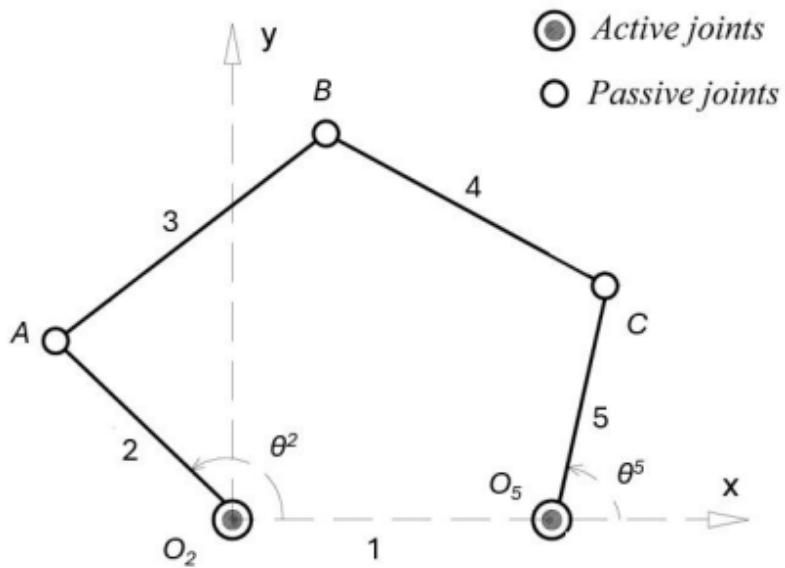


COMPUTATIONAL KINEMATICS ASSIGNMENT

MS2105 – Kinematics and Dynamics

Nama : Dhiwa Hafizh Wirianputra
NIM : 13124123

1. Pendahuluan

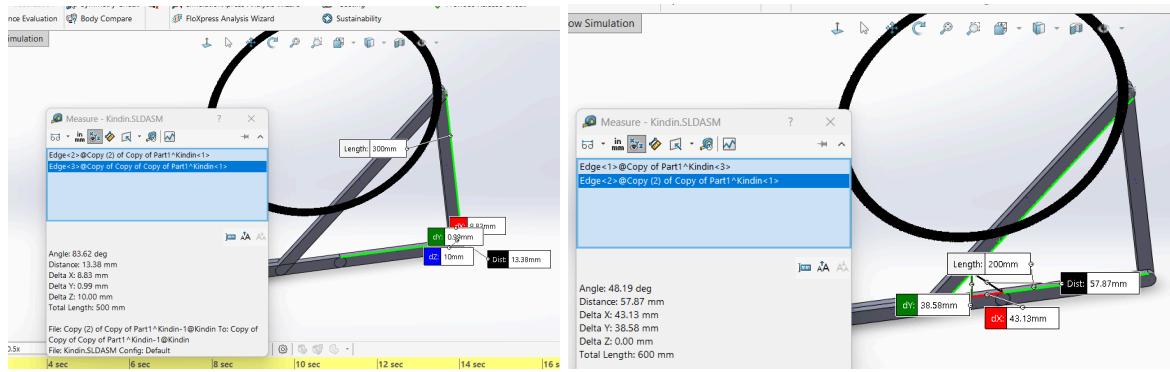


Tugas ini bertujuan untuk melakukan analisis kinematika komputasional pada sebuah manipulator planar dua derajat kebebasan (2-DoF) sesuai dengan spesifikasi yang diberikan pada dokumen penugasan. Analisis dilakukan untuk menentukan posisi, kecepatan, dan percepatan titik sambung (joint) B dalam sistem koordinat global, serta kecepatan sudut dan percepatan sudut pada link 3 dan link 4 sebagai fungsi waktu.

