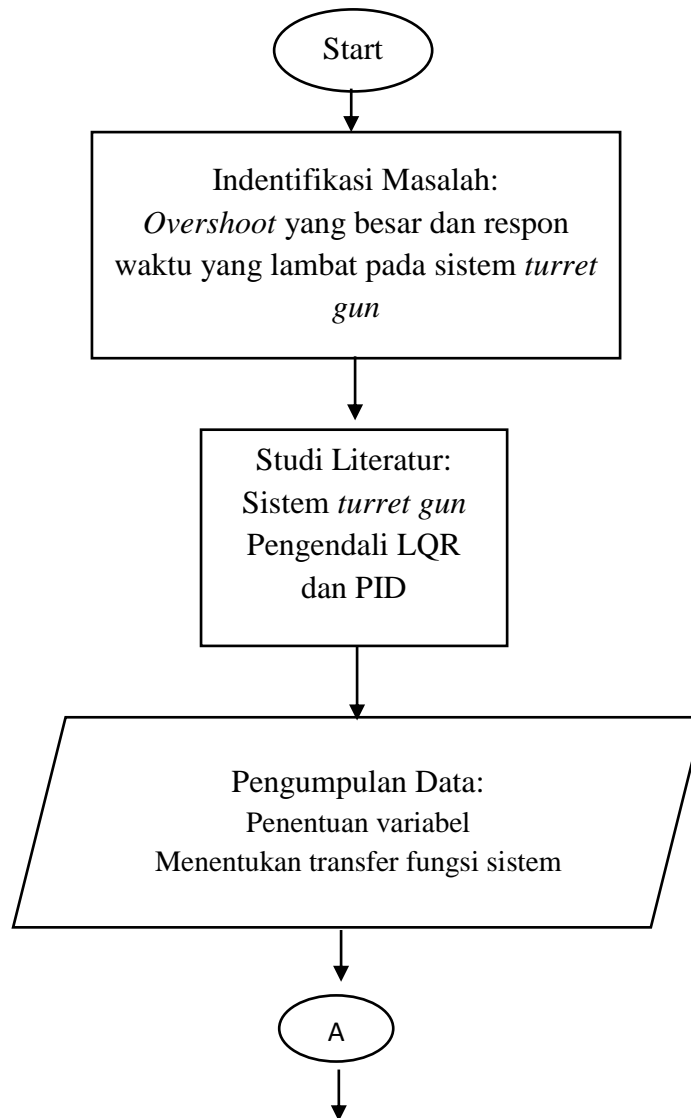


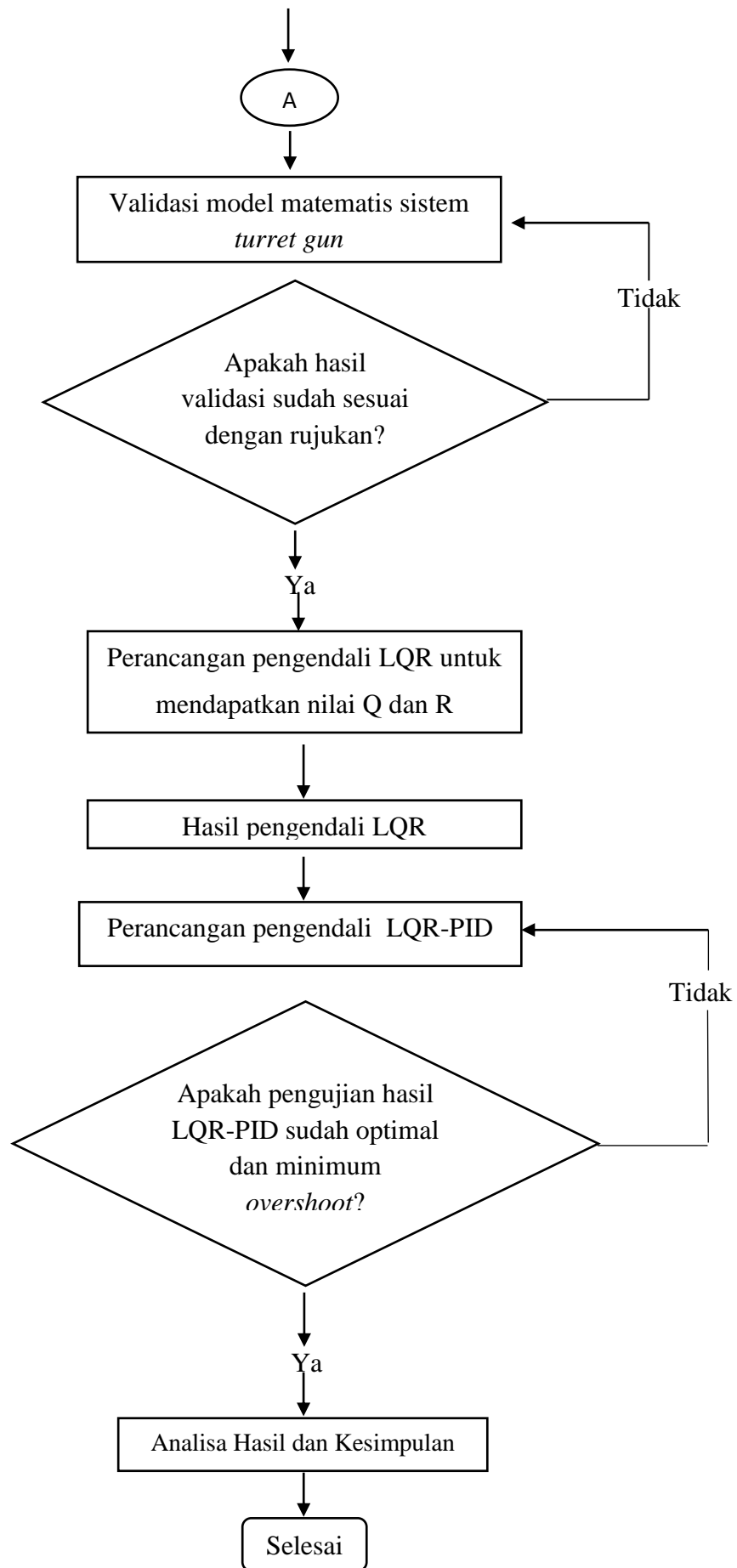
BAB III

METODOLOGI DAN PENELITIAN

3.1 Proses Alur Penelitian

Pada penelitian ini terdapat beberapa tahapan alur penelitian yang dilakukan, dimulai dari studi literature, identifikasi sistem, penurunan model matematis, perancangan pengendali LQR dan PID, dan analisa akhir dari penelitian ini. Adapun tahapan alur penelitian ini adalah sebagai berikut:





Gambar 3.1 Flow Chart Penelitian

Agar tujuan dalam melakukan penelitian ini dapat tercapai, maka pada penelitian ini perlu dilakukan perumusan masalah, penentuan judul serta mencapai tujuan yang diharapkan dari suatu penelitian. Oleh karena itu, terdapat beberapa tahapan rencana yang harus dilakukan yaitu:

1. Identifikasi masalah

Mencari masalah yang terdapat pada sistem *turret gun* dengan melakukan *review* dari beberapa referensi yang berkaitan dengan pengendalian pada sistem *turret gun*.

2. Studi literatur

Melakukan *review* dan mempelajari referensi yang berkaitan dengan judul penelitian ini, mengenai sistem *turret gun* dan perancangan pengendali LQR dan PID.

3. Pengumpulan data pra design

Tujuan dari pengumpulan data pra design adalah untuk mengetahui karakteristik sistem yang akan diteliti, dan transfer fungsi sistem yaitu:

$$G = \frac{c}{s^2 + sa + b}$$

4. Penentuan variabel

Data-data yang didapat dari pengumpulan data pra design diubah dalam bentuk model matematis dengan persamaan transfer fungsi sebagai berikut:

$$TF = \frac{0,05294}{s^2+0.3971} \quad (3.1)$$

5. Validasi dan pengujian model matematis *turret gun*

Tahapan ini adalah pengujian model matematis *plant* dalam bentuk transfer fungsi dari sistem *turret gun* yang diubah dalam bahasa program *matlab simulink* untuk menyesuaikan hasil keluaran dari sistem dengan referensi yang digunakan.

6. Hasil validasi dan pengujian model matematis *turret gun*

Tahapan ini melihat apakah hasil keluaran dari sistem sama dengan hasil keluaran dari referensi yang digunakan

7. Desain pengendali LQR

Dalam tahapan ini perancangan pengendali LQR untuk mendapatkan nilai matriks Q dan R

8. Analisa hasil pengendali LQR

Pada tahapan ini pengendali LQR yang telah dirancang disimulasikan untuk mengetahui respon dari sistem menggunakan pengendali LQR

9. Desain pengendali LQR-PID

Tahapan ini adalah mendesain pengendali LQR yang dikombinasikan dengan PID sampai mendapat hasil *output* yang diharapkan.

10. Analisa hasil pasca desain

Pada tahapan ini akan dilakukan analisa terhadap hasil pengujian *turret* yang telah dipasangkankan pengendali LQR-PID, untuk pemeriksaan hasil keluaran respon apakah sudah optimal atau belum optimal?. Analisa keluaran respon yang optimal dapat diketahui dari beberapa hal berikut ini: pencapaian setpoint yang diinginkan, *settling time* (t_s) yang cepat, waktu naik atau *rise time* (t_r) yang cepat, waktu tunda atau *delay time* (t_d), waktu puncak (t_p), dan terjadi maksimum overshoot.

11. Hasil desain dan kesimpulan

Pada tahapan ini penulis menyesuaikan antara hasil design dengan tujuan yang ingin dicapai. Apabila tujuan dari penelitian ini telah terpenuhi maka penelitian dapat dinyatakan berhasil, namun apabila hasil dari penelitian belum mencapai tujuan yang ingin dicapai maka perlu dilakukan pemeriksaan kembali terhadap perancangan pengendali. Setelah masalah telah dapat diselesaikan dan tujuan awal dari penelitian ini telah sesuai dengan yang diharapkan maka dapat ditarik kesimpulan dari penelitian ini.

3.2 Pengumpulan Data

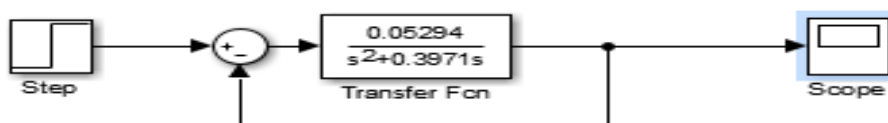
Dalam perancangan pengendali LQR dan LQR-PID persamaan yang digunakan adalah persamaan *state space* dari transfer fungsi sistem *turret gun*. *Set point* 20° , model matematis sistem *turret gun* serta nilai Q dan R

3.3 Penentuan Variabel

Data-data yang didapat dibuat dalam bentuk model matematis sistem dalam bentuk transfer fungsi yaitu:

$$G(s) = \frac{0.05294}{s^2 + 0.3971s} \quad (3.2)$$

3.4 Validasi Model Matematis



Gambar 3.2 Blok Simulasi *Turret Gun* Secara *Open Loop*

Gambar 3.2 adalah blok pengujian simulasi turret gun secara *open loop* tanpa menggunakan pengendali, pengujian *open loop* ini bertujuan untuk mengetahui perilaku dan karakteristik sistem sebelum perancangan pengendali [5].

3.5 Perancangan Pengendali *Linear Quadratic Regulator* (LQR)

Persamaan transfer fungsi dari *turret gun* diubah dalam bentuk *state space* karena pada perancangan pengendali LQR dibutuhkan pemodelan *state space* dari sistem. Adapun pemodelan *state space* tersebut adalah:

$$\frac{Y(s)}{U(s)} = \frac{0.05294}{s^2 + 0.3971s}$$

$$s^2Y(s) + 0.3971sY(s) = 0.05294U(s)$$

Dari turunan transfer fungsi pada persamaan *turret gun* tersebut kemudian diturunkan dengan menggunakan metode *inverse transformasi laplace* balik sehingga diperoleh persamaan diferensi orde dua:

$$\frac{d^2y}{dt^2} + 0.3971 \frac{dy}{dt} = 0.05294U$$

$$\ddot{y} + 0.3971\dot{y} = 0.05294U$$

$$\ddot{y} = -0.3971\dot{y} + 0.05294U$$

Di definisikan:

$$x_1 = y$$

$$x_2 = \dot{y}$$

$$\dot{x}_1 = \dot{y} = x_2$$

$$\dot{x}_2 = \ddot{y}$$

$$\text{Jadi, } \dot{x}_2 = -0.3971\dot{y} + 0.05294U$$

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & -0.3971 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 0.05294 \end{bmatrix} U \quad (3.3)$$

$$Y = [1 \quad 0] \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \quad (3.4)$$

Jadi, dari persamaan di atas didapatkan nilai matriks A, B, C, D yaitu sebagai berikut:

$$A = \begin{bmatrix} 0 & 1 \\ 0 & -0.3971 \end{bmatrix} \quad (3.5)$$

$$B = \begin{bmatrix} 0 \\ 0.05294 \end{bmatrix} \quad (3.6)$$

$$C = [1 \ 0] \quad (3.7)$$

$$D = [0] \quad (3.8)$$

Perhitungan yang dilakukan pengendali LQR dilakukan dengan memasukan persamaan state space yang telah didapat kedalam software matlab. Untuk melakukan simulasi menggunakan pengendali LQR dibutuhkan matriks Q dan R yang optimal. Untuk mendapatkan matriks Q dan R yang optimal dilakukan dengan cara menetapkan matriks R bernilai konstan yaitu 1, sehingga didapatkan nilai matriks Q adalah bilangan itu sendiri, karena hal inilah yang diinginkan. Penentuan matriks Q menggunakan *trial and error* dengan analisa IAE. Percobaan *trial and error* dengan analisa IAE dapat dilihat pada tabel 3.1

Tabel 3.1 Matriks R Konstan dengan Analisa IAE

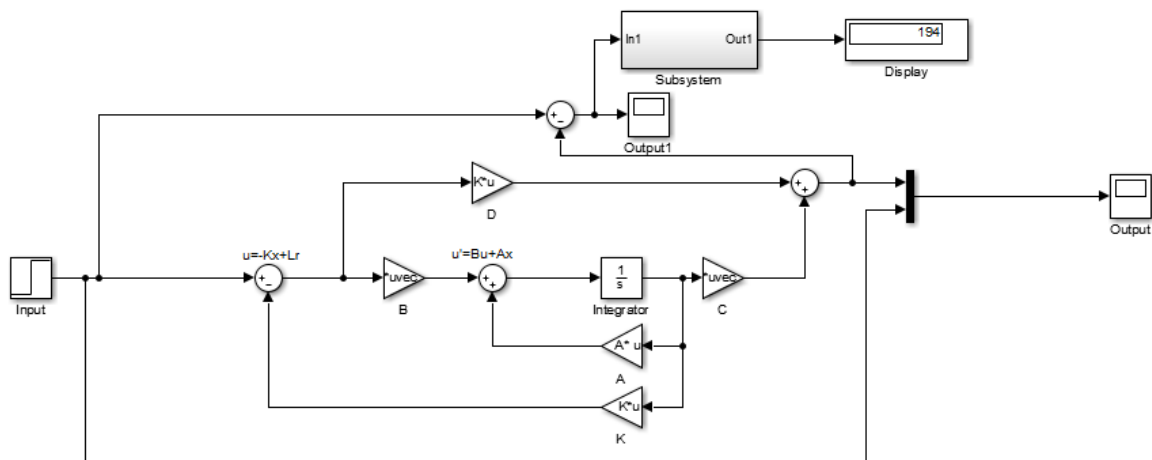
NO.	Matriks R	Matriks Q	IAE
1	[1]	$\begin{bmatrix} 0.01 & 0 \\ 0 & 0 \end{bmatrix}$	6381
2	[1]	$\begin{bmatrix} 0.02 & 0 \\ 0 & 0 \end{bmatrix}$	5728
3	[1]	$\begin{bmatrix} 0.03 & 0 \\ 0 & 0 \end{bmatrix}$	5034
4	[1]	$\begin{bmatrix} 0.04 & 0 \\ 0 & 0 \end{bmatrix}$	4531
5	[1]	$\begin{bmatrix} 0.05 & 0 \\ 0 & 0 \end{bmatrix}$	4139
6	[1]	$\begin{bmatrix} 0.06 & 0 \\ 0 & 0 \end{bmatrix}$	3821
7	[1]	$\begin{bmatrix} 0.07 & 0 \\ 0 & 0 \end{bmatrix}$	3554
8	[1]	$\begin{bmatrix} 0.08 & 0 \\ 0 & 0 \end{bmatrix}$	3327
9	[1]	$\begin{bmatrix} 0.09 & 0 \\ 0 & 0 \end{bmatrix}$	3128
10	[1]	$\begin{bmatrix} 0.1 & 0 \\ 0 & 0 \end{bmatrix}$	2954
11	[1]	$\begin{bmatrix} 0.2 & 0 \\ 0 & 0 \end{bmatrix}$	1888
12	[1]	$\begin{bmatrix} 0.3 & 0 \\ 0 & 0 \end{bmatrix}$	1346

13	[1]	$\begin{bmatrix} 0.4 & 0 \\ 0 & 0 \end{bmatrix}$	1004
14	[1]	$\begin{bmatrix} 0.5 & 0 \\ 0 & 0 \end{bmatrix}$	763.7
15	[1]	$\begin{bmatrix} 0.6 & 0 \\ 0 & 0 \end{bmatrix}$	584.4
16	[1]	$\begin{bmatrix} 0.7 & 0 \\ 0 & 0 \end{bmatrix}$	445,6
17	[1]	$\begin{bmatrix} 0.8 & 0 \\ 0 & 0 \end{bmatrix}$	336,1
18	[1]	$\begin{bmatrix} 0.9 & 0 \\ 0 & 0 \end{bmatrix}$	250.4
19	[1]	$\begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}$	194
20	[1]	$\begin{bmatrix} 2 & 0 \\ 0 & 0 \end{bmatrix}$	690.5
21	[1]	$\begin{bmatrix} 3 & 0 \\ 0 & 0 \end{bmatrix}$	918.8
22	[1]	$\begin{bmatrix} 4 & 0 \\ 0 & 0 \end{bmatrix}$	1057
23	[1]	$\begin{bmatrix} 5 & 0 \\ 0 & 0 \end{bmatrix}$	1153

Pada tabel percobaan 3.1 dilakukan *trial and error* dengan memasukkan nilai matriks Q yaitu $\begin{bmatrix} 0.01 & 0 \\ 0 & 0 \end{bmatrix}$ kemudian didapatkan nilai IAE namun respon keluaran sistem melebihi *set point*. Selanjutnya dilakukan penambahan nilai matriks Q sebesar 0.1, nilai IAE semakin kecil namun respon keluaran sistem masih melebihi *set point*. Penambahan terus dilakukan sebesar 0.1, sampai pada matriks $Q = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}$ pada matriks ini respon keluaran sistem telah mencapai *set point* dan didapatkan nilai IAE. Selanjutnya dilakukan penambahan nilai matriks Q sebesar 1, namun nilai IAE semakin besar dan respon keluaran sistem tidak mencapai *set point*. Penambahan terus dilakukan sampai pada matriks $Q = \begin{bmatrix} 5 & 0 \\ 0 & 0 \end{bmatrix}$ nilai IAE lebih besar dari pada matriks $Q = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}$ dan respon keluaran sistem semakin tidak mencapai *set point*. Sehingga dilakukan perancangan pengendali LQR dengan menggunakan nilai matriks $Q = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}$, matriks $R = [1]$ dan nilai IAE = 193.9 pada sistem *turret gun* sumbu *azimuth*.

Perancangan pengendali LQR adalah berdasarkan penurunan persamaan matematis LQR pada bab 2. Sehingga berdasarkan turunan matematis dan skema LQR yang sudah

didapat akan dibuat ke Simulink. Blok-blok Simulink pengendali LQR pada sistem *turret gun* sumbu *azimuth* diperlihatkan gambar berikut:



Gambar 3.3 Blok Diagram *Turret Gun* Sumbu *Azimuth* dengan Pengendali LQR

Adapun program yang dimasukkan dalam M – file adalah seperti yang ditunjukkan pada gambar (3.4) dibawah ini:

```

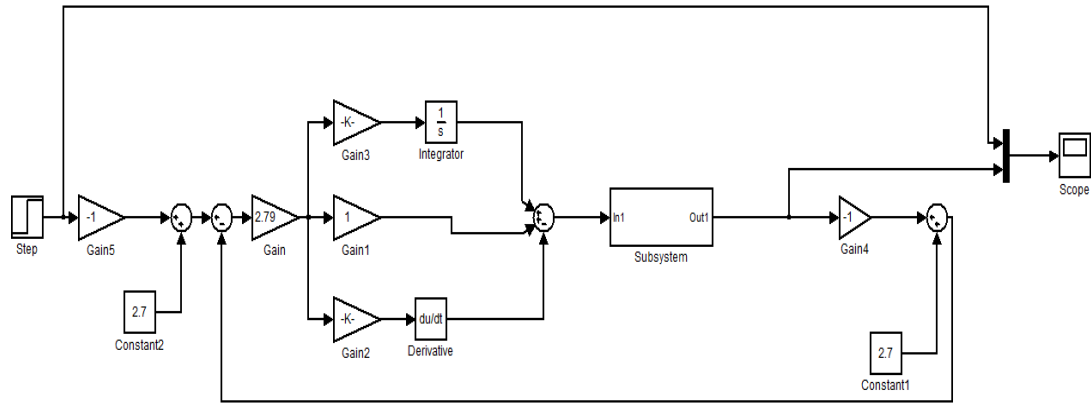
Editor - D:\Barakallah\MATLAB ALISIA\cobacobalilis.m
cobacobalilis.m
1 - A=[0 1 ;0 -0.3971];
2 - B=[0;0.05294];
3 - C=[1 0];
4 - D=[0];
5 - R=[1];
6 - Q=[1 0; 0 0];
7 - N=[0];
8 - sys=ss(A,B,C,D)
9 - [K,S,e]=lqr(sys,Q,R,N)

```

Gambar 3.4 Tampilan Program M-File Sistem *Turret Gun* Sumbu *Azimuth* Menggunakan Pengendali LQR

3.6 Perancangan Pengendali Proporsional Integral Derivatif (PID)

Berdasarkan studi literatur yang telah ditentukan maka desain pengendali PID dapat dibuat dengan bentuk *simulink* yang terdapat pada gambar (3.5) dengan memasukkan nilai *transfer function* yang diperoleh dari sistem *turret gun* sumbu *azimuth* dengan kondisi *setpoint* 20 yang telah dimasukkan kedalam blok desain pengendali PID. Penentuan nilai konstanta K_p , K_i dan K_d diperoleh dengan menggunakan metode *heuristic*.

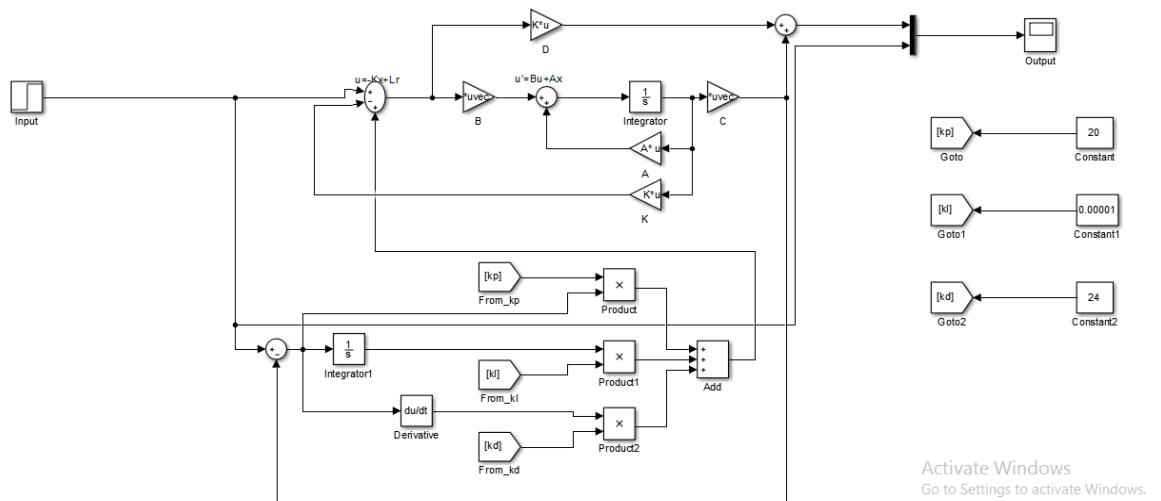


Gambar 3.5 Blok Diagram pengendali PID

3.7 Perancangan Pengendali LQR – PID

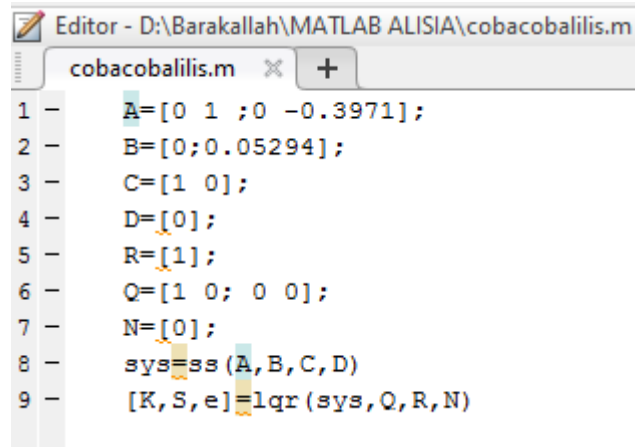
Setelah mendapatkan dan mempelajari desain pengendali LQR dan desain pengendali PID maka dapat mengkombinasikan pengendali optimal LQR dan pengendali PID agar kinerja dari pengendali LQR menjadi lebih baik dan optimal. Optimal difokuskan pada respon waktu yang cepat, *overshoot* yang kecil, *error* yang kecil dan tidak ada osilasi.

Nilai parameter PID dituning menggunakan metode heuristik, adapun tabel *heuristic* parameter PID terdapat pada lampiran. Sehingga didapatkan nilai $K_p = 20$, $K_i = 0.00001$ dan $K_d = 24$. Blok diagram desain pengendali LQR – PID pada sistem *turret gun* sumbu *azimuth* adalah sebagai berikut:



Gambar 3.6 Blok Diagram Desain Pengendali LQR – PID

Adapun program yang dimasukkan dalam M – file adalah seperti yang ditunjukkan pada gambar (3.11) dibawah ini:



```
Editor - D:\Barakallah\MATLAB ALISIA\cobacobalilis.m
cobacobalilis.m x +
1 - A=[0 1 ;0 -0.3971];
2 - B=[0;0.05294];
3 - C=[1 0];
4 - D=[0];
5 - R=[1];
6 - Q=[1 0; 0 0];
7 - N=[0];
8 - sys=ss(A,B,C,D)
9 - [K,S,e]=lqr(sys,Q,R,N)
```

Gambar 3.7 Tampilan Program M-File Sistem *Turret Gun* Sumbu *Azimuth* dengan Pengendali Optimal LQR-PID