LAMPIRAN

5.3 Main Program

```
clc
clear all
close all
addpath('DataLog')

%% Get Input
%%% Get Control Method
ControlMethod = input('Pilih Metode Kontrol (1.Consensus Based, 2. Vector
Feld Based): ');

%%% Get initial Leader
firstInitialLeader = input('Masukan Posisi awal virtual leader(VL)
[x,y,theta]:');
firstInitialLeader
vvirtualLeader = input('Masukan kecepatan linear dan Angular VL (v,w)
[v,w]:');
vvirtualLeader

switch ControlMethod
    case 1
        SettingMultiMobileRobot.control = 'DistributedControl';
        %% Distributed Consensus Method Control
            %%% Control Variable
            k0 = input('Masukan nilai k0 : ');
            if isempty(k0)
             k0 = 5;
            end
            alpha = input('Masukan nilai alpha : ');
            if isempty(alpha)
             alpha = 10;
            end
            beta = input('Masukan nilai beta : ');
            if isempty(beta)
             beta = 0.5;
            end
            bj = input('Masukan nilai matriks bj : ');
            if isempty(bj)
```

```
            bj = [10.0 10.0 10.0 10.0 10.0];  %% Weighted from leader to
follower for all Agent
            end
            A = input('Masukan nilai matriks a_ij : ');
            if isempty(A)
             %%%%% Average Consensus
            A =[ 0.00    1.00    1.00    1.00    1.00;....      1  \
                 1.00       0    1.00    1.00    1.00;....      2   |
                 1.00    1.00       0    1.00    1.00;....      3   p = agent
                 1.00    1.00    1.00       0    1.00;....      4   |
                 1.00    1.00    1.00    1.00    0.00];....     5  /
            end
    case 2
        SettingMultiMobileRobot.control = 'VectorField';
        %% Vector Field Method Control
            %%% Control Variable
            kl = input('Masukan nilai kl : ');
            if isempty(kl)
             kl = 1;
            end
            kd = input('Masukan nilai kd : ');
            if isempty(kd)
             kd = 0.01;
            end
            ktheta = input('Masukan nilai ktheta : ');
            if isempty(ktheta)
             ktheta = 1;
            end
            sigma = input('Masukan nilai sigma : ');
            if isempty(sigma)
             sigma = 0.1;
            end


end

t = 0:0.01:30;
%% Inisialisasi Model Multi Robot
SettingMultiMobileRobot.numRobots = 5;
SettingMultiMobileRobot.visualization  = true;
SettingMultiMobileRobot.recordData= true;
SettingMultiMobileRobot.figure = figure(...
```

```
    'Units','normalized' ....
    ....'pos', [0 0 600 600] ...
);
SettingMultiMobileRobot.space = axes( ...
    'Units','normalized', ....
    ....'pos',[50 50 1000 1000], ....
    ....'Xlim', [0 1000], ....
    ....'Ylim', [0 1000], ....
    'XLimMode', 'auto',....
    'YLimMode', 'auto',....
    'XGrid', 'on', ....
    'YGrid', 'on' ...
);

%% Option Control
%%% Mohon Gunakan syntax comment untuk me-non-aktifkan control
% SettingMultiMobileRobot.control = 'DistributedControl'; %% Opsi
DistributedControl/VectorField
% SettingMultiMobileRobot.control = 'VectorField'; %% Opsi VectorField

%%-Init Robot
initMultiRobot;

%% Make Virtual Leader Center robot
virtualLeader.robot         = f_unicycle;
if isempty(firstInitialLeader)
    virtualLeader.robot.x       = 0;
    virtualLeader.robot.y       = -250;
    virtualLeader.robot.theta   = 0;
else
    virtualLeader.robot.x       = firstInitialLeader(1);
    virtualLeader.robot.y       = firstInitialLeader(2);
    virtualLeader.robot.theta   = firstInitialLeader(3);
end


if isempty(firstInitialLeader)
    virtualLeader.robot.v       = 100;
    virtualLeader.robot.w       = 0.2;
else

    virtualLeader.robot.v       = vvirtualLeader(1);
```

```
    virtualLeader.robot.w       = vvirtualLeader(2);
end


%%% Record Virtual Leader
virtualLeader.rcx(1)      = 0; %record posisi x
virtualLeader.rcy(1)      = 0; %record posisi y
virtualLeader.rctheta(1)  = 0; %record posisi y
virtualLeader.rcv(1)      = 0; %record control v
virtualLeader.rcw(1)      = 0; %record control w


%% Formation dengan acuan Virtual Center
u(1).px = 0;
u(1).py = 0;

u(2).px = 0;
u(2).py = 50;

u(3).px = 25;
u(3).py = 25;

u(4).px = 50;
u(4).py = 0;

u(5).px = 50;
u(5).py = 50;

%% Simulasi Robot terhadap Waktu
for i= 1:(length(t)-1)
    %% Stop looping

    % Control Schema
    switch SettingMultiMobileRobot.control
        case 'DistributedControl'
            MultiDistributedControl;
        case 'VectorField'
            MultiVectorField;
    end

    % Forward to Robot Model
    for l=1:SettingMultiMobileRobot.numRobots
```

```
        %% Saturation Signal Control
        u(l).robot.v = min(max(-300,u(l).robot.v),300);
        u(l).robot.w = min(max(-300,u(l).robot.w),300);

        %% Update Position
        u(l).robot.updatePosition(t(i+1) - t(i));

        %Record
        u(l).rcv(i) = u(l).robot.v;
        u(l).rcw(i) = u(l).robot.w;

        u(l).rcx(i)     = u(l).robot.x;
        u(l).rcy(i)     = u(l).robot.y;
        u(l).rctheta(i) = u(l).robot.theta;
    end

    % Visualization
    MultiRobotVisualization;
end

% DokumentasiGraphic
```

5.4 Init Multi Robot

```
%% init Unit Robot
colorRobot = ['c','m','y','k','r'];

for i=1:SettingMultiMobileRobot.numRobots
    u(i).robot = f_unicycle;

    u(i).rcx(1)     = 0; %record posisi x
    u(i).rcy(1)     = 0; %record posisi y
    u(i).rctheta(1) = 0; %record posisi y
    u(i).rcv(1)     = 0; %record control v
    u(i).rcw(1)     = 0; %record control w


    u(i).rtheta = pi/2; %referensi theta
    u(i).rv     = 0; %referensi v
    u(i).rw     = 0; %referensi w
    u(i).rx     = 0; %referensi x
```

```
    u(i).ry      = 0; %referensi y

    u(i).rcrv(1)      = 0; %record referensi v
    u(i).rcrw(1)      = 0; %record referensi w
    u(i).rcrx(1)      = 0; %record referensi x
    u(i).rcry(1)      = 0; %record referensi y
    u(i).rcrtheta(1) = 0; %record referensi theta

    % Visualisasi Robot
    u(i).marker = annotation( ...
