure('name','Respon Posisi','numbertitle','off',...
            'units','norm','pos',[325,10,305,200]./scrn,'menubar','none',...
            'vis','off');

% window respon sudut
f3 = figure('name','Respon Sudut','numbertitle','off',...
            'units','norm','pos',[10,10,305,200]./scrn,'menubar','none',...
            'vis','off');

% CONTROL EFFORT WINDOW
f5 = figure('name','Sinyal Kontrol', 'numbertitle','off',...
            'units','norm','pos',[325,240,305,200]./scrn,'menubar','none',...
            'vis','off');

windows=[info_win,f1,f2,f3,f5];

set(info_win,'vis','off');
set(info_str,'string','');

% anime_view.m
% Tampilkan respon atau tidak ?
left=(scrnsz(3)-300)/2;
botm=(scrnsz(4)-80)/2;
set(info_win,'pos',[left,botm,300,80]);
set(info_str(1),'string','Tampilkan Grafik Respon..?','pos',[20,55,260,20]);
set(info_str(2),'pos',[20,30,260,20]);
set(b(1),'pos',[7,5,90,20],'vis','on');
set(b(2),'vis','off','pos',[105,5,90,20]);
set(b(3),'string','Respon','callback','anime_figs','pos',[203,5,90,20],...
         'vis','on');
set(info_win,'vis','on');
```

```
% anime_plot.m
function anime_plot(x,u,t,fig,flag,pmodel)
% Plot pendulum
%
%   x - state pendulum.
%   u - sinyal kontrol.
%   t - waktu.
%   fig - pointer gambar
%   flag - set = true pada saat pertamakali memanggil fungsi ini

% konstanta
L1 = 2*pmodel(1);
tinggi = 0.6;

bkclr=[0,0,0];

% state
pos  = x(1);
vel  = x(2);
phi1 = x(3);
cp=0.16*exp(i*[0:0.1:2*pi]);

% plot batas tepi
 midx = 0;
 minx = 1.5*(midx-2.5);
 maxx = 1.5*(midx+2.5);
 maxy = 1.5*(4*.66);

% ujung pendulum
 xp=sin(phi1);
 yp=cos(phi1);
 deg =  180/pi;

% plot
% ========
if flag,
  figure(fig);axis('image');
  pen_win=[400,200,400,200];
  lft=70; btm= 12;
  lft2 = 40;

uicontrol(fig,'style','text','units','norm','pos',[10+lft2,170+btm,60,...
   12]./pen_win,'string','Posisi             :','horizontalalignment','left');
txt(1)=uicontrol(fig,'style','text','units','norm','pos',[40+lft2,...
   170+btm,20,12]./pen_win,'string',sprintf('%3.2f',pos));

uicontrol(fig,'style','text','units','norm','pos',[10+lft2,160+btm,60,...
   12]./pen_win,'string','Sudut              :','horizontalalignment','left');
txt(2)=uicontrol(fig,'style','text','units','norm','pos',[40+lft2,...
   160+btm,20,12]./pen_win,'string',sprintf('%3.2f',phi1*deg));

uicontrol(fig,'style','text','units','norm','pos',[10+lft2,150+btm,60,...
   12]./pen_win,'string','Sinyal Kontrol :','horizontalalignment','left');
txt(3)=uicontrol(fig,'style','text','units','norm','pos',[40+lft2,...
   150+btm,20,12]./pen_win,'string',sprintf('%3.2f',u));
```

```
uicontrol(fig,'style','text','units','norm','pos',[10+lft2,140+btm,60,...
   12]./pen_win,'string','Waktu            :','horizontalalignment','left');
txt(4)=uicontrol(fig,'style','text','units','norm','pos',[40+lft2,...
   140+btm,20,12]./pen_win,'string',sprintf('%5.2f',t));

uicontrol(fig,'style','text','units','norm','pos',[220+lft,170+btm,60,...
   12]./pen_win,'string','Panjang Pendulum :','horizontalalignment','left');
txt(5)=uicontrol(fig,'style','text','units','norm','pos',[270+lft,...
   170+btm,15,12]./pen_win,'string',sprintf('%5.2f',L1));

uicontrol(fig,'style','text','units','norm','pos',[220+lft,160+btm,60,...
   12]./pen_win,'string','Massa Pendulum    :','horizontalalignment','left');
txt(6)=uicontrol(fig,'style','text','units','norm','pos',[270+lft,...
   160+btm,15,12]./pen_win,'string',sprintf('%5.2f',pmodel(2)));

uicontrol(fig,'style','text','units','norm','pos',[220+lft,150+btm,60,...
 12]./pen_win,'string','Massa Cart            :','horizontalalignment','left');
txt(7)=uicontrol(fig,'style','text','units','norm','pos',[270+lft,...
   150+btm,15,12]./pen_win,'string',sprintf('%5.2f',pmodel(3)));


  set(txt,'horizontalalignment','right');
  ax=get(fig,'currentaxes');
  set(ax,'box','off','xtick',[],'ytick',[],'xcolor',[0 0 0],'ycolor',[0 0 0],...
          'xlim',[minx,maxx],'ylim',[0,maxy],'vis','off',...
          'nextplot','add');

  figure(fig);

% Gambar Cart
ln(1)=patch('xdata',[-0.5,0.5,0.5,-0.5]+pos,'ydata',[0.3,0.3,0.6,0.6],...
   'facecolor','r','erase','xor');

% Gambar Pendulum
 x1 = xp*2*L1 + pos;
 y1 = yp*2*L1 + tinggi;
 x1 = xp*2*(L1+0.1) + pos;
 y1 = yp*2*(L1+0.1) + tinggi;


 ln(4)=line('xdata',[pos x1],'ydata',[tinggi y1],...
             'linewidth',4,'color','b','erase','xor');
 ln(3)=patch('xdata',(real(cp)+x1),'ydata',imag(cp)+y1,'facecolor','b',...
              'erase','xor');

% Gambar sumbu axis
ln(5)=line('xdata',[minx minx],'ydata',[0 .5],'color','black','erase','xor');
ln(6)=line('xdata',[maxx maxx],'ydata',[0 0.5],'color','black','erase','xor');
ln(7)=line('xdata',[midx midx],'ydata',[0 0.25],'color','black','erase','xor');
ln(8)=line('xdata',[minx maxx],'ydata',[0.25 0.25],'color','black','erase','xor');

txt(7)=text('pos',[minx-.1,-0.1],'string',sprintf('%g',minx),'erase','xor');
txt(8)=text('pos',[maxx-.1,-0.1],'string',sprintf('%g',maxx),'erase','xor');
set(fig,'userdata',[ln,txt]);
```

```
else
 han=get(fig,'userdata');
 ln=han(1:8);
 txt=han(9:16);

 set(ln(1),'xdata',[-0.5,0.5,0.5,-0.5]+pos);

 x1 = xp*2*L1 + pos;
 y1 = yp*2*L1 + tinggi;
 x1 = xp*2*(L1+0.1) + pos;
 y1 = yp*2*(L1+0.1) + tinggi;

 set(ln(4),'xdata',[pos x1],'ydata',[tinggi y1]);
 set(ln(3),'xdata',real(cp)+x1,'ydata',imag(cp)+y1);

 set(txt(1),'string',sprintf('%5.2f',pos));
 set(txt(2),'string',sprintf('%5.2f',phi1*deg));
 set(txt(3),'string',sprintf('%5.2f',u));
 set(txt(4),'string',sprintf('%5.2f',t));

end
```