

BAB IV

PERANCANGAN DAN PEMBUATAN ALAT

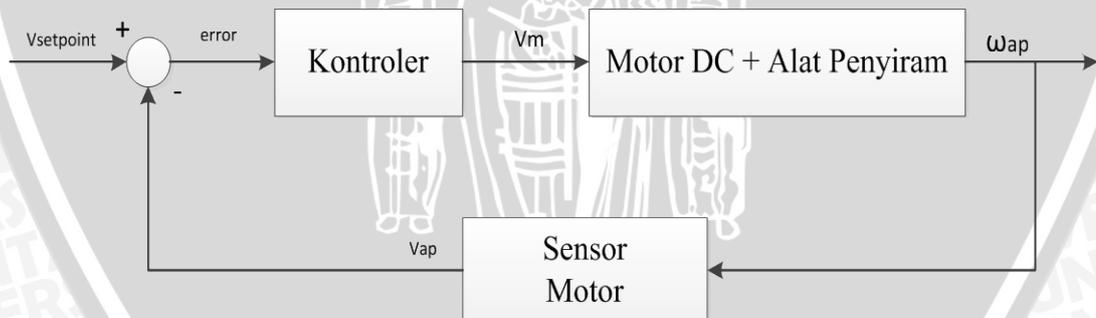
4.1 Perancangan Sistem

Perancangan dan pembuatan alat ini terdiri dari dua bagian, yaitu perancangan dan pembuatan perangkat keras serta perancangan dan pembuatan perangkat lunak. Perancangan dan pembuatan alat dilakukan secara bertahap untuk memudahkan analisis sistem. Beberapa aspek yang perlu dijelaskan dalam bab ini meliputi:

- Cara kerja alat.
- Spesifikasi alat.
- Perancangan perangkat keras terdiri atas perancangan elektrik dan mekanik alat penyiram, *Soil Moisture Sensor* YL69/LM393, rel dan mediator tanah.
- Perancangan perangkat lunak terdiri atas perancangan algoritma kontrol PID pada *software arduino ERW 1.0.5*.

4.1.1 Blok Diagram Sistem

Blok diagram sistem yang dirancang ditunjukkan dalam Gambar 4.1.



Gambar 4.1 Blok Diagram Sistem

Keterangan dari blok diagram diatas adalah sebagai berikut :

- *Soil Moisture Sensor* berfungsi sebagai pendeteksi kelembaban air dalam tanah saat itu.

- Pusat pengendalian sistem menggunakan Arduino Uno Rev3 yang memberikan sinyal keluaran berupa *duty cycle* PWM kepada *driver* motor.
- Motor yang digunakan adalah 2 (dua) buah motor DC yang berfungsi sebagai aktuator. *Driver* motor menggunakan modul EMS 5A H-Bridge.
- Perancangan alat berupa miniatur tanpa skala pada proses penyiraman tanaman sehingga pompa dan selang yang digunakan bertekanan dan berukuran lebih kecil.

4.1.2 Prinsip Kerja Sistem

Cara kerja sistem adalah sebagai berikut :

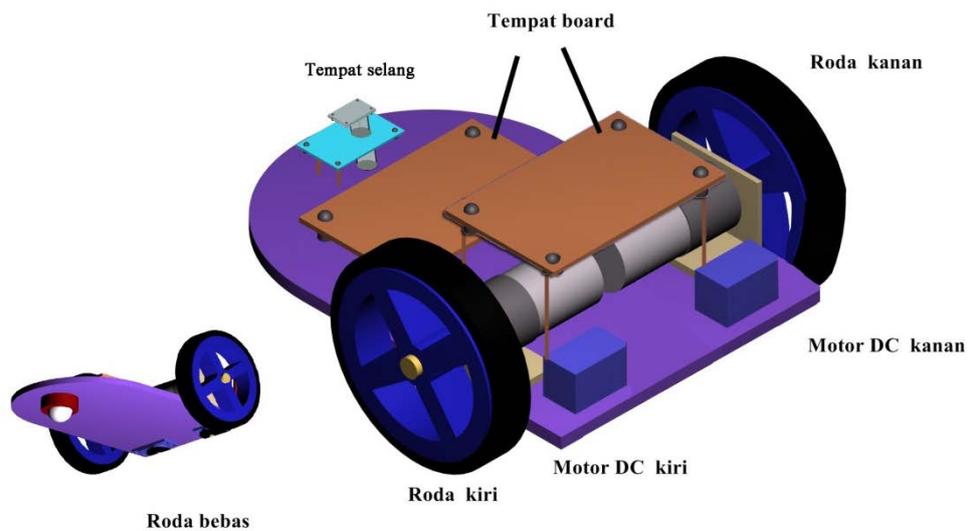
1. Catu daya yang digunakan untuk menggerakkan motor sebesar 12 volt dari baterai lipo. Dan catu daya untuk Arduino Uno Rev3 sebesar 5 volt.
2. *Soil Moisture Sensor* diletakkan di dalam tanah. Sensor ini digunakan sebagai selektor yang akan digunakan untuk menentukan petak mana yang akan disiram.
3. Tahap pertama yaitu ketika sistem pada alat diaktifkan, maka sensor akan mendeteksi kandungan air pada tanah yang terbaca kemudian memberikan masukan ke arduino berupa nilai yang ditampilkan pada serial monitor, dan di titik mana saja sensor itu terdeteksi.
4. Tahap selanjutnya yaitu masukan pada arduino akan diolah. Dan arduino akan memberikan sinyal kontrol berupa PWM ke *driver* motor agar motor dapat berputar dengan *setpoint* yang diinginkan.
5. Saat motor sudah mulai berputar dengan kondisi air selang sudah menyala, maka alat akan berjalan menuju titik yang sudah terdeteksi dengan *setpoint* kecepatan yang sudah ditentukan.
6. Ketika alat sudah mencapai titik sensor yang sudah terdeteksi, maka alat akan berhenti di titik itu dan akan menyiram selama 5 detik.
7. Jika sudah mencapai 5 detik maka alat akan berjalan kembali menuju titik terdeteksi yang lainnya. Jika tidak ada maka alat akan kembali ke kondisi awal.