        SettingMultiMobileRobot.figure,'arrow', ....
        'Units', 'pixels', ....
        'X',[u(i).robot.x+50-(10.1*cos(u(i).robot.theta))
u(i).robot.x+50+(10.1*cos(u(i).robot.theta))], ....
        'Y',[u(i).robot.y+50-(10.1*sin(u(i).robot.theta))
u(i).robot.y+50+(10.1*sin(u(i).robot.theta))], ....
        'HeadWidth', 10, ....
        'Headlength', 10, ....
        'color', colorRobot(i) ...
    );
end

%% Set first position robot
% u(1).x = 0; u(1).y = 0; u(1).theta = 0;
% u(2).x = 0; u(2).y = 0; u(2).theta = 0;
% u(3).x = 0; u(3).y = 0; u(3).theta = 0;
% u(4).x = 0; u(4).y = 0; u(4).theta = 0;
% u(5).x = 0; u(5).y = 0; u(5).theta = 0;
u(1).robot.x =    0;      u(1).robot.y =    0; u(1).robot.theta = 0;
u(2).robot.x =   50;      u(2).robot.y =  100; u(2).robot.theta = 0;
u(3).robot.x =  100;      u(3).robot.y =   50; u(3).robot.theta = 0;
u(4).robot.x =  150;      u(4).robot.y =    0; u(4).robot.theta = 0;
u(5).robot.x =    0;      u(5).robot.y =  150; u(5).robot.theta = 0;
% u(1).x =  200;      u(1).y =  200; u(1).theta = 0;
% u(2).x =  250;      u(2).y =  300; u(2).theta = 0;
% u(3).x =  300;      u(3).y =  250; u(3).theta = 0;
% u(4).x =  350;      u(4).y =  200; u(4).theta = 0;
% u(5).x =  200;      u(5).y =  250; u(5).theta = 0;
% u(1).x = 0; u(1).y = 0; u(1).theta = 0;
% u(2).x = 0; u(2).y = 350; u(2).theta = (7/4)*pi;
% u(3).x = 350; u(3).y = 350; u(3).theta = (5/4)*pi;
% u(4).x = 350; u(4).y = 0; u(4).theta = (3/4)*pi;
```

```
% u(5).x = 0; u(5).y = 150; u(5).theta = 0;
% for l=1:SettingMultiMobileRobot.numRobots
%     set(u(l).marker,....
%     'X',[u(l).x+50-(10.1*cos(u(l).theta))
u(l).x+50+(10.1*cos(u(l).theta))], ....
%     'Y',[u(l).y+50-(10.1*sin(u(l).theta)) u(l).y+50+(10.1*sin(u(l).theta))]
...
%    );
% end
```

5.5 MultiDistributedControl

```
%% Virtual Leader Variable
%%% Transformation Position to Leader Orientation
virtualLeader.robot.theta = normalizeAngle(virtualLeader.robot.theta);
z_10 = virtualLeader.robot.theta;
z_30 = virtualLeader.robot.x*sin(virtualLeader.robot.theta) -
virtualLeader.robot.y*cos(virtualLeader.robot.theta);
z_20 = virtualLeader.robot.x*cos(virtualLeader.robot.theta) +
virtualLeader.robot.y*sin(virtualLeader.robot.theta) +
k0*sign(virtualLeader.robot.w)*z_30;

u_10 = virtualLeader.robot.w;
u_20 = virtualLeader.robot.v - (1 + k0.^2)*u_10*z_30;

dz_10 = u_10;
dz_20 = u_20 + k0*abs(u_10)*z_20;
dz_30 = u_10*z_20 - k0*abs(u_10)*z_30;

%% Forward to Virtual Leader
virtualLeader.robot.updatePosition(t(i+1) - t(i));

%Record
virtualLeader.rcx(i)      = virtualLeader.robot.x; %record posisi x
virtualLeader.rcy(i)      = virtualLeader.robot.y; %record posisi y
virtualLeader.rctheta(i) = virtualLeader.robot.theta; %record posisi y
virtualLeader.rcv(i)      = virtualLeader.robot.v; %record control v
virtualLeader.rcw(i)      = virtualLeader.robot.w; %record control w

x_target        = virtualLeader.robot.x;
y_target        = virtualLeader.robot.y;
```

```
theta_target    =  normalizeAngle(virtualLeader.robot.theta);


%% Membuat Transformasi antar robot
for pp=1:SettingMultiMobileRobot.numRobots
   %%
%    u(pp).rx = x_target*;
%    u(pp).ry = x_target*sin(theta_target) + y_target*cos(theta_target) +
u(pp).ryTrans;
%    u(pp).rtheta = theta_target;

   u(pp).rx = x_target + u(pp).px;
   u(pp).ry = y_target + u(pp).py;
   u(pp).rtheta = theta_target;

   u(pp).rcrx(i)        = u(pp).rx;
   u(pp).rcry(i)        = u(pp).ry;
   u(pp).rcrtheta(i)    = u(pp).rtheta;

end

%% Pre Control
for p=1:SettingMultiMobileRobot.numRobots
    %% Multi-Unicycle Variable
    u(p).robot.theta = normalizeAngle(u(p).robot.theta);
    v = u(p).robot.v;
    w = u(p).robot.w;
    theta =  u(p).robot.theta;
    x = u(p).robot.x;
    y = u(p).robot.y;
    px = u(p).px;
    py = u(p).py;

    %%% Transformation Position to Robot Orientation
    u(p).z_1 = theta;
    u(p).z_3 = (x-px)*sin(theta) - (y-py)*cos(theta);
    u(p).z_2 = (x-px)*cos(theta) + (y-py)*sin(theta) + k0*sign(w)*u(p).z_3;

    u(p).rcz_1(i) = theta;
    u(p).rcz_3(i) = (x-px)*sin(theta) - (y-py)*cos(theta);
    u(p).rcz_2(i) = (x-px)*cos(theta) + (y-py)*sin(theta) +
k0*sign(virtualLeader.robot.w)*u(p).z_3;
```

```
    u(p).u_1 = w;
    u(p).u_2 = v - (1 + k0.^2)*u(p).u_1*u(p).z_3;
%
%     u(p).dz_1 = u(p).u_1;
%     u(p).dz_2 = u(p).u_2 + k0*abs(u(p).u_1)*u(p).z_2;
%     u(p).dz_3 = u(p).u_1*u(p).z_2 - k0*abs(u(p).u_1)*u(p).z_3;

end

%% Begin Distributed Consensus Control
for p = 1:SettingMultiMobileRobot.numRobots

    DC_z_1 = 0;
    DC_z_2 = 0;
    for pp=1:5
%         u(p).f = u(p).f + (dccc(p,j) * u(p).transform(:,j));
        DC_z_1 = DC_z_1 + (A(p,pp) * ( u(p).z_1 - u(pp).z_1 ));
        DC_z_2 = DC_z_2 + (A(p,pp) * ( u(p).z_2 - u(pp).z_2 ));
    end

    %%% Control Signal
    u1 = u_10 - alpha*DC_z_1 - alpha*bj(p)*(u(p).z_1 - z_10)  ...
