

Estimasi Filter Kalman Pada Sistem Motor DC

Implementasi Pada Sistem Orde 1

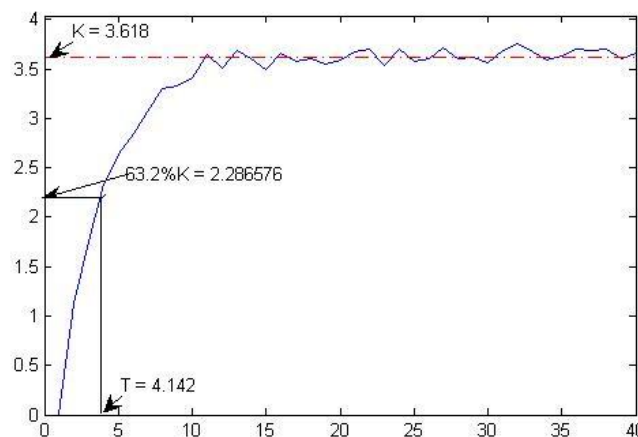
Oleh : Fahmizal (2209 105 010)

Teknik Sistem Pengaturan, Teknik Elektro ITS Surabaya

Bentuk umum fungsi alih dari sistem orde satu ialah sebagai berikut:

$$\frac{C(s)}{R(s)} = \frac{K}{\tau s + 1}$$

Misal hasil identifikasi *plant* motor DC menggunakan perangkat lunak LabVIEW ialah sebagai berikut (Gambar 1), kemudian diberi masukan sebesar *step* 1 volt dengan waktu *sampling* 1 ms.



Gambar 1. Identifikasi *plant* motor DC dengan waktu *sampling* sebesar 1ms.

