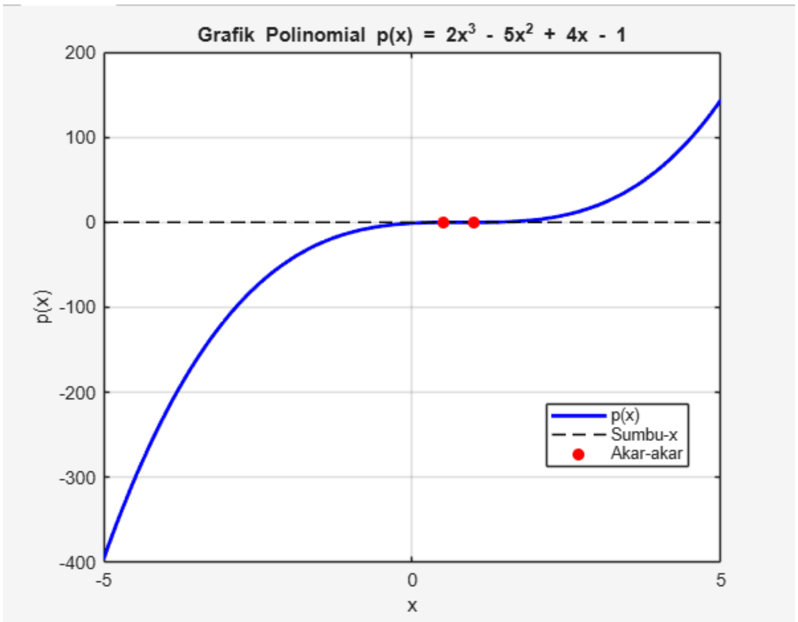
Nama : Yoga Prasetiyo

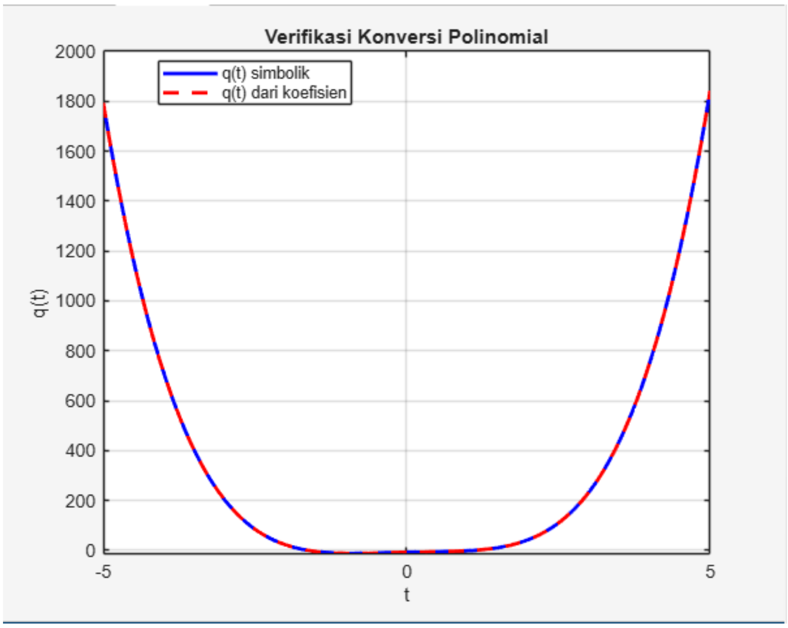
NIM : 2411010002

MAKUL : Praktik system kendali 1 (UTS)

```
>> % soal no. 1
>> p = [2 -5 4 -1];
>>
>> % Cari akar-akar polinomial
>> r = roots(p)

r =

   1.0000 + 0.0000i
   1.0000 - 0.0000i
   0.5000 + 0.0000i

>> % Buat rentang x untuk plot
>> x = linspace(-5,5,500);
>> y = polyval(p,x);
>> % Plot grafik p(x)
>> figure;
>> plot(x,y,'b','LineWidth',2);
>> hold on;
>> plot(x,zeros(size(x)),'k--');
>> plot(r,zeros(size(r)),'ro','MarkerFaceColor','r');
Warning: Imaginary parts of complex X and/or Y arguments ignored.
>> title('Grafik Polinomial p(x) = 2x^3 - 5x^2 + 4x - 1');
>> xlabel('x');
>> ylabel('p(x)');
>> grid on;
>> legend('p(x)','Sumbu-x','Akar-akar','Location','best');
>> % Interpretasi hasil
>> disp('Interpretasi:');
Interpretasi:
>> disp('Akar-akar adalah titik potong grafik p(x) dengan sumbu-x.');
Akar-akar adalah titik potong grafik p(x) dengan sumbu-x.
>>
```