4.2 Perancangan Perangkat Keras (*Hardware*)

4.2.1 Spesifikasi Alat

Spesifikasi alat yang dirancang adalah sebagai berikut:

1. Alat penyiram beroda 3 dengan ukuran:
 - Tinggi : 14 cm
 - Panjang : 24 cm
 - Lebar : 16,5 cm
2. Mesin penggerak berupa 2 motor DC dengan catu daya 12 volt dan kecepatan maksimal 285 rpm.
3. Lintasan tempat alat berjalan dengan ukuran:
 - Panjang : 340 cm
 - Lebar : 21 cm
4. Sensor yang digunakan untuk mendeteksi kandungan air dalam tanah adalah *Soil Moisture Sensor YL69/LM393* dengan acuan nilai ADC:
 - Kering : > 614
 - Basah : < 614
5. Kontrol yang digunakan adalah kontroler PID.
6. Menggunakan satu buah mikrokontroler Arduino Uno Rev3.
7. *Driver* motor yang digunakan adalah EMS 5A H-Bridge
8. *Software* yang digunakan sebagai pemrograman yaitu Arduino ERW 1.0.5.



Gambar 4.2 Rancangan Alat Penyiram

4.2.2 Soil Moisture Sensor YL69/LM393

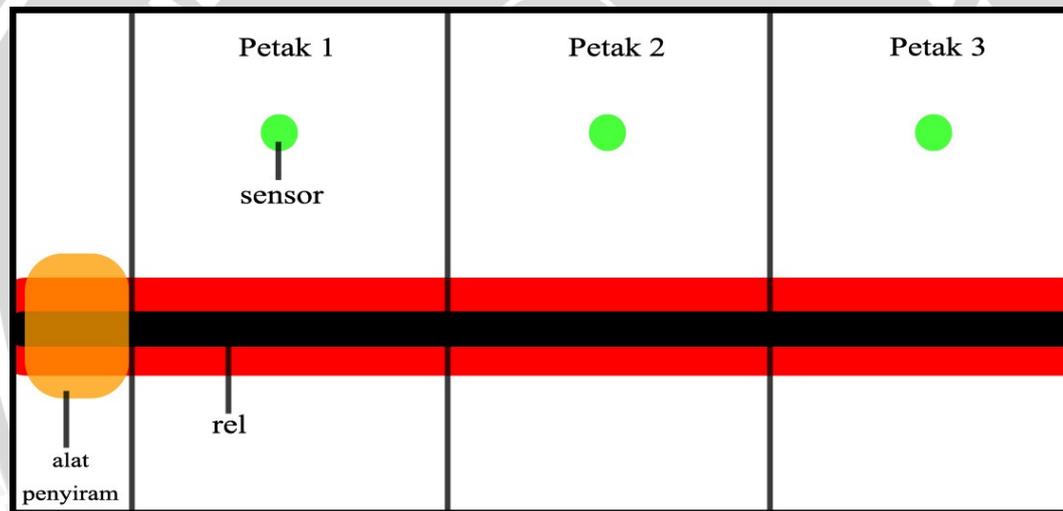
Soil Moisture Sensor YL69/LM393 adalah sensor yang digunakan untuk mendeteksi kadar air dalam tanah, yang kemudian bisa menjadi acuan dalam sistem pengairan / penyiraman tanaman secara otomatis. Penempatan sensor ini tepat di dalam tanah dengan kedalaman dari permukaan tanah adalah 5 cm, sehingga setiap perubahan kandungan air di dalam tanah akan mempengaruhi nilai resistansinya. jarak tiap sensor ke sensor lainnya adalah 150 cm dengan lebar tiap petak adalah :