        - beta*sign( DC_z_1 + alpha*bj(p)*(u(p).z_1 - z_10) );
    u2 = - alpha*DC_z_2 - alpha*bj(p)*(u(p).z_2 - z_20)  ...
        - beta*sign( DC_z_2 + alpha*bj(p)*(u(p).z_2 - z_20) ) ...
        - k0*abs(u1)*u(p).z_2;

    u(p).robot.v = u2;
    u(p).robot.w = u1;

end
```

5.6 MultiVectorField

```
x_offset = 0;
y_offset = 0;
r = 200;

%% Forward to Virtual Leader
```

```
virtualLeader.robot.updatePosition(t(i+1) - t(i));

%Record
virtualLeader.rcx(i)      = virtualLeader.robot.x; %record posisi x
virtualLeader.rcy(i)      = virtualLeader.robot.y; %record posisi y
virtualLeader.rctheta(i) = virtualLeader.robot.theta; %record posisi y
virtualLeader.rcv(i)      = virtualLeader.robot.v; %record control v
virtualLeader.rcw(i)      = virtualLeader.robot.w; %record control w

x_target        = virtualLeader.robot.x;
y_target        = virtualLeader.robot.y;
theta_target    =  normalizeAngle(virtualLeader.robot.theta);
% theta_target = normalizeAngle(theta_target);

%% Membuat Transformasi antar robot
for pp=1:SettingMultiMobileRobot.numRobots
   %%
%    u(pp).rx = x_target*;
%      u(pp).ry  =  x_target*sin(theta_target)  +  y_target*cos(theta_target)  +
u(pp).ryTrans;
%    u(pp).rtheta = theta_target;

   u(pp).rx = x_target + u(pp).px;
   u(pp).ry = y_target + u(pp).py;
   u(pp).rtheta = theta_target;

   u(pp).rcrx(i)        = u(pp).rx;
   u(pp).rcry(i)        = u(pp).ry;
   u(pp).rcrtheta(i)    = u(pp).rtheta;
end
    vt = virtualLeader.robot.v;
    wt = virtualLeader.robot.w;
for p=1:SettingMultiMobileRobot.numRobots
    %% Posisi Aktual
    xc = u(p).robot.x;
    yc = u(p).robot.y;
    theta_c = u(p).robot.theta;
%     theta_c = mod((theta_c + 2*pi),(2*pi));
    theta_c = normalizeAngle(theta_c);
    %% Posisi Target
    xt = u(p).rx;
    yt = u(p).ry;
```

```
    %% Control untuk setiap robot
    [xcr,ycr] = VectorCOR( u(p).rx,u(p).ry,u(p).rtheta,vt,wt );
    if(wt == 0)
        rt = 0;
    else
        rt = vt/abs(wt); %% vt/wt tentang radius target,
%         rt = abs(rt);
    end


    %% Menghitung Gamma C
    gammac = atan2(( u(p).robot.y - ycr ),( u(p).robot.x - xcr ));
%     gammac = mod((gammac + 2*pi),(2*pi));
    gammac = normalizeAngle(gammac);

    %% Measure rc sebagai radius current Position terhadap posisi COR
    rc = (yc - ycr).^2 + (xc - xcr).^2;
    rc = sqrt(rc);
    %% Mengukur selisih radius antara (target - COR) terhadap (current - COR)
    er = rc-rt;
    u(p).rc_er(i) = er;

    %% Mengukur selisih jarak antara posisi aktual dengan posisi target
    el = sqrt( (yt-yc).^2 + (xt-xc).^2 );

    u(p).rc_er(i) = er;
    u(p).rc_el(i) = el;

    gainError = atan(kd*er);
%     gainError = mod((gainError + 2*pi),(2*pi));
    gainError = normalizeAngle(gainError);
    %% Mengukur Orientasi target dan theta d
    if(vt >= 0)

        theta_t = gammac + sign(wt)*(pi/2);
        theta_d = gammac + sign(wt)*( pi/2 + gainError );
    else
        theta_t = gammac + sign(wt)*(pi/2) - pi;
        theta_d = gammac + sign(wt)*( pi/2 + gainError ) - pi;
    end
%     theta_t = mod((theta_t + 2*pi),(2*pi));
```

```
%     theta_d = mod((theta_d + 2*pi),(2*pi));

    theta_t = normalizeAngle(theta_t);
    theta_d = normalizeAngle(theta_d);

    %% Mengukur selisih antara orientasi aktual dengan lc
    qc = theta_c - atan2((yt-yc),(xt-xc));

    %% Mengukur selisih antara orientasi target dengan lc
    qt = theta_t - atan2((yt-yc),(xt-xc));


% %     u(p).rcwv(i) = rad2deg(theta_t);
%     qc = mod( (qc+2*pi), (2*pi) );
%     qt = mod( (qt+2*pi), (2*pi) );
    qc = normalizeAngle(qc);
    qt = normalizeAngle(qt);

    %% Mengukur myu
    myu = 1 - exp( -( abs(qc) - (pi/2) ).^2 / sigma.^2 );

    %% Mengukur error theta
    etheta = theta_c - theta_d;
%     etheta = mod( (etheta+2*pi), (2*pi) );
    etheta = normalizeAngle(etheta);

    thetac_gammac = theta_c-gammac;
%     thetac_gammac = mod( (thetac_gammac+2*pi), (2*pi) );
    thetac_gammac = normalizeAngle(thetac_gammac);

    %% Menghitung Nilai kontrol ke Unicycle
    vc = myu *(1/cos(qc))*( vt * cos(qt) + kl*el );
                wc    =    ((vc/rc)*(sin(thetac_gammac)))    +    sign(wt)    *
( (kd*vc*cos(thetac_gammac))/(1 + (kd*er).^2) ) - ktheta*etheta;

    u(p).robot.v = vc;