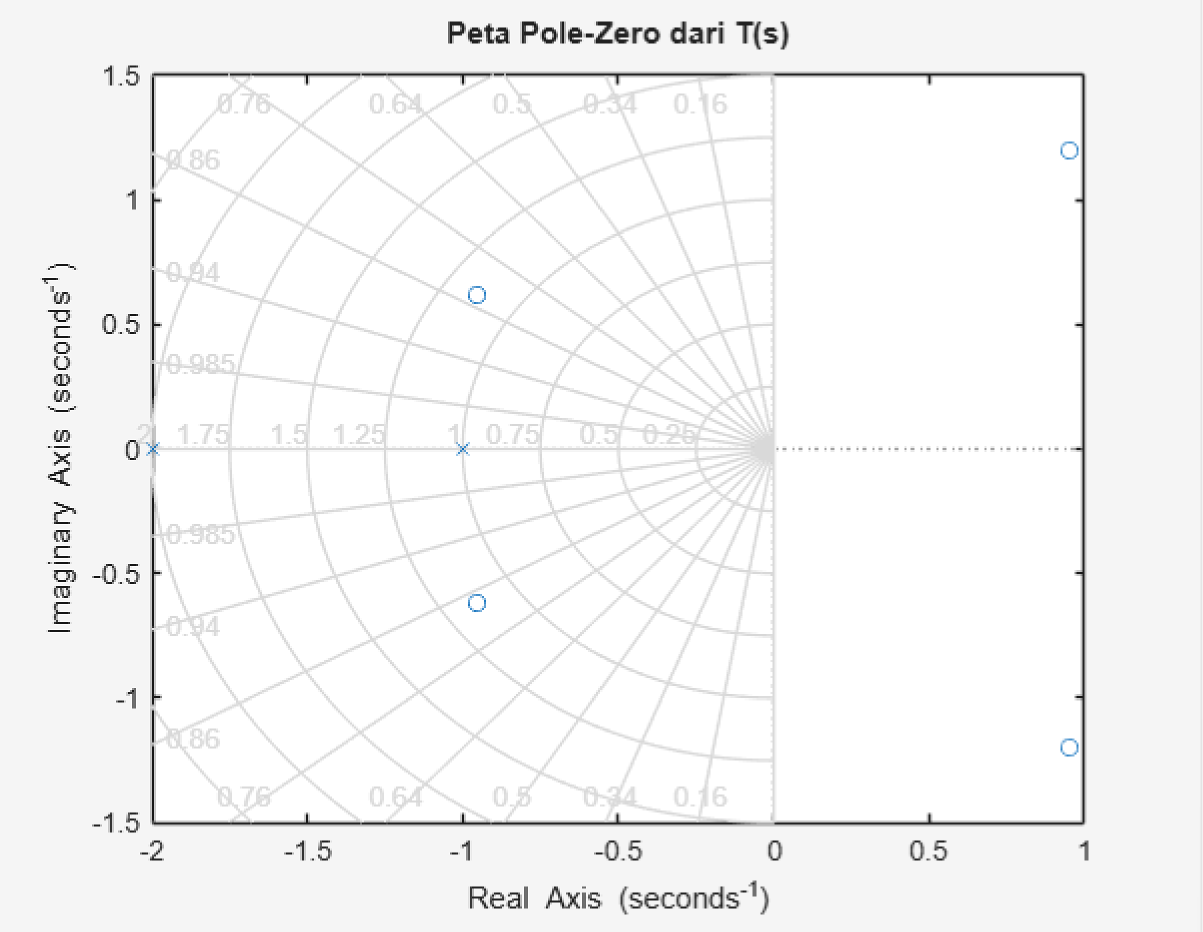



```
>> % Soal no. 2
>> syms t
>> q = 3*t^4 - 2*t^2 + 5*t - 7;
>> % Ubah ke vektor koefisien
>> coeff_q = sym2poly(q);
>> disp('Vektor koefisien q(t):');
Vektor koefisien q(t):
>> disp(coeff_q);
     3     0    -2     5    -7

>> % Kembalikan ke bentuk simbolik
>> q_sym = poly2sym(coeff_q, t);
>> disp('Bentuk simbolik kembali dari vektor:');
Bentuk simbolik kembali dari vektor:
>> disp(q_sym);
3*t^4 - 2*t^2 + 5*t - 7

>> % Verifikasi dengan plot
>> x = linspace(-5,5,500);
>> y1 = double(subs(q,t,x));
>> y2 = polyval(coeff_q,x);
>> figure;
>> plot(x,y1,'b','LineWidth',2);
>> hold on;
>> plot(x,y2,'r--','LineWidth',2);
>> legend('q(t) simbolik','q(t) dari koefisien','Location','best');
>> title('Verifikasi Konversi Polinomial');
>> xlabel('t');
>> ylabel('q(t)');
>> grid on;
>>
```