Petak 1 : 100 cm

Petak 2 : 100 cm

Petak 3 : 100 cm

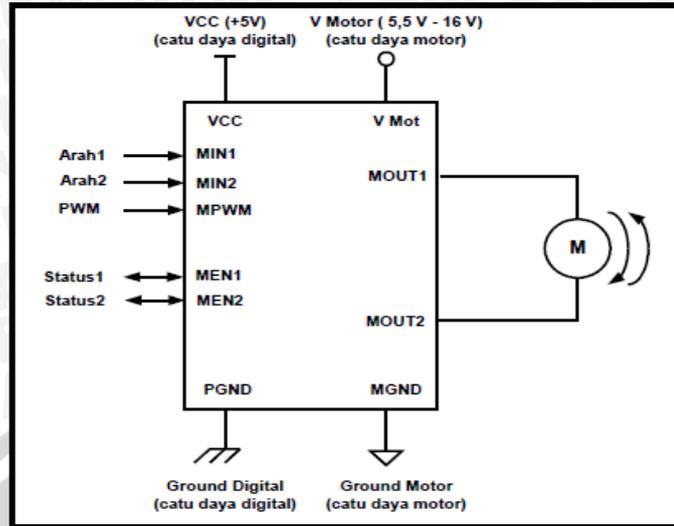
Letak penempatan sensor ditunjukkan pada Gambar 4.3.



Gambar 4.3 Penempatan Sensor

4.2.3 Perancangan Driver Motor DC

Modul pengendali motor DC yang digunakan adalah modul *EMS 5A H-Bridge*. Secara garis besar, fungsi modul pengendali motor ini adalah untuk mengendalikan arah dan kecepatan putaran motor DC sesuai instruksi kendali dari Arduino Uno ERW 1.0.5. Gambar koneksi modul pengendali motor DC *EMS 5A H-Bridge* ditunjukkan dalam Gambar 4.4.



Gambar 4.4 Koneksi Modul EMS 5A H-Bridge

Rangkaian *driver* ini sanggup bekerja dengan tegangan maksimal 40 volt, serta kapasitas arus maksimum yang dapat dilewatkan pada modul ini sebesar 5 ampere. Gambar modul rangkaian driver *EMS 5A H-Bridge* dapat dilihat pada Gambar 4.5.

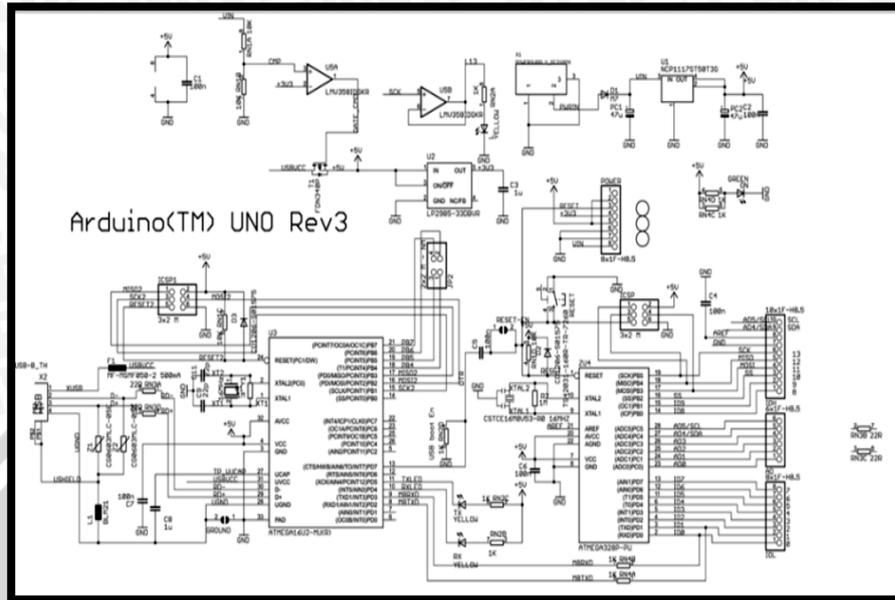


Gambar 4.5 Modul Rangkaian Driver EMS 5A H-Bridge

4.2.4 Modul Arduino Uno Rev.3

Pada sistem pengaturan kecepatan putaran alat penyiraman tanaman ini digunakan Arduino Uno Rev.3 sebagai pengolah data dalam proses pengaturan

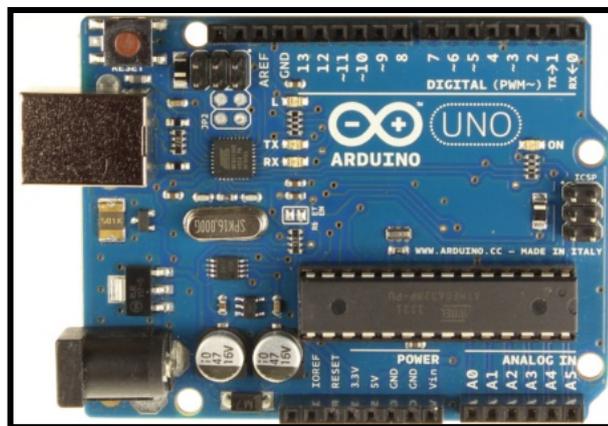
kecepatan terdapat pada Driver motor *H-Bridge* untuk menggerakkan motor DC. Konfigurasi kaki I/O dari Arduino Uno Rev3 ditunjukkan dalam Gambar 4.6.