    u(p).robot.w = wc;

end
% figure(1001);
%     plot(u(1).rcwv);
%     figure(1002);
```

```
%     plot(u(2).rcwv);
%     figure(1003);
%     plot(u(3).rcwv);
%     figure(1004);
%     plot(u(4).rcwv);
%     figure(1005);
%     plot(u(5).rcwv);
```

5.7 UnicycleModel

```
classdef f_unicycle < handle
    %Robot Model f_unicycle merupakan object robot dengan model unicycle
    %   Pada Object ini memuat beberapa properties dari unicycle dan
    %   kinematicsnya

    properties
        v = 0;
        w = 0;

        theta = 0;
        x = 0;
        y = 0;
    end

    methods
        function u = f_unicycle(initial_X, initial_Y, initial_th, initial_V,
initial_W)
%           u.x         = initial_X;
%           u.y         = initial_Y;
%           u.theta     = initial_th;
%           u.v         = initial_V;
%           u.w         = initial_W;

         % Calling a static method requires the class name
         %%% none Calling
        end

        function updatePosition(u,change_t)
            %UNTITLED Method untuk mengupdate posisi dari robot unicycle
            %   Detailed explanation goes here
            dx      = u.v * cos(u.theta);
```

```
            dy      = u.v * sin(u.theta);
            dtheta  = u.w;

            u.x      = dx * (change_t) + u.x;
            u.y      = dy * (change_t) + u.y;
            u.theta  = dtheta * (change_t) + u.theta;

        end
    end

end
```

5.8 Dokumentasi Grafik

```
clc
clear all

addpath('..');
% f_unicycle;

namanya = 'VectorField_3_LingkaranBesar_';
isDocumented = false;
% isGray = true;

%% Document Visualization Select
naskah = 1;
laporan = 2;
poster = 3;
% docVisType = laporan;
docVisType = naskah;


%% Load Datalog
%%% Track Lingkaran dengan W berlawan jarum jam
% load('19_1_2019_DataLog_VectorFieldMethod_Circle_6.mat');
% load('19_1_2019_DataLog_DistributionMethod_Circle_6.mat');

%% Track Lingkaran dengan W searah jarum jam
% load('19_1_2019_DataLog_VectorFieldMethod_Circle_7.mat');
% load('23_2_2019_DataLog_Consensus_2.mat');
```

```
% load('2019_2_27_Consensus Jauh.mat');
% load('2019_2_27_VectorField Jauh.mat');
% load('2019_2_27_vectorField inside circe v3.mat');
% load('2019_2_27_Consensus inside circe v3.mat');
% load('5_5_2019_DataLog_DistributionMethod_Circle_3.mat')
load('10_5_2019_DataLog_VectorFieldMethod_Circle_3.mat')

%% Delete Last Time
t(length(t)-1) = [];

% Pembagi Resolusi
actual_length = length(t);
resolution_div = 20;
new_data_length = actual_length/resolution_div;

tt(1:new_data_length) = t(1:resolution_div:actual_length);
% tt = t(1:new_data_length);

for(i=1:5)
    u(i).rcxx(1:new_data_length) = u(i).rcx(1:resolution_div:actual_length);
    u(i).rcyy(1:new_data_length) = u(i).rcy(1:resolution_div:actual_length);
                                   u(i).rcthetaa(1:new_data_length)          =
u(i).rctheta(1:resolution_div:actual_length);

    u(i).rcvv(1:new_data_length) = u(i).rcv(1:resolution_div:actual_length);
    u(i).rcww(1:new_data_length) = u(i).rcw(1:resolution_div:actual_length);

%     u(i).pxx(1:new_data_length) = u(i).px(1:resolution_div:actual_length);
%     u(i).pyy(1:new_data_length) = u(i).py(1:resolution_div:actual_length);

                                     u(i).rcrxx(1:new_data_length)          =
u(i).rcrx(1:resolution_div:actual_length);
                                     u(i).rcryy(1:new_data_length)          =
u(i).rcry(1:resolution_div:actual_length);
                                   u(i).rcrthetaa(1:new_data_length)         =
u(i).rcrtheta(1:resolution_div:actual_length);
end

u(1).s = '-oc';
u(2).s = '--sm';
u(3).s = ':dy';
u(4).s = '-^k';
```

```
u(5).s = '-.vr';

%% Error Position Terhadap Potition
for i=1:length(u)
    u(i).error_pos_y = u(i).rcryy - u(i).rcyy;
%
%         subplot(3,2,i)
%         plot(t,u(i).error_pos_y);
%         xlabel('Time')
%         ylabel('Error')
%         graph_title = sprintf('Error of Y of Robot %d', i);
%         title(graph_title)
end
if docVisType == laporan
    figure(1)
    plot(...
        tt,u(1).error_pos_y,u(1).s,...
        tt,u(2).error_pos_y,u(2).s,...
        tt,u(3).error_pos_y,u(3).s,...
        tt,u(4).error_pos_y,u(4).s,...
        tt,u(5).error_pos_y,u(5).s ...
