

# Kendali Kecepatan dan Posisi Motor DC

Priyova M. Rafief<sup>1</sup>, Karunia Dini F.<sup>1</sup>, Octsana Dhiyaa W.<sup>1</sup>, Bodhi Setiawan<sup>1</sup>

Departemen Teknik Elektro dan Informatika, Sekolah Vokasi, Universitas Gadjah Mada<sup>1</sup>

Praktikum Sistem Kendali Lanjut



UNIVERSITAS  
GADJAH MADA

# Anggota Kelompok 1

Priyoya M. Rafie  
(20/457197/SV/17644)

Bodhi Setiawan  
(20/464239/SV/18558)

Karunia Dini F.  
(20/464248/SV/18567)

Octsana Dhiyaa W.  
(20/464253/SV/18572)

Dosen Pembimbing : Fahmizal, S.T., M.Sc.

# Isi Pembahasan

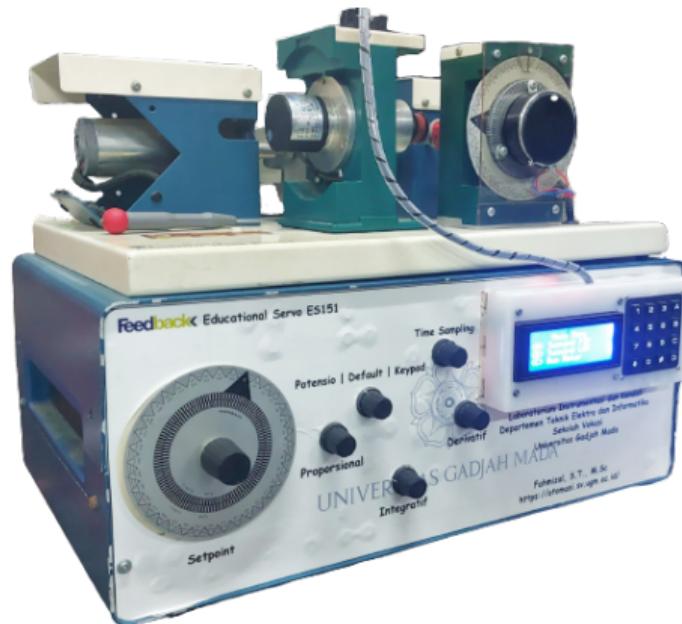
- 1 Pengenalan Sistem
- 2 Pemodelan Sistem Motor DC
- 3 Perancangan Kendali PID Motor DC
- 4 Simulasi Kendali PID
- 5 Sistem Mekanikal dan Elektrikal
- 6 *Graphical User Interface (GUI)*
- 7 Data Hasil Percobaan
- 8 Penutup

# Pengenalan Sistem

# Feedback Educational Servo ES151

*Feedback Educational Servo ES151* merupakan sebuah modul motor DC servo yang digunakan untuk edukasi dalam penerapan kendali posisi dan kendali kecepatan.

Pada modul ini potensiometer digunakan untuk kendali posisi, tachogenerator digunakan untuk kendali kecepatan, dan *rotary encoder* yang dapat digunakan untuk kendali posisi maupun kecepatan.



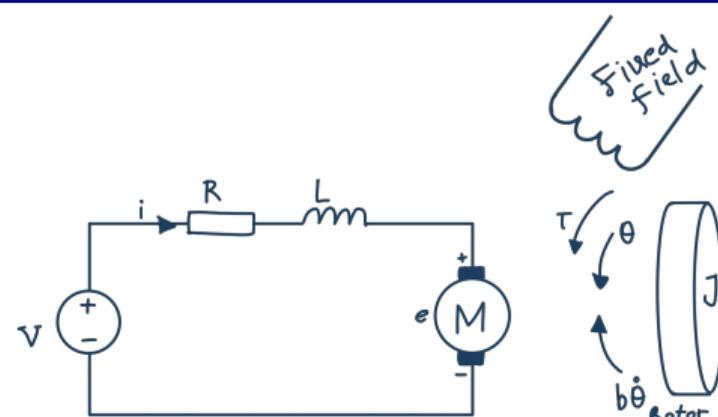
# Pemodelan Sistem Motor DC

# Struktur Fisik

Parameter:

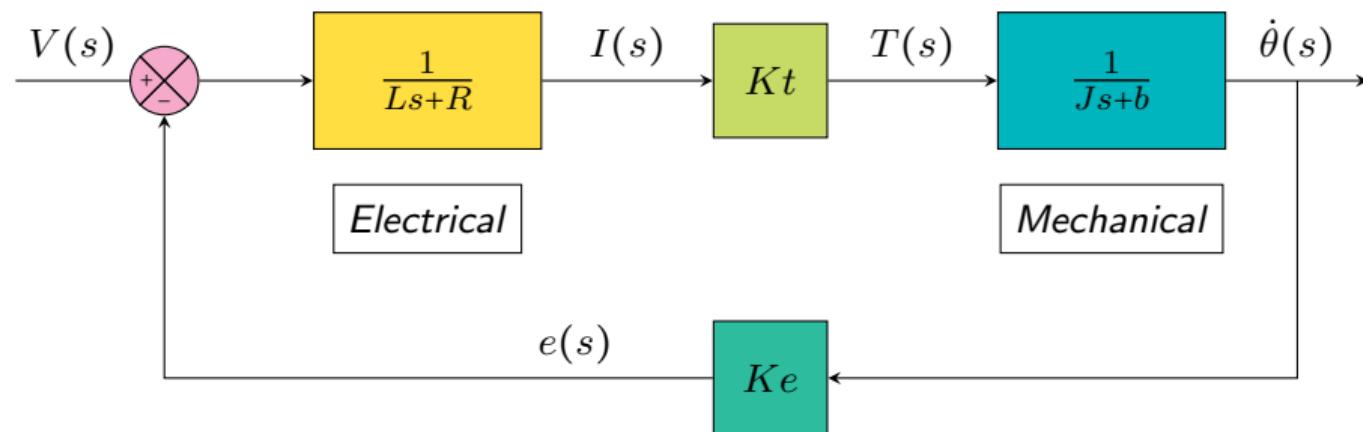
- $J$  : Momen inersia rotor ( $Kg.m^2$ )
- $b$  : Koefisien gaya gesek viskos ( $N.m.s$ )
- $Ke$  : Koefisien gaya elektromotif ( $V/rad/sec$ )
- $Kt$  : Koefisien torsi motor ( $N.m/Amp$ )
- $R$  : Resistansi kumparan ( $Ohm$ )
- $L$  : Induktansi kumparan ( $H$ )

Struktur:



# Diagram Blok Plant Motor DC

Struktur motor DC dengan parameter-parameter sebelumnya memiliki diagram blok sebagai berikut:



# Fungsi Alih

Diagram blok *plant* motor DC menghasilkan persamaan fungsi alih berikut:

$$\frac{\dot{\theta}(s)}{V(s)} = \frac{Kt}{(Js + b)(Ls + R) + KtKe} \quad \left[ \frac{\text{rad/sec}}{V} \right] \quad (1)$$

Persamaan di atas merupakan fungsi alih kecepatan motor DC. Dengan mengintegralkan fungsi alih tersebut, maka diperoleh fungsi alih untuk posisi motor DC:

$$\frac{\theta(s)}{V(s)} = \frac{Kt}{s((Js + b)(Ls + R) + KtKe)} \quad \left[ \frac{\text{rad}}{V} \right] \quad (2)$$

# State Space

Kecepatan:

$$\frac{d}{dt} \begin{bmatrix} \dot{\theta} \\ i \end{bmatrix} = \begin{bmatrix} -\frac{b}{J} & \frac{Kt}{J} \\ -\frac{Ke}{L} & -\frac{R}{L} \end{bmatrix} \begin{bmatrix} \dot{\theta} \\ i \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{L} \end{bmatrix} V$$

$$y = [1 \quad 0] \begin{bmatrix} \dot{\theta} \\ i \end{bmatrix} \quad (3)$$

Posisi:

$$\frac{d}{dt} \begin{bmatrix} \theta \\ \dot{\theta} \\ i \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & -\frac{b}{J} & \frac{Kt}{J} \\ 0 & -\frac{Ke}{L} & -\frac{R}{L} \end{bmatrix} \begin{bmatrix} \theta \\ \dot{\theta} \\ i \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \frac{1}{L} \end{bmatrix} V$$

$$y = [1 \quad 0 \quad 0] \begin{bmatrix} \theta \\ \dot{\theta} \\ i \end{bmatrix} \quad (4)$$

# Perancangan Kendali PID Motor DC

# Apa Itu Kendali PID?

- **PID=Proportional-Integral-Derivative**
- Kendali mekanisme umpan balik yang biasanya dipakai pada sistem kontrol industri
- Secara kontinu menghitung nilai kesalahan sebagai beda antara setpoint yang diinginkan dan variabel proses terukur. Persamaan:

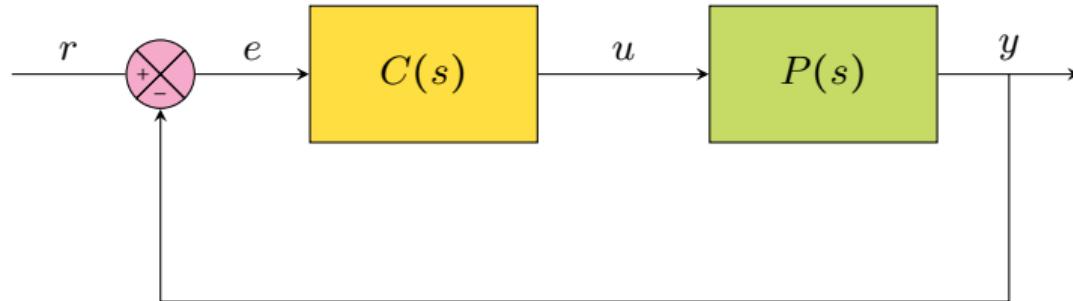
$$u(t) = K_p e(t) + K_i \int_0^t e(t) dt + K_d \frac{de(t)}{dt} \quad (5)$$

# Mengapa Kendali PID?

Kendali PID berfungsi untuk meminimalkan nilai kesalahan (*error*) setiap waktu dengan penyetelan variabel kontrol, seperti posisi, kecepatan, damper, daya, dan lain sebagainya.

Contoh perbandingan sistem dengan dan tanpa PID:

# Diagram Blok Kendali



## Keterangan:

$C(s)$  : Controller

$P(s)$  : Plant

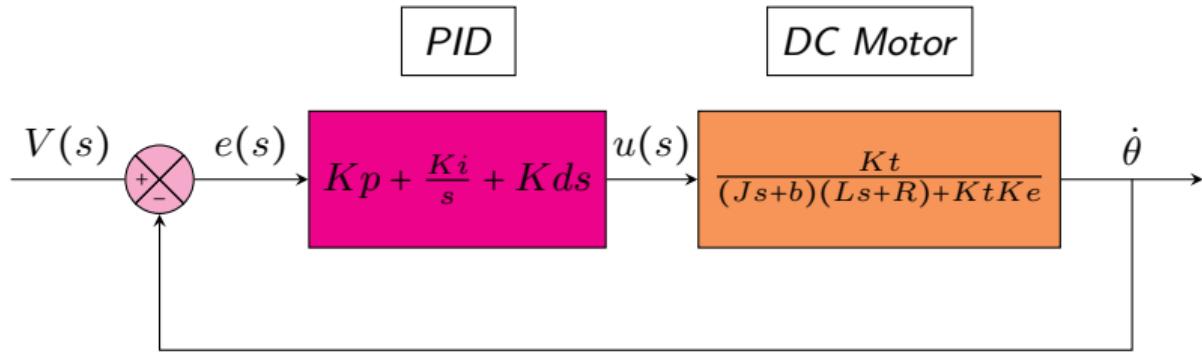
$r(s)$  : Set Point

$e(s)$  : Nilai error

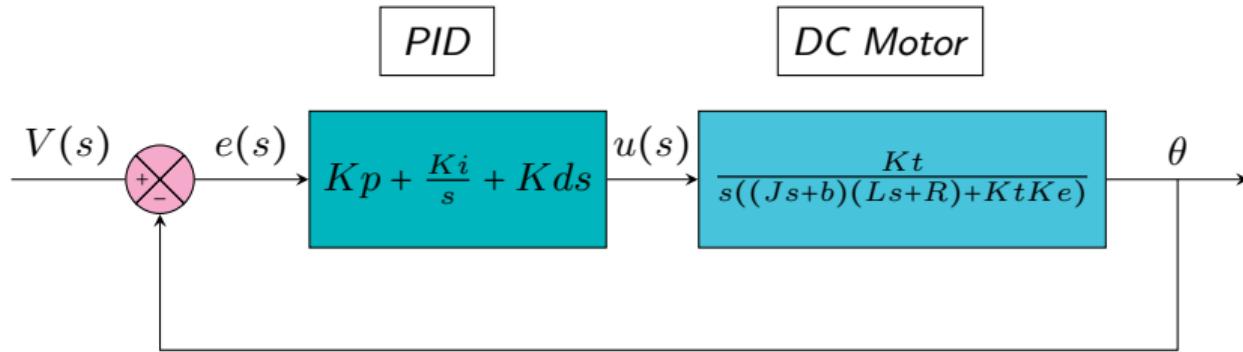
$u(s)$  : Sinyal kendali

$y(s)$  : Output sesungguhnya

# Diagram Blok Kendali PID Motor DC: Kecepatan



# Diagram Blok Kendali PID Motor DC: Posisi



# Simulasi Kendali PID

# Uji Perbandingan Sistem *Open-loop* dengan *Closed-loop* Motor DC

## Program *Open-loop*:

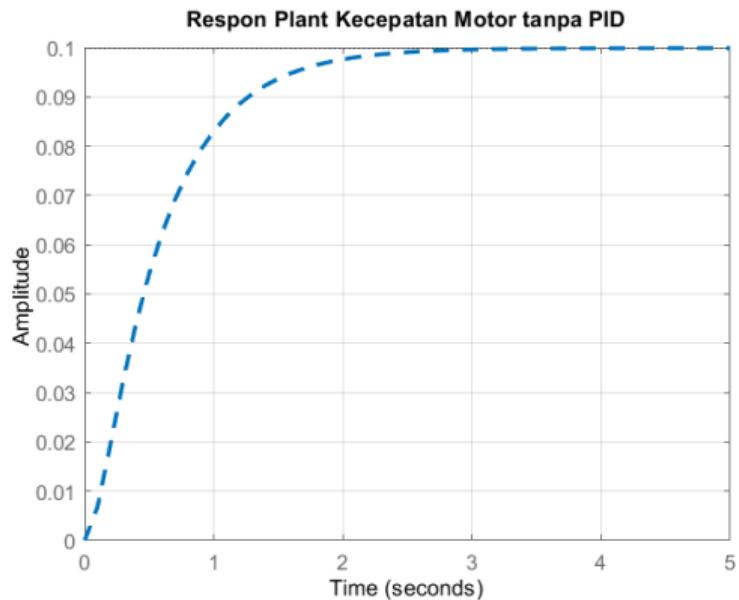
```
J = 0.01;
b = 0.1;
Kt = 0.01;
Ke = 0.01;
R = 1;
L = 0.5;
s = tf('s');
P_motor = Kt/((J*s+b)*(L*s+R)+Kt*Ke);
rP_motor = 0.1/(0.5*s+1)
step(P_motor, 0:0.1:5);
grid
title('Respon Plant Kecepatan Motor tanpa PID')
```

## Program *Closed-loop*:

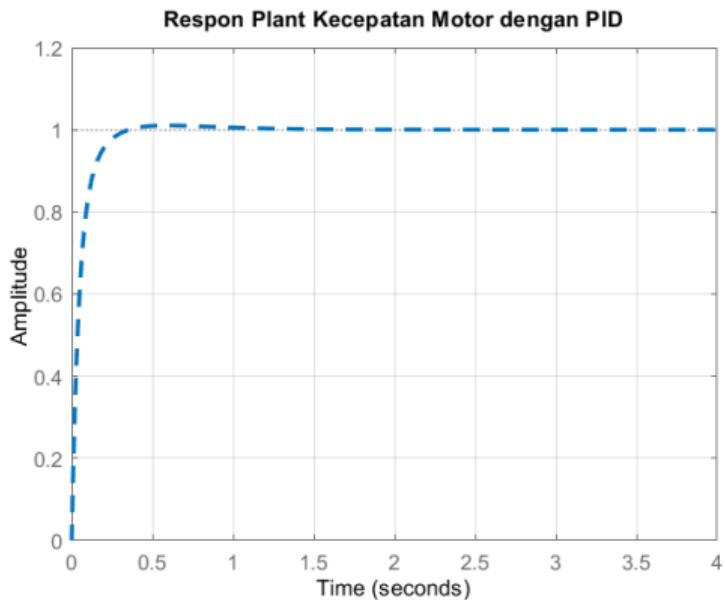
```
J = 0.01;
b = 0.1;
Kt = 0.01;
Ke = 0.01;
R = 1;
L = 0.5;
s = tf('s');
P_motor = Kt/((J*s+b)*(L*s+R)+Kt*Ke);
Kp = 100;
Ki = 200;
Kd = 10;
C = pid(Kp,Ki,Kd);
sys_cl = feedback(C*P_motor,1);
step(sys_cl, 0:0.01:4)
grid
title('Respon Plant Kecepatan Motor dengan PID')
```