Gambar 4.6 Desain Sistem Arduino Uno Rev3

Sumber : Datasheet Arduino Rev3

Arduino Uno adalah *board* mikrokontroler berbasis ATmega328. Memiliki 14 pin *input* dari *output* digital dimana 6 pin *input* tersebut dapat digunakan sebagai *output* PWM dan 6 pin *input* analog, 16 MHz osilator kristal, koneksi USB, *jack power*, ICSP *header*, dan tombol *reset*. Untuk mendukung mikrokontroler agar dapat digunakan, cukup hanya menghubungkan *Board* Arduino Uno ke komputer dengan menggunakan kabel USB atau listrik dengan AC ke adaptor DC atau baterai untuk menjalankannya. Modul arduino uno ditunjukkan pada Gambar 4.7.



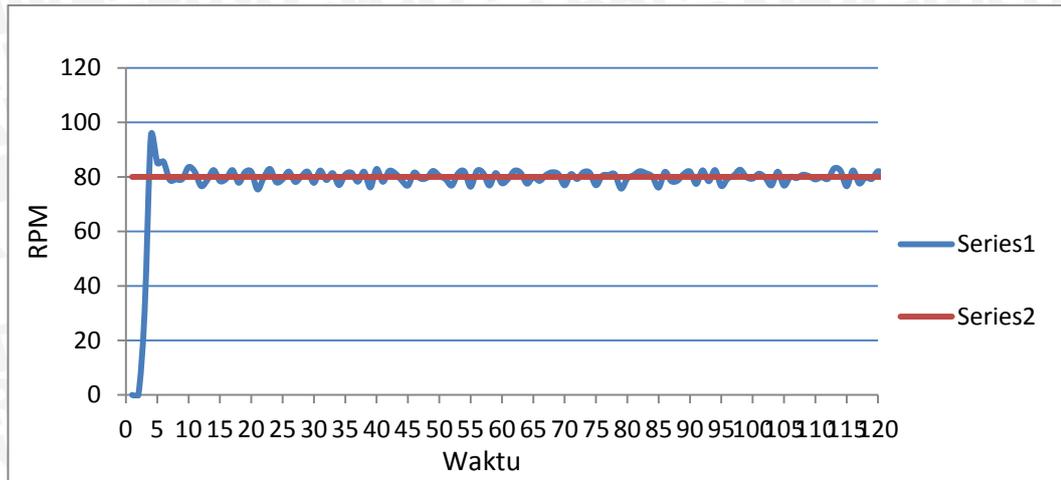
Gambar 4.7 Modul Arduino Uno

Tabel 4.1 Fungsi Pin Arduino Uno Rev3

No	Pin	Fungsi
1	2	Jalur masukan data dari sensor
2	3	LCD pin D6
3	4	LCD pin D5
4	5	LCD pin D4
5	6	Jalur masukan PWM
6	7	Pin input untuk MOUT1
7	8	Pin input untuk MOUT2
8	9	Pin enable untuk MOUT1
9	10	LCD pin D7
10	11	LCD pin Enable
11	12	LCD pin RS
12	5V	Jalur keluaran 5V
13	GND	Jalur keluaran ground

4.3 Perancangan Kontroler PID

Perancangan kontroler dilakukan dengan menggunakan microsoft excel untuk mengamati respon silinder. Langkah pertama adalah dengan melihat respon motor setelah diberi kenaikan $K_{cr} = 35$ seperti dalam Gambar 4.8.



Gambar 4.8 Respon PID Zygler Nichols

Berdasarkan Gambar 4.8, didapatkan nilai P_{cr} sebesar 4. Nilai parameter kontroler PID ditentukan berdasarkan tabel aturan dasar Ziegler-Nichols dengan *critical gain* K_{cr} dan *critical period* P_{cr} yang ditunjukkan dalam Tabel 4.2.