    );
    legend('q1','q2','q3','q4','q5');
    grid on
    xlabel('Waktu(detik)','FontName','Times New Roman','FontSize',15)
    ylabel('Kesalahan (cm)','FontName','Times New Roman','FontSize',15)
    graph_title = sprintf('Kesalahan posisi robot pada sumbu Y');
                                title(graph_title,'FontName','Times        New
Roman','FontSize',16,'FontWeight','bold')
    set(gca,'FontName','Times New Roman','FontSize',14)
    if isDocumented
        name = strcat(namanya,'ErrorY');
        savefig(name);
        saveas(gcf,name,'png');
    end
elseif docVisType == naskah
    figure(1)
    subplot(4,2,1)
        plot(...
            tt,u(1).error_pos_y,u(1).s,...
            tt,u(2).error_pos_y,u(2).s,...
            tt,u(3).error_pos_y,u(3).s,...
```

```
            tt,u(4).error_pos_y,u(4).s,...
            tt,u(5).error_pos_y,u(5).s ...
        );
        legend('q1','q2','q3','q4','q5');
        grid on
        xlabel('Waktu(detik)','FontName','Times New Roman','FontSize',15)
        ylabel('Kesalahan (cm)','FontName','Times New Roman','FontSize',15)
        graph_title = sprintf('Kesalahan posisi robot pada sumbu Y');
                                title(graph_title,'FontName','Times     New
Roman','FontSize',16,'FontWeight','bold')
        set(gca,'FontName','Times New Roman','FontSize',14)
end



for i=1:length(u)
    u(i).error_pos_x = u(i).rcrxx - u(i).rcxx;
%     figure(2)
%         subplot(3,2,i)
%         plot(t,u(i).error_pos_x);
%         xlabel('Time')
%         ylabel('Error')
%         graph_title = sprintf('Error of X of Robot %d', i);
%         title(graph_title)
end

if docVisType == laporan
    figure(2)
    plot(tt,u(1).error_pos_x,u(1).s,...
        tt,u(2).error_pos_x,u(2).s,...
        tt,u(3).error_pos_x,u(3).s,...
        tt,u(4).error_pos_x,u(4).s,...
        tt,u(5).error_pos_x,u(5).s);
    legend('q1','q2','q3','q4','q5')
    grid on
    xlabel('Waktu(detik)','FontName','Times New Roman','FontSize',15)
    ylabel('Kesalahan (cm)','FontName','Times New Roman','FontSize',15)
    graph_title = sprintf('Kesalahan posisi robot pada sumbu X');
                                title(graph_title,'FontName','Times        New
Roman','FontSize',16,'FontWeight','bold')
    set(gca,'FontName','Times New Roman','FontSize',14)
    if isDocumented
```

```
        name = strcat(namanya,'ErrorX');
        savefig(name);
        saveas(gcf,name,'png');
    end
elseif docVisType == naskah
    subplot(4,2,2)
        plot(tt,u(1).error_pos_x,u(1).s,...
        tt,u(2).error_pos_x,u(2).s,...
        tt,u(3).error_pos_x,u(3).s,...
        tt,u(4).error_pos_x,u(4).s,...
        tt,u(5).error_pos_x,u(5).s);
    legend('q1','q2','q3','q4','q5')
    grid on
    xlabel('Waktu(detik)','FontName','Times New Roman','FontSize',15)
    ylabel('Kesalahan (cm)','FontName','Times New Roman','FontSize',15)
    graph_title = sprintf('Kesalahan posisi robot pada sumbu X');
                                title(graph_title,'FontName','Times        New
Roman','FontSize',16,'FontWeight','bold')
    set(gca,'FontName','Times New Roman','FontSize',14)
end
for i=1:length(u)
    u(i).error_pos_theta = u(i).rcrthetaa - u(i).rcthetaa;
%     figure(3)
%         subplot(3,2,i)
%         plot(tt,u(i).error_pos_theta);
%         xlabel('Time')
%         ylabel('Error')
%         graph_title = sprintf('Error of theta of Robot %d', i);
%         title(graph_title)
end

if docVisType == laporan
    figure(3)
    plot(tt,u(1).error_pos_theta,u(1).s,...
        tt,u(2).error_pos_theta,u(2).s,...
        tt,u(3).error_pos_theta,u(3).s,...
        tt,u(4).error_pos_theta,u(4).s,...
        tt,u(5).error_pos_theta,u(5).s);
    legend('q1','q2','q3','q4','q5')
    grid on
    xlabel('Waktu(detik)','FontName','Times New Roman','FontSize',16)
    ylabel('Kesalahan (rad)','FontName','Times New Roman','FontSize',16)
```

```
    graph_title = sprintf('Kesalahan orientasi robot');
                                title(graph_title,'FontName','Times        New
Roman','FontSize',15,'FontWeight','bold')
    set(gca,'FontName','Times New Roman','FontSize',14)
    if isDocumented
        name = strcat(namanya,'ErrorTheta');
        savefig(name);
        saveas(gcf,name,'png');
    end
elseif docVisType == naskah
    subplot(4,2,3)
    plot(tt,u(1).error_pos_theta,u(1).s,...
        tt,u(2).error_pos_theta,u(2).s,...
        tt,u(3).error_pos_theta,u(3).s,...
        tt,u(4).error_pos_theta,u(4).s,...