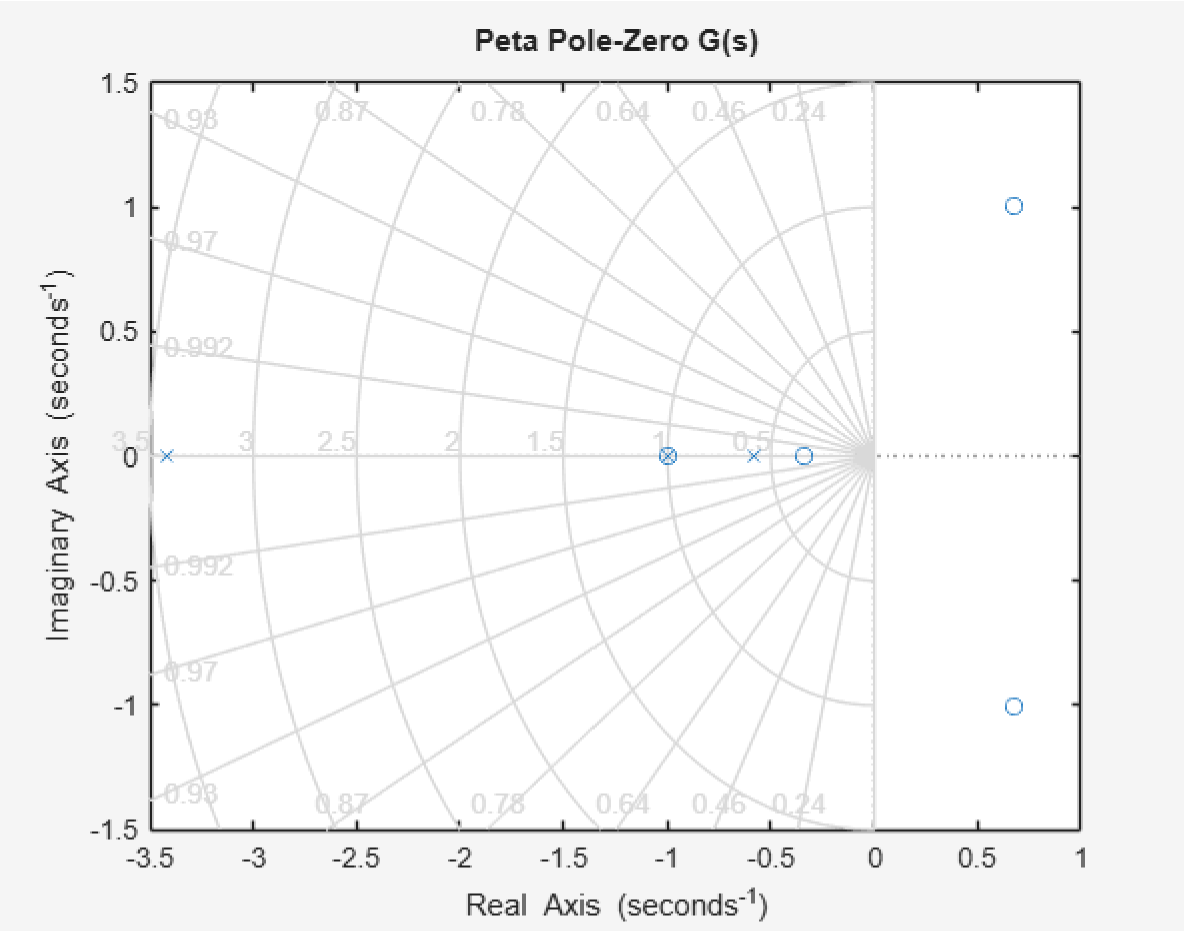



```
>> % soal no.3
>> num = [1 0 0 2 3];
>> den = [1 4 5 2];
>>
>> [r,p,k] = residue(num,den);
>> disp('Residu (r):');
Residu (r):
>> disp(r);
15.0000
-4.0000
2.0000

>> disp('Pole (p):');
Pole (p):
>> disp(p);
   -2.0000
   -1.0000
   -1.0000

>> disp('Hasil bagi (k):');
Hasil bagi (k):
>> disp(k);
     1    -4

>> [num2,den2] = residue(r,p,k)

num2 =

    1.0000   -0.0000   -0.0000    2.0000    3.0000
```

```
den2 =

    1.0000    4.0000    5.0000    2.0000

>> disp('Rekonstruksi kembali num dan den:');
Rekonstruksi kembali num dan den:
>> disp(num2);
    1.0000   -0.0000   -0.0000    2.0000    3.0000

>> disp(den2);
    1.0000    4.0000    5.0000    2.0000

>> figure;
>> pzmap(tf(num,den));

>> title('Peta Pole-Zero dari T(s)');
>> grid on;
>>
```

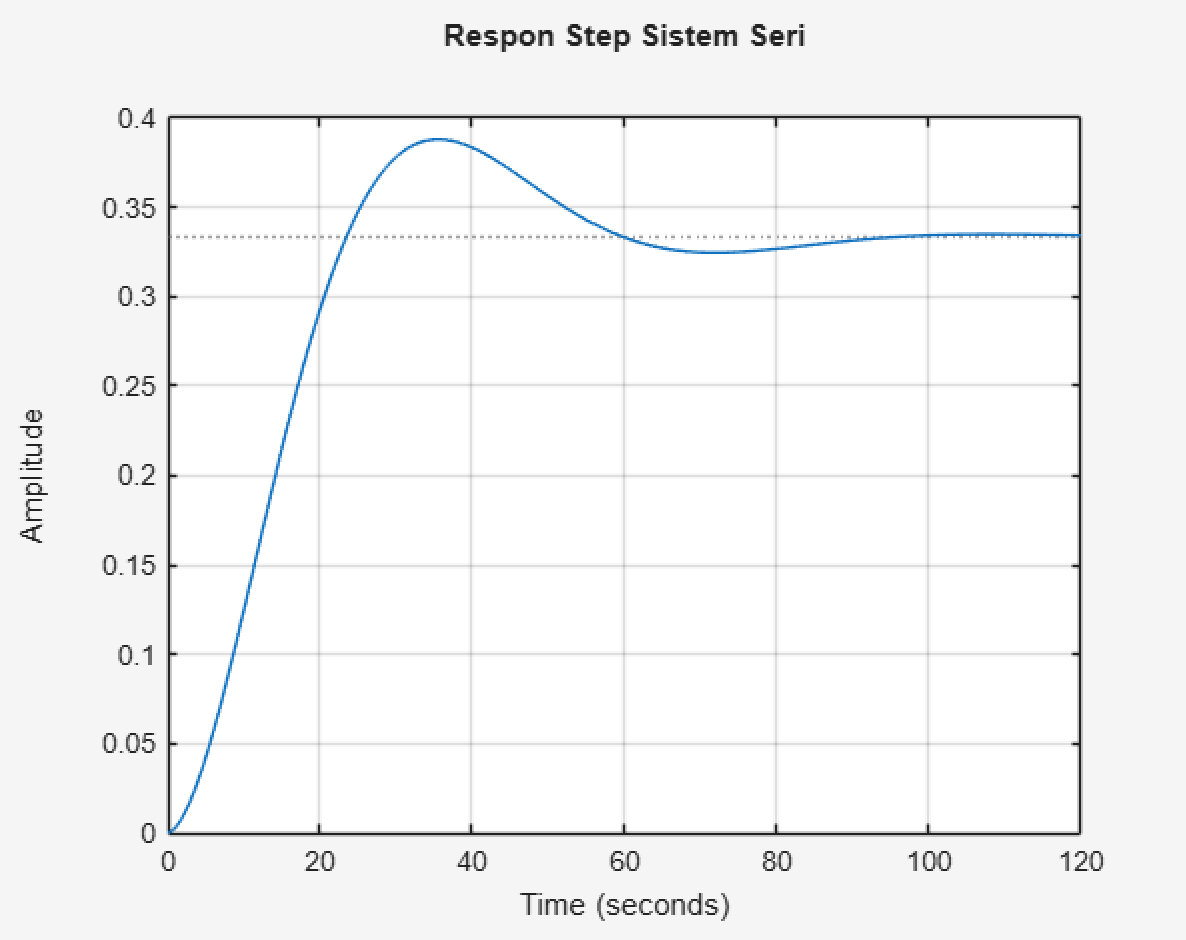


```
>> % soal no. 4
>> num = [2 0 0 3 1];
>> den = [1 5 6 2];
>> G = tf(num, den)

G =

     2 s^4 + 3 s + 1
  ---------------------
  s^3 + 5 s^2 + 6 s + 2

Continuous-time transfer function.
Model Properties
>> G_zpk = zpk(G)

G_zpk =

  2 (s+1) (s+0.3425) (s^2 - 1.343s + 1.46)
  ----------------------------------------
         (s+3.414) (s+1) (s+0.5858)

Continuous-time zero/pole/gain model.
Model Properties
>> % Visualisasi pole-zero
>> figure;
>> pzmap(G);
>> title('Peta Pole-Zero G(s)');
>> grid on;
>> p = pole(G);
>> if all(real(p) < 0)
disp('Sistem STABIL (semua pole di sebelah kiri s-plane)');
else
    disp('Sistem TIDAK STABIL (ada pole di sebelah kanan s-plane)');
end
Sistem STABIL (semua pole di sebelah kiri s-plane)
>>
```