Pendekatan yang digunakan memodelkan setiap batang menggunakan koordinat translasi dan rotasi. Hubungan kinematika antarbatang dinyatakan melalui persamaan yang disusun dalam bentuk vektor constraint dan matriks Jacobian. Penyelesaian sistem persamaan nonlinier dilakukan secara numerik menggunakan metode Newton-Raphson pada setiap langkah waktu simulasi.

Parameter masukan berupa dimensi geometris mekanisme serta kecepatan sudut link 2 dan link 5 ditentukan berdasarkan ketentuan soal, yaitu dari tiga digit terakhir Nomor Induk Mahasiswa (NIM). Simulasi dilakukan pada rentang waktu 0 hingga 5 detik, dan hasil analisis disajikan dalam bentuk grafik sesuai dengan keluaran yang diminta.

- **Input sudut awal**



2. Kode

```
%% 1. Identitas dan Intro

%Input Identitas
Nama = 'Dhiwa H. Wirianputra';
NIM = '13124055';

% Ambil nama depan
tokens = strsplit(strtrim(Nama));
namaDepan = tokens{1};

% Ambil 3 digit terakhir
s = char(NIM);
if strlength(s) < 3
    error('NIM harus minimal 3 karakter.');
end
XYZ = s(end-2:end);
X = XYZ(1); Y = XYZ(2); Z = XYZ(3);

% Covert jadi double
x = double(XYZ(1)) - '0';
y = double(XYZ(2)) - '0';
z = double(XYZ(3)) - '0';

% Bentuk bilangan dua digit dari pasangan digit
XY = 10*x + y;
YZ = 10*y + z;
ZY = 10*z + y;

%% 2. Parameter Geometri dan Setup Awal

%Input Panjang batang
L2 = 200; %mm
L3 = 400; %mm
L4 = 300; %mm
L5 = 200; %mm
O2O5 = 300; %mm

%Input sudut awal batang
Initeta2 = 0;
Initeta5 = 0;

% Hitung Omega
Omega2 = XY + YZ;
Omega5 = XY + ZY;
t = 0:0.01:5; %Waktu simulasi 5 Detik
NumData = size(t,2);
```

```

% Buat menampilkan window
txt = sprintf(['Hello %s\n' ...
    '      NIM:%d%d%d\n' ...
    '      Omega2=%d\n' ...
    '      Omega5=%d\n\n' ...
    '      Klik OK untuk melanjutkan'] ...
, namaDepan, x, y, z, Omega2, Omega5);
h = msgbox(txt, 'Intro');
drawnow;
uiwait(h, 10); % tunggu sampai ditutup atau 10 detik berlalu
if ishandle(h); close(h); end

% Definisikan q
%q = {Rx1, Ry1, Theta1, Rx2, Ry2, Theta2,
% Rx3, Ry3, Theta3, Rx4, Ry4, Theta4, Rx5, Ry5, Theta5,}
q = zeros(15,1);
q_all = zeros(15,NumData);
qdot_all = zeros(15,NumData);
qdot2_all = zeros(15,NumData);

% Perkiraan awal sudut batang (gambar 1.1)
q(6) = deg2rad(0);
q(9) = deg2rad(48.19);
q(12) = deg2rad(-83.62);
q(15) = deg2rad(0);

%% 3. Loop Analisis Kinematika
for j = 1:NumData
    for i = 1:30 %Jumlah iterasi
        C = [q(1); q(2); q(3);...
            %Joint O2
            q(4)-(L2*cos(q(6)));...
            q(5)-(L2*sin(q(6)));...
            %Joint A
            q(4)-q(7)+(L3*cos(q(9)));...
            q(5)-q(8)+(L3*sin(q(9)));...
            %Joint B
            q(7)-q(10)+(L4*cos(q(12)));...
            q(8)-q(11)+(L4*sin(q(12)));...
            %Joint C
            q(10)-q(13)-(L5*cos(q(15)));...
            q(11)-q(14)-(L5*sin(q(15)));...
            %Joint O5
            q(13)-O205;...
            q(14);...
            q(6)-Initeta2-Omega2*t(j);...
            q(15)-Initeta5-Omega5*t(j)];
    %Jacobian Matrix