Dari grafik respon diperoleh fungsi alih Plant Motor DC ialah

$$K = \text{gain overall} = 3,618$$

$$\tau = 4,142 * 1 \text{ ms} = 0,004142 \text{ detik}$$

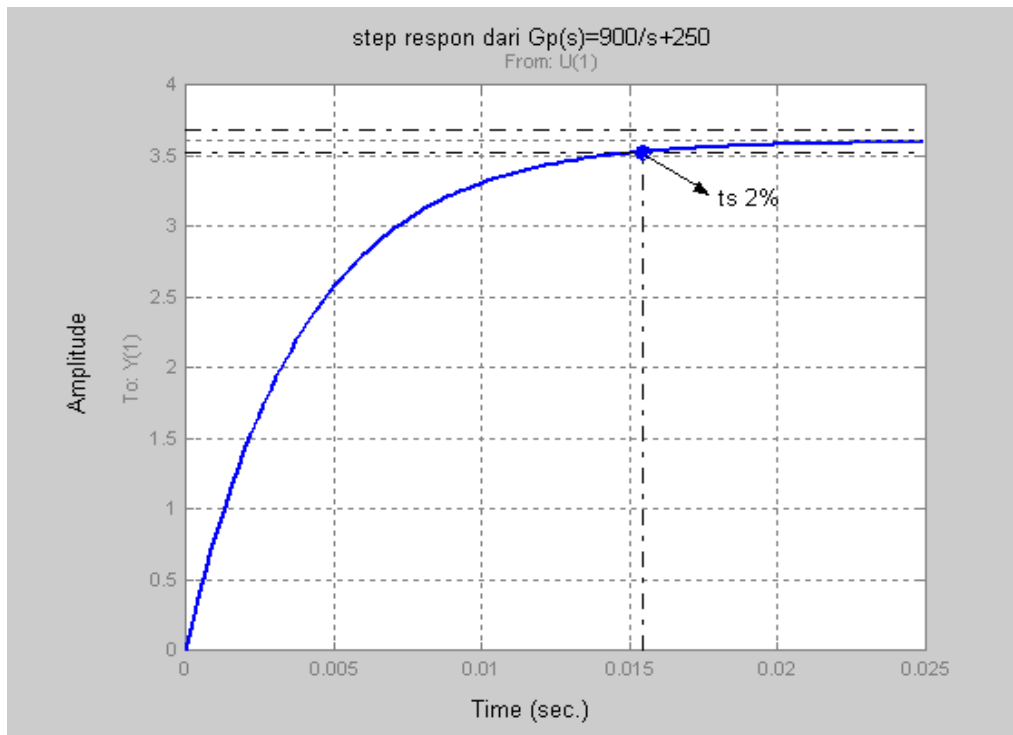
$$Gp(s) = \frac{3.618}{0.004142s + 1}$$

Untuk kemudahan dalam menganalisa maka dilakukan pembulatan data matematis dari fungsi alih, berikut hasil pembulatan fungsi alih dari Plant:

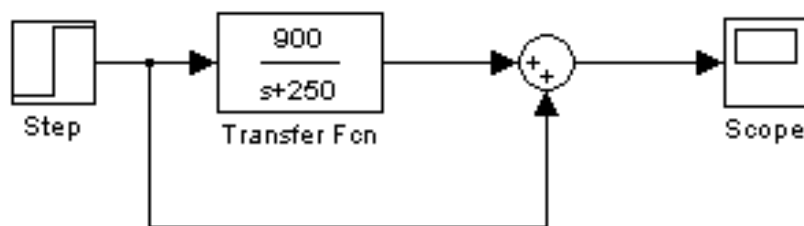
$$Gp(s) = \frac{3.6}{0.004s + 1} = \frac{900}{s + 250}$$

Dengan menggunakan matlab dapat digambarkan step respon sebagai berikut:

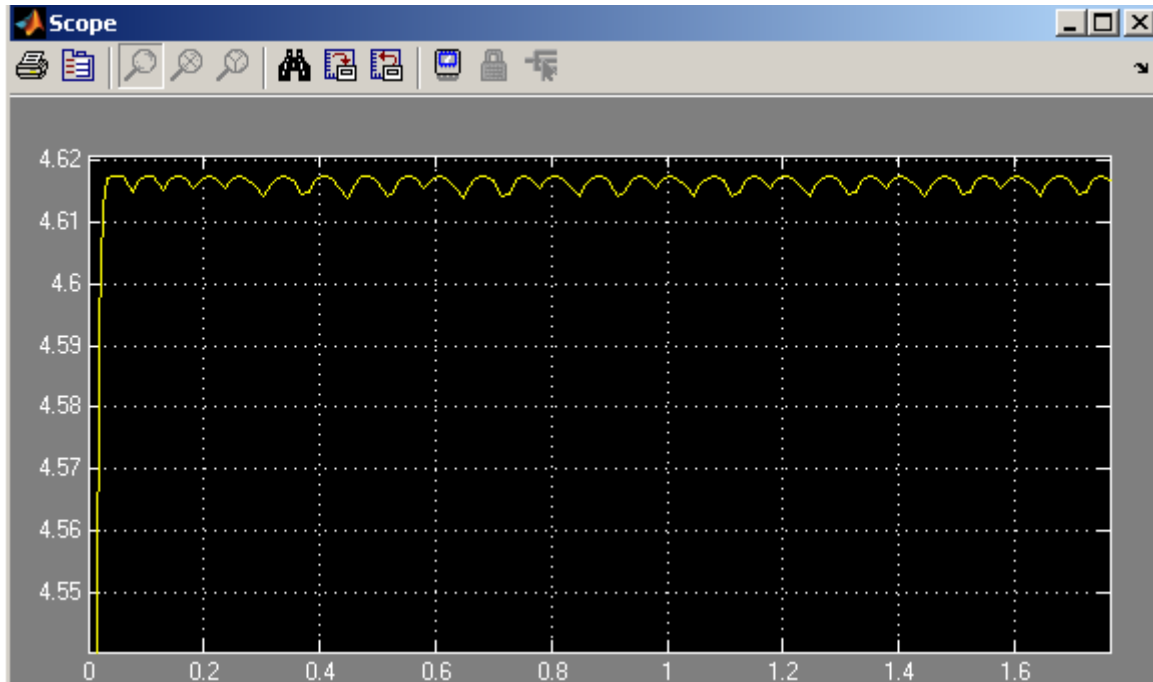
```
%plot step respon Gp(s)
num=900;
den=[1 250];
Gp=tf(num,den);
step(Gp)
grid
title ('step respon dari Gp(s)=900/s+250')
```