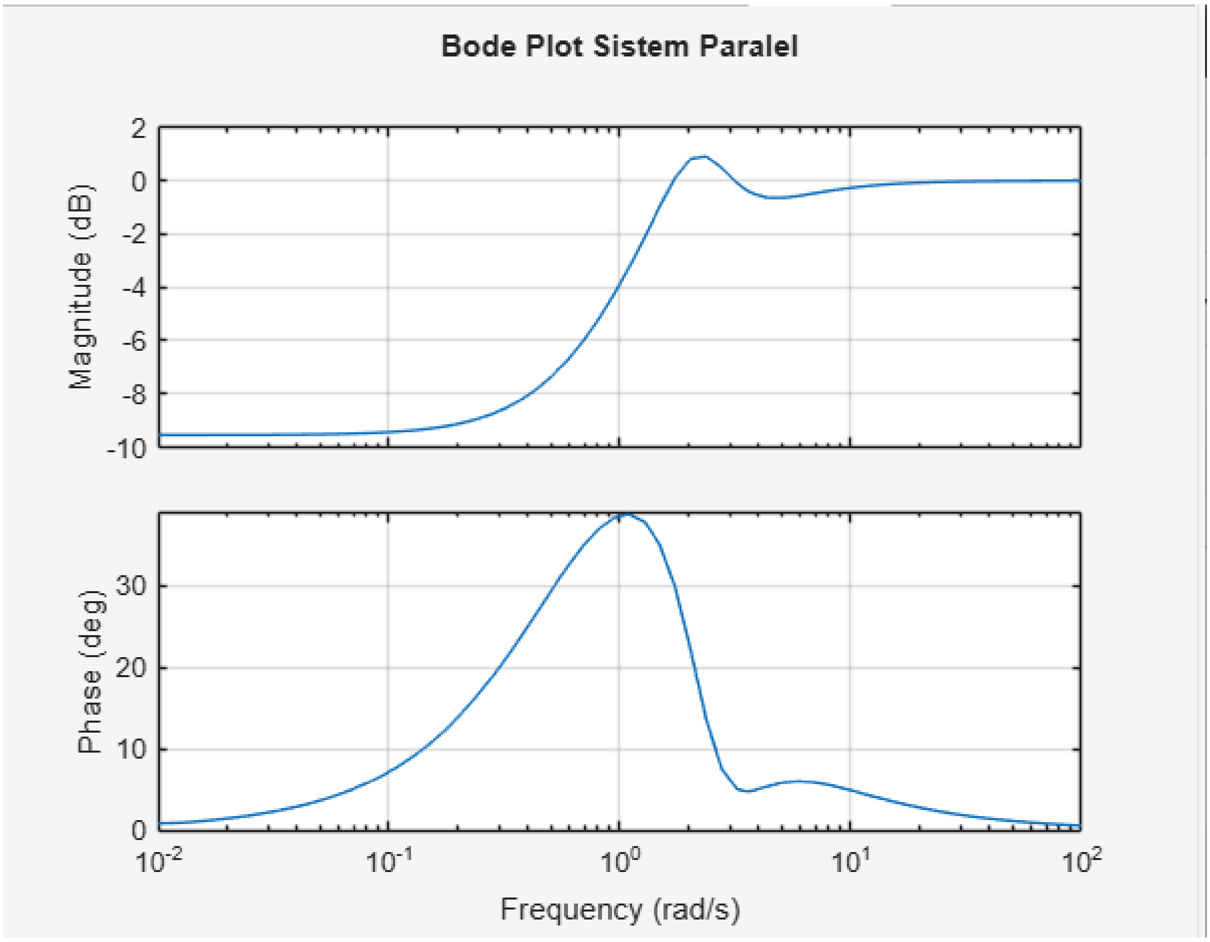

```
>> % soal no. 5
>> G1 = tf([1 1], [1 3]);
>>
>> G2 = tf([1], [100 10 1]);
>> sys_series = series(G1, G2);
>> disp('Fungsi Transfer Total (Seri):');
Fungsi Transfer Total (Seri):
>> sys_series

sys_series =

             s + 1
  ----------------------------
  100 s^3 + 310 s^2 + 31 s + 3

Continuous-time transfer function.
Model Properties
>> % Plot respon step
>> figure;
>> step(sys_series);
>> title('Respon Step Sistem Seri');
```

```
>> grid on;
>>
```