        tt,u(5).error_pos_theta,u(5).s);
    legend('q1','q2','q3','q4','q5')
    grid on
    xlabel('Waktu(detik)','FontName','Times New Roman','FontSize',16)
    ylabel('Kesalahan (rad)','FontName','Times New Roman','FontSize',16)
    graph_title = sprintf('Kesalahan orientasi robot');
                                title(graph_title,'FontName','Times        New
Roman','FontSize',15,'FontWeight','bold')
    set(gca,'FontName','Times New Roman','FontSize',14)
end


% for i=1:length(u)
%     figure(4)
%         subplot(3,2,i)
%         plot(t,u(i).rcv);
%         xlabel('Time')
%         ylabel('V')
%         graph_title = sprintf('V of Robot %d', i);
%         title(graph_title)
% end

% for i=1:length(u)
%     figure(5)
%         subplot(3,2,i)
%         plot(t,u(i).rcw);
%         xlabel('Time')
```

```
%         ylabel('W')
%         graph_title = sprintf('w of Robot %d', i);
%         title(graph_title)
% end

% for i=1:length(u)
%     figure(6)
%         subplot(3,2,i)
%         plot(t,u(i).rcw);
%         xlabel('Time')
%         ylabel('W')
%         graph_title = sprintf('w of Robot %d', i);
%         title(graph_title)
% end

for i=1:length(u)
    %%% Jika dianggap bahwa robot-1 denga formasi pxy(0,0) sebagai leader
    %%% maka kecepatan formasi yang dibentuk oleh setiap robot
    %%% Transformasi terdahulu terhadap robot-1 selanjutnya,
%
%     u(i).err_of_formation_x = u(i).rcx - (u(1).rcx + u(i).px);
%     u(i).err_of_formation_y = u(i).rcy - (u(1).rcy + u(i).py);
    u(i).err_of_formation_x = u(i).rcxx - (u(1).rcxx + u(i).px);
    u(i).err_of_formation_y = u(i).rcyy - (u(1).rcyy + u(i).py);

%     figure(7)
%         subplot(3,2,i)
%         plot(t,u(i).err_of_formation_x);
%         xlabel('Time')
%         ylabel('Error X')
%         graph_title = sprintf('Error of x of Formation for Robot %d', i);
%         title(graph_title)
%
%         figure(8)
%         subplot(3,2,i)
%         plot(t,u(i).err_of_formation_y);
%         xlabel('Time')
%         ylabel('Error Y')
%         graph_title = sprintf('Error of y of Formation for Robot %d', i);
%         title(graph_title)
end
```

```
if docVisType == laporan
    figure(7)
    plot(tt,u(1).err_of_formation_x,u(1).s,...
        tt,u(2).err_of_formation_x,u(2).s,...
        tt,u(3).err_of_formation_x,u(3).s,...
        tt,u(4).err_of_formation_x,u(4).s,...
        tt,u(5).err_of_formation_x,u(5).s);
    legend('q1','q2','q3','q4','q5');
    grid on
    xlabel('Waktu(detik)','FontName','Times New Roman','FontSize',15)
    ylabel('Kesalahan (cm)','FontName','Times New Roman','FontSize',15)
        graph_title = sprintf('Kesalahan formasi posisi robot pada sumbu X');
                                title(graph_title,'FontName','Times     New
Roman','FontSize',16,'FontWeight','bold')
        set(gca,'FontName','Times New Roman','FontSize',14)
    if isDocumented
        name = strcat(namanya,'ErrorX_Formasi');
        savefig(name);
        saveas(gcf,name,'png');
    end
elseif docVisType == naskah
    subplot(4,2,4)
    plot(tt,u(1).err_of_formation_x,u(1).s,...
        tt,u(2).err_of_formation_x,u(2).s,...
        tt,u(3).err_of_formation_x,u(3).s,...
        tt,u(4).err_of_formation_x,u(4).s,...
        tt,u(5).err_of_formation_x,u(5).s);
    legend('q1','q2','q3','q4','q5');
    grid on
xlabel('Waktu(detik)','FontName','Times New Roman','FontSize',15)
ylabel('Kesalahan (cm)','FontName','Times New Roman','FontSize',15)
    graph_title = sprintf('Kesalahan formasi posisi robot pada sumbu X');
                            title(graph_title,'FontName','Times        New
Roman','FontSize',16,'FontWeight','bold')
    set(gca,'FontName','Times New Roman','FontSize',14)
end

if docVisType == laporan
    figure(8)
    plot(tt,u(1).err_of_formation_y,u(1).s,...
        tt,u(2).err_of_formation_y,u(2).s,...
```

```
        tt,u(3).err_of_formation_y,u(3).s,...
        tt,u(4).err_of_formation_y,u(4).s,...
        tt,u(5).err_of_formation_y,u(5).s);
    legend('q1','q2','q3','q4','q5');
    grid on
    xlabel('Waktu(detik)','FontName','Times New Roman','FontSize',15)
    ylabel('Kesalahan (cm)','FontName','Times New Roman','FontSize',15)
        graph_title = sprintf('Kesalahan formasi posisi robot pada sumbu Y');
                                title(graph_title,'FontName','Times     New
Roman','FontSize',16,'FontWeight','bold')
        set(gca,'FontName','Times New Roman','FontSize',14)
    if isDocumented
        name = strcat(namanya,'ErrorY_Formasi');
        savefig(name);
        saveas(gcf,name,'png');
    end
elseif docVisType == naskah
    subplot(4,2,5)
    plot(tt,u(1).err_of_formation_y,u(1).s,...
        tt,u(2).err_of_formation_y,u(2).s,...
        tt,u(3).err_of_formation_y,u(3).s,...
        tt,u(4).err_of_formation_y,u(4).s,...
        tt,u(5).err_of_formation_y,u(5).s);
    legend('q1','q2','q3','q4','q5');
    grid on
    xlabel('Waktu(detik)','FontName','Times New Roman','FontSize',15)
    ylabel('Kesalahan (cm)','FontName','Times New Roman','FontSize',15)
        graph_title = sprintf('Kesalahan formasi posisi robot pada sumbu Y');
                                title(graph_title,'FontName','Times     New
Roman','FontSize',16,'FontWeight','bold')
        set(gca,'FontName','Times New Roman','FontSize',14)
end

if docVisType == laporan
    figure(9);
    plot(tt,u(1).rcvv,u(1).s,...
        tt,u(2).rcvv,u(2).s,...
        tt,u(3).rcvv,u(3).s,...
        tt,u(4).rcvv,u(4).s,...
        tt,u(5).rcvv,u(5).s...