```

```

Cq = [1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0;
      0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0;
      0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0;
      0 0 0 1 0 (L2*sin(q(6))) 0 0 0 0 0 0 0 0 0 0; %Joint O2
      0 0 0 0 1 -(L2*cos(q(6))) 0 0 0 0 0 0 0 0 0 0;
      0 0 0 1 0 0 -1 0 -(L3*sin(q(9))) 0 0 0 0 0 0 0; %Joint A
      0 0 0 0 1 0 0 -1 (L3*cos(q(9))) 0 0 0 0 0 0 0;
      0 0 0 0 0 0 1 0 0 -1 0 -(L4*sin(q(12))) 0 0 0 0; %Joint B
      0 0 0 0 0 0 0 1 0 0 -1 (L4*cos(q(12))) 0 0 0 0;
      0 0 0 0 0 0 0 0 0 1 0 0 -1 0 (L5*sin(q(15))); %Joint C
      0 0 0 0 0 0 0 0 0 0 1 0 0 -1 -(L5*cos(q(15)));
      0 0 0 0 0 0 0 0 0 0 0 0 1 0 0; %Joint O5
      0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0;
      0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0;
      0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1];
r = rank(Cq);
q_delta = -Cq\C; %Newton-Raphson Update
q = q + q_delta;
%velocity Analysis
Ct = zeros(15,1);
Ct(14) = -Omega2;
Ct(15) = -Omega5;
qdot = -Cq\ Ct;
Omega3 = qdot(9);
Omega4 = qdot(12);
%Acceleration Analysis
Cq_qdot_q = [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0;
              0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0;
              0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0;
              0 0 0 0 0 (Omega2*L2*cos(q(6))) 0 0 0 0 0 0 0 0 0 0 0 0; %Joint O2
              0 0 0 0 0 (Omega2*L2*sin(q(6))) 0 0 0 0 0 0 0 0 0 0 0 0;
              0 0 0 0 0 0 0 0 -(Omega3*L3*cos(q(9))) 0 0 0 0 0 0 0 0 0; %Joint A
              0 0 0 0 0 0 0 0 -(Omega3*L3*sin(q(9))) 0 0 0 0 0 0 0 0 0;
              0 0 0 0 0 0 0 0 0 0 0 -(Omega4*L4*cos(q(12))) 0 0 0 0; %Joint B
              0 0 0 0 0 0 0 0 0 0 0 -(Omega4*L4*sin(q(12))) 0 0 0 0;
              0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0; %Joint C
              0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0;
              0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0; %Joint O5
              0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0;
              0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0;
              0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0];
Qd = -Cq_qdot_q*qdot;
qdot2 = Cq\Qd;
end
%Final Parameter
q_all(:,j) = q; %Posisi
qdot_all(:,j) = qdot; %Kecepatan
qdot2_all(:,j) = qdot2; %Percepatan
end

```

```

%% 4. Plotting Hasil Simulasi

% Cek simulasi apakah benar atau tidak
if r < size(Cq,1)
    error('Jacobian singular: rank=%d, expected=%d, SALAH\n', r, size(Cq,1));
else
    fprintf('rank=%d, expected=%d, BENAR\n', r, size(Cq,1));
end

% 6 plot dalam satu figure 3 baris x 2 kolom
tiledlayout(3,2,"TileSpacing","compact","Padding","compact");

% Plot 1: Position B
ax1 = nexttile;
plot(ax1, t, q_all(7,:), t, q_all(8,:));
xlabel(ax1,'t'); ylabel(ax1,'Position B');
legend(ax1,'x_B','y_B');

% Plot 2: Velocity B
ax2 = nexttile;
plot(ax2, t, qdot_all(7,:), t, qdot_all(8,:));
xlabel(ax2,'t'); ylabel(ax2,'Velocity B');
legend(ax2,'vx_B','vy_B');

% Plot 3: Acceleration B
ax3 = nexttile;
plot(ax3, t, qdot2_all(7,:), t, qdot2_all(8,:));
xlabel(ax3,'t'); ylabel(ax3,'Acceleration B');
legend(ax3,'ax_B','ay_B');

% Plot 4: Angular velocity
ax4 = nexttile;
plot(ax4, t, qdot_all(9,:), t, qdot_all(12,:));
xlabel(ax4,'t'); ylabel(ax4,'Angular velocity');
legend(ax4,'Omega_3','Omega_4');

% Plot 5: Angular acceleration
ax5 = nexttile;
plot(ax5, t, qdot2_all(9,:), t, qdot2_all(12,:));
xlabel(ax5,'t'); ylabel(ax5,'Angular acceleration');
legend(ax5,'Alpha_3','Alpha_4');

% Plot 6: Trajectory B (untuk tes)
ax6 = nexttile;
plot(ax6, q_all(7,:), q_all(8,:));
xlabel(ax6,'x_B'); ylabel(ax6,'y_B');

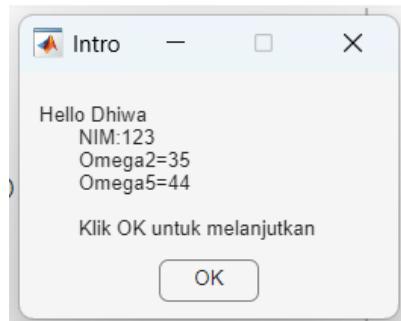
```

```
% Judul
sgtitle('Ringkasan Hasil Simulasi', 'FontSize', 14, 'FontWeight', 'bold');

% Easter egg
for k = 1:3
    disp('Yellboys');
end
```

3. Feature Highlight

Program MATLAB yang saya kembangkan dilengkapi dengan beberapa fitur pendukung untuk meningkatkan kejelasan dan keandalan proses simulasi. Pada saat program dijalankan, sistem akan menampilkan sebuah pop-up window yang memuat informasi identitas pengguna, seperti nama dan Nomor Induk Mahasiswa (NIM), serta nilai kecepatan sudut input yang digunakan. Pop-up ini juga berfungsi sebagai bentuk konfirmasi awal sebelum simulasi kinematika dijalankan.



Setelah proses simulasi dilakukan, program dilengkapi dengan mekanisme pengecekan kebenaran model kinematika melalui evaluasi rank dari matriks Jacobian. Pengecekan ini digunakan untuk memastikan bahwa sistem constraint yang dibentuk bersifat independen dan tidak berada pada kondisi singular. Dengan adanya fitur ini, pengguna dapat memastikan bahwa hasil simulasi yang diperoleh berasal dari formulasi kinematika yang valid.

```

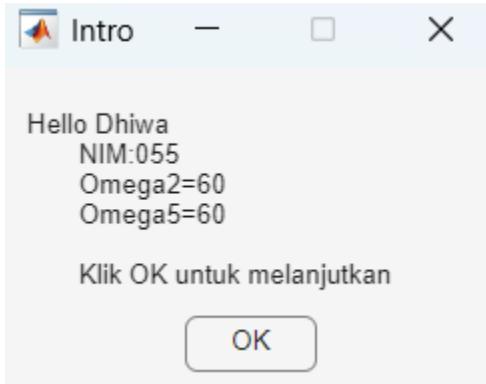
>> KindinFinalProject
rank=15, expected=15, BENAR
Yellboys
Yellboys
Yellboys

% Cek simulasi apakah benar atau tidak
if r < size(Cq,1)
    error('Jacobian singular: rank=%d, expected=%d, SALAH\n', r, size(Cq,1));
else
    fprintf('rank=%d, expected=%d, BENAR\n', r, size(Cq,1));
end

```

4. Data

- Input data



- Output data