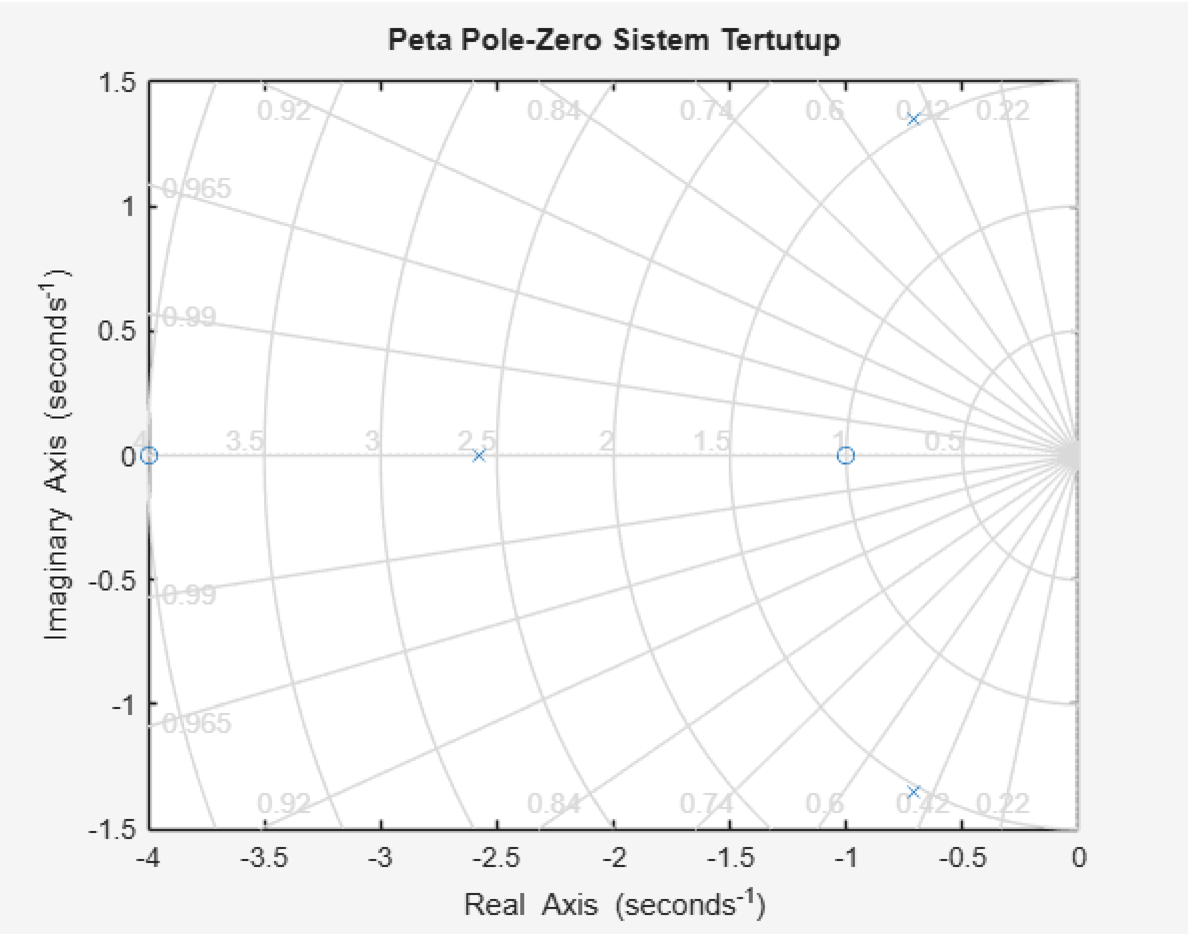

```
>> %soal no. 6
>> G1 = tf([1 1], [1 3]);
>> G2 = tf([1 0], [1 2 5]);
>> sys_parallel = parallel(G1, G2)

sys_parallel =

  s^3 + 4 s^2 + 10 s + 5
  -----------------------
  s^3 + 5 s^2 + 11 s + 15

Continuous-time transfer function.
Model Properties
>> % Visualisasi dengan Bode dan impulse
>> figure;
>> bode(sys_parallel);
>> grid on;
```

```
>>
```



