

BAB III

METODE PENELITIAN

3.1 ALAT YANG DIGUNAKAN

Terdapat beberapa alat dan bahan yang digunakan dalam penelitian ini diantaranya:

Tabel 3.1 Alat dan Bahan

No	Alat dan Bahan	Jumlah
1	Robot <i>Line follower</i>	1
2	Software Arduino IDE	1
3	Matlab	1

1. Robot *Line follower*

Dalam penelitian ini robot *line follower* menggunakan board STM32F103C8T6 Blue Pill sebagai mikrokontroler, TB6612FNG yang merupakan sebuah motor driver yang difungsikan sebagai pengatur kecepatan roda dengan sumber *input* berupa sinyal *Pulse Width Modulation* (PWM), CD74HC4067 Multiplexer yang berfungsi sebagai pin extender untuk sensor *line tracer*, dengan demikian penggunaan pin terhadap 10 sensor *line tracer* dapat diperkecil. Motor DC yang berfungsi sebagai penggerak robot *line follower* ini memiliki kecepatan hingga 1000 RPM yang dilengkapi dengan *encoder*, sumber daya robot *line follower* pada penelitian ini menggunakan baterai Li-Po 3s.

2. Software Arduino IDE

Software ini digunakan untuk membuat kode program dalam bahasa C++ dan selanjutnya diunggah pada board STM32F103C8T6 Blue Pill.

3. Matlab

Software ini digunakan untuk menganalisis tanggapan waktu (Respon Transient) berdasarkan hasil data yang sudah didapat.

3.2 ALUR PENELITIAN

3.2.1 Flowchart Penelitian

Pada penelitian ini dimulai dari studi literatur, perancangan dari sistem, sampai dengan kesimpulan penelitian yang ditunjukkan pada *flowchart* dibawah ini.



Gambar 3.1 Flowchart penelitian

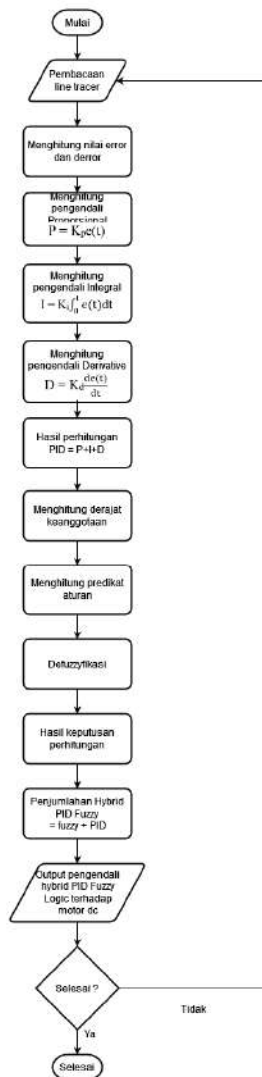
Pada *flowchart* penelitian tersebut menunjukkan ringkasan alur penelitian yang akan dilakukan. Penelitian dimulai dari studi literatur. Studi literatur dilakukan untuk memahami dan membandingkan penelitian yang akan dilakukan penulis dengan penelitian-penelitian sebelumnya. Selain itu, studi literatur juga dilakukan untuk mencari sumber-sumber yang akan dijadikan referensi pendukung penelitian yang akan dilakukan oleh penulis. Sumber-sumber referensi tersebut akan membantu penulis dalam tahap perancangan sistem.

Perancangan sistem dalam penelitian ini meliputi perancangan *hardware* dan *software*. Dalam perancangan *hardware* penulis merancang purwarupa model robot *line follower* menggunakan komponen-komponen dari alat dan bahan yang sudah disiapkan guna menunjang penelitian, sedangkan perancangan *software*

meliputi pembuatan *flowchart* sistem dan perancangan sistem kendali *hybrid* PID-Fuzzy Logic yang akan digunakan pada *hardware*. Setelah dilakukan perancangan sistem penulis akan melakukan pengecekan terhadap sistem yang sudah dibuat, jika sistem berjalan dengan baik penulis akan melanjutkan langkah berikutnya diantaranya mengambil data penelitian, analisis data penelitian, serta membuat kesimpulan penelitian. Jika sistem yang dibuat belum berjalan dengan baik penulis akan melakukan perbaikan sistem sampai sistem berjalan dengan baik sehingga dapat melanjutkan ke langkah berikutnya.