Tabel 4.2 Dasar Ziegler-Nichols Berdasarkan *Critical Gain* K_{cr} dan *Critical Period* P_{cr} (Ogata K., 1997)

Tipe Kontroler	K_p	T_i	T_d
P	$0.5 K_{cr}$	∞	0
PI	$0.45 K_{cr}$	$\frac{1}{1.2} P_{cr}$	0
PID	$0.6 K_{cr}$	$0.5 P_{cr}$	$0.125 P_{cr}$

maka nilai parameter PID yang diperoleh adalah :

$$K_p = 0.6 \times 35 = 21$$

$$T_i = 0.5 \times 4 = 2$$

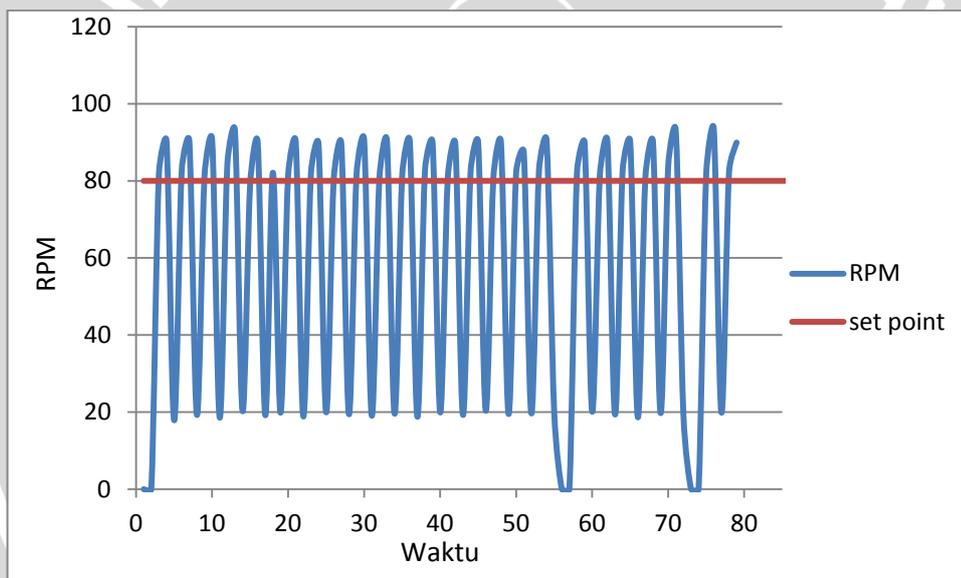
$$T_d = 0.125 \times 4 = 0,5$$

Dengan demikian dapat ditentukan nilai K_p , K_i , dan K_d adalah:

$$K_p = 21$$

$$\begin{aligned}K_i &= \frac{K_p}{T_i} \\ &= 21/2 \\ &= 10,5 \\ K_d &= K_p \times T_d \\ &= 21 \times 0,5 = 10,5\end{aligned}$$

Dari penghitungan penentuan nilai penguatan dari metode kedua Ziegler-Nichols di atas diperoleh $K_p = 21$, $K_i = 10,5$, $K_d = 10,5$ yang akan digunakan untuk pengendali kecepatan motor DC. Dengan nilai K_p , K_i , dan K_d yang sudah diperoleh, maka grafik sistem respon yang diperoleh ditunjukkan oleh Gambar 4.9.



Gambar 4.9 Respon Sistem dengan K_p , K_i , dan K_d Berdasarkan Ziegler-Nichols

Dari grafik Gambar 4.9 bisa dilihat sistem tidak dapat menggunakan perhitungan Ziegler-Nichols tidak diperoleh respon osilasi yang divergen sebagaimana yang di kriteriakan oleh zygler -nichols tipe 2. Maka dilakukan *hand tuning* (*hand* eksperimen) untuk mendapatkan respon yang lebih baik.

Tuning eksperimen adalah proses yang dilakukan untuk mendapatkan hasil kontroler yang optimal dengan cara suatu percobaan. Inti dari *tuning* eksperimen adalah menentukan nilai dari tiga buah parameter yang terdapat pada kontroler PID yaitu konstanta proporsional (K_p), konstanta integral (K_i) dan konstanta

diferensial (Kd). Pada perancangan kontroler PID sistem pengaturan kecepatan ini, menggunakan metode *handtuning* untuk menentukan parameter Kp, Ki, dan Kd.

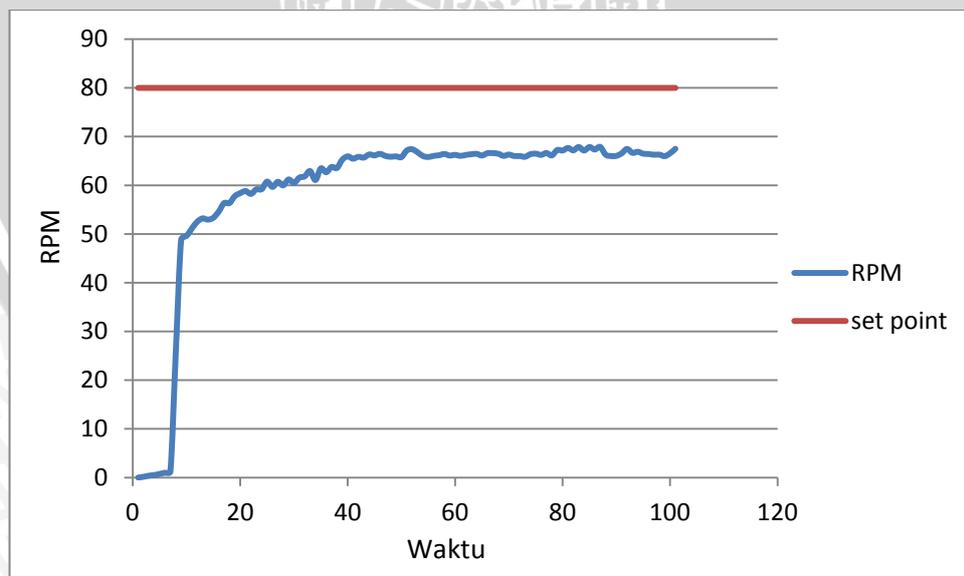
Menurut Smith (1979), untuk melakukan *hand tuning* prosedur yang dilakukan adalah sebagai berikut :

1. Melepaskan kontroler integral dan diferensial dengan memberikan nilai $K_i = 0$ dan $K_d = 0$.
2. Mengatur nilai Kp hingga didapatkan respon yang diinginkan, dengan mengabaikan *offset* dari *setpoint*.
3. Dengan terus menaikkan nilai Kp nilai dari Kd dinaikkan untuk mengurangi *overshoot* yang terjadi.
4. Naikkan nilai Ki untuk mengurangi *offset*.

Dengan menggunakan metode *hand tuning* nilai parameter PID perlu diubah-ubah secara *trial* dan *error* agar respon yang diperoleh sesuai dengan harapan.

Proses *tuning* parameter PID ini dilakukan dengan cara mengatur nilai Kp hingga didapatkan respon sistem yang mendekati *setpoint* 80 rpm.

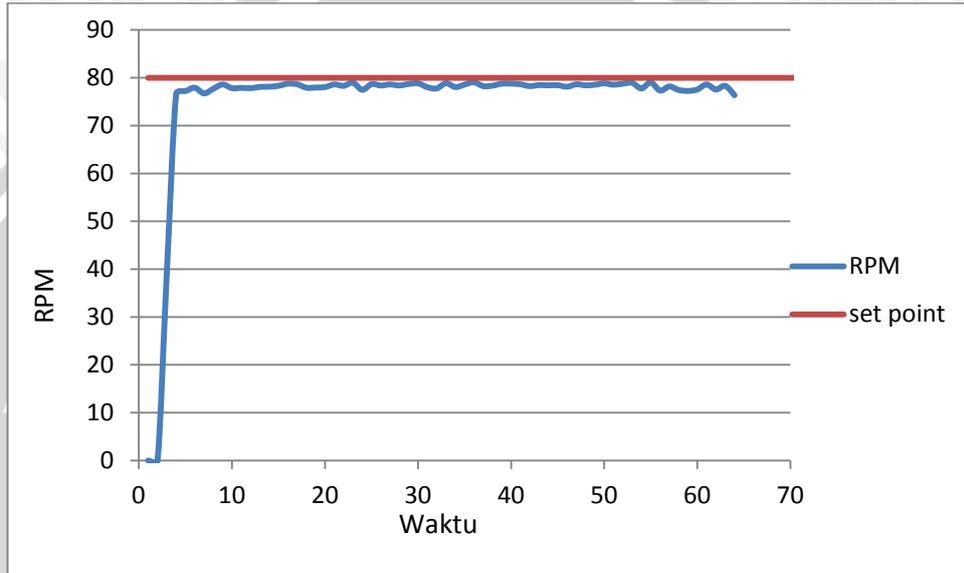
Untuk nilai $K_p = 0.1$, $K_i = 0$, dan $K_d = 0$ diperoleh grafik respon seperti pada Gambar 4.10.



Gambar 4.10 Grafik Respon untuk $K_p = 0.1$

$$\begin{aligned} \% E_{ss} &= \frac{|Average\ speed\ steady - setpoint|}{setpoint} \times 100 \% \\ &= \frac{|59,594 - 80|}{80} \times 100\% \\ &= 25,5\% \text{ dan memiliki waktu } steady \text{ 40 detik.} \end{aligned}$$

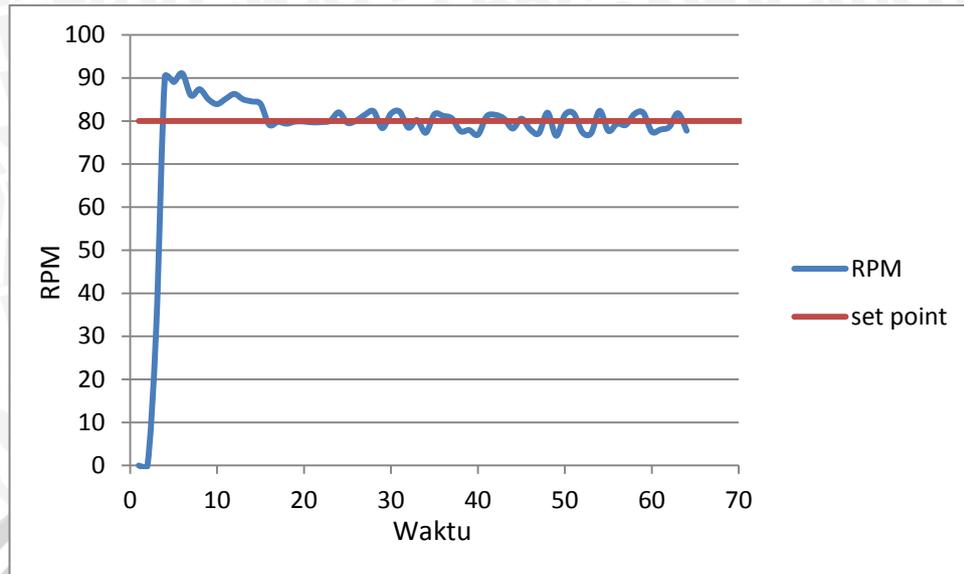
Untuk nilai $K_p = 2$, $K_i = 0$, dan $K_d = 0$ diperoleh grafik respon seperti pada Gambar 4.11.



Gambar 4.11 Grafik Respon untuk $K_p = 2$

$$\begin{aligned} \% E_{ss} &= \frac{|Average\ speed\ steady - setpoint|}{setpoint} \times 100 \% \\ &= \frac{|76,33226 - 80|}{80} \times 100\% \\ &= 4,58\% \text{ dan memiliki waktu } steady \text{ 4 detik.} \end{aligned}$$

Untuk nilai $K_p = 9$, $K_i = 0$, dan $K_d = 0$ diperoleh grafik respon seperti pada Gambar 4.12.

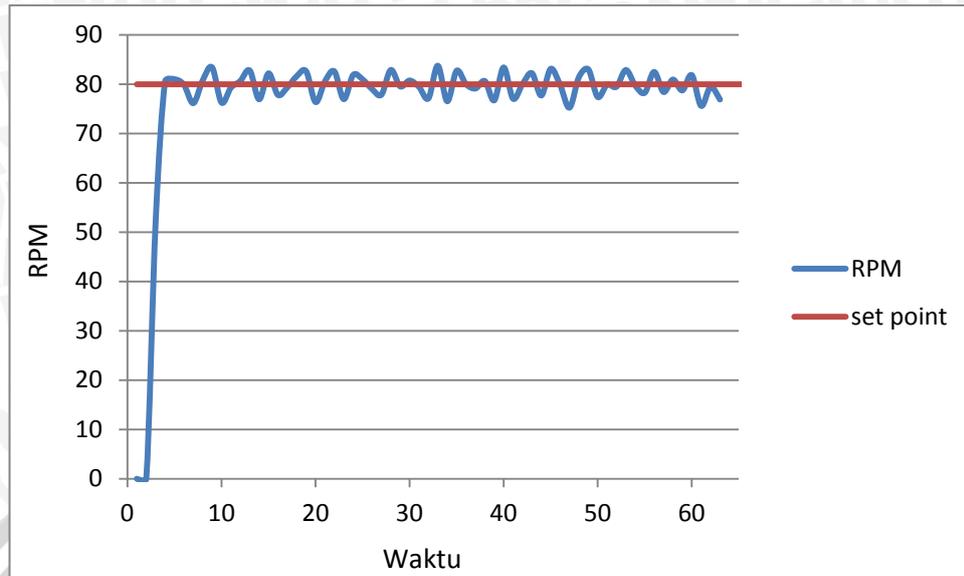


Gambar 4.12 Grafik Respon untuk $K_p = 10$

$$\begin{aligned} \% E_{ss} &= \frac{|Average\ speed\ steady - setpoint|}{setpoint} \times 100 \% \\ &= \frac{|78,80219 - 80|}{80} \times 100\% \\ &= 0,24\% \text{ dan memiliki waktu } steady \text{ 4 detik.} \end{aligned}$$

Dari hasil perancangan tersebut, didapatkan nilai $K_p = 10$ dapat mencapai stabil dan memiliki *error steady state* paling kecil walaupun masih terjadi *overshoot* dari titik stabil sekarang dan akan terjadi osilasi jika K_p dinaikkan. Setelah diperoleh nilai K_p yang cukup baik, Untuk mengurangi *overshoot* maka digunakanlah kontroler *derivative*.

Untuk nilai $K_p = 10$, $K_i = 0$, dan $K_d = 100$ diperoleh grafik respon seperti pada Gambar 4.13.



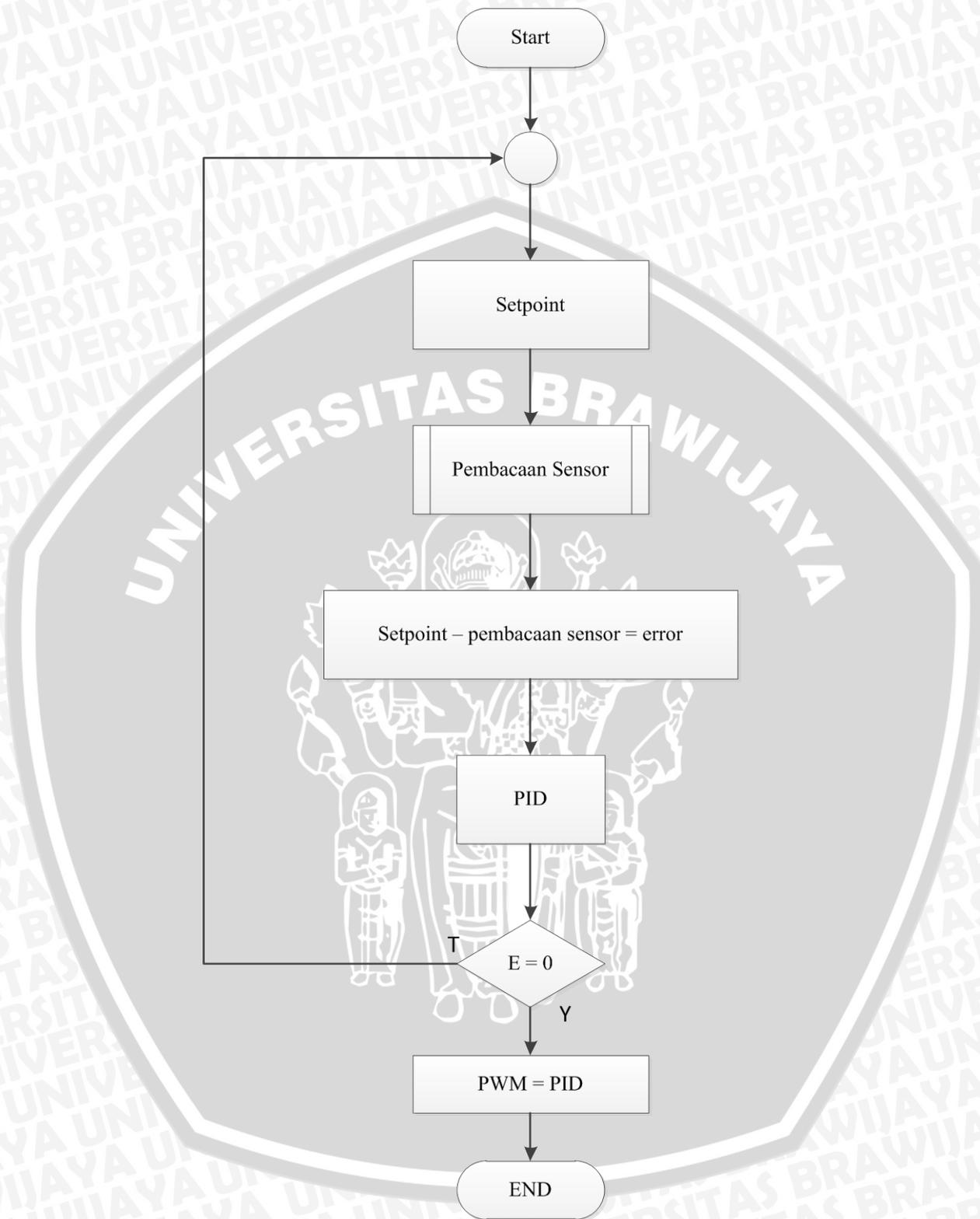
Gambar 4.13 Grafik Respon untuk $K_p = 10$ dan $K_d = 100$

$$\begin{aligned} \% E_{ss} &= \frac{|Average\ speed\ steady - setpoint|}{setpoint} \times 100\% \\ &= \frac{|76,88742 - 80|}{80} \times 100\% \\ &= 3,89\% \text{ dan memiliki waktu } steady \text{ 4 detik.} \end{aligned}$$

Dari hasil *tuning* parameter kontroler diperoleh nilai $K_p = 10$, $K_i = 0$ dan $K_d = 100$. Dengan nilai $K_p = 10$, $K_i = 0$ dan $K_d = 15$ didapatkan respon yang baik karena tidak ada *overshoot*, waktu *steady* 4 detik dan hanya memiliki *error steady state* 3,89%.

4.4 Perancangan Perangkat Lunak

Flowchart perancangan perangkat lunak ditunjukkan pada Gambar 4.14 berikut:



Gambar 4.14 Flowchart Program