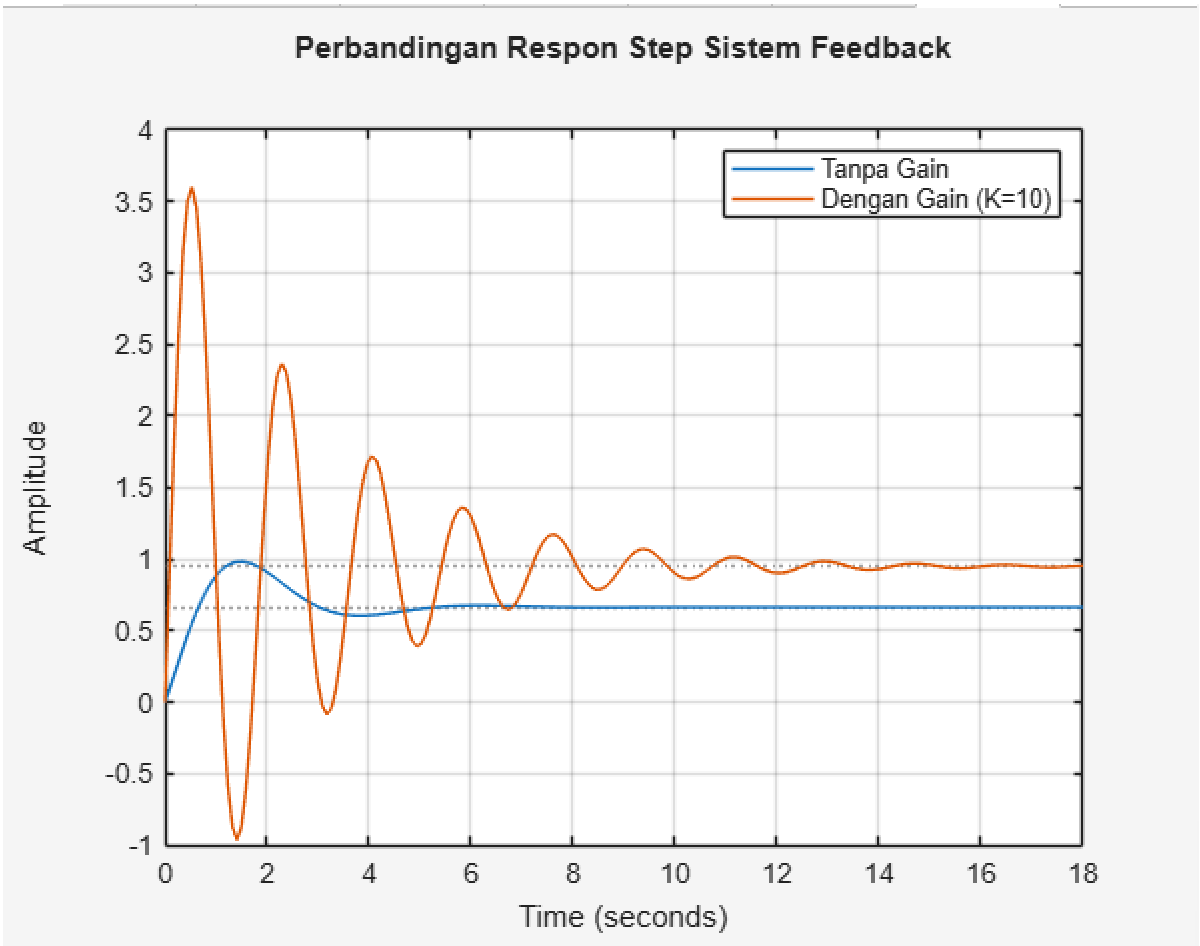

```
>> % soal no. 7
>> G1 = tf([1 4], [1 3 2]);
>> G2 = tf([1], [1 1]);
>> sys_cl = feedback(G1, G2)

sys_cl =

      s^2 + 5 s + 4
  ---------------------
  s^3 + 4 s^2 + 6 s + 6

Continuous-time transfer function.
Model Properties
>> % Analisis kestabilan
>> poles_cl = pole(sys_cl);
>> disp('Pole sistem tertutup:');
Pole sistem tertutup:
>> disp(poles_cl);
```

```
  -2.5747 + 0.0000i
  -0.7126 + 1.3500i
  -0.7126 - 1.3500i

>> if all(real(poles_cl) < 0)
    disp('Sistem Tertutup STABIL.');
else
    disp('Sistem Tertutup TIDAK STABIL.');
end
Sistem Tertutup STABIL.
>> % Visualisasi
>> figure;
>> pzmap(sys_cl);
>> title('Peta Pole-Zero Sistem Tertutup');
>> grid on;
>>
```




```
>> % soal no. 8
>> K = 10;
>> G1_gain = tf(K*[1 4], [1 3 2]);
```

```
>> sys_cl_gain = feedback(G1_gain, G2)

sys_cl_gain =

    10 s^2 + 50 s + 40
  -----------------------
  s^3 + 4 s^2 + 15 s + 42

Continuous-time transfer function.
Model Properties
>> % Analisis kestabilan
>> poles_gain = pole(sys_cl_gain)

poles_gain =

  -0.3471 + 3.5475i
  -0.3471 - 3.5475i
  -3.3058 + 0.0000i

>> if all(real(poles_gain) < 0)
    disp('Sistem DENGAN GAIN STABIL.');
else
    disp('Sistem DENGAN GAIN TIDAK STABIL.');
end
Sistem DENGAN GAIN STABIL.
>> % Bandingkan respon step dengan sistem sebelumnya
>> figure;
>> step(sys_cl, sys_cl_gain);
>> legend('Tanpa Gain', 'Dengan Gain (K=10)');
>> title('Perbandingan Respon Step Sistem Feedback');
>> grid on;
>>
```




```
>> % soal no. 9
>> num = [1 2];
>> den = [1 4 3];
>> % Bentuk fungsi transfer
>> G = tf(num, den)

G =

s + 2
-------------
s^2 + 4 s + 3

Continuous-time transfer function.
```