3.2.2 Flowchart Sistem

Berikut merupakan *flowchart* sistem kendali *hybrid* pid-fuzzy logic pada robot beroda *line follower*:



Gambar 3.2 Flowchart sistem

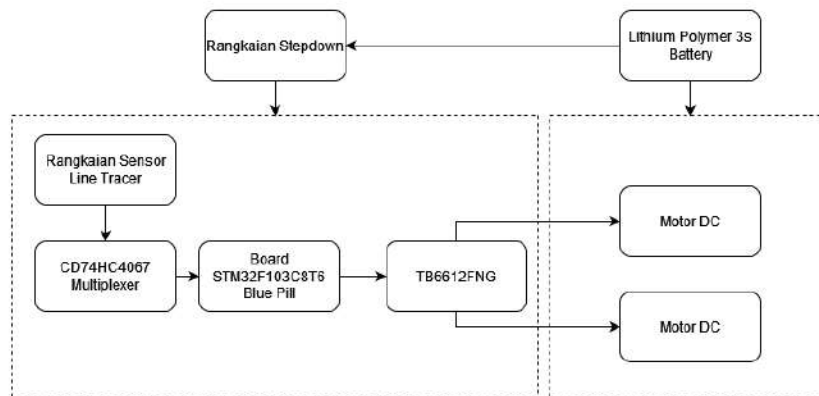
Dalam *flowchart* sistem tersebut diawali dengan pemacaan sensor *line tracer* yang terdiri dari led dan photodiode yang akan mendeteksi posisi robot *line follower* pada sebuah garis. Nilai pembacaan *line tracer* akan bernilai negatif apabila *line tracer* mendeteksi garis pada kiri robot *line follower* sedangkan nilai pembacaan *line tracer* akan bernilai positif apabila *line tracer* mendeteksi garis pada kanan robot *line follower*, dan nilai pembacaan akan bernilai nol apabila robot *line follower* berada pada tengah garis.

Nilai *error* didapat dari selisih set poin dengan nilai pembacaan *line tracer*, sedangkan *derror* didapat dari selisih eror sekarang dengan *error* sebelumnya. Untuk selanjutnya akan dilakukan perhitungan pengendali PID dan Fuzzy logic. Nilai pengendali PID diperoleh dari penjumlahan antara pengendali Proporsional, Integral, dan Derivatif. Selain perhitungan pengendali PID akan dilakukan perhitungan pengendali Fuzzy logic dengan menghitung derajat keanggotaan, kemudian menghitung predikat aturan, defuzzyfikasi, sehingga memperoleh hasil keputusan perhitungan. Setelah masing masing pengendali PID dan Fuzzy sudah dihitung, dilakukan penjumlahan diantara keduanya dan hasilnya akan menjadi *output* terhadap motor dc kanan dan kiri pada robot *line follower*.