        );
    legend('q1','q2','q3','q4','q5');
```

```
    grid on
    xlabel('Waktu(detik)','FontName','Times New Roman','FontSize',15)
             ylabel('Kecepatan   Linear   (cm/detik)','FontName','Times   New
Roman','FontSize',15)
    graph_title = sprintf('Sinyal Kendali Kecepatan Linear Robot');
                              title(graph_title,'FontName','Times        New
Roman','FontSize',16,'FontWeight','bold')
    set(gca,'FontName','Times New Roman','FontSize',14)
    if isDocumented
        name = strcat(namanya,'Linear');
        savefig(name);
        saveas(gcf,name,'png');
    end
elseif docVisType == naskah
    subplot(4,2,6)
    plot(tt,u(1).rcvv,u(1).s,...
        tt,u(2).rcvv,u(2).s,...
        tt,u(3).rcvv,u(3).s,...
        tt,u(4).rcvv,u(4).s,...
        tt,u(5).rcvv,u(5).s...
        );
    legend('q1','q2','q3','q4','q5');
    grid on
    xlabel('Waktu(detik)','FontName','Times New Roman','FontSize',15)
             ylabel('Kecepatan   Linear   (cm/detik)','FontName','Times   New
Roman','FontSize',15)
    graph_title = sprintf('Sinyal Kendali Kecepatan Linear Robot');
                              title(graph_title,'FontName','Times        New
Roman','FontSize',16,'FontWeight','bold')
    set(gca,'FontName','Times New Roman','FontSize',14)
end

if docVisType == laporan
    figure(10);
    plot(tt,u(1).rcww,u(1).s,...
        tt,u(2).rcww,u(2).s,...
        tt,u(3).rcww,u(3).s,...
        tt,u(4).rcww,u(4).s,...
        tt,u(5).rcww,u(5).s...
        );
    legend('q1','q2','q3','q4','q5');
    grid on
```

```
    xlabel('Waktu(detik)','FontName','Times New Roman','FontSize',15)
            ylabel('Kesalahan   angular   (rad/detik)','FontName','Times   New
Roman','FontSize',15)
    graph_title = sprintf('Sinyal Kendali kecepatan angular Robot');
                                title(graph_title,'FontName','Times        New
Roman','FontSize',16,'FontWeight','bold')
        set(gca,'FontName','Times New Roman','FontSize',14)
    if isDocumented
        name = strcat(namanya,'Angular');
        savefig(name);
        saveas(gcf,name,'png');
    end
elseif docVisType == naskah
    subplot(4,2,7)
    plot(tt,u(1).rcww,u(1).s,...
        tt,u(2).rcww,u(2).s,...
        tt,u(3).rcww,u(3).s,...
        tt,u(4).rcww,u(4).s,...
        tt,u(5).rcww,u(5).s...
        );
    legend('q1','q2','q3','q4','q5');
    grid on
    xlabel('Waktu(detik)','FontName','Times New Roman','FontSize',15)
            ylabel('Kesalahan   angular   (rad/detik)','FontName','Times   New
Roman','FontSize',15)
    graph_title = sprintf('Sinyal Kendali kecepatan angular Robot');
                                title(graph_title,'FontName','Times        New
Roman','FontSize',16,'FontWeight','bold')
        set(gca,'FontName','Times New Roman','FontSize',14)
end
```

```
% figure(11);
%
plot(tt,u(1).rcy,'c',tt,u(2).rcy,'m',tt,u(3).rcy,'y',tt,u(4).rcy,'k',tt,u(5).
rcy,'r');
% title('Track X')
% figure(12);
```

```
%
plot(tt,u(1).rcy,'c',tt,u(2).rcy,'m',tt,u(3).rcy,'y',tt,u(4).rcy,'k',tt,u(5).
rcy,'r');
% title('Track Y')




% figure(12);
% plot(....
%     u(1).rcxx,u(1).rcyy,'-oc',...
%     u(1).rcxx(1),u(1).rcyy(1),'oc',...
%     u(1).rcxx(new_data_length),u(1).rcyy(new_data_length),'oc',...
%     u(2).rcxx,u(2).rcyy,'--sm',...
%     u(2).rcxx(1),u(2).rcyy(1),'sm',...
%     u(2).rcxx(new_data_length),u(2).rcyy(new_data_length),'sm',...
%     u(3).rcxx,u(3).rcyy,':dy',...
%     u(3).rcxx(1),u(3).rcyy(1),'dy',...
%     u(3).rcxx(new_data_length),u(3).rcyy(new_data_length),'dy',...
%     u(4).rcxx,u(4).rcyy,'-^k',...
%     u(4).rcxx(1),u(4).rcyy(1),'^k',...
%     u(4).rcxx(new_data_length),u(4).rcyy(new_data_length),'^k',...
%     u(5).rcxx,u(5).rcyy,'-.vr',...
%     u(5).rcxx(1),u(5).rcyy(1),'vr',...
%     u(5).rcxx(new_data_length),u(5).rcyy(new_data_length),'vr'...
%     );

% figure(13);
% plot(....
%     u(1).rcxx,u(1).rcyy,':oc',...
%     u(1).rcrxx,u(1).rcryy,'-c');plot(...
%     u(1).rcxx(1),u(1).rcyy(1),'Parent','oc',...
%     u(1).rcxx(new_data_length),u(1).rcyy(new_data_length),'oc',...
%     u(2).rcxx,u(2).rcyy,':sm',...
%     u(2).rcrxx,u(2).rcryy,'-m',...
%     u(2).rcxx(1),u(2).rcyy(1),'sm',...
%     u(2).rcxx(new_data_length),u(2).rcyy(new_data_length),'sm',...
%     u(3).rcxx,u(3).rcyy,':dy',...
%     u(3).rcrxx,u(3).rcryy,'-y',...
%     u(3).rcxx(1),u(3).rcyy(1),'dy',...
%     u(3).rcxx(new_data_length),u(3).rcyy(new_data_length),'dy',...
```

```
%     u(4).rcxx,u(4).rcyy,':^k',...
%     u(4).rcrxx,u(4).rcryy,'-k',...
%     u(4).rcxx(1),u(4).rcyy(1),'^k',...
%     u(4).rcxx(new_data_length),u(4).rcyy(new_data_length),'^k',...
%     u(5).rcxx,u(5).rcyy,':vr',...
