# Percakapan tentang Kinematika Robot dan Kontrol Pengikut

User: Tolong modifikasi kode yang saya berikan untuk membuat robot kedua mengikuti robot pertama.

Assistant: Untuk membuat robot kedua mengikuti robot pertama, kita akan memodifikasi kode untuk menambahkan kontrol pengikut (follower) pada robot kedua. Robot kedua akan mencoba 'mengejar' posisi robot pertama di setiap waktu, mengikuti lintasan yang sama.  
  
Berikut adalah modifikasi kode Anda untuk membuat robot kedua mengikuti robot pertama dengan menggunakan kontrol berbasis posisi dan orientasi. Pada setiap langkah waktu, robot kedua akan 'mengejar' posisi sebelumnya dari robot pertama, sehingga bergerak pada jalur yang sama.  
  
### Kode yang Dimodifikasi  
```matlab  
clc; close all; clear;  
  
% Parameter Robot  
r = 0.075; % Jari-jari roda (dalam meter)  
L = 0.3; % Jarak antar roda (dalam meter)  
tf = 20; % Waktu akhir simulasi (dalam detik)  
  
% Kondisi Awal  
theta0\_1 = 0; % Orientasi awal robot 1  
x0\_1 = 0; % Posisi awal robot 1 pada sumbu x  
y0\_1 = 0; % Posisi awal robot 1 pada sumbu y  
  
theta0\_2 = 0; % Orientasi awal robot 2  
x0\_2 = -1; % Posisi awal robot 2 pada sumbu x  
y0\_2 = 0; % Posisi awal robot 2 pada sumbu y  
  
% Vektor Waktu dan Time Increment  
t = linspace(0, tf, 200); % Vektor waktu dari 0 hingga tf dengan 200 titik  
dt = t(2) - t(1); % Increment waktu  
  
% Inisialisasi Variabel Pose untuk Robot 1 dan Robot 2  
x\_1 = zeros(1, length(t));  
y\_1 = zeros(1, length(t));  
theta\_1 = zeros(1, length(t));  
x\_2 = zeros(1, length(t));  
y\_2 = zeros(1, length(t));  
theta\_2 = zeros(1, length(t));  
  
x\_1(1) = x0\_1;  
y\_1(1) = y0\_1;  
theta\_1(1) = theta0\_1;  
  
x\_2(1) = x0\_2;  
y\_2(1) = y0\_2;  
theta\_2(1) = theta0\_2;  
  
% Pengaturan Kecepatan untuk Robot 1 (kecepatan tetap)  
vA\_1 = 0.5 \* ones(1, length(t)); % Kecepatan roda kiri konstan  
vB\_1 = 0.5 \* ones(1, length(t)); % Kecepatan roda kanan konstan  
  
% Loop Update Pose untuk Robot 1  
for ii = 2:length(t)  
 x\_1(ii) = x\_1(ii-1) + cos(theta\_1(ii-1)) \* (vA\_1(ii-1) + vB\_1(ii-1))/2 \* dt;  
 y\_1(ii) = y\_1(ii-1) + sin(theta\_1(ii-1)) \* (vA\_1(ii-1) + vB\_1(ii-1))/2 \* dt;  
 theta\_1(ii) = theta\_1(ii-1) + ((vA\_1(ii-1) - vB\_1(ii-1)) / L) \* dt;  
end  
  
% Kontrol untuk Robot 2 agar mengikuti jalur Robot 1  
kp = 1.0; % Gain untuk kontrol posisi  
ktheta = 1.0; % Gain untuk kontrol orientasi  
  
% Loop Update Pose dengan Kinematik untuk Robot 2 agar mengikuti Robot 1  
for ii = 2:length(t)  
 ex = x\_1(ii-1) - x\_2(ii-1);  
 ey = y\_1(ii-1) - y\_2(ii-1);  
 e\_distance = sqrt(ex^2 + ey^2);  
 e\_theta = atan2(ey, ex) - theta\_2(ii-1);  
 v = kp \* e\_distance;  
 omega = ktheta \* e\_theta;  
 vA\_2 = v - (omega \* L) / 2;  
 vB\_2 = v + (omega \* L) / 2;  
 x\_2(ii) = x\_2(ii-1) + cos(theta\_2(ii-1)) \* (vA\_2 + vB\_2) / 2 \* dt;  
 y\_2(ii) = y\_2(ii-1) + sin(theta\_2(ii-1)) \* (vA\_2 + vB\_2) / 2 \* dt;  
 theta\_2(ii) = theta\_2(ii-1) + ((vB\_2 - vA\_2) / L) \* dt;  
end  
  
% Plot dan Animasi  
figure(1);  
plot(x\_1, y\_1, 'k-', 'DisplayName', 'Robot 1 (Leader)');  
hold on;  
plot(x\_2, y\_2, 'r--', 'DisplayName', 'Robot 2 (Follower)');  
title('Workspace Robot');  
xlabel('x (m)');  
ylabel('y (m)');  
legend;  
  
axis equal;  
axis([-1 3 -1 3]);  
  
% Animasi Pergerakan  
for ii = 2:length(t)  
 wlsrot\_1 = [cos(theta\_1(ii)) -sin(theta\_1(ii)); sin(theta\_1(ii)) cos(theta\_1(ii))] \* wls;  
 wlsrot\_2 = [cos(theta\_2(ii)) -sin(theta\_2(ii)); sin(theta\_2(ii)) cos(theta\_2(ii))] \* wls;  
 set(h1\_1, 'XData', wlsrot\_1(1,1) + x\_1(ii));  
 set(h1\_1, 'YData', wlsrot\_1(2,1) + y\_1(ii));  
 set(h1\_2, 'XData', wlsrot\_2(1,1) + x\_2(ii));  
 set(h3\_2, 'YData', y\_2(ii));  
 drawnow;  
 pause(0.05);  
end  
```