3.2.3 Perancangan Sistem

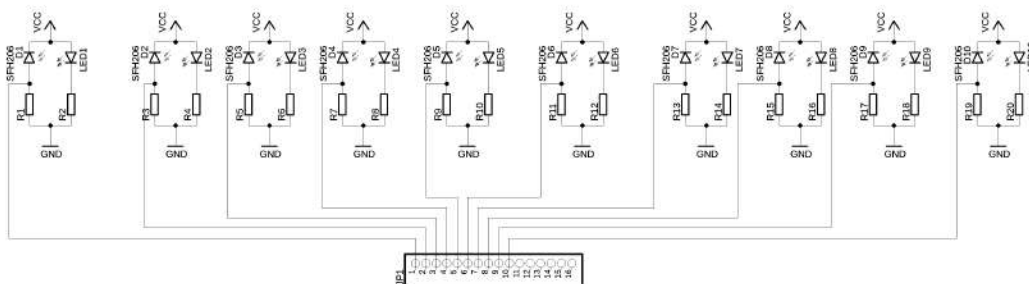
3.2.3.1 Robot *Line follower*

Robot *line follower* dalam penelitian ini dirancang menggunakan sensor *line tracer* yang berjumlah 10 channel yang dihubungkan ke Board STM32F103C8T6 Blue Pill dengan menggunakan CD74HC4067 Multiplexer. Robot ini juga dilengkapi dengan dua buah DRV8871 sebagai motor driver yang dapat mengatur laju kecepatan motor dc dengan menggunakan sinyal Pulse Width Modulation (PWM) dari Board STM32F103C8T6 Blue Pill.



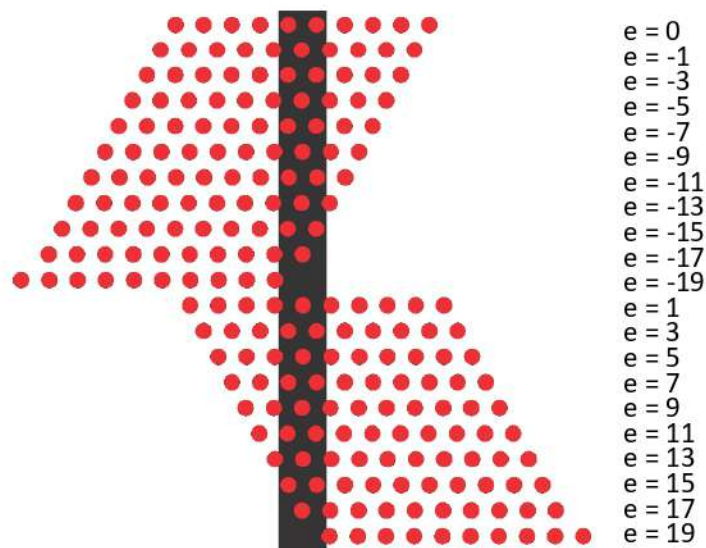
Gambar 3.3 Diagram Blok *Hardware*

Rangkaian sensor *line tracer* terdiri dari beberapa komponen diantaranya led 3mm, photodioda, dan resistor, rangkaian tersebut dirancang untuk dapat mengenali garis pada jalur yang akan dilewati robot. Sensor *line tracer* berjumlah 10 sensor yang nantinya akan dihubungkan dengan CD74HC4067 Multiplexer.



Gambar 3.4 Rangkaian Skematik *Line tracer* sensor

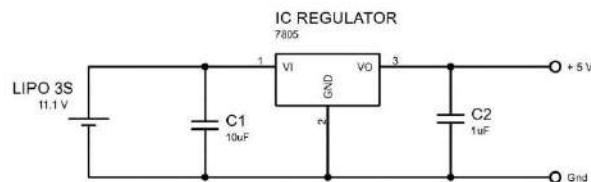
Untuk mendapatkan nilai *error* pembacaan, 10 sensor tersebut dipetakan menjadi beberapa bagian seperti sebagai berikut:



Gambar 3.5 Mapping Nilai *Error*

Error tersebut akan digunakan sebagai *input* dari sistem pengendali dan *output* berupa nilai *pwm* yang akan mengendalikan roda robot sehingga robot tetap berada pada garis lintasan.

Semua perangkat dalam robot *line follower* mendapatkan supply daya dari baterai *Lithium Polymer* (Li-Po) 3s. Tegangan kerja pada masing-masing komponen sebesar 5v kecuali motor dc yang memiliki tegangan kerja sebesar 12v, dengan demikian dibutuhkan rangkaian *stepdown* untuk menurunkan tegangan baterai Li-Po 3s yang memiliki tegangan sekitar 11,1v-12,6v menjadi 5v. Rangkaian *stepdown* dirancang dengan menggunakan sebuah IC regulator LM7805.



Gambar 3.6 Rangkaian Stepdown

3.2.3.2 Pengendali PID

Perancangan sistem kendali *hybrid pid-fuzzy logic* pada robot beroda *line follower*, menggunakan pengendali PID yang berperan sebagai pengendali utama meskipun sistem pengendali dirancang secara *hybrid*. Pengendali PID sebelumnya sudah melewati proses *tunning* menggunakan metode *tunning trial and error* untuk mendapatkan nilai konstanta k_p , k_i , dan k_d , dalam implementasinya pengendali PID yang berupa rumus persamaan matematika harus diubah kedalam suatu perintah program, perubahan ini mengacu pada sifat-sifat dan algoritma dari rumus tersebut sehingga hanya akan merubah bentuknya saja. Perancangan sistem pengendali PID ini, akan menjelaskan implementasi pengendali PID dalam sebuah perintah program sebagai berikut:

1. Pengendali Proporsional

Rumus pengendali proporsional adalah:

$$P = K_p e(t) \tag{3.1}$$

Keterangan: P = Proporsional.

K_p = Konstanta proporsional.

$e(t)$ = *Error* berdasarkan waktu (t).

Sehingga dapat kita ubah menjadi perintah program sebagai berikut:

```
error = Sv-Pv;  
P = Kp*error;
```

Keterangan: P = Proporsional.

K_p = Konstanta proporsional.

Sv = Set value.

Pv = Preset value.

2. Pengendali Integral

Rumus pengendali integral adalah:

$$I = K_i \int_0^t e(t) dt \quad (3.2)$$

Keterangan: I = Integral.

K_i = Konstanta integral.

e(t) = *Error* berdasarkan waktu (t).

dt = Nilai perubahan waktu (detik).

Sehingga dapat kita ubah menjadi perintah program sebagai berikut:

```
error = Sv-Pv;  
integral_e = (integral_e + e)*delta_t;  
I = Ki*integral_e;
```

Keterangan: I = Integral.

K_i = Konstanta integral.

Sv = Set value.

Pv = Preset value.

integral_e = Integral *error*.

delta_t = Perubahan waktu (detik).

3. Pengendali Derivatif

Rumus pengendali derivatif adalah:

$$D = K_d \frac{de(t)}{dt} \quad (3.3)$$

Keterangan: D = Derivatif.

K_d = Konstanta derivatif.

de(t) = Perubahan *error* berdasarkan waktu (t).

dt = Nilai perubahan waktu (detik).

Sehingga dapat kita ubah menjadi perintah program sebagai berikut:

```

error = Sv-Pv;
derivatif_e = (e-last_e)/delta_t;
D = Kd*derivatif_e;
last_e = error;

```

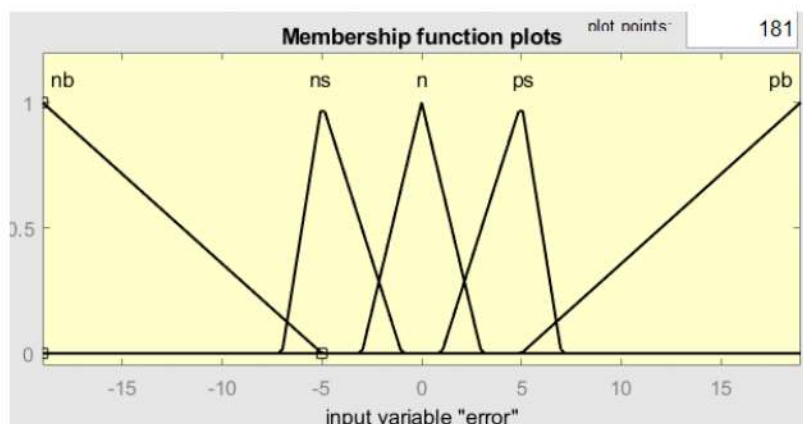
Keterangan: D = Derivatif.
 K_d = Konstanta derivatif.
Sv = Set value.
Pv = Preset value.
derivatif_e = Derivatif *error*.
last_error = *Error* sebelumnya.
delta_t = Perubahan waktu (detik).

3.2.3.3 Pengendali Fuzzy

Metode fuzzy yang digunakan dalam sistem kendali *hybrid pid-fuzzy logic* pada robot beroda *line follower* ini menggunakan fuzzy tsukamoto, dikarenakan metode fuzzy tsukamoto memiliki rumus yang sederhana dari proses fuzzyfikasi sampai defuzzyfikasi sehingga memudahkan untuk diubah kedalam sebuah perintah program. Dalam pengendali fuzzy tsukamoto ini menggunakan dua buah *input* yaitu *error* dan *derror* dengan satu variabel *output*.

1. Proses Fuzzyfikasi

Berikut merupakan rancangan membership function untuk masing-masing variabel *input* dan variable *output*:



Gambar 3.7 Membership Function error

Persamaan yang digunakan dalam mencari masing masing derajat keanggotaan pada kurva segitiga *input* variabel *error*:

a. Persamaan Derajat Keanggotaan Negatif Besar

$$\mu[\text{Negatif Besar}] = \begin{cases} 0 & ; x \geq -5 \\ \frac{-5-x}{-5+19} & ; -19 < x < -5 \\ 1 & ; x = -19 \end{cases} \quad (3.4)$$

b. Persamaan Derajat Keanggotaan Negatif Sedikit

$$\mu[\text{Negatif Sedikit}] = \begin{cases} 0 & ; x \leq -7 \text{ atau } x \geq -1 \\ \frac{x+7}{-5+7} & ; -7 < x < -5 \\ \frac{-1-x}{-1+5} & ; -5 < x < -1 \\ 1 & ; x = -5 \end{cases} \quad (3.5)$$

c. Persamaan Derajat Keanggotaan Normal

$$\mu[\text{Normal}] = \begin{cases} 0 & ; x \leq -1 \text{ atau } x \geq 1 \\ \frac{x+1}{0+1} & ; -1 < x < 0 \\ \frac{1-x}{1-0} & ; 0 < x < 1 \\ 1 & ; x = 0 \end{cases} \quad (3.6)$$

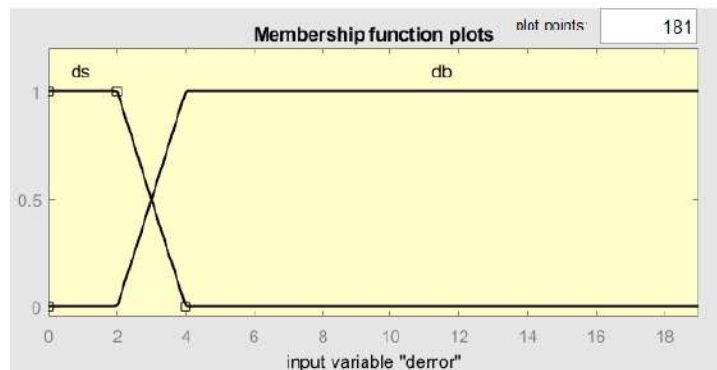
d. Persamaan Derajat Keanggotaan Positif Sedikit

$$\mu[\text{Positif Sedikit}] = \begin{cases} 0 & ; x \leq 1 \text{ atau } x \geq 7 \\ \frac{x-1}{7-1} & ; 1 < x < 5 \\ \frac{7-x}{7-5} & ; -5 < x < -1 \\ 1 & ; x = 5 \end{cases} \quad (3.7)$$

e. Persamaan Derajat Keanggotaan Positif Besar

$$\mu[\text{Positif Besar}] = \begin{cases} 0 & ; x \leq 5 \\ \frac{x-5}{19-5} & ; 5 < x < 19 \\ 1 & ; x = 19 \end{cases} \quad (3.8)$$

Sedangkan rancangan membership function pada variable *error* dengan menggunakan kurva trapesium sebagai berikut:



Gambar 3.8 Membership Function *error*

Persamaan yang digunakan dalam mencari masing masing derajat keanggotaan pada kurva trapesium *input* variabel *derror*:

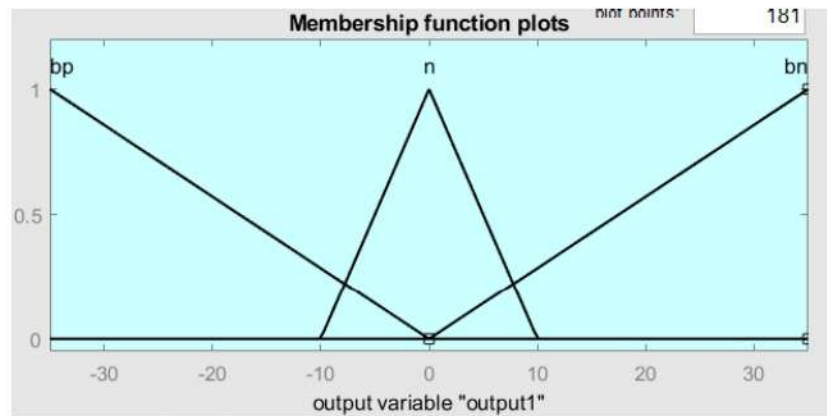
- a. Persamaan Derajat Keanggotaan *Derror* Sedikit

$$\mu[\text{Derror Sedikit}] = \begin{cases} 0 & ; x \geq 4 \\ \frac{4-x}{4-0} & ; 2 < x < 4 \\ 1 & ; x \leq 2 \end{cases} \quad (3.9)$$

- b. Persamaan Derajat Keanggotaan *Derror* Banyak

$$\mu[\text{Derror Banyak}] = \begin{cases} 0 & ; x \leq 2 \\ \frac{x-2}{4-2} & ; 2 < x < 4 \\ 1 & ; x \geq 4 \end{cases} \quad (3.10)$$

Sedangkan rancangan membership function pada variable *output* dengan menggunakan kurva segitiga sebagai berikut:



Gambar 3.9 Membership Function output

Persamaan yang digunakan dalam mencari masing masing derajat keanggotaan pada kurva segitiga variabel *output*:

- a. Persamaan Derajat Keanggotaan Besar Positif

$$\mu[\text{Besar Positif}] = \begin{cases} 0 & ; x \geq 0 \\ \frac{0-x}{0+35} & ; -35 < x < 0 \\ 1 & ; x = -35 \end{cases} \quad (3.11)$$

- b. Persamaan Derajat Keanggotaan Normal

$$\mu[\text{Normal}] = \begin{cases} 0 & ; x \leq -10 \text{ atau } x \geq 10 \\ \frac{x-0}{10-0} & ; 0 < x < 10 \\ \frac{0-x}{0+10} & ; -10 < x < 0 \\ 1 & ; x = 0 \end{cases} \quad (3.12)$$

c. Persamaan Derajat Keanggotaan Besar Negatif

$$\mu[\text{Besar Negatif}] = \begin{cases} 0 & ; x \leq 0 \\ \frac{x-0}{35-0} & ; 0 < x < 35 \\ 1 & ; x = 35 \end{cases} \quad (3.13)$$

2. Proses Inferensi Pada Tabel Aturan Fuzzy

Dalam proses inferensi terdapat aturan-aturan yang perlu dibuat untuk menentukan *output* berdasarkan *input* dari pengendali fuzzy, aturan-aturan tersebut dibuat dalam sebuah tabel aturan fuzzy.

Tabel 3.2 Tabel Aturan Fuzzy

$\begin{matrix} u_error \\ u_derror \end{matrix}$	Negatif Banyak	Negatif Sedikit	Normal	Positif Sedikit	Positif Banyak
Derror Sedikit	Banyak Negatif	Normal	Normal	Normal	Banyak Positif
Derror Banyak	Banyak Negatif	Banyak Negatif	Normal	Banyak Positif	Banyak Positif

Tabel 3.2 merupakan aturan-aturan fuzzy yang berupa aturan IF-THEN sebagai berikut:

1. IF *error* = Negatif Banyak AND *derror* = Derror Sedikit THEN *output* = Banyak Negatif.
2. IF *error* = Negatif Banyak AND *derror* = Derror Banyak THEN *output* = Banyak Negatif.
3. IF *error* = Negatif Sedikit AND *derror* = Derror Sedikit THEN *output* = Normal.
4. IF *error* = Negatif Sedikit AND *derror* = Derror Banyak THEN *output* = Banyak Negatif.
5. IF *error* = Normal AND *derror* = Derror Sedikit THEN *output* = Normal.
6. IF *error* = Normal AND *derror* = Derror Banyak THEN *output* = Normal.
7. IF *error* = Positif Sedikit AND *derror* = Derror Sedikit THEN *output* = Normal.
8. IF *error* = Positif Sedikit AND *derror* = Derror Banyak THEN *output* = Banyak Positif.

9. IF $error = \text{Positif Banyak}$ AND $derror = \text{Derror Sedikit}$ THEN $output = \text{Banyak Positif}$.
10. IF $error = \text{Positif Banyak}$ AND $derror = \text{Derror Banyak}$ THEN $output = \text{Banyak Positif}$.

Proses selanjutnya mengubah nilai linguistik dalam aturan tersebut menjadi $output$ crisp (nilai pasti) melalui proses defuzzyfikasi. Metode fuzzy yang digunakan dalam sistem ini merupakan metode fuzzy tsukamoto oleh karena itu, persamaan defuzzyfikasi dalam sistem ini menggunakan metode rata-rata seperti sebagai berikut:

$$Z = \frac{\sum \mu_i * z_i}{\sum \mu_i} \quad (3.10)$$

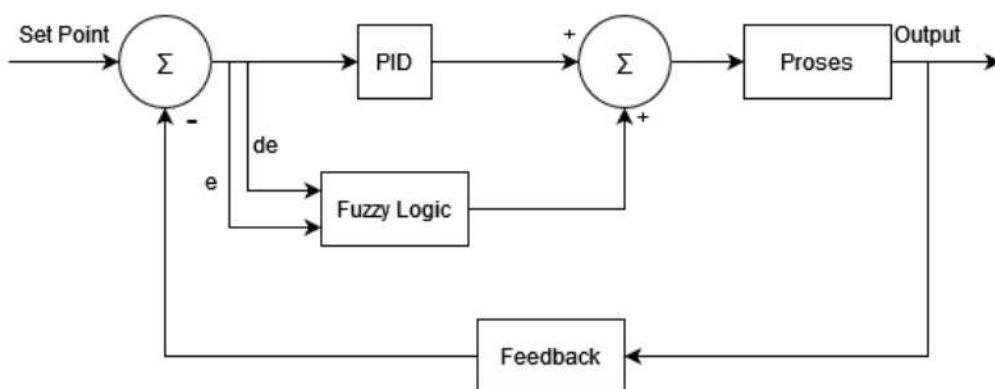
Keterangan: $Z = \text{Output Cripst}$.

$\mu_i = \text{Membership Function}$.

$z_i = \text{Nilai hasil inferensi pada membership function}$.

3.2.3.4 Pengendali *Hybrid* PID-Fuzzy

Pengendali *hybrid* PID-Fuzzy merupakan pengendali yang menggabungkan pengendali PID dan pengendali PID secara *hybrid*. Penerapan sistem pengendali *hybrid* dalam pengendali sistem kendali *hybrid* pid-fuzzy logic pada robot beroda *line follower* ini dilakukan secara paralel dengan hasil $output$ berupa penjumlahan dari masing-masing pengendali PID dan pengendali Fuzzy.

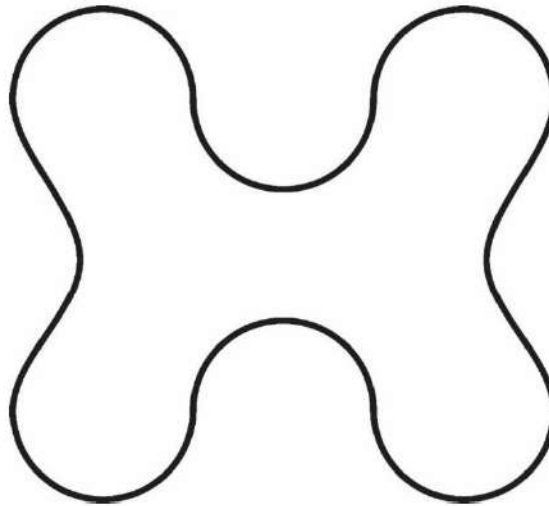


Gambar 3.10 Diagram Blok Sistem

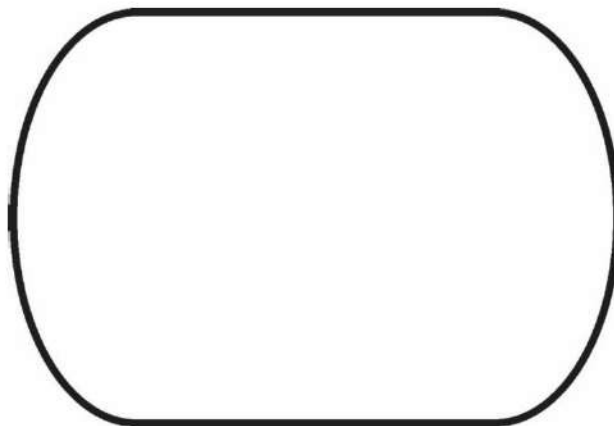
3.3 METODE PENGUJIAN

3.3.1 Pengujian Sistem Kendali PID dan Sistem Kendali *Hybrid* PID-Fuzzy

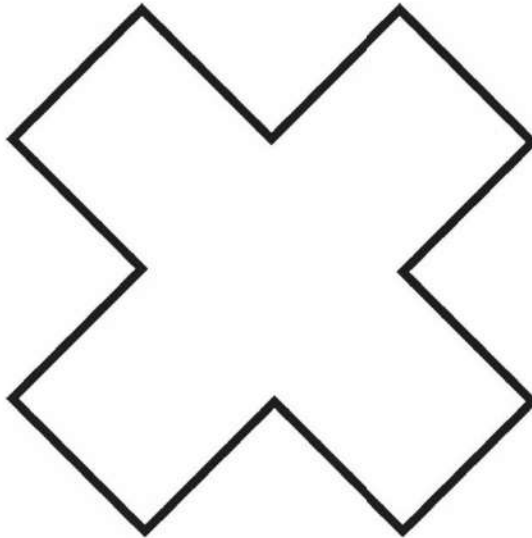
Pengujian ini bertujuan untuk mengetahui perbandingan antara sistem dengan kendali dan sistem kendali *hybrid* PID-Fuzzy dengan membandingkan kecepatan masing-masing sistem dengan parameter waktu tempuh dalam satu putaran penuh lintasan robot *line follower* yang sudah dirancang, dan melakukan analisis tanggapan waktu terhadap pergerakan robot di lintasan berdasarkan nilai *error*. Pengujian dilakukan dengan menggunakan tiga skema lintasan, lintasan pertama merupakan lintasan berkelok setengah lingkaran (Lintasan A), lintasan kedua merupakan lintasan lurus berbelok (Lintasan B), dan lintasan ketiga merupakan lintasan yang memiliki tikungan tajam sebesar 90 derajat.



Gambar 3.11 Lintasan Robot *Line follower* berkelok setengah lingkaran.



Gambar 3.12 Lintasan Robot *Line follower* Lurus Berkelok.



Gambar 3.13 Lintasan Robot *Line follower* Tikungan Tajam 90 Derajat.

Dalam pengambilan data untuk menganalisis tanggapan waktu, data akan dikirim melalui *bluetooth* yang dipasang pada robot.