%     u(5).rcrxx,u(5).rcryy,'-r',...
%     u(5).rcxx(1),u(5).rcyy(1),'vr',...
%     u(5).rcxx(new_data_length),u(5).rcyy(new_data_length),'vr'...
%     );
% legend('q1','','','q2','q3','q4','q5');
% grid on
% xlabel('Sumbu X (cm)')
% ylabel('Sumbu Y (cm)')
% title('Lintasan robot')

% Create figure
figure1 = figure(13);

% Create axes
axes1 = axes('Parent',figure1,...
    'Position',[0.13 0.111915708812261 0.646765375854214 0.814999999999998]);
box(axes1,'on');
grid(axes1,'on');
hold(axes1,'all');

%% q1
% Create plot
plot(u(1).rcxx,u(1).rcyy,'Parent',axes1,'Marker','o','LineStyle',':','Color',
[0 1 1],...
    'DisplayName','q1');
plot(u(1).rcrxx,u(1).rcryy,'Parent',axes1,'Color',[0                    1
1],'DisplayName','Lintasan referensi q1');
plot(u(1).rcxx(1),u(1).rcyy(1),'Parent',axes1,'MarkerFaceColor',[0          1
1],'Marker','o','Color',[0 1 1],...
    'DisplayName','Posisi awal q1');
plot(u(1).rcxx(new_data_length),u(1).rcyy(new_data_length),'Parent',axes1,'Ma
rkerFaceColor',[0 1 1],'Marker','o','Color',[0 1 1],'MarkerSize',15,...
    'DisplayName','Posisi akhir q1');

%% q2
% Create plot
```

```
plot(u(2).rcxx,u(2).rcyy,'Parent',axes1,'Marker','square','LineStyle',':','Co
lor',[1 0 1],...
    'DisplayName','q2');
plot(u(2).rcrxx,u(2).rcryy,'Parent',axes1,'Color',[1                          0
1],'DisplayName','Lintasan referensi q2');
% Create plot
plot(u(2).rcxx(1),u(2).rcyy(1),'Parent',axes1,'MarkerFaceColor',[1            0
1],'MarkerSize',10,...
    'Marker','square',...
    'LineStyle','none',...
    'Color',[1 0 1],...
    'DisplayName','Posisi awal q2');
plot(u(2).rcxx(new_data_length),u(2).rcyy(new_data_length),'Parent',axes1,'Ma
rkerFaceColor',[1 0 1],'Marker','square','Color',[1 0 1],'MarkerSize',15,...
    'DisplayName','Posisi akhir q2');

%% q3
% Create plot
plot(u(3).rcxx,u(3).rcyy,'Parent',axes1,'Marker','diamond','LineStyle',':',..
.
    'Color',[1 1 0],...
    'DisplayName','q3');
% Create plot
plot(u(3).rcrxx,u(3).rcryy,'Parent',axes1,'Color',[1 1 0],...
    'DisplayName','Lintasan referensi q3');
% Create plot
plot(u(3).rcxx(1),u(3).rcyy(1),'Parent',axes1,'MarkerFaceColor',[1            1
0],'MarkerSize',10,...
    'Marker','diamond',...
    'LineStyle','none',...
    'Color',[1 1 0],...
    'DisplayName','Posisi awal q3');
% Create plot
plot(u(3).rcxx(new_data_length),u(3).rcyy(new_data_length),'Parent',axes1,'Ma
rkerFaceColor',[1 1 0],'MarkerSize',15,...
    'Marker','diamond',...
    'LineStyle','none',...
    'Color',[1 1 0],...
    'DisplayName','Posisi akhir q3');

%% q4
% Create plot
```

```
plot(u(4).rcxx,u(4).rcyy,'Parent',axes1,'Marker','^','LineStyle',':','Display
Name','q4',...
    'Color',[0 0 0]);
plot(u(4).rcrxx,u(4).rcryy,'Parent',axes1,'DisplayName','Lintasan    referensi
q4','Color',[0 0 0]);
% Create plot
plot(u(4).rcxx(1),u(4).rcyy(1),'Parent',axes1,'MarkerFaceColor',[0           0
0],'MarkerSize',10,...
    'Marker','^',...
    'LineStyle','none',...
    'DisplayName','Posisi awal q4',...
    'Color',[0 0 0]);
plot(u(4).rcxx(new_data_length),u(4).rcyy(new_data_length),'Parent',axes1,'Ma
rkerFaceColor',[0 0 0],'Marker','^','MarkerSize',15,...
    'DisplayName','Posisi akhir q4',...
    'Color',[0 0 0]);

%% q5
% Create plot
plot(u(5).rcxx,u(5).rcyy,'Parent',axes1,'Marker','v','LineStyle',':','Color',
[1 0 0],...
    'DisplayName','q5');
plot(u(5).rcrxx,u(5).rcryy,'Parent',axes1,'Color',[1                         0
0],'DisplayName','Lintasan referensi q5');
plot(u(5).rcxx(1),u(5).rcyy(1),'Parent',axes1,'MarkerFaceColor',[1           0
0],'Marker','v','Color',[1 0 0],...
    'DisplayName','Posisi awal q5');
plot(u(5).rcxx(new_data_length),u(5).rcyy(new_data_length),'Parent',axes1,'Ma
rkerFaceColor',[1 0 0],'Marker','v','Color',[1 0 0],'MarkerSize',15,...
    'DisplayName','Posisi akhir q5');

% Create xlabel
xlabel('Sumbu X (cm)','FontName','Times New Roman','FontSize',13);

% Create ylabel
ylabel('Sumbu Y (cm)','FontName','Times New Roman','FontSize',13);

% Create title
graph_title = sprintf('Lintasan robot');
title(graph_title,'FontName','Times                                        New
Roman','FontSize',14,'FontWeight','bold')
    set(gca,'FontName','Times New Roman','FontSize',12)
```

```
% Create legend
legend1 = legend(axes1,'show');
set(legend1,...
          'Position',[0.785349701979002   0.159247041313318   0.205780346820809
0.765182186234818]);
if isDocumented
    name = strcat(namanya,'Lintasan');
    savefig(name);
    saveas(gcf,name,'png');
end
```