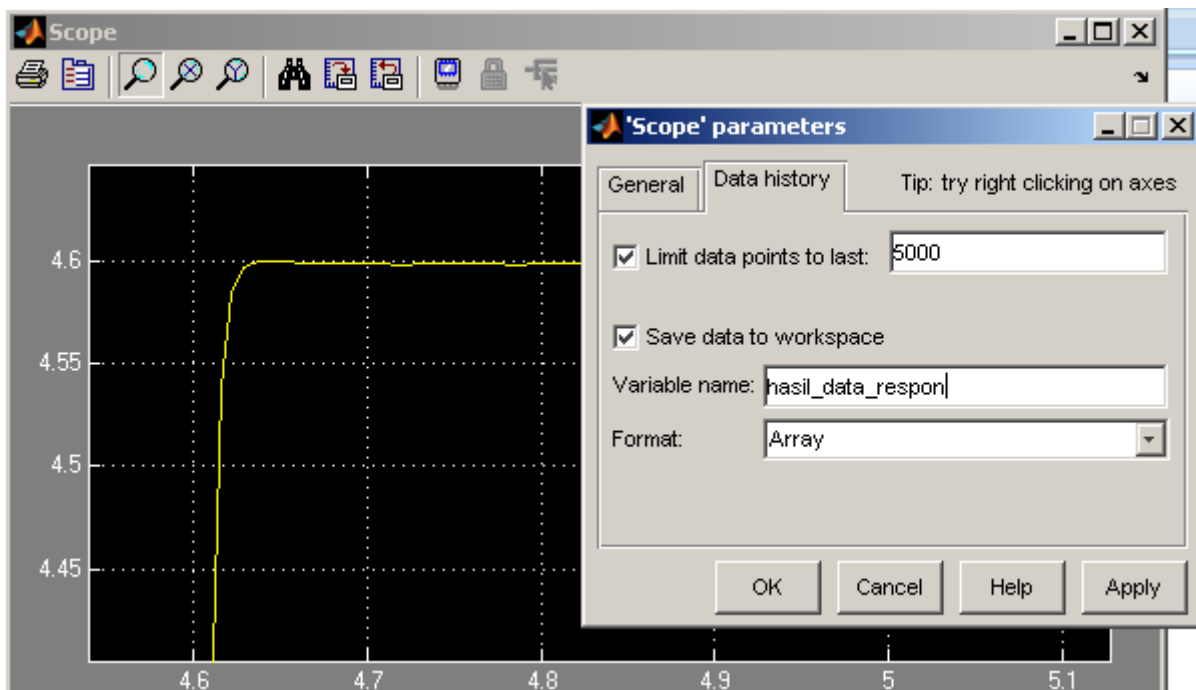
Gambar 2. Step Respon Open Loop dari Gp(s).



Gambar 2. Respon Open Loop Plant Menggunakan Simulink



Menyimpan respon output pada workspace:





ESTIMASI FILTER KALMAN

Digunakan untuk estimasi state x pada time discrete controlled process, oleh karena itu model matematis plant motor DC di ubah ke bentuk State Space, langkah tersebut dapat dilakukan dengan cara sebagai berikut:

$$[A, B, C, D] = \text{tf2ss}(\text{num}, \text{den})$$

```
num=900;  
den=[1 250];  
[A,B,C,D]=tf2ss(num,den)
```

Hasil:

$$A =$$

$$-250$$

$$B =$$

$$1$$

$$C =$$

$$900$$

$$D =$$

$$0$$

Bentuk persamaan “State Space”:

$$x_{k+1} = A_k X_k + B U_k$$

$$y = C_k + D U_k$$

maka diperoleh:

$$x_{k+1} = -250x_k + U_k$$

$$y = 900x_k$$



ALGORITMA FILTER KALMAN

```
disp('                                Fahmizal');
disp('                                2209.105.010');
disp(' ');
disp('Program estimasi KALMAN FILTER');
disp(' ');
disp(' ');
disp('Kondisi State Space');
disp(' ');
disp('Xk+1=Ak Xk + B Uk');
disp('y= Ck');
disp(' ');
A = input('masukkan nilai A orde (n*n)= ');
B = input('masukkan nilai B orde (n*1)= ');
C = input('masukkan nilai C orde (1*n)= ');
disp(' ');
disp('Masukan nilai awal KALMAN FILTER ');
Q = input('masukkan nilai Q = ');
R = input('masukkan nilai R = ');
H = input('masukkan nilai H = ');
x_min(1) = input('masukkan nilai X|k-1 = ');
p_min(1) = input('masukkan nilai P|k-1 = ');
iterasi= input('masukkan nilai iterasi k=');
disp(' ');
disp('Hasil Masukan');
disp('-----');

%state space
A
B
C
%kondisi awal kalman filter
Q
R
H
x_min
p_min
iterasi

%data pengukuran
for n=1:1:iterasi;
    z(n)=hasil_pengukuran(n);
end;

%indeks buat plotting
for n=1:1:iterasi;
    index(n)=n;
end;

%program perulangan estimasi kalman filter

for n=1:1:iterasi;
%=====Prediksi=====

%proyeksi kondisi yang akan datang
x_kalman(n)=[A*x_min(n)]+B
    %proyeksi kovarian error yang akan datang
p_kalman(n)=[A*p_min(n)*A'+Q]
```

```

%=====Koreksi=====

%hitung Kalman Gain
kalman_gain(n)=[p_kalman(n)*H']*inv([[H*p_kalman(n)*H']]+R])

%update estimasi dengan pengukuran Zk
x_kalman_estimasi(n)=x_kalman(n)+[kalman_gain(n)*[z(n)-x_kalman(n)]]

%update kovarian error
p_min(n+1)=[1-kalman_gain(1,n)]*p_kalman(n)
x_min(n+1)=x_kalman_estimasi(n)

end;

%figure
plot(index,x_kalman,'r-',index,hasil_pengukuran(1:n,1),'b-')
grid
legend('x estimasi','Hasil Pengukuran')
title('Kalman filter ')
xlabel('iterasi')
ylabel('output')

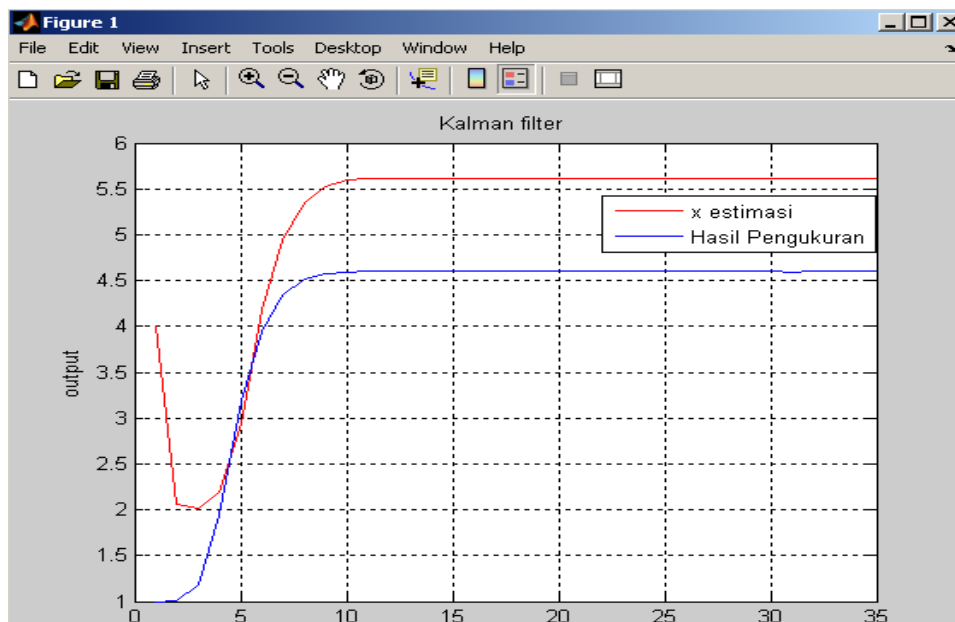
```

Hasil Percobaan 1 dengan memberikan nilai:

```

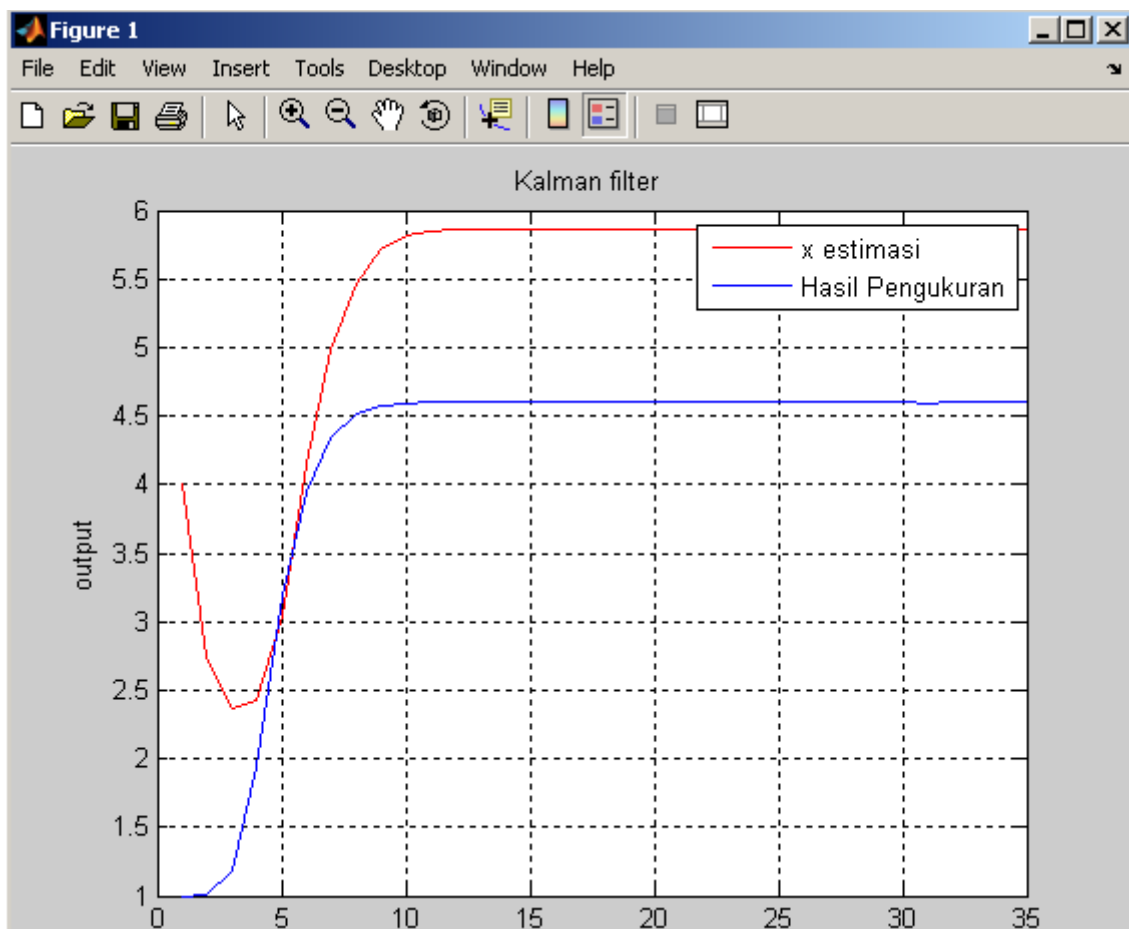
%state space
A
B
C
%kondisi awal kalman filter
Q=50
R=1
H=1
x_min=3
p_min=1
iterasi=35

```



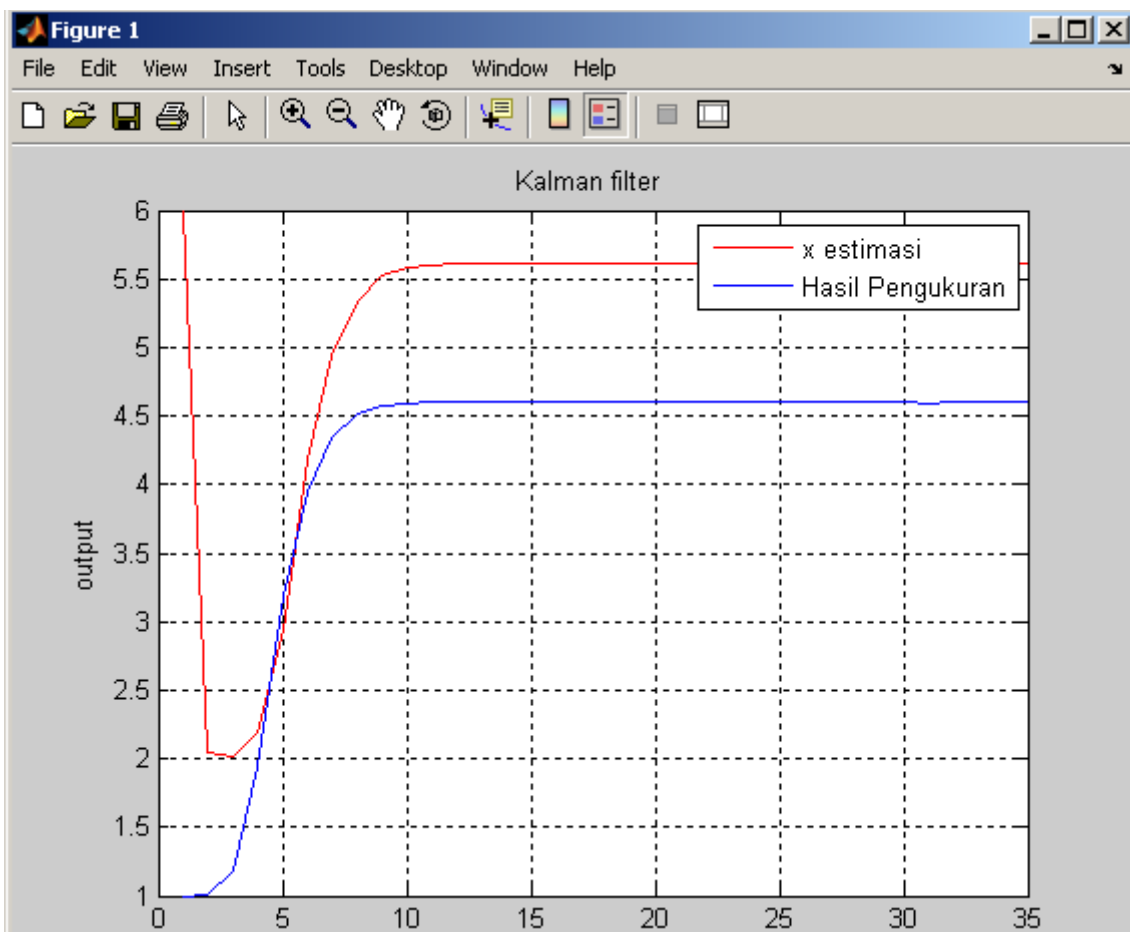
Hasil Percobaan 2 dengan memberikan nilai:

```
%state space  
A  
B  
C  
%kondisi awal kalman filter  
Q=75  
R=25  
H=1  
x_min=3  
p_min=1  
iterasi=35
```



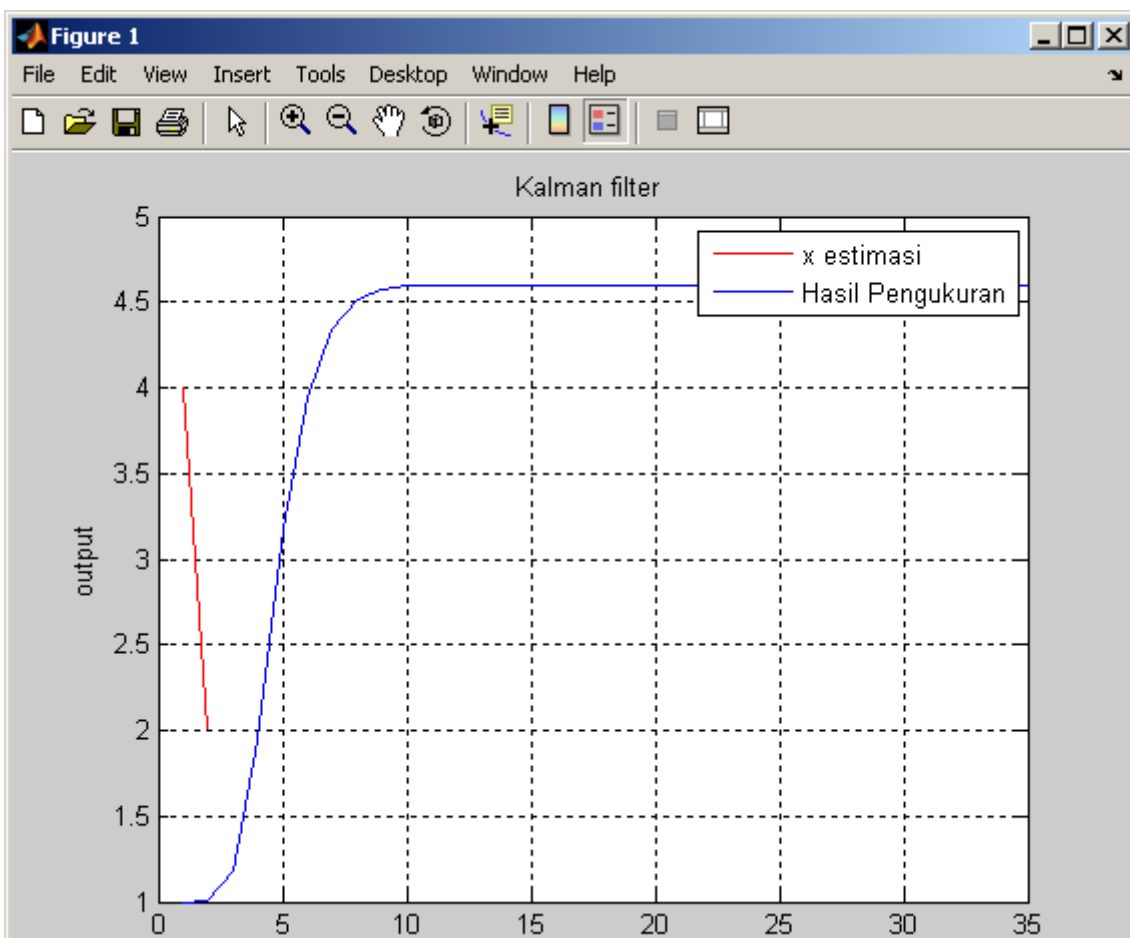
Hasil Percobaan 3 dengan memberikan nilai:

```
%state space  
A  
B  
C  
%kondisi awal kalman filter  
Q=75  
R=1  
H=1  
x_min=9  
p_min=1  
iterasi=35
```



Hasil Percobaan 4 dengan memberikan nilai:

```
%state space  
A  
B  
C  
%kondisi awal kalman filter  
Q=0  
R=0  
H=1  
x_min=3  
p_min=1  
iterasi=35
```



Hasil Percobaan 5 dengan memberikan nilai:

```
%state space  
A  
B  
C  
%kondisi awal kalman filter  
Q=1  
R=50  
H=1  
x_min=9  
p_min=1  
iterasi=35
```

