

BAB II

TINJAUAN PUSTAKA

2.1. Artificial Intelligence

Di awal abad 20, Seorang penemu Spanyol, Torres y Quevedo, membuat sebuah mesin yang dapat men'skak-mat' raja lawannya dengan ratu dan raja, Perkembangan secara sistematis kemudian dimulai segera setelah ditemukannya komputer digital. Artikel ilmiah pertama tentang Kecerdasan Buatan ditulis oleh Alan Turing pada tahun 1950, dan kelompok riset pertama dibentuk tahun 1954 di Carnegie Mellon University oleh Allen Newell dan Hebert Simon. Namun bidang Kecerdasan Buatan baru dianggap sebagai bidang tersendiri di konferensi Dartmouth tahun 1956, di mana 10 peneliti muda memimpikan mempergunakan komputer untuk memodelkan bagaimana cara berfikir manusia. Hipotesa mereka adalah "*Mekanisme berfikir manusia dapat secara tepat dimodelkan dan disimulasikan pada komputer digital*". dan ini yang menjadi landasan dasar Kecerdasan Buatan.

Tidak ada kesepakatan mengenai definisi Kecerdasan Buatan, di antaranya adalah:

- a. Sebuah studi tentang bagaimana membuat komputer mengerjakan sesuatu yang dapat dikerjakan manusia (rich, 1991)
- b. Cabang ilmu komputer yang mempelajari otomatisasi tingkah laku cerdas (setiawan, 1993)
- c. Suatu perilaku sebuah mesin yang jika dikerjakan oleh manusia akan disebut cerdas (Turing, et al, 1996)

Menurut Winston dan Prendergast (1984), tujuan dari Kecerdasan Buatan adalah:

- a. Membuat mesin menjadi lebih pintar.
- b. Memahami apakah kecerdasan (*intelligence*) itu.
- c. Membuat mesin menjadi lebih berguna

Sebuah ujian yang dapat dilakukan untuk menentukan apakah sebuah komputer/mesin menunjukkan perilaku cerdas didesain oleh Alan Turing Tes Turing menyatakan sebuah mesin dikatakan pintar hanya apabila seorang pewawancara (manusia) yang berbicara dengan orang lain dan mesin yang dua-duanya tidak terlihat olehnya, tidak mampu menentukan mana yang manusia dan mana yang mesin, meskipun dia telah berulang-ulang melontarkan pertanyaan yang sama.

Secara garis besar, AI terbagi menjadi dua paham pemikiran yaitu AI Konvensional dan Kecerdasan Komputasional (CI, *Computational Intelligence*). AI konvensional kebanyakan melibatkan metoda-metoda yang sekarang diklasifikasikan sebagai pembelajaran mesin, yang ditandai dengan formalisme dan analisis statistik. Dikenal juga sebagai AI simbolis, AI logis, AI murni dan AI cara lama (GOFAI, *Good Old Fashioned Artificial Intelligence*). Metoda-metodanya meliputi:

1. Sistem pakar: menerapkan kapabilitas pertimbangan untuk mencapai kesimpulan. Sebuah sistem pakar dapat memproses sejumlah besar informasi yang diketahui dan menyediakan kesimpulan-kesimpulan berdasarkan pada informasi-informasi tersebut.
2. Petimbangan berdasar kasus
3. Jaringan Bayesian
4. AI berdasar tingkah laku: metoda modular pada pembentukan sistem AI secara manual

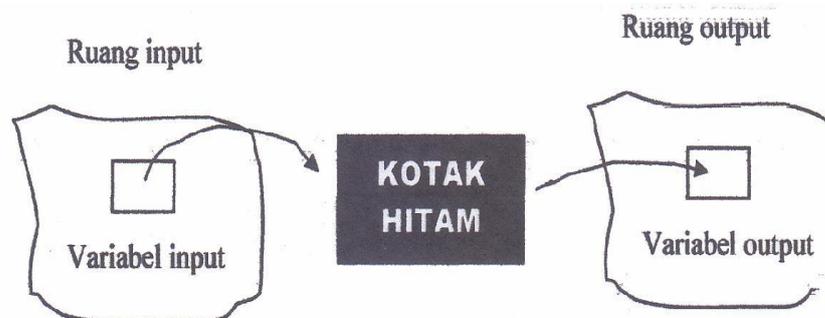
Kecerdasan komputasional melibatkan pengembangan atau pembelajaran interaktif (misalnya penalaran parameter seperti dalam sistem koneksionis). Pembelajaran ini berdasarkan pada data empiris dan diasosiasikan dengan AI non-simbolis, AI yang tak teratur dan perhitungan lunak. Metode-metode pokoknya meliputi:

1. Jaringan Syaraf: sistem dengan kemampuan pengenalan pola yang sangat kuat
2. Sistem *Fuzzy*: teknik-teknik untuk pertimbangan di bawah ketidakpastian, telah digunakan secara meluas dalam industri modern dan sistem kendali produk konsumen.

3. Komputasi Evolusioner: menerapkan konsep-konsep yang terinspirasi secara biologis seperti populasi, mutasi dan “*survival of the fittest*” untuk menghasilkan pemecahan masalah yang lebih baik.

2.2. Logika Fuzzy

Logika Fuzzy merupakan sebuah logika yang memiliki nilai kekaburan atau kesamaran (fuzzyness) antara benar dan salah. Dalam teori logika fuzzy sebuah nilai bisa bernilai benar dan salah secara bersamaan namun berapa besar kebenaran dan kesalahan suatu nilai tergantung kepada bobot keanggotaan yang dimilikinya.



Gambar 2.1 Konsep Dasar Logika Fuzzy

Profesor Lotfi A. Zadeh [1965] adalah guru besar pada *University of California* yang merupakan pencetus se kaligus yang memasarkan ide tentang cara mekanisme pengolahan atau manajemen ketidakpastian yang kemudian dikenal dengan logika *fuzzy*. Banyak model DBMS dan *query fuzzy* yang ada, salah satunya adalah model Tahani yang ditemukan pada tahun 1977. Prof. Lutfi Zadeh berpendapat bahwa logika benar dan salah dari logika *boolean*/konvensional tidak dapat mengatasi masalah gradasi yang ada di dunia nyata. Untuk mengatasi masalah gradasi tersebut maka ia mengembangkan sebuah himpunan samar (*fuzzy*). Dalam penyajiannya variabel-variabel yang akan digunakan harus cukup menggambarkan ke-*fuzzy*-an tetapi di lain pihak persamaan-persamaan yang dihasilkan dari variabel-variabel itu haruslah cukup sederhana sehingga komputasinya menjadi

cukup mudah. Karena itu Profesor Lotfi A Zadeh kemudian memperoleh ide untuk menyajikannya dengan menentukan “derajat keanggotaan” (*membership function*) dari masing-masing variabelnya.

Fungsi keanggotaan (*membership function*), Sudradjat (2008) adalah suatu kurva yang menunjukkan pemetaan titik input data kedalam nilai keanggotaanya (sering juga disebut dengan derajat keanggotaan) yang memiliki interval antara 0 sampai 1.

2.2.1. Himpunan dan Fungsi Keanggotaan Logika Fuzzy

Pada himpunan tegas (Crisp), nilai keanggotaan suatu variabel x dalam suatu himpunan A , yang sering ditulis dengan $\mu_A[x]$, memiliki dua kemungkinan, (Kusumadewi S, Purnomo H, 2010) yaitu

1. Satu (1), yang berarti bahwa suatu item menjadi anggota dalam suatu himpunan atau
2. Nol (0), yang berarti bahwa suatu item tidak menjadi anggota dalam suatu himpunan.

Kalau pada himpunan tegas (crisp), nilai keanggotaan hanya ada dua kemungkinan yaitu 0 atau 1, maka pada himpunan fuzzy nilai keanggotaan terletak pada rentang 0 sampai 1. Apabila x memiliki nilai keanggotaan fuzzy $\mu_A[x]=0$ berarti x tidak menjadi anggota himpunan A , demikian pula apabila x memiliki nilai keanggotaan fuzzy $\mu_A[x]=1$ berarti x menjadi anggota penuh pada himpunan A .

Himpunan fuzzy memiliki dua atribut, yaitu :

1. Linguistik, yaitu penamaan suatu grup yang mewakili suatu keadaan atau kondisi tertentu dengan menggunakan bahasa alami seperti : Dekat, Sedang, Jauh.
2. Numeris, yaitu suatu nilai (angka) yang menunjukkan ukuran dari suatu variabel seperti : 25, 35, 40 dan sebagainya.

Ada beberapa hal yang perlu diketahui dalam memahami sistem *fuzzy* (Aplikasi logika *fuzzy* untuk pendukung keputusan, Sri Kusumadewi, Hari Purnomo, Edisi kedua, Graha Ilmu, 2010), yaitu:

a. Variabel fuzzy

Variabel fuzzy merupakan variabel yang hendak dibahas dalam suatu sistem fuzzy, contoh : kecepatan dan jarak.

b. Himpunan fuzzy

Himpunan fuzzy merupakan suatu grup yang mewakili suatu kondisi atau keadaan tertentu dalam suatu variabel fuzzy. Contohnya adalah variabel jarak terbagi menjadi tiga himpunan fuzzy, yaitu : Dekat, Sedang, Jauh.

c. Semesta Pembicaraan

Semesta pembicaraan adalah keseluruhan nilai yang diperbolehkan untuk dioperasikan dalam suatu variabel fuzzy. Semesta pembicaraan merupakan himpunan bilangan real yang senantiasa naik (bertambah) secara monoton dari kiri ke kanan. Nilai semesta pembicaraan dapat berupa bilangan positif maupun negatif. Pada suatu kondisi tertentu nilai semesta pembicaraan ini tidak dibatasi batas atasnya.

Contoh :

1. Semesta pembicaraan untuk variabel kecepatan $[0 \ 100]$

2. Semesta pembicaraan untuk variabel jarak $[0 \ 20]$

d. Domain

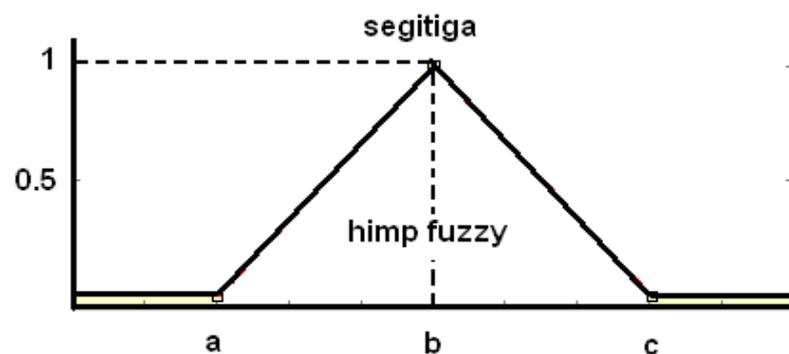
Domain himpunan fuzzy adalah keseluruhan nilai yang diijinkan dalam semesta pembicaraan dan boleh dioperasikan dalam suatu himpunan fuzzy. Seperti halnya semesta pembicaraan, domain merupakan himpunan bilangan real yang senantiasa naik (bertambah) secara monoton dari kiri ke kanan. Nilai domain dapat berupa bilangan positif maupun negatif.

Contoh domain himpunan fuzzy :

1. DEKAT = [0, 5]
2. SEDANG = [7, 10]
3. JAUH = [15, 20]

Fungsi keanggotaan adalah suatu kurva yang menunjukkan pemetaan titik-titik *input* data kedalam nilai keanggotaannya (sering juga disebut dengan derajat keanggotaan) yang memiliki interval antara 0 sampai 1. Salah satu cara yang dapat digunakan untuk mendapatkan nilai keanggotaan adalah dengan melalui pendekatan fungsi. Ada beberapa fungsi yang bisa digunakan.

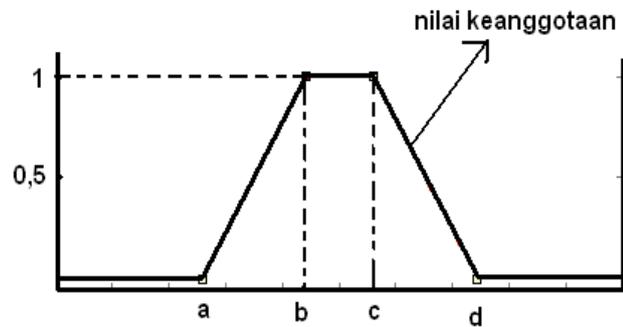
1. fungsi keanggotaan segitiga



Gambar 2.2. Himpunan fungsi keanggotaan

$$S(u; a, b, c) = \begin{cases} 0 & \text{untuk } u < a \\ (u - a)/(b - a) & \text{untuk } a \leq u \leq b \\ (c - u)/(c - b) & \text{untuk } b \leq u \leq c \\ 0 & \text{untuk } u > c \end{cases}$$

2. fungsi keanggotaan trapezium

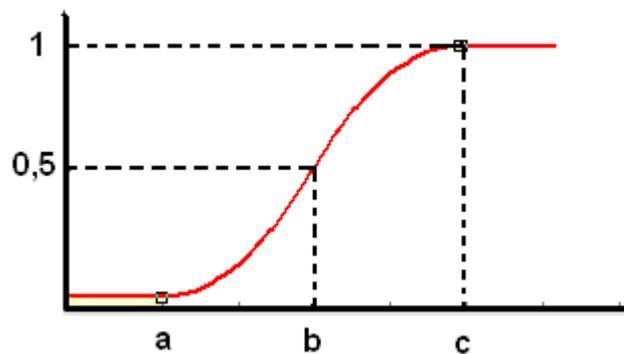


Gambar 2.3. Himpunan Keanggotaan trapesium

$$S(u : a, b, c, d) = \begin{cases} 0 & \text{untuk } u < a \\ (u - a)/(b - a) & \text{untuk } a \leq u \leq b \\ 1 & \text{untuk } b \leq u \leq c \\ (d - u)/(d - c) & \text{untuk } c \leq u \leq d \\ 0 & \text{untuk } u > d \end{cases}$$

3. fungsi keanggotaan sigmoid

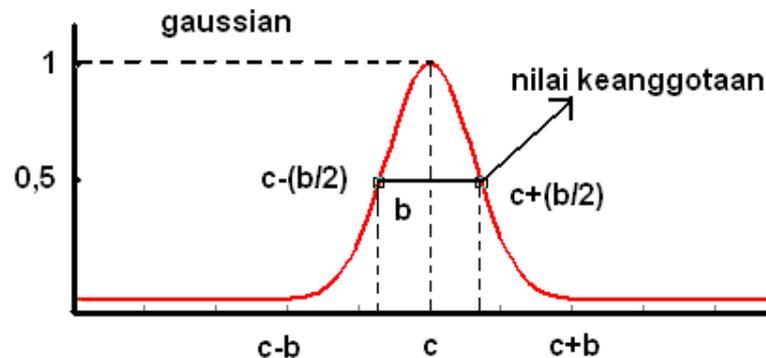
fungsi keanggotaan sigmoid



Gambar 2.4. Himpunan keanggotaan sigmoid

$$S(u; a, b, c) = \begin{cases} 0 & \text{untuk } u < a \\ 2[(u - a)/(c - a)]^2 & \text{untuk } a \leq u \leq b \\ 1 - 2[(u - c)/(c - u)]^2 & \text{untuk } b \leq u \leq c \\ 1 & \text{untuk } u > c \end{cases}$$

4. fungsi keanggotaan gaussian



Gambar 2.5. Himpunan keanggotaan gaussian

$$\pi(u; b, c) = \begin{cases} S(u; c-b, c-b/2, c) \\ 1-S(u; c, c+b/2, c+b) \end{cases}$$

Untuk mengerti sistem fuzzy, kita harus mengenal konsep dasar yang berhubungan dengan logika fuzzy.

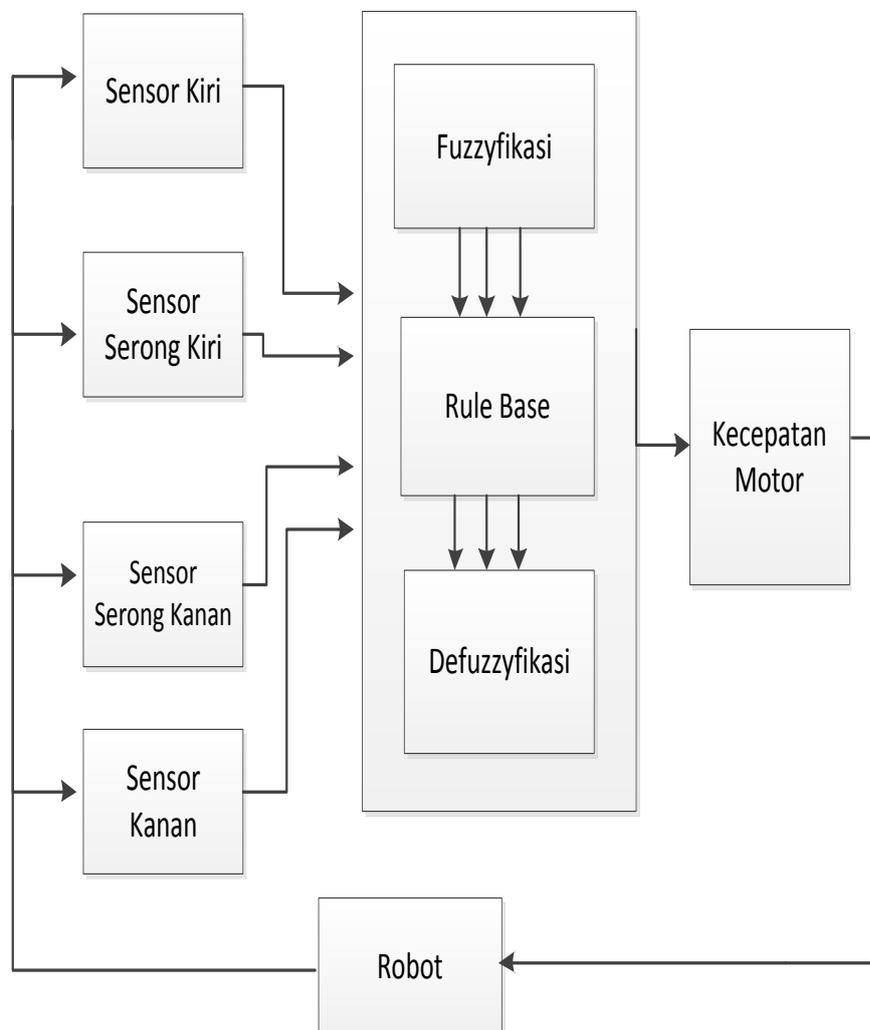
1. Derajat keanggotaan adalah derajat dimana nilai crisp compatible dengan fungsi keanggotaan (dari 0 sampai 1), juga mengacu sebagai tingkat keanggotaan, nilai kebenaran, atau masukan fuzzy.
2. Label adalah nama deskriptif yang digunakan untuk mengidentifikasi sebuah fungsi keanggotaan.
3. Fungsi keanggotaan adalah mendefinisikan fuzzy set dengan memetakan masukan crisp dari domainnya ke derajat keanggotaan.
4. Masukan crisp adalah masukan yang tegas dan tertentu.
5. Lingkup / domain adalah fungsi keanggotaan. Jangkauan konsep, biasanya bilangan tempat dimana fungsi keanggotaan dipetakan.
6. Daerah batasan crisp adalah jangkauan seluruh nilai yang mungkin dapat diaplikasikan pada variabel sistem. menggunakan logika fuzzy untuk mencapai penyelesaian crisp pada masalah khusus biasanya melibatkan tiga langkah : fuzzyfikasi, evaluasi rule, dan defuzzyfikasi.

2.2.2. Cara Kerja Kontrol Logika Fuzzy

Dalam system control logika fuzzy terdapat beberapa tahapan operasional yang meliputi :

1. Fuzzyfikasi
2. Aturan Dasar (Rule Baseed)
3. Defuzzyfikasi

Blok diagram control logika fuzzy ditunjukkan pada gambar berikut :



Gambar 2.6 Blok Diagram Fuzzy Logic Control

2.2.2.1. Fuzzyfikasi

Fuzzifikasi yaitu suatu proses untuk mengubah suatu masukan dari bentuk tegas (crisp) menjadi fuzzy (variabel linguistik) yang biasanya disajikan dalam bentuk himpunan-himpunan fuzzy dengan suatu fungsi keanggotaannya masing-masing. Masukan crisp ditransformasikan kedalam masukan fuzzy. misalkan masukan crisp 9 cm akan ditransformasikan sebagai dekat dalam bentuk fuzzy. Untuk mengubah bentuk masukan crisp kedalam masukan fuzzy, hal yang perlu dilakukan menentukan fungsi keanggotaan untuk tiap masukan. Sekali fungsi keanggotaan ditentukan, fuzzifikasi mengambil nilai masukan secara *realtime*, seperti jarak, dan membandingkannya dengan informasi fungsi keanggotaan yang tersimpan untuk menghasilkan nilai masukan fuzzy.

Langkah pertama dalam fuzzyfikasi adalah menentukan label-label fuzzy pada daerah batasan crisp dari setiap masukan crisp. Masukan fungsi keanggotaan dihasilkan dengan menuliskan satu demi satu bilangan, yaitu derajat keanggotaan, untuk setiap masukan yang mungkin dari label yang diberikan.

2.2.2.2. Evaluasi Rule

Sistem fuzzy menggunakan aturan linguistik untuk menentukan aksi kontrol apa yang harus dilakukan dalam merespon nilai masukan yang diberikan. Evaluasi rule terdiri dari sejumlah aturan yang biasanya dinyatakan secara *linguistic*. Evaluasi rule juga mengacu pada fuzzy interface, mengaplikasikan aturan pada masukan fuzzy yang dihasilkan dalam proses fuzzyfikasi, kemudian mengevaluasi tiap aturan dengan masukan yang dihasilkan dari proses fuzzyfikasi. Aturan fuzzy seringkali dinyatakan dengan “IF... THEN....”.

Pada sistem fuzzy alat ini menggunakan aturan-aturan dalam bentuk bahasa alami yangn dibatasi oleh istilah linguistik, serta sintaksis yang baku. Sintaksi tersebut adalah :

If antecedent 1 And antecedent 2 Then consequent 1 And consequent 2

Dimana :

And adalah salah satu operator logika fuzzy yang diizinkan

Antecedent adalah bentuk dari : Variabel masukan = Label (contohnya : jarak sensor kiri = dekat, dimana jarak sensor kiri adalah variabel masukan dan dekat adalah salah satu label fungsi keanggotaan yang berhubungan dengan jarak sensor kiri).

Consequent adalah bentuk : variabel keluaran = label (contohnya : kecepatan motor = cepat)

Ada 2 jenis proposisi fuzzy yaitu “*condition fuzzy proposition*” dan “*uncondition fuzzy proposition*”.

1. *Condition Fuzzy Proposition*

Jenis ini dicirikan dengan penggunaan IF.

IF x is A THEN y is B

2. *Uncondition Fuzzy Proposition.*

Jenis *uncondition* ditandai dengan tidak adanya pernyataan IF.

x is A

proposisi *uncondition* selalu diaplikasi dengan model AND.

Komposisi aturan logika fuzzy pada laporan ini menggunakan metode sugeno. Metode sugeno ini hampir sama dengan metode mamdani hanya saja keluaran tidak berupa himpunan fuzzy, melainkan berupa konstanta atau persamaan linear. Metode ini diperkenalkan oleh Takagi- Sugeno Kang pada tahun 1985. Menurut Cox (1994), metode TSK terdiri dari 2 jenis yaitu:

1. Model Fuzzy Sugeno Orde-Nol

Secara umum bentuk model fuzzy SUGENO Orde-Nol adalah :

IF (x1 is A1) • (x2 is A2) • (x3 is A3) • • (xN is AN) THEN z=k

dengan A_i adalah himpunan fuzzy ke- i sebagai anteseden, dan k adalah suatu konstanta (tegas) sebagai konsekuen.

2. Model Fuzzy Sugeno Orde-Satu

Secara umum bentuk model fuzzy SUGENO Orde-Satu adalah :

IF (x_1 is A_1) \cdot \cdot (x_N is A_N) THEN $z = p_1 * x_1 + \dots + p_N * x_N + q$

dengan A_i adalah himpunan fuzzy ke- i sebagai anteseden, dan p_i adalah suatu konstanta (tegas) ke- i dan q juga merupakan konstanta dalam konsekuen. Apabila komposisi aturan menggunakan metode SUGENO, maka defuzzifikasi dilakukan dengan cara mencari nilai rata-ratanya (Kusumadewi, 2004).

Pada evaluasi rule menggunakan operasi logika fuzzy. Ada dua operator utama dalam crisp sebagaimana diketahui dengan baik dalam teori set fuzzy adalah IRISAN (dasar untuk operator logika AND) dan GABUNGAN (dasar untuk logika OR). Dengan operator fuzzy AND zadeh, minimum nilai kebenaran antecedent dipilih untuk menentukan rule strength keseluruhan. Dengan operator fuzzy OR zadeh, maksimum nilai kebenaran antecedent dipilih tetapi direkomendasikan menggunakan operator AND dalam keadaan apapun bila memungkinkan. Operator logika fuzzy yang lainnya adalah NOT. jika X mempunyai nilai kebenaran 0.3 maka not X mempunyai kebenaran 0.7 sebagai catatan dalam fungsi tertentu, sebagai contoh, "Dekat" menghasilkan nilai 0.9 pada derajat keanggotaan, maka NOT dekat akan menghasilkan nilai 0.1 pada derajat keanggotaan.

2.2.2.3. Defuzzifikasi

Merupakan proses pemetaan himpunan fuzzy ke himpunan tegas (crisp). Proses ini merupakan kebalikan dari proