

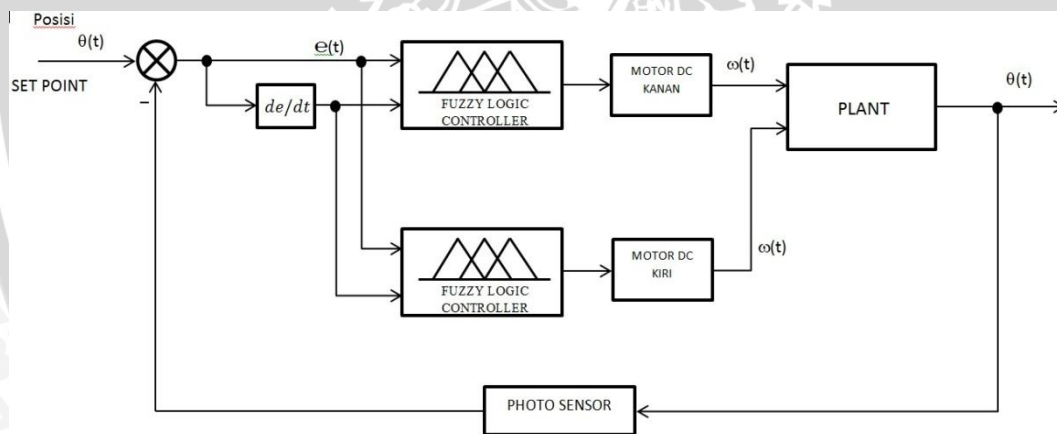
BAB IV

PERANCANGAN DAN PEMBUATAN ALAT

Pada bab ini akan dijelaskan mengenai perancangan dan pembuatan robot *line follower* mulai dari diagram blok sistem, desain mekanik robot, perancangan perangkat keras, dan perancangan perangkat lunak. Perancangan dan pembuatan dilakukan secara bertahap dan sistematis, sehingga akan memudahkan dalam analisis sistem.

4.1 Blok Diagram Sistem

Pada perancangan alat diperlukan perancangan blok diagram sistem yang dapat menjelaskan sistem secara garis besar dan diharapkan alat dapat bekerja sesuai dengan yang direncanakan. Blok diagram tersebut dapat dilihat pada Gambar 4.1.



Gambar 4.1 Diagram Blok Sistem

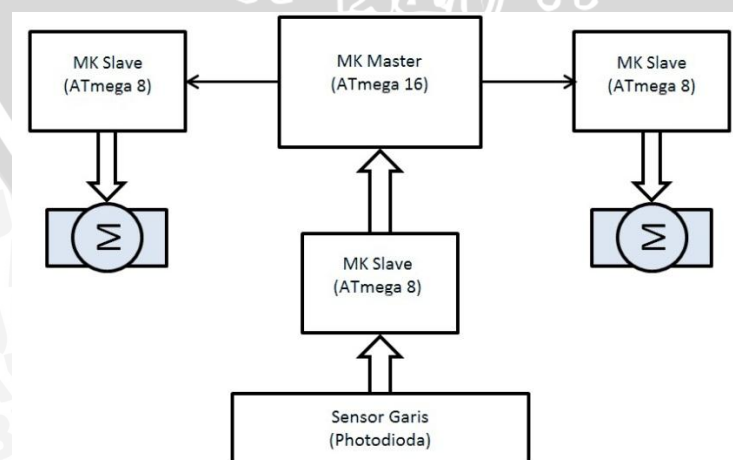
Robot yang dirancang menggunakan sistem gerak differensial, yaitu robot *line follower* digerakkan oleh dua motor DC secara terpisah. Walaupun masing-masing motor DC memiliki karakteristik yang identik, tidak menutup kemungkinan perhitungan kontrol dari dua motor tersebut berbeda. Maka dari itu digunakan dua FLC untuk mengatur kecepatan masing-masing motor. Masukan FLC berupa simpangan sudut antara robot dengan *line*.

Sebagai sensor garis digunakan photodiode yang berfungsi mendeteksi warna hitam yang nantinya akan digunakan sebagai data masukan kontroler sekaligus sebagai detektor apakah posisi(θ) robot terhadap garis (*line*) sudah sesuai dengan *set point* yang telah ditentukan.

4.2 Prinsip Kerja Alat

Robot digerakkan oleh dua motor DC magnet permanen. Sistem dari alat ini berjalan mengikuti garis hitam sebagai lintasan (*line*). Saat photodiode mendeteksi garis hitam, data logika digital dari sensor photodiode dikonversi menjadi nilai sudut simpangan (θ°) oleh mikrokontroler *slave sensor* (ATmega8). Pada kondisi ini robot masih dalam keadaan persiapan (*prepare*). Data hasil konversi dari mikrokontroler *slave sensor* (ATmega8) selanjutnya diteruskan ke ATmega 16. Mikrokontroler ATmega16 sebagai mikrokontroler *master* berfungsi mengolah data dari mikrokontroler *slave sensor* (ATmega8), mikrokontroler *slave motor* kiri serta mikrokontroler *slave motor* kanan menggunakan *Fuzzy Logic Control* (FLC). Kemudian mikrokontroler *master* mengirimkan data berupa nilai PWM motor ke mikrokontroler *slave* (ATmega8) motor kiri dan mikrokontroler *slave* (ATmega8) motor kanan.

Pengaturan kecepatan motor dilakukan oleh mikrokontroler *slave motor* berdasar perintah dari mikrokontroler *master* untuk mempertahankan batas tepi badan robot terhadap *line*. Untuk selanjutnya robot akan berjalan saat motor penggerak dicatu tegangan. Lebih jelasnya dapat dilihat pada Gambar 4.2 berikut.

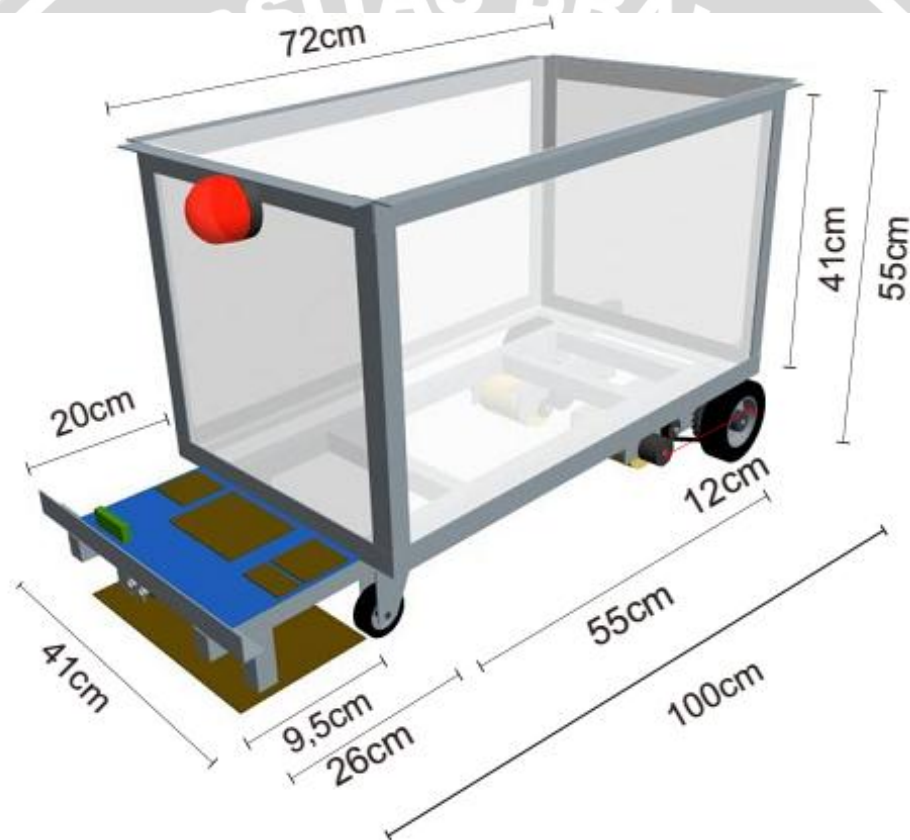


Gambar 4.2 Prinsip Kerja Alat

4.3 Perancangan Perangkat Keras (*Hardware*)

4.3.1 Perancangan Mekanik Robot

Mekanik robot *line follower* dirancang menyerupai miniatur truk pengangkut dengan panjang 100cm, lebar 40cm dan tinggi 45cm. Robot memiliki *box* dengan ukuran 70cm x 40cm x 40cm, dua roda depan merupakan roda bebas dan dua roda belakang yang dikopel motor DC sebagai penggerak dengan sistem differensial (terpisah). Gambar 4.3 menunjukkan desain mekanik robot yang digambar dengan menggunakan *software 3ds Max 9* tampak perspektif depan samping.



Gambar 4.3 Desain Mekanik Robot Tampak Perspektif Depan Samping

Sedangkan robot *line follower* hasil perancangan dapat dilihat dalam Gambar 4.4

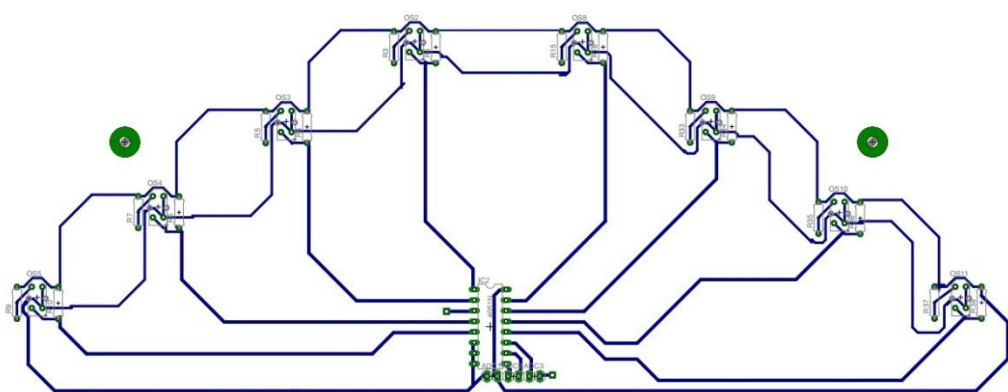


Gambar 4.4 Robot *Line Follower*

Dalam Gambar 4.4 terdapat kerangka *box* yang berfungsi sebagai tempat *box* sampah, rangkaian elektrik diletakkan dibagian kepala robot, sensor photodiode diletakkan dibagian depan bawah robot.

4.3.2 Sensor Photodiode

Dalam perancangan robot ini digunakan photodiode sebagai sensor garis (*tracer*). Photodiode yang digunakan sebagai sensor berjumlah delapan buah dan jarak antara sensor yang satu dengan yang lain ditentukan sebesar 3cm dengan desain penempatan posisi seperti dalam Gambar 4.5 berikut.



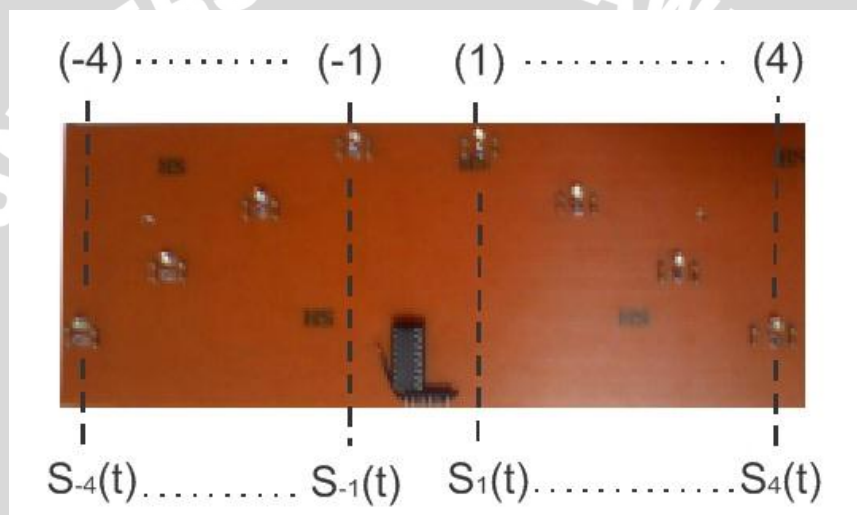
Gambar 4.5 *Layout* Sensor Photodiode

Data keluaran dari sensor photodiode menjadi masukan dari MK *slave* sensor dan selanjutnya digunakan dalam proses pengaturan di MK *master* menggunakan FLC.

Identifikasi sensor diperlukan untuk mendapatkan nilai keluaran sensor berupa sudut (θ). Sistem ini menggunakan metode penalaran (*reasoning*). Untuk mendapatkan fungsi keanggotaan masukan sudut kendali (*input membership function of steering angle*) menggunakan fungsi keanggotaan segitiga himpunan fuzzy. Keadaan logika sensor saat (t) di asumsikan $S_a(t)$. Subscript a merepresentasikan posisi setiap sensor berdasarkan rancangan sensor dalam Gambar 4.5.

(Hiroshi Takimoto & Yokinobu Hoshino, 2009)

Identifikasi sensor dapat dilihat dalam Gambar 4.6



Gambar 4.6 Identifikasi Sensor

Sensor akan berlogika 1 jika mendeteksi warna hitam dan berlogika 0 saat mendeteksi warna putih.

$$S_a(t) = \begin{cases} 1 & \text{sensor ON} \\ 0 & \text{sensor OFF} \end{cases}$$

Sedangkan nilai dari masing-masing sensor yang mendeteksi warna hitam di asumsikan sebagai $N_a(t)$. Hubungan antara $S_a(t)$ dan $N_a(t)$ dapat dilihat pada persamaan 4.1

$$\begin{aligned} S_a(t) = 1 & \leftrightarrow N_a(t) = a \\ S_a(t) = 0 & \leftrightarrow N_a(t) = 0 \end{aligned} \tag{4.1}$$

Dari persamaan 4.1 dapat dihitung nilai P :

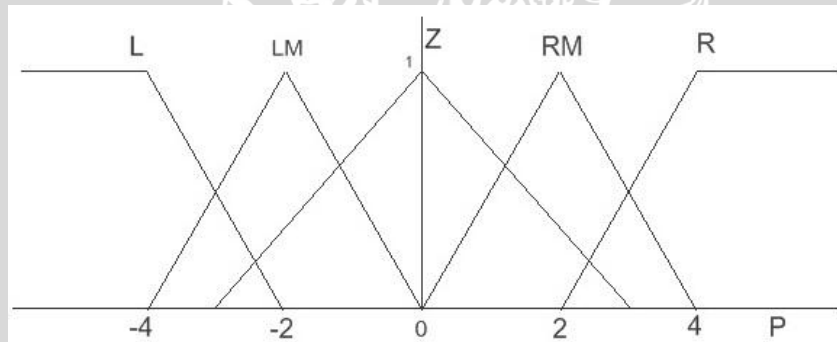
$$P = \frac{\sum_{a=-4}^4 N_a(t)}{\sum_{a=-4}^4 S_a(t)} \tag{4.2}$$

dimana $a = \{-4,-3,-2,-1,1,2,3,4\}$

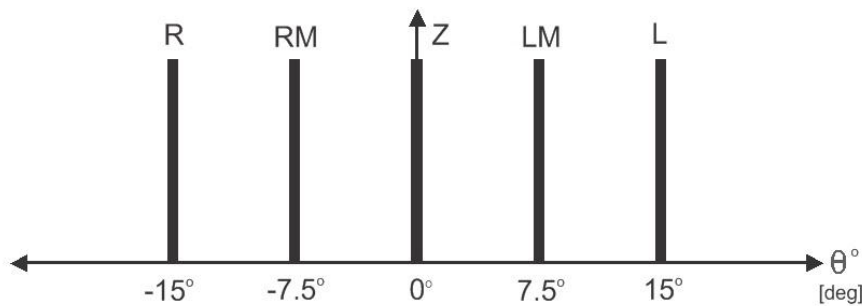
Keterangan:

- P = Fungsi keanggotaan masukan sudut kendali
- $N_a(t)$ = Nilai sensor
- $S_a(t)$ = Posisi sensor

Rentang nilai P diambil dari batas a , yaitu dari -4 sampai 4. Sedangkan keluaran menggunakan *singleton* yang bukan fungsi keanggotaan tetapi nilai numerik yang nyata. Fungsi keanggotaan keluaran sensor merupakan representasi dari sudut simpangan robot yang dikeluarkan oleh MK *slave* sensor. Konstanta *singleton* ditentukan $-15^\circ, -7,5^\circ, 0^\circ, 7,5^\circ$ dan 15° . Fungsi keanggotaan masukan P ditunjukkan pada Gambar 4.7 sedangkan konstanta *singleton* sensor ditunjukkan pada Gambar 4.8



Gambar 4.7 Fungsi Keanggotaan Masukan P



Gambar 4.8 Konstanta Singleton Sensor

Keterangan :

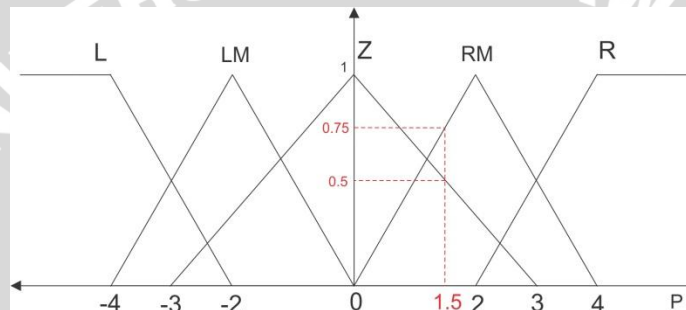
L = Left
 LM = Left Middle
 Z = Zero
 RM = Right Middle
 R = Right

Contoh kasus:

Misalkan sensor $S_1(t)$ dan $S_2(t)$ mendeteksi warna hitam, maka;

$$\begin{array}{l} S_1(t) = 1 \leftrightarrow N_1(t) = 1 \\ S_2(t) = 1 \leftrightarrow N_2(t) = 2 \end{array} \quad \Rightarrow \quad \begin{array}{l} P = \frac{1+2}{2} \\ = 1.5 \end{array}$$

Derajat keanggotaan P untuk nilai $P = 1.5$ dapat dilihat dalam Gambar 4.9



Gambar 4.9 Derajat keanggotaan P untuk nilai $P = 1.5$

Dengan $\mu(Z) = \frac{3-1.5}{3-0} = 0,5$; $\mu P(RM) = \frac{1.5-0}{2-0} = 0,75$, maka didapatkan

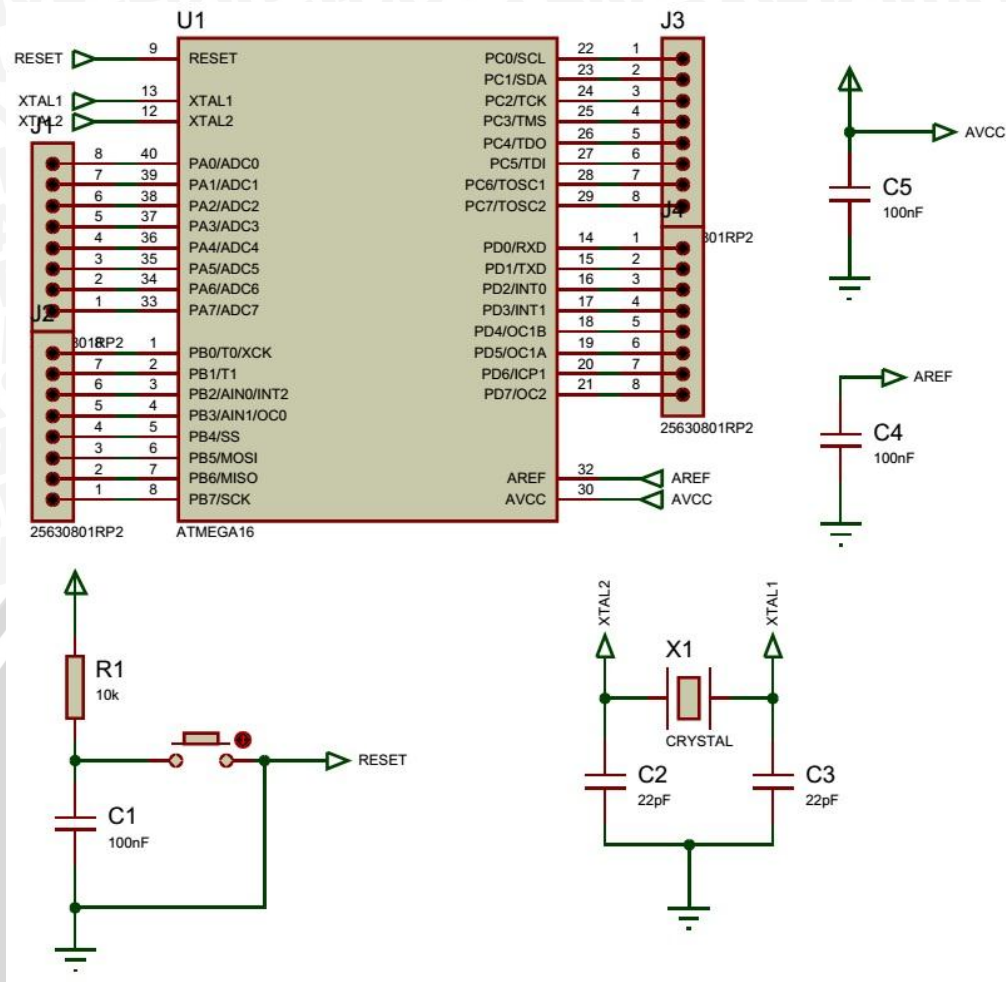
$\mu P(Z) = 0,5$ dan $\mu P(RM) = 0,75$

$$\theta^\circ = \frac{0,5[Z] + 0,75[RM]}{0,5 + 0,75} = -4,5^\circ$$

Jadi, besarnya simpangan (θ°) untuk nilai $P=1,5$ adalah $-4,5^\circ$

4.3.3 Rangkaian Minimum Sistem Mikrokontroler Master

Pada robot ini digunakan mikrokontroler ATmega16 sebagai pengolah data utama. mikrokontroler *master* menerima data berupa posisi robot terhadap garis (θ°) dari mikrokontroler *slave* sensor (ATmega8), data tersebut di olah menggunakan *fuzzy* kemudian hasil dari pengolahan data berupa PWM motor akan dikirimkan ke mikrokontroler *slave* (ATmega8) motor kanan dan Mikrokontroler *slave* (ATmega8) motor kiri. Berikut konfigurasi dari pin I/O dari mikrokontroler ATmega16 (mikrokontroler *master*) ditunjukkan dalam Gambar 4.10



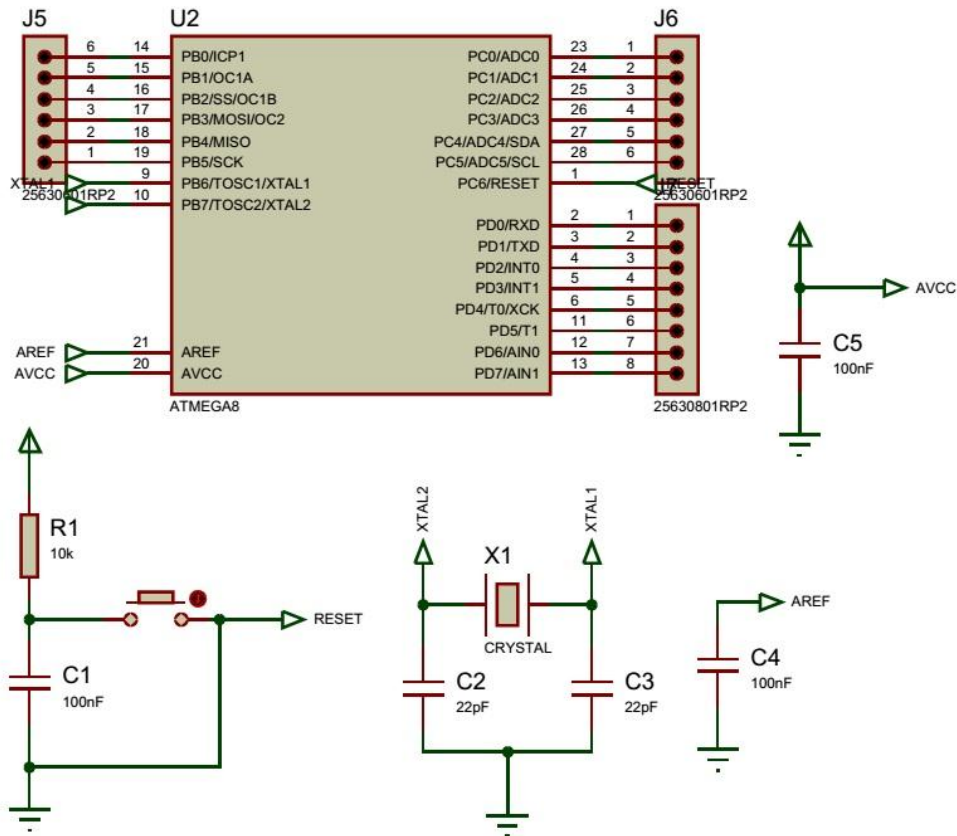
Gambar 4.10 Rangkaian Minimum Sistem Mikrokontroler Master

Rangkaian minimum sistem mikrokontroler *master* yang digunakan mempunyai 4 port, dengan 32 jalur I/O yang dapat di program menjadi masukan atau keluaran. Pin dan kaki IC yang digunakan antara lain:

- Pin B5(MOSI) : Sebagai pin keluaran data SPI.
- Pin B6(MISO) : Sebagai pin masukan data SPI.
- Pin B7(SCK) : Sebagai pin keluaran *clock* SPI.
- Pin Reset : Dihubungkan dengan rangkaian *switch reset*.
- Pin XTAL1 : Dihubungkan dengan kaki osilator 11059200 Hz.
- Pin XTAL2 : Dihubungkan dengan kaki osilator 11059200 Hz.
- Pin AREF : Dihubungkan dengan rangkaian AREF.
- Pin AVCC : Dihubungkan dengan rangkaian AVCC.

4.3.4 Rangkaian Minimum Sistem Mikrikontroler *Slave*

Pada robot ini juga digunakan mikrokontroler ATmega8 sebagai pengolah data sensor photodiode, pengaturan rpm motor kanan dan motor kiri. Konfigurasi kaki I/O dari mikrokontroler *slave* (ATmega8) ditunjukkan dalam Gambar 4.11



Gambar 4.11 Rangkaian Minimum Sistem Mikrokontroler *Slave*

Mikrokontroler *slave* (ATmega8) mempunyai 3 port, dengan 21 jalur I/O.

Pin yang digunakan antara lain:

- Pin B3(MOSI) : Sebagai pin masukan data SPI.
- Pin B4(MISO) : Sebagai pin keluaran data SPI.
- Pin B5(SCK) : Sebagai pin masukan *clock* SPI.
- Pin Reset : Dihubungkan dengan rangkaian *switch* Reset.
- Pin XTAL1 : Dihubungkan dengan kaki osilator 11059200 Hz.
- Pin XTAL2 : Dihubungkan dengan kaki osilator 11059200 Hz.
- Pin AREF : Dihubungkan dengan rangkaian AREF.

- Pin AVCC : Dihubungkan dengan rangkaian AVCC.

4.4 Perancangan *Fuzzy Logic Control*

Berdasarkan diagram blok sistem pada gambar 4.1, dibutuhkan dua buah FLC. Satu FLC sebagai kontroler untuk motor kiri dan satu FLC sebagai kontroler motor kanan. Perbedaan dari keduanya terletak pada penentuan fungsi keanggotaan keluaran pada proses defuzzifikasi.

4.4.1 Variabel Masukan dan Variabel Keluaran

Masukan FLC berupa posisi robot terhadap *line*. Variabel masukan yang digunakan dalam perancangan kontroler ada dua, yaitu *error* (e) dan *delta error* (Δe) posisi robot terhadap *line* (θ°), sedangkan variabel keluaran kontroler berupa *Pulse Width Modulation* (PWM) motor. *Error* merupakan selisih antara nilai *set point* dan nilai *present value* atau nilai sekarang. Sedangkan *delta error* (Δe) adalah selisih antara *present error* $e(t)$ dan *past error* $e(t-1)$. *Error* dan *delta error* di representasikan dalam perumusan sebagai berikut :

$$e(t) = \text{set poin} - \text{present value}(t) \quad (4.3)$$

$$\Delta e(t) = e(t) - e(t - 1) \quad (4.4)$$

Keterangan :

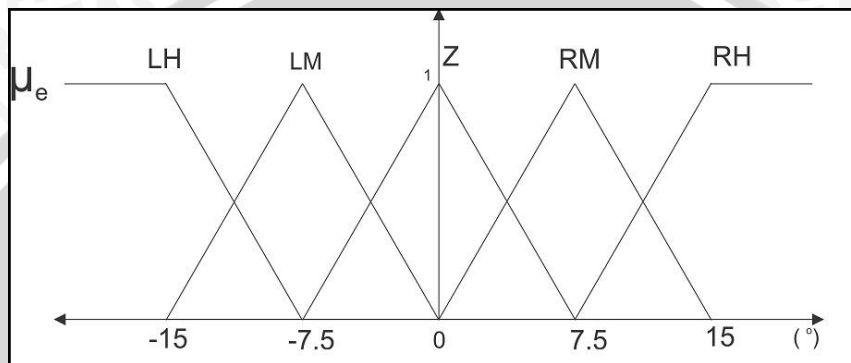
$e(t)$: <i>error</i> saat t
present value(t)	: nilai saat t
$e(t - 1)$: <i>error</i> sebelumnya
$\Delta e(t)$: selisih <i>error</i> saat t

dimana variabel *error* dan *delta error* merupakan fungsi waktu.

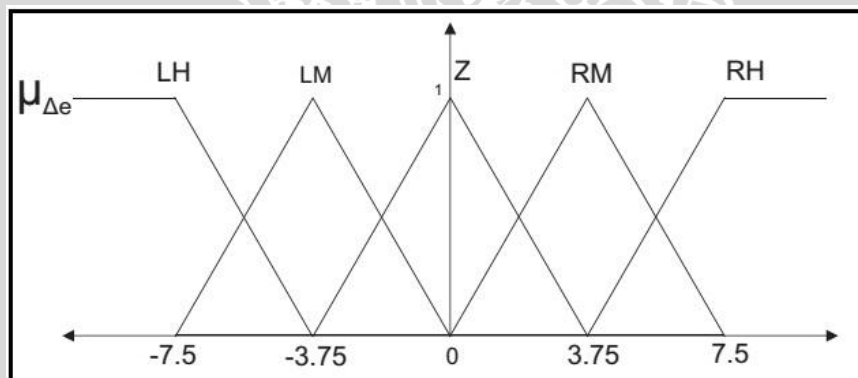
4.4.2 Fungsi Keanggotaan Masukan

Input Membership Function atau fungsi keanggotaan masukan kontroler terbagi menjadi dua fungsi yaitu, fungsi keanggotaan *error* dan fungsi keanggotaan *delta error*. Masing-masing fungsi keanggotaan terbagi lagi menjadi lima label yaitu, *Left High* (LH), *Left Middle* (LM), *Zero* (Z), *Right Middle* (RM) dan *Right High* (RH).

Data masukan fungsi keanggotaan didapat dari nilai yang dikeluarkan oleh sensor photodiode. Label masing-masing fungsi keanggotaan masukan berjumlah lima dengan tujuan agar *rule* yang dihasilkan semakin banyak. Semakin banyak *rule* maka semakin banyak juga kemungkinan kondisi yang tercipta. Gambar 4.12 dan 4.13 merupakan fungsi keanggotaan masukan error dan delta error. (ZHANG Xiao Ming & LONG Shi Yu, 2012; Ahmed Rubaai., Abdul R. Ofoli., Marcel J. Castro-Sitiriche, 2008)



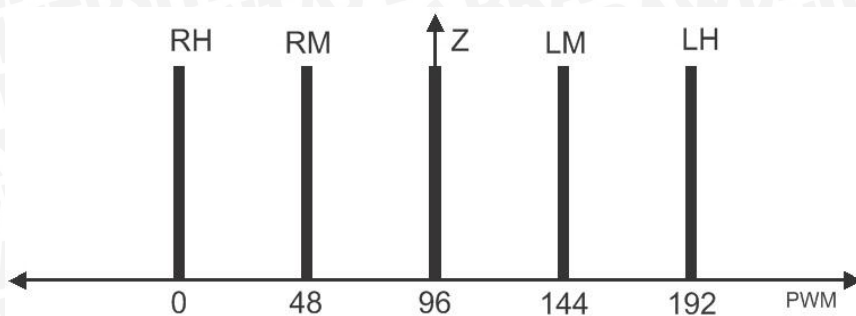
Gambar 4.12 Fungsi Keanggotaan Masukan Error.



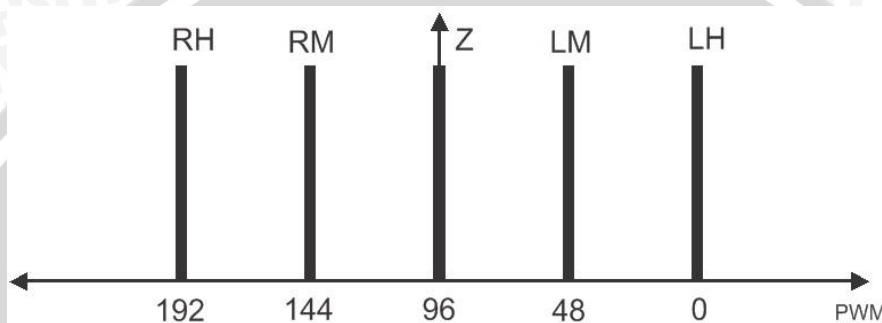
Gambar 4.13 Fungsi Keanggotaan Delta Error

4.4.3 Fungsi Keanggotaan Keluaran

Fungsi keanggotaan keluaran motor DC merupakan representasi dari besarnya nilai PWM yang dikeluarkan oleh mikroprosesor berupa hasil dari perhitungan metode defuzzifikasi *Center Of Gravity (COG)* yang telah dibulatkan oleh *integer* di dalam pemrograman. Besarnya nilai PWM untuk motor kanan dirancang berbanding terbalik dengan motor kiri. Hal ini dimaksudkan untuk pergerakan robot saat berbelok. Fungsi keanggotaan keluaran motor kiri dan motor kanan dapat dilihat dalam Gambar 4.14 dan Gambar 4.15



Gambar 4.14 Fungsi Keanggotaan Keluaran Motor Kiri



Gambar 4.15 Fungsi Keanggotaan Keluaran Motor Kanan

Keterangan:

- RH : *Right High*
- RM : *Right Middle*
- Z : *Zero*
- LM : *Left Middle*
- LH : *Left High*

4.4.4 Perancangan Aturan Fuzzy

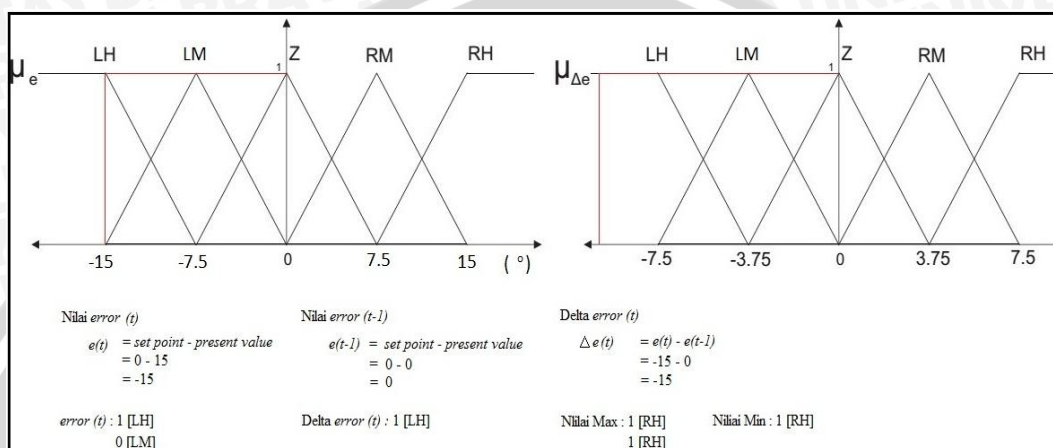
Rule (aturan) fuzzy digunakan sebagai penentu keluaran dari fuzzifikasi yang akan diolah dalam proses defuzzifikasi, dengan jumlah *rule* 25 macam. *Rule* tersebut didapat dari 5 *membership function error* dan 5 *membership function delta error*. Perancangan aturan fuzzy terdapat dalam Tabel 4.1

Tabel 4.1 Aturan Fuzzy

e de	LH	LM	Z	RM	RH
LH	RH	RH	RH	RM	Z
LM	RH	RH	RM	Z	LM
Z	RH	RM	Z	LM	LH
RM	RM	Z	LM	LH	LH
RH	Z	LM	LH	LH	LH

4.4.5 Metode Inferensi Max-Min

Untuk mengetahui metode ini pada sistem, perlu melakukan beberapa perhitungan pembuktian metode Max-Min pada sistem. Pengujian dilakukan pada nilai *steering angle* 15° dengan *set point* 0° sebagai perwakilan atas keseluruhan sistem. Inferensi Max-Min pada *steering angle* 15° ditunjukkan dalam Gambar 4.16



Gambar 4.16 Inferensi Max-Min pada *Steering Angle* 15°

4.4.6 Defuzzifikasi

Defuzzifikasi adalah proses untuk mengubah keluaran *fuzzy* menjadi keluaran *crisp*. Pada skripsi ini metode defuzzifikasi yang digunakan adalah *Center Of Gravity* (COG) dengan hasil defuzzifikasi berupa nilai PWM motor. Perhitungan defuzzifikasi untuk inferensi Max-Min pada *steering angle* 15° adalah sebagai berikut :

$$U = \frac{\sum_{i=1}^n w_i u_i}{\sum_{i=1}^n w_i} \quad (4.5)$$

Defuzzifikasi untuk motor kanan :

$$\begin{aligned} &= \frac{1 \times RH}{1} \\ &= \frac{1 \times 192}{1} \\ &= 192 \end{aligned}$$

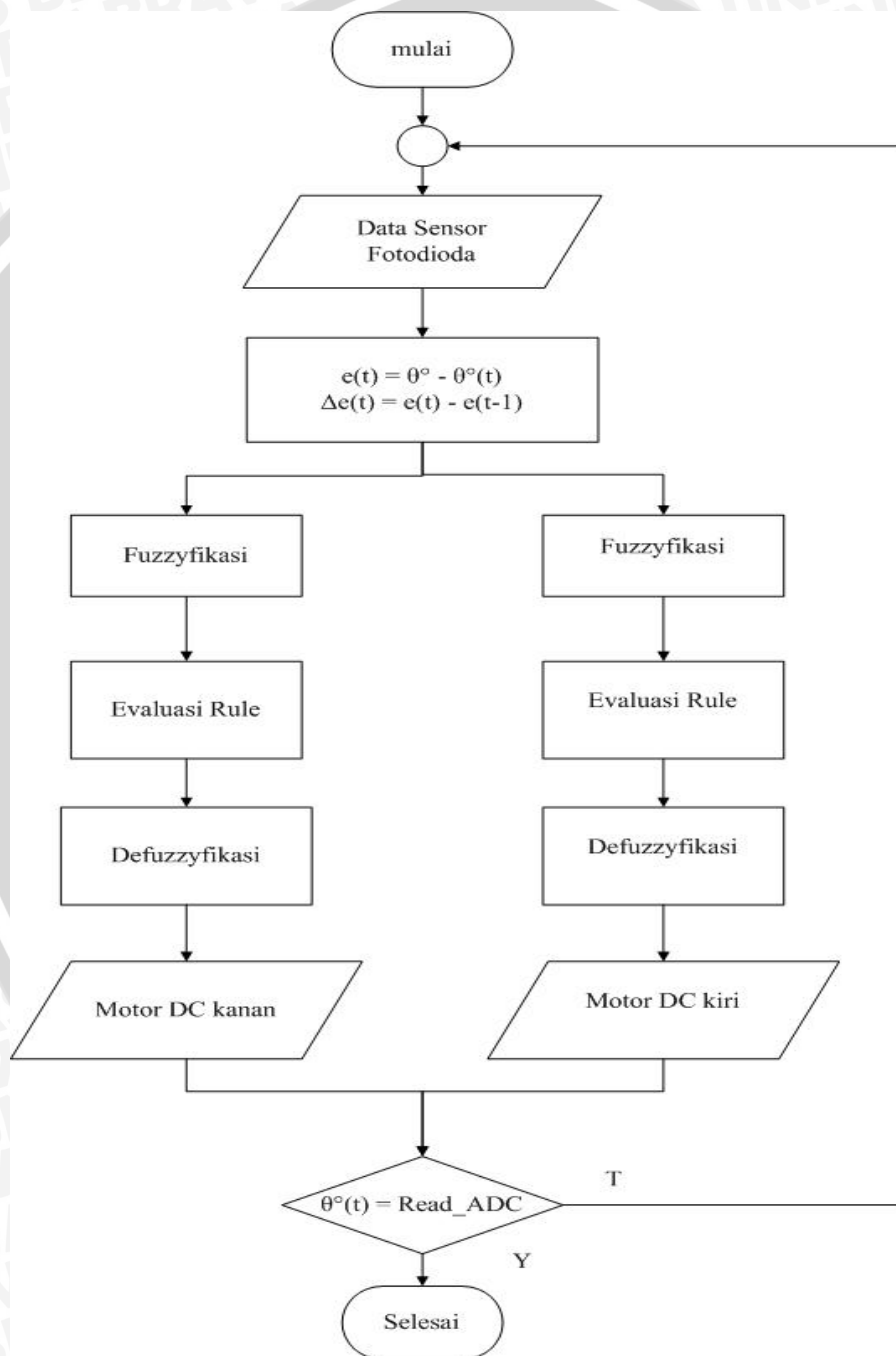
Defuzzifikasi untuk motor kiri :

$$\begin{aligned} &= \frac{1 \times RH}{1} \\ &= \frac{1 \times 0}{1} \\ &= 0 \end{aligned}$$

Dari perhitungan defuzzifikasi tersebut dihasilkan PWM untuk motor kanan sebesar 192 dan PWM untuk motor kiri sebesar 0.

4.5 Perancangan Perangkat Lunak (*software*)

Flowchart perancangan *software* pada skripsi ini ditunjukkan dalam Gambar 4.17



Gambar 4.17 *Flowchart* Perancangan *Software*