[Model Properties]

```
>> % Konversi ke bentuk State-Space
>> [A, B, C, D] = tf2ss(num, den)

A =
```

-4 -3
1 0

B =

1
0

C =

1 2

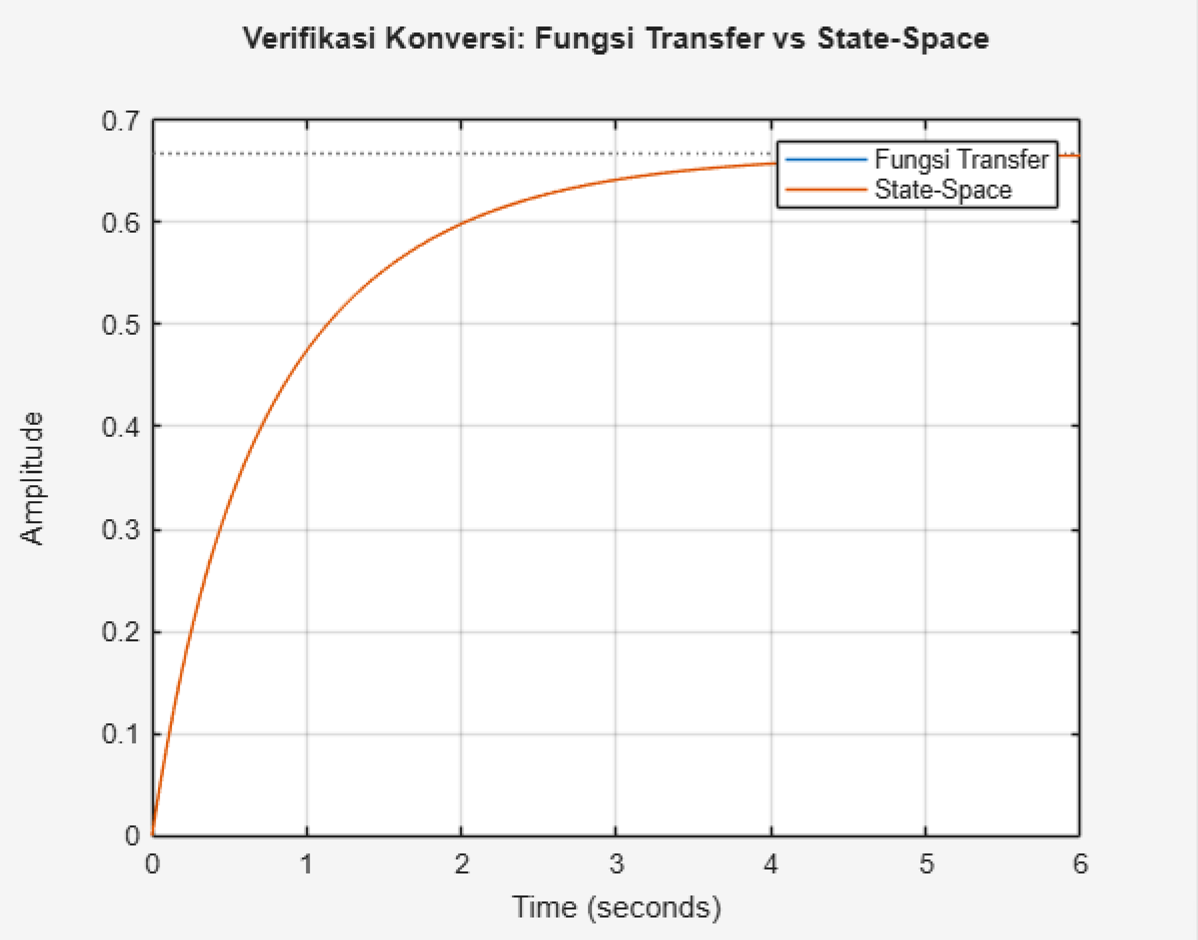
D =

0

>> % Buat model state-space
>> sys_ss = ss(A, B, C, D)

```
sys_ss =

  A =
       x1   x2
   x1  -4   -3
   x2   1    0

  B =
       u1
   x1   1
   x2   0

  C =
       x1   x2
   y1   1    2

  D =
       u1
   y1   0

Continuous-time state-space model.
Model Properties
>> % Verifikasi dengan plot step
>> figure;
>> step(G, sys_ss);
>> legend('Fungsi Transfer','State-Space');
>> title('Verifikasi Konversi: Fungsi Transfer vs State-Space');
>> grid on;
>> % Interpretasi makna fisik
>> disp('--- Interpretasi Fisik Matriks ---')
```

```
--- Interpretasi Fisik Matriks ---
>> disp('A → Matriks dinamika internal (menggambarkan interaksi antar state).');
A → Matriks dinamika internal (menggambarkan interaksi antar state).
>> disp('B → Matriks input (bagaimana input mempengaruhi state).');
B → Matriks input (bagaimana input mempengaruhi state).
>> disp('C → Matriks output (bagaimana state dikombinasikan menjadi output).');
C → Matriks output (bagaimana state dikombinasikan menjadi output).
>> disp('D → Feedthrough langsung (pengaruh langsung input ke output, biasanya
0).');
D → Feedthrough langsung (pengaruh langsung input ke output, biasanya 0).
>>
```




```
>> % soal no. 10
>> A = [0 1; -4 -3];
>> B = [0; 1];
>> C = [1 0];
>>
>> D = 0;
>> disp('Model State-Space diberikan:');
```

Model State-Space diberikan:
>> A, B, C, D

A =

0 1
-4 -3

B =

0
1

C =

1 0

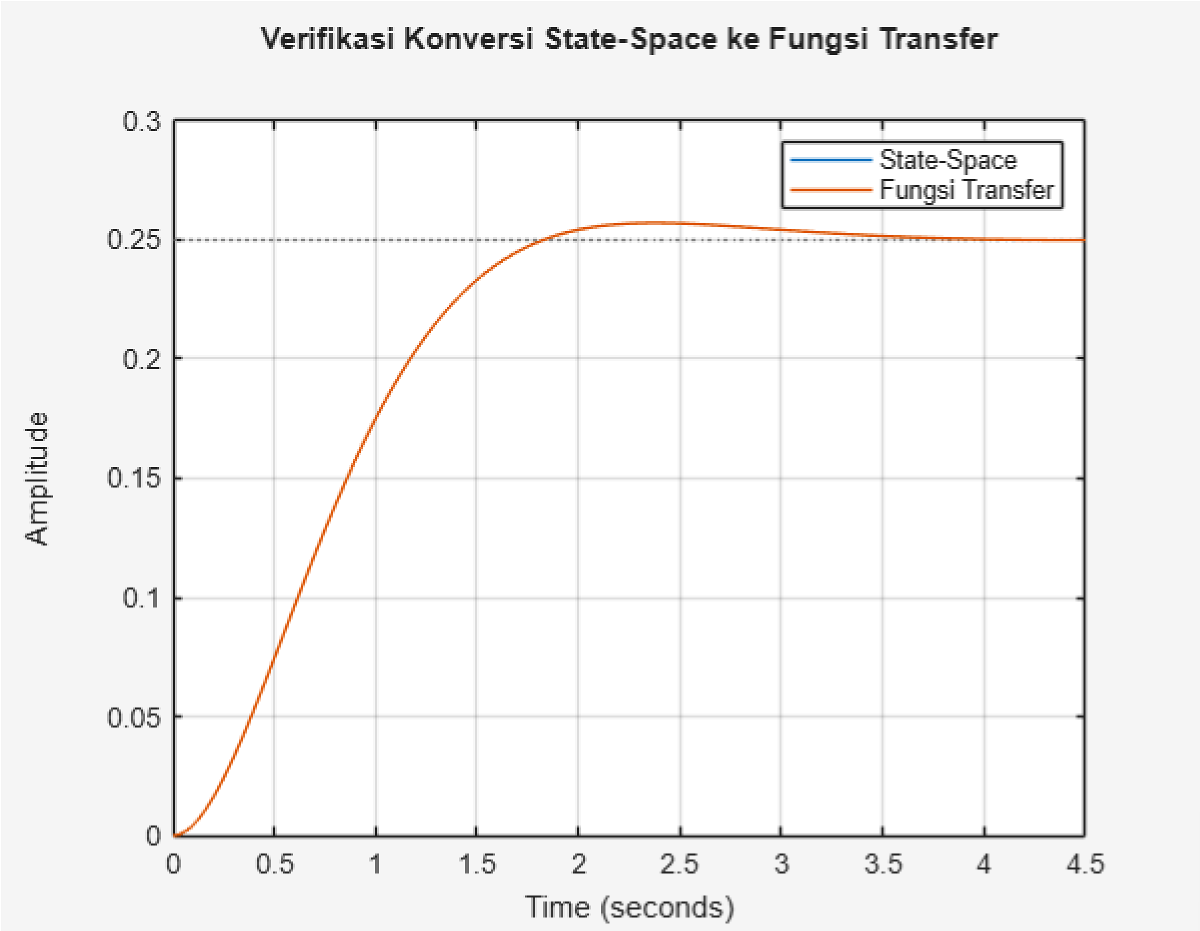
D =

0

>> % Buat model state-space
>> sys = ss(A, B, C, D)

```
sys =

  A =
       x1  x2
   x1   0   1
   x2  -4  -3

  B =
       u1
   x1   0
   x2   1

  C =
       x1  x2
   y1   1   0

  D =
       u1
   y1   0

Continuous-time state-space model.
Model Properties
>> % Konversi kembali ke fungsi transfer
>> [num2, den2] = ss2tf(A, B, C, D);
>> G2 = tf(num2, den2)
```

```
G2 =

        1
  -------------
  s^2 + 3 s + 4

Continuous-time transfer function.
Model Properties
>> % Verifikasi dengan plot respon step
>> figure;
>> step(sys, G2);
>> legend('State-Space','Fungsi Transfer');
>> title('Verifikasi Konversi State-Space ke Fungsi Transfer');
>> grid on;
>> % Analisis kestabilan
>> poles_sys = pole(sys)

poles_sys =

  -1.5000 + 1.3229i
  -1.5000 - 1.3229i

>> if all(real(poles_sys) < 0)
    disp('Sistem STABIL (semua pole di sebelah kiri s-plane).');
else
    disp('Sistem TIDAK STABIL (ada pole di sebelah kanan s-plane).');
end
Sistem STABIL (semua pole di sebelah kiri s-plane).
>>
```


Verifikasi Konversi State-Space ke Fungsi Transfer
State-Space
Fungsi Transfer
Amplitude
Time (seconds)
0.3
0.25
0.2
0.15
0.1
0.05
0
0
0.5
1
1.5
2
2.5
3
3.5
4
4.5