# Perbandingan Respon Sistem *Open-loop* dengan *Closed-loop* pada Motor DC

Hasil *Open-loop*:



Hasil *Closed-loop*:



# Kendali PID: Posisi

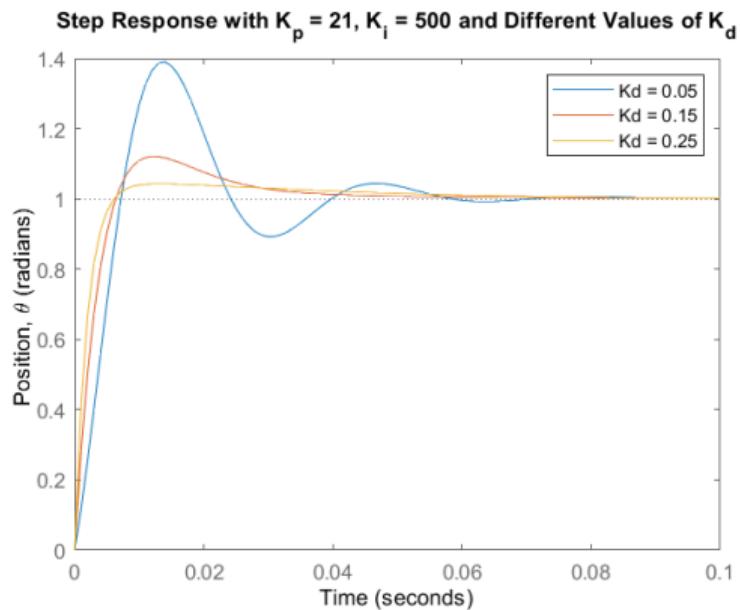
Program:

```
J = 3.2284E-6;
b = 3.5077E-6;
K = 0.0274;
R = 4;
L = 2.75E-6;
s = tf('s');
P_motor = K/(s*((J*s+b)*(L*s+R)+K^2));
Kp = 21;
Ki = 500;
Kd = 0.05;

for i = 1:3
    C(:,:,i) = pid(Kp,Ki,Kd);
    Kd = Kd + 0.1;
end

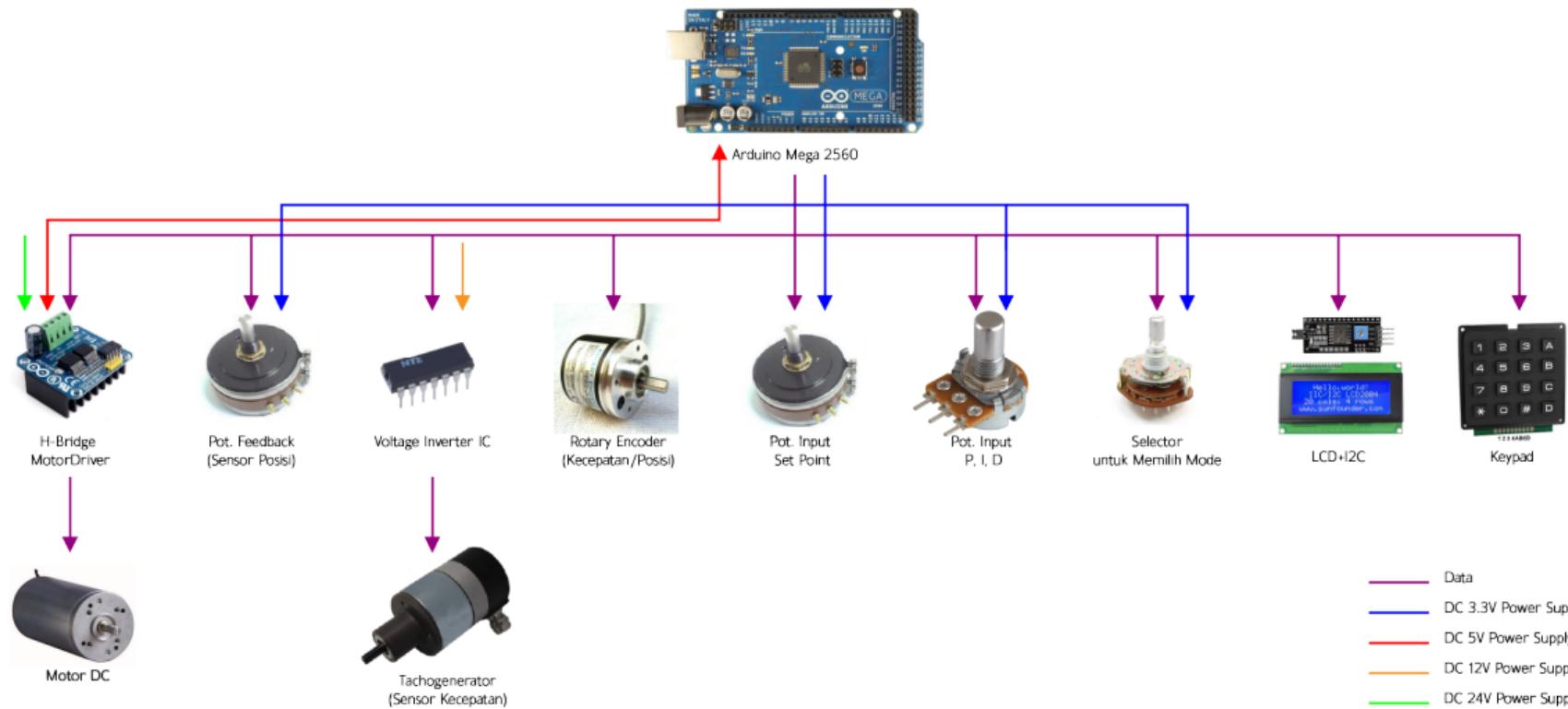
sys_cl = feedback(C*P_motor,1);
t = 0:0.001:0.1;
step(sys_cl(:,:,1), sys_cl(:,:,2), sys_cl(:,:,3), t)
ylabel('Position, \theta (radians)')
title('Step Response with K_p = 21, K_i = 500 and
        Different Values of K_d')
legend('Kd = 0.05', 'Kd = 0.15', 'Kd = 0.25')
```

Hasil:

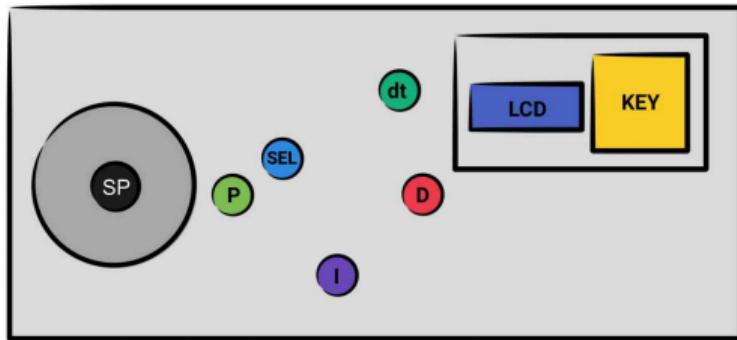


# Sistem Mekanikal dan Elektrikal

# Diagram Sistem



# Panel Kendali

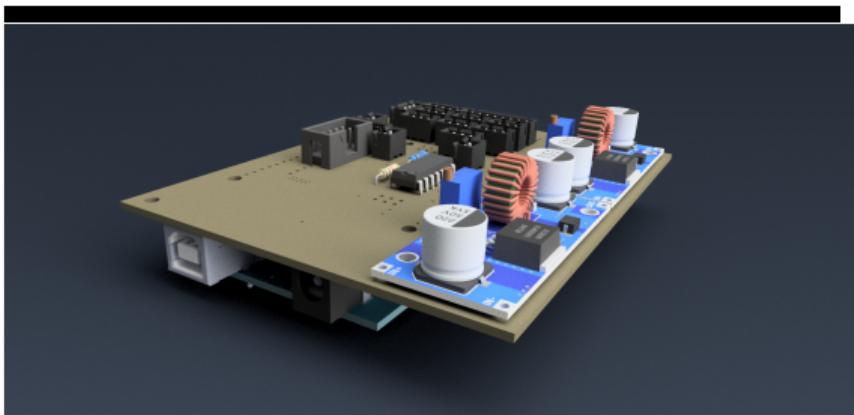


Kontroler

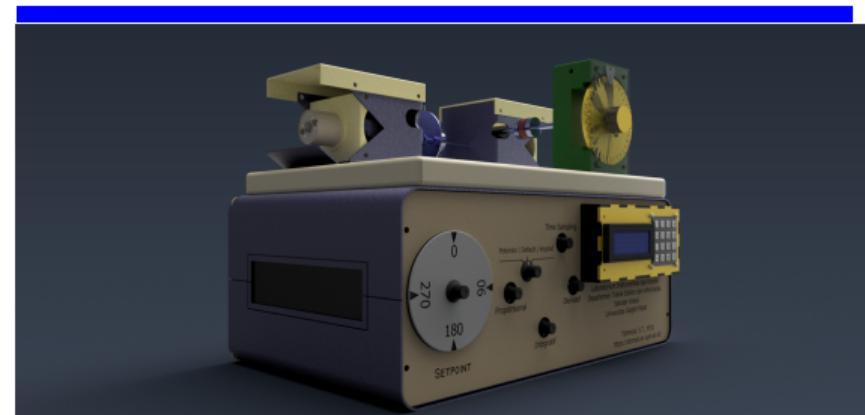
Input-Output	Fungsi	Pin
SP	Set Poin	A7
P	Proporsional	A5
I	Integral	A8
D	Derivatif	A6
SEL	<i>Selector</i>	34, 32, 30
dt	<i>Time Sampling</i>	A4
LCD	LCD	0, 1
KEY	Keypad	52, 50, 48, 46, 44, 42, 40, 38

# Desain *Hardware*

## Desain Elektronis

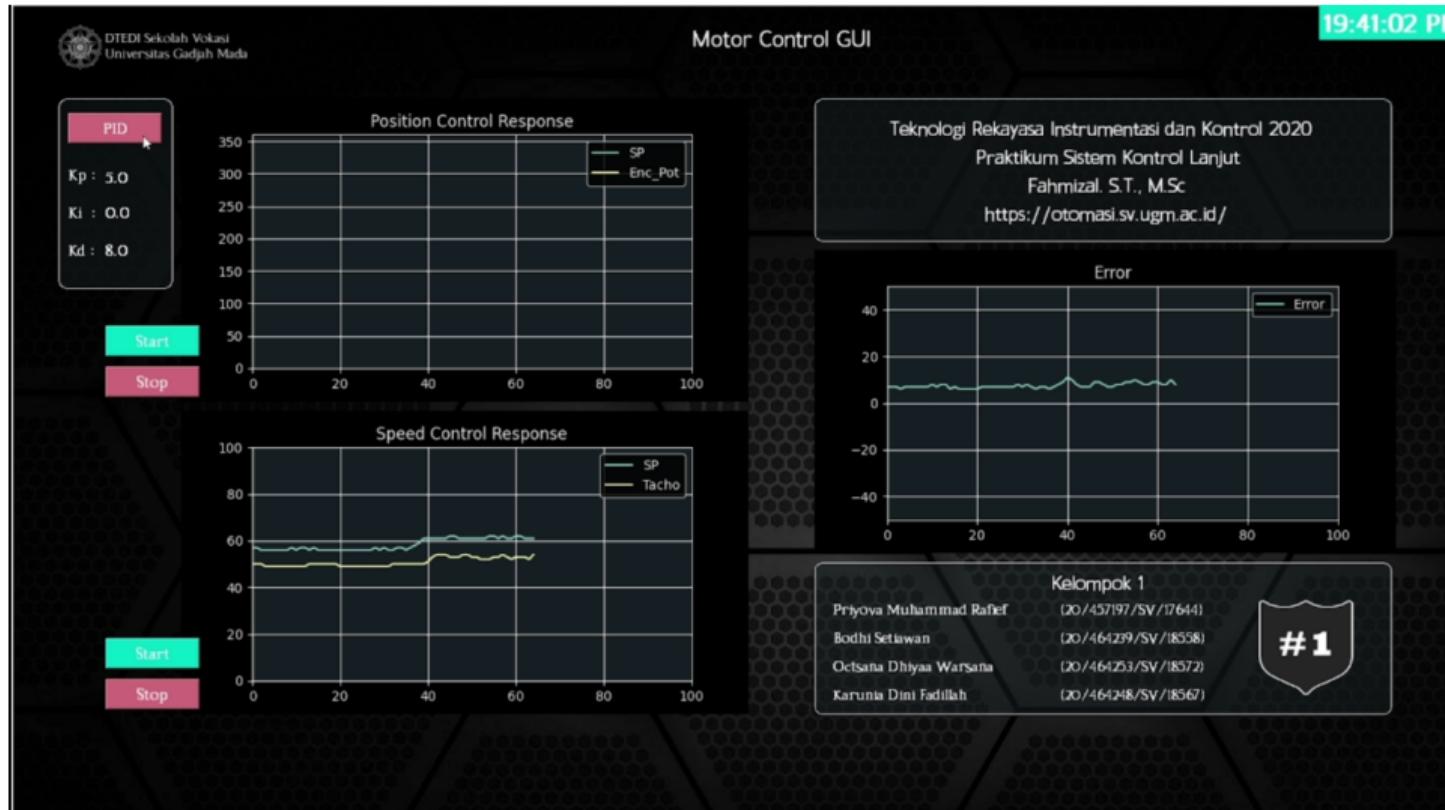


## Desain Mekanis



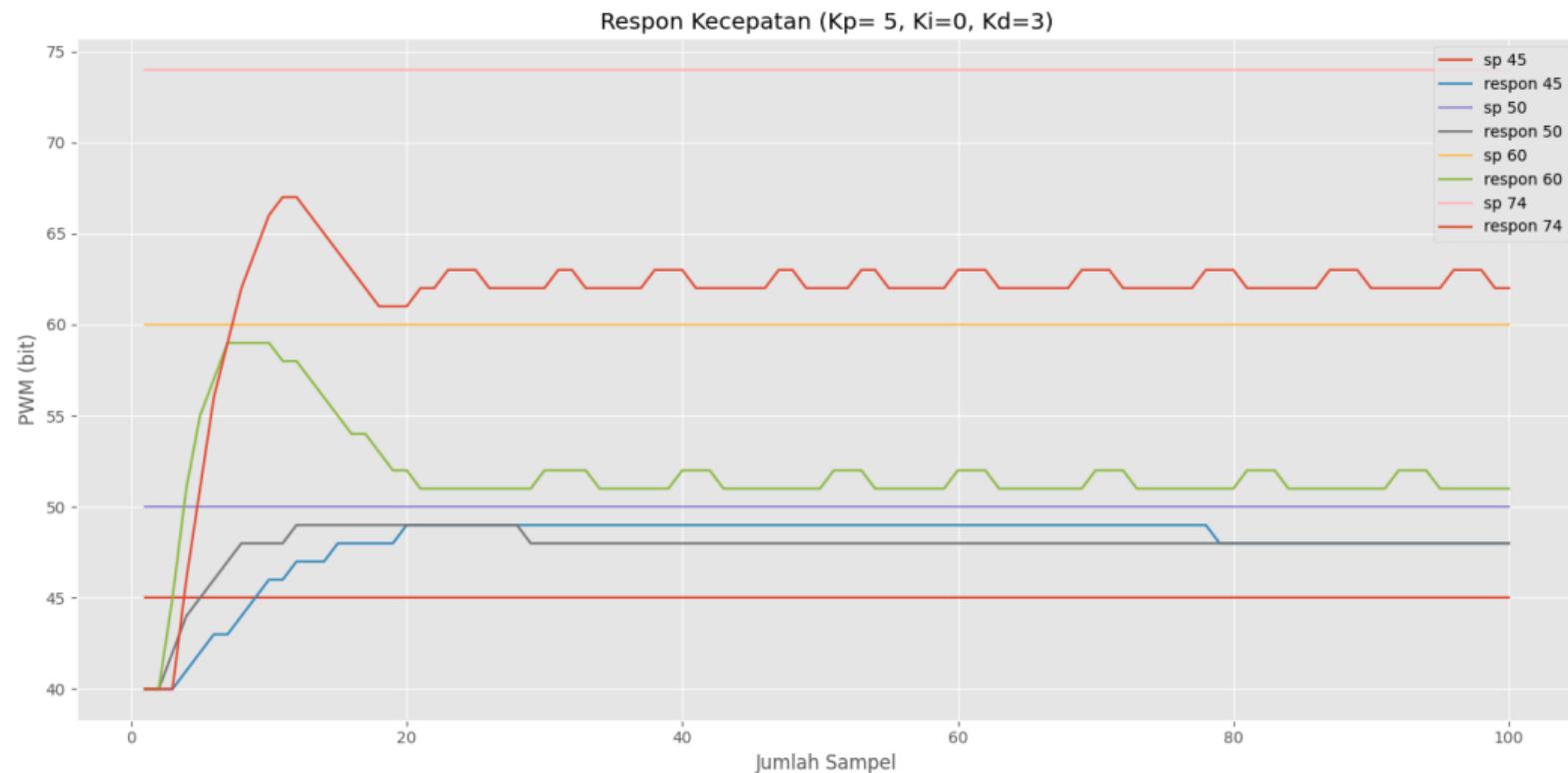
# *Graphical User Interface (GUI)*

# GUI Menggunakan Python

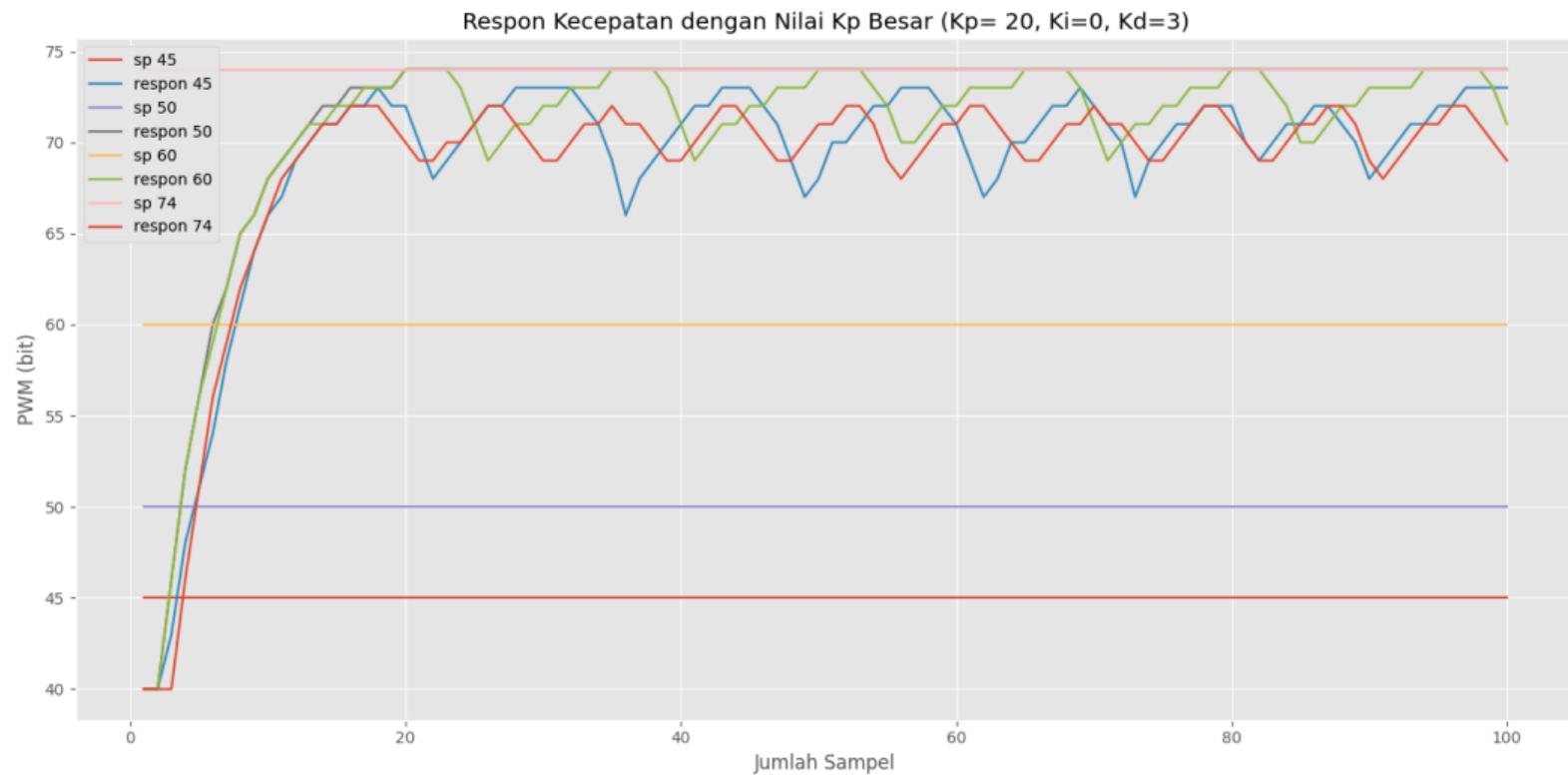


# Data Hasil Percobaan

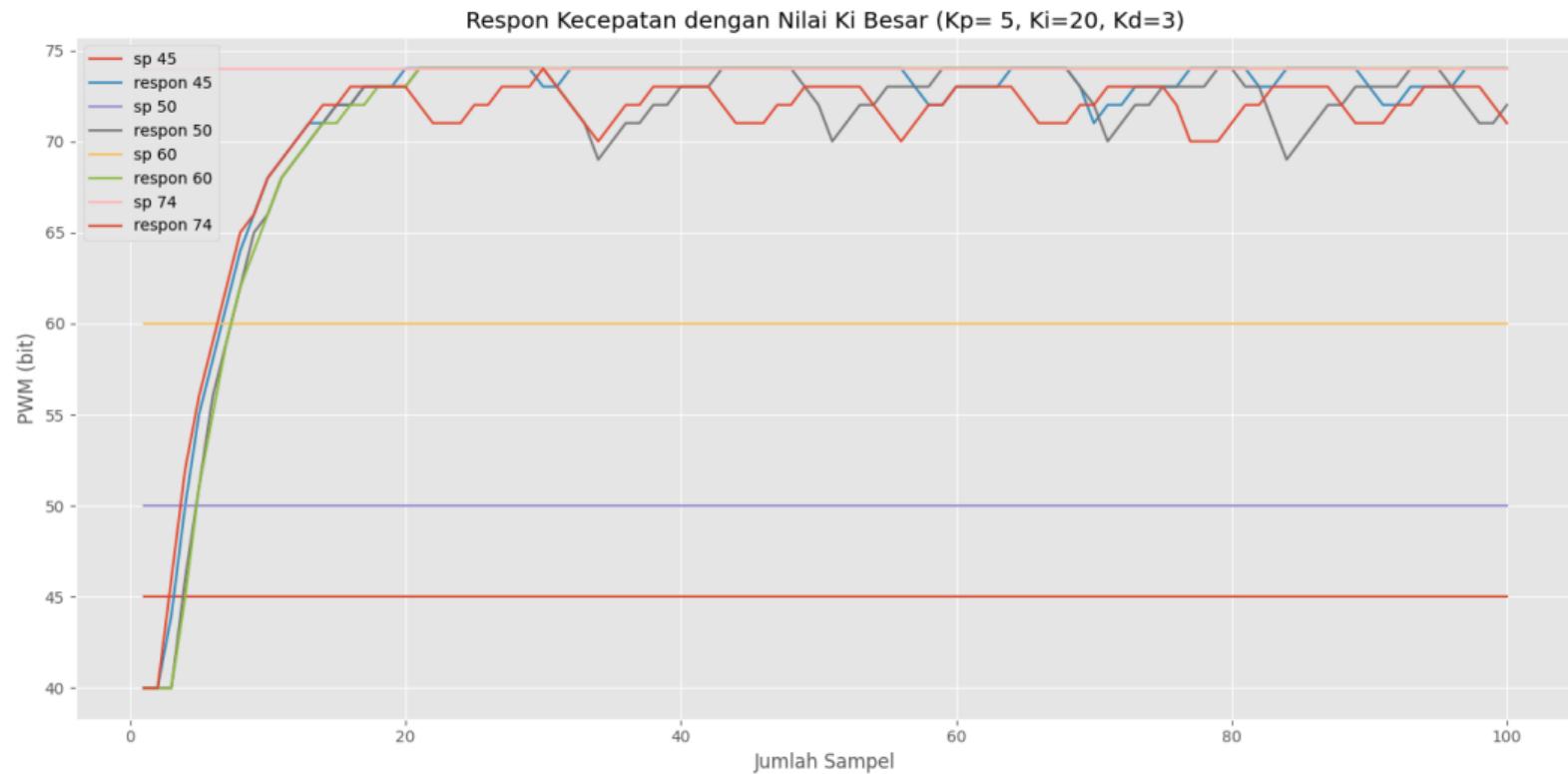
# Percobaan Kontroler Kecepatan Motor DC



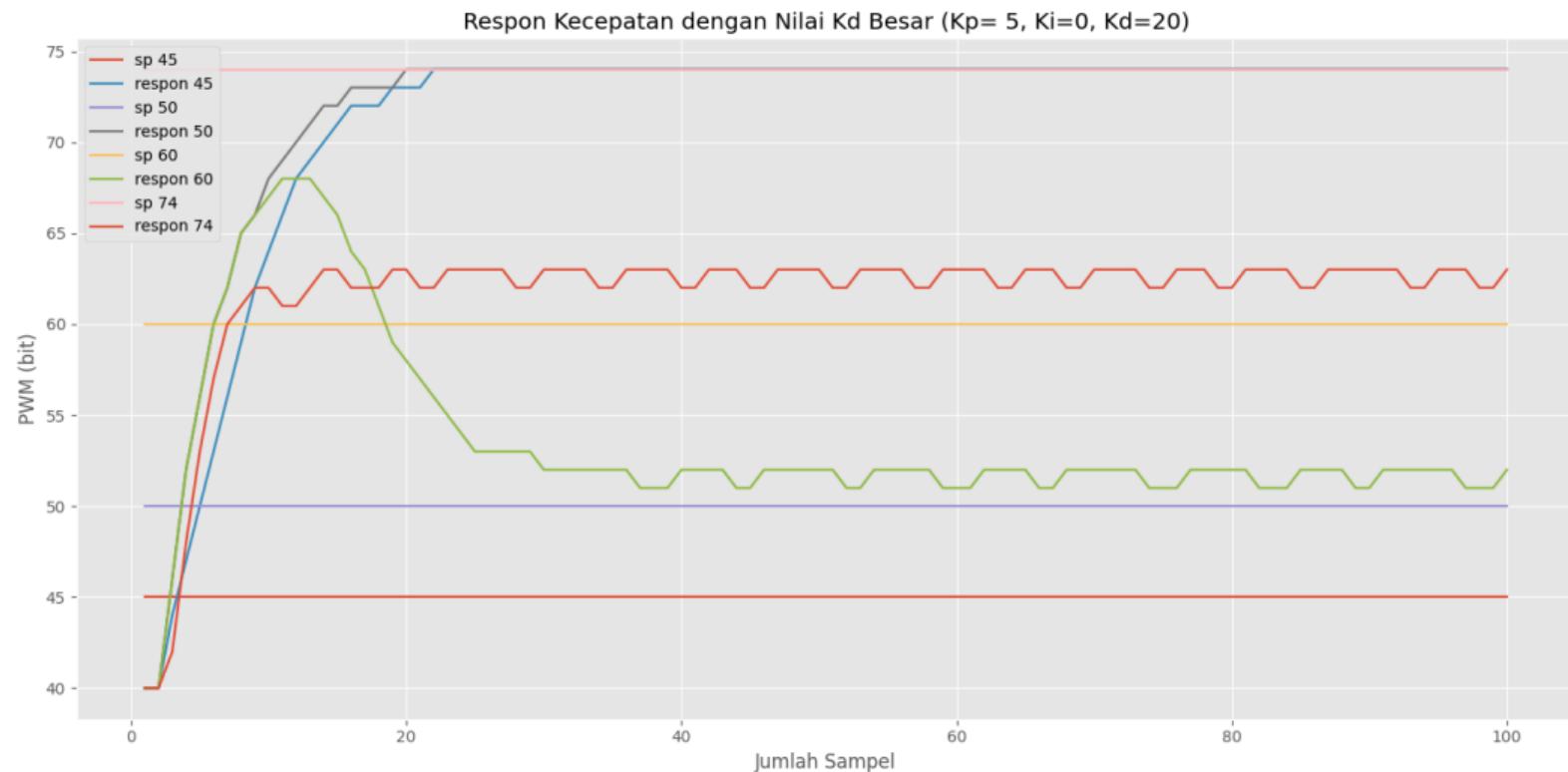
# Percobaan Kontroler Kecepatan Motor DC



# Percobaan Kontroler Kecepatan Motor DC



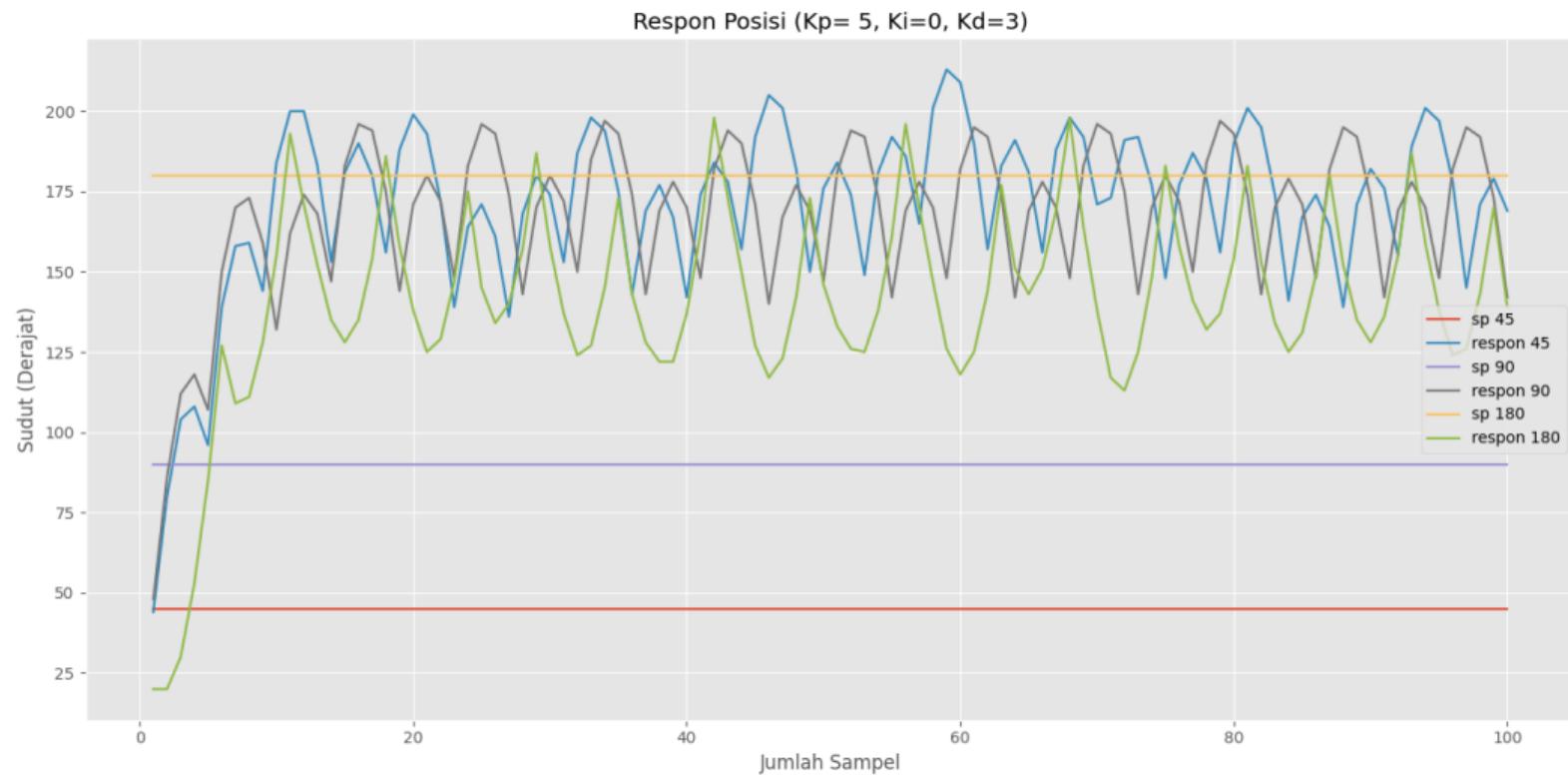
# Percobaan Kontroler Kecepatan Motor DC



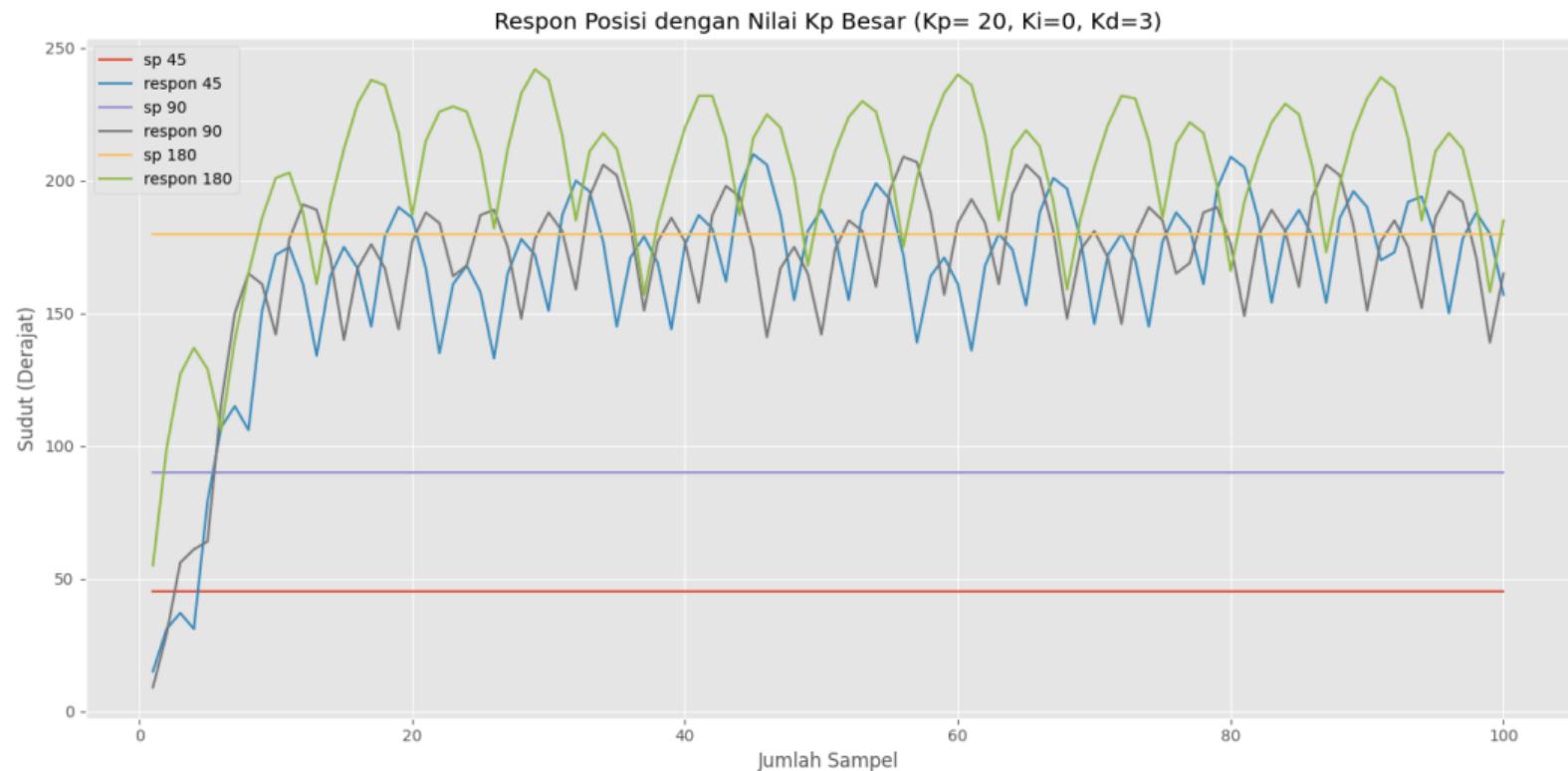
# Nilai RMSE dari Respon Kendali Kecepatan (*Potentiometer Feedback*)

$K_p, K_i, K_d$	Set Point (PWM)	RMSE
5, 0, 3	45	3.6193
	50	2.5903
	60	8.6931
	74	13.2860
20, 0, 3	45	24.8897
	50	22.8576
	60	12.3036
	74	8.2987
5, 20, 3	45	27.2939
	50	21.7294
	60	13.6198
	74	6.98641
5, 0, 20	45	27.3744
	50	22.8576
	60	8.2243
	74	13.0831

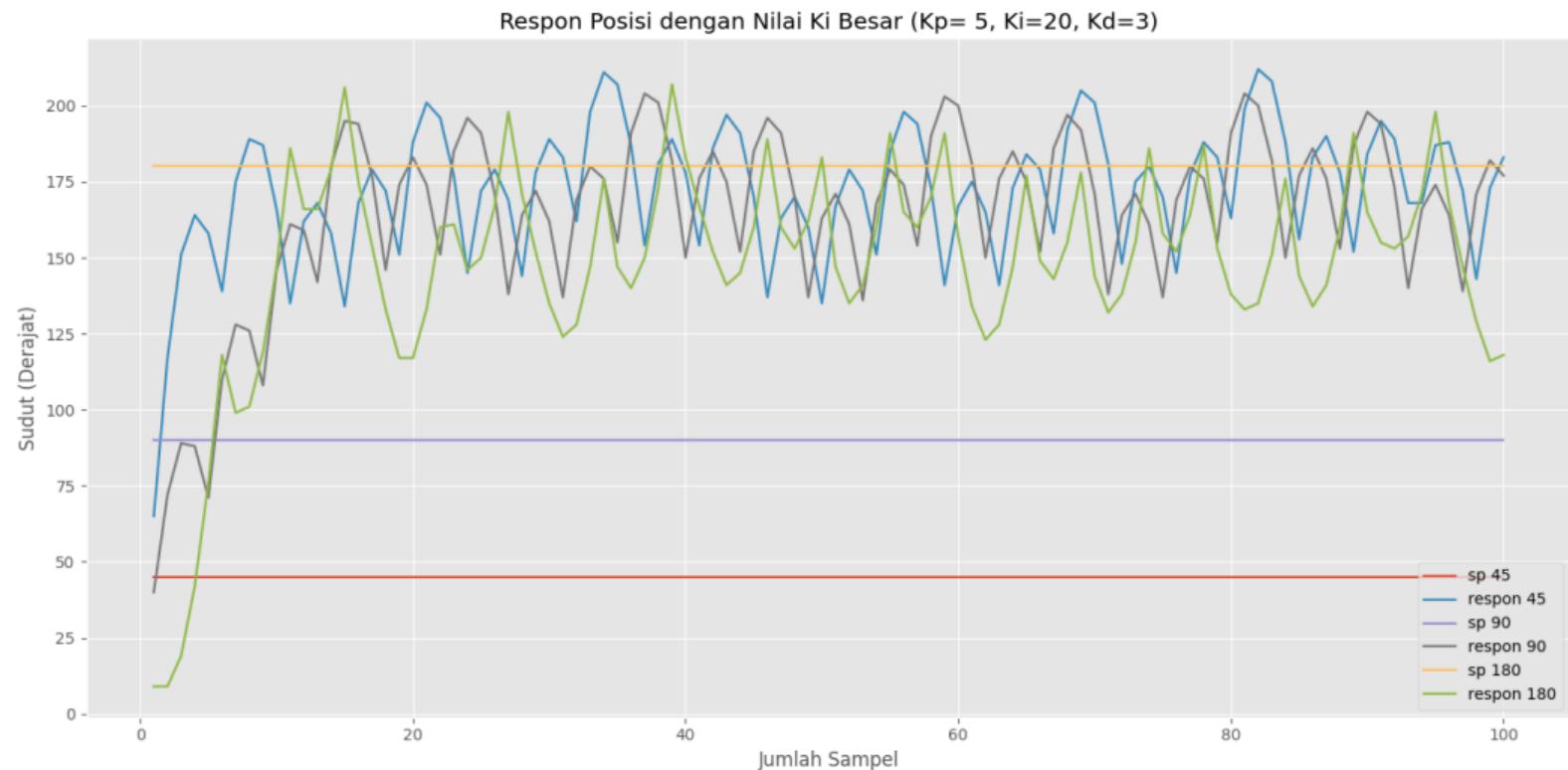
# Percobaan Kontroler Posisi Motor DC



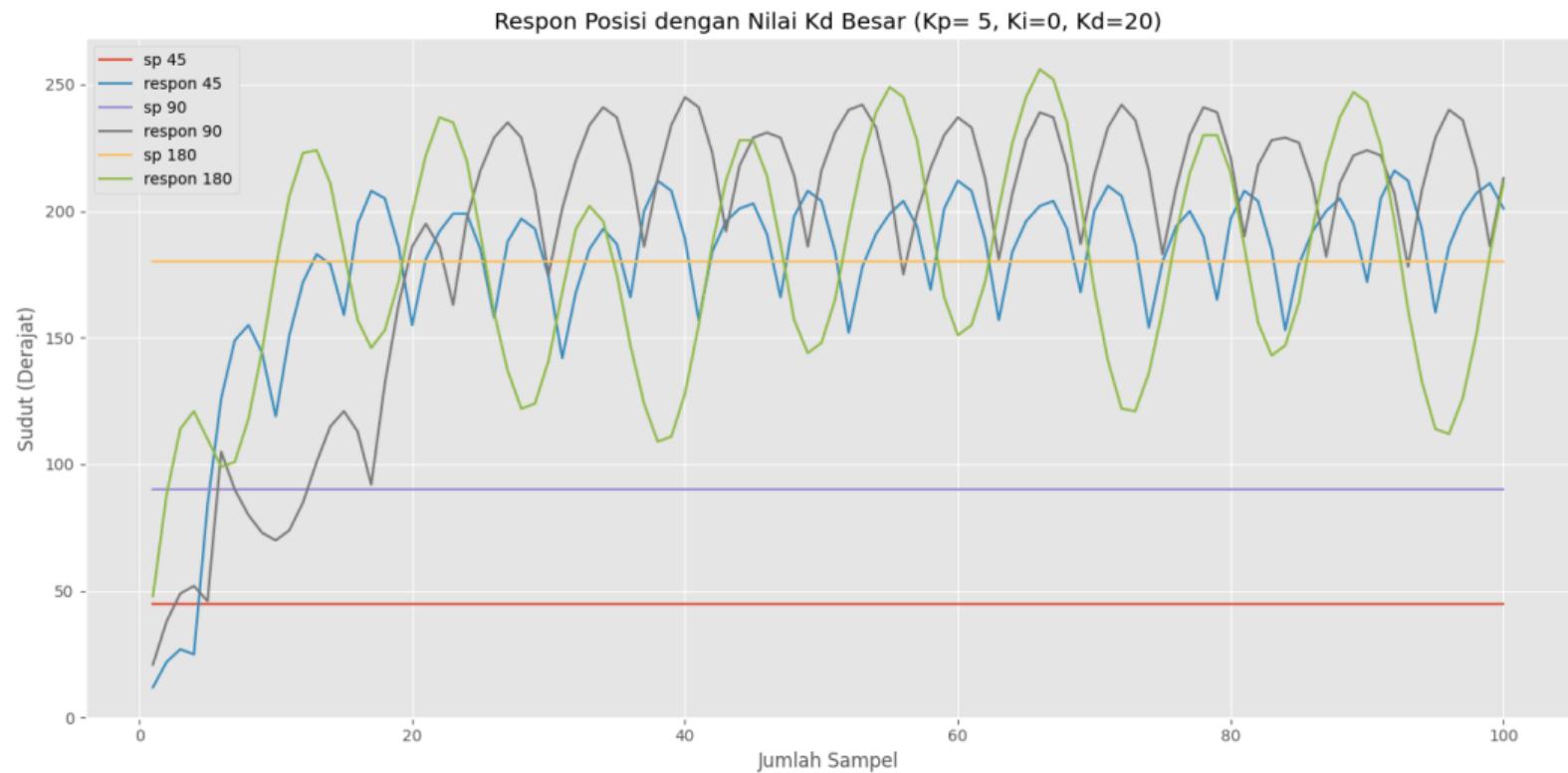
# Percobaan Kontroler Posisi Motor DC



# Percobaan Kontroler Posisi Motor DC



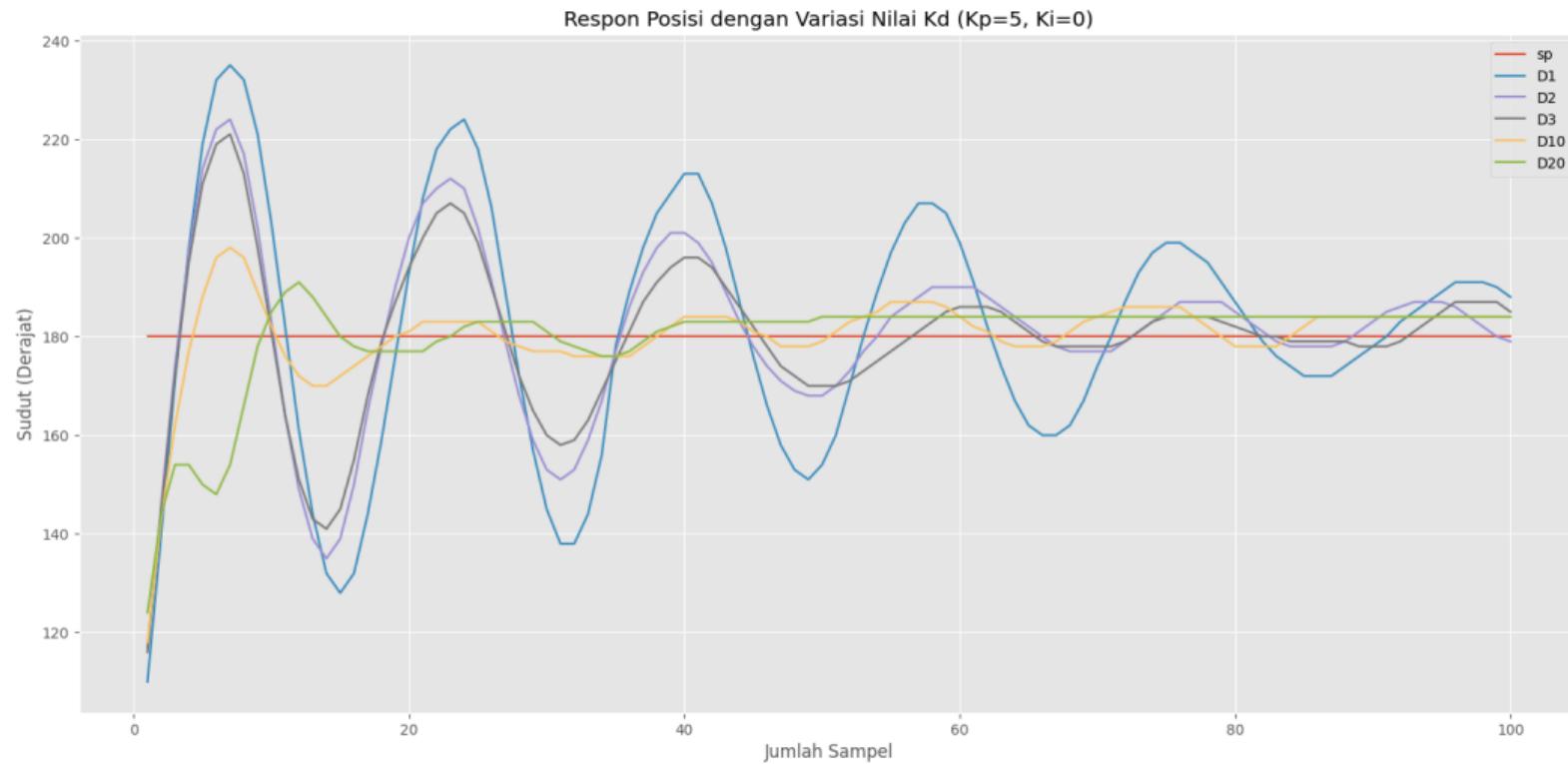
# Percobaan Kontroler Posisi Motor DC



# Nilai RMSE dari Respon Kendali Posisi (*Potentiometer Feedback*)

$K_p, K_i, K_d$	Set Point	RMSE
5, 0, 3	45°	128.8989
	90°	81.5136
	180°	50.0722
20, 0, 3	45°	125.3849
	90°	86.0077
	180°	39.1094
5, 20, 3	45°	129.0604
	90°	80.7569
	180°	47.6050
5, 0, 20	45°	138.9283
	90°	116.1425
	180°	45.4912

# Respon Kontroler Posisi Motor DC dengan Gangguan dan Variasi Nilai Kd



# Nilai RMSE dari Respon Kendali Posisi dengan Gangguan dan Variasi Nilai Kd (*Potentiometer Feedback*)

$K_p, K_i, K_d$	Set Point	RMSE
5, 0, 1	180°	24.9295
5, 0, 2	180°	17.9092
5, 0, 3	180°	15.7003
5, 0, 10	180°	8.9520
5, 0, 20	180°	9.9789

# Penutup

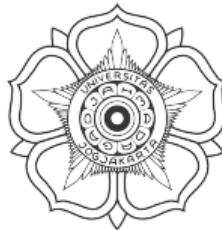
# Kesimpulan

Dari percobaan-percobaan dan pembahasan-pembahasan yang telah dilakukan, diperoleh beberapa kesimpulan yaitu:

- ➊ Kontroler yang sebelumnya sirkuit yang bersifat analog dirombak menjadi digital dengan Arduino Mega 2560. Instrumen mampu menerapkan kendali PID posisi dan kecepatan pada motor DC, namun pada kendali kecepatan masih memiliki kekurangan berupa rentang PWM yang digunakan yaitu sebesar 44-74 yang mana harusnya adalah 0-255.
- ➋ GUI yang didesain khusus untuk instrumen ini mampu menampilkan data berupa grafik respon dan *error*.
- ➌ Pengembangan GUI menggunakan Python memiliki potensi keberlanjutan yang sangat besar karena *library-library* yang dapat digunakan sangat beragam. Oleh karena itu, GUI dapat ditingkatkan lagi dengan berbagai fitur baru sesuai dengan kebutuhan.

# Daftar Pustaka

- Control Tutorial for Matlab & Simulink. <https://ctms.engin.umich.edu/CTMS/index.php?example=Introduction&section=SystemModeling>
- Proportional-Integral-Derivative (PID) Controllers. <https://www.mathworks.com/help/control/ug/proportional-integral-derivative-pid-controllers.html>



Visit us on <https://otomasi.sv.ugm.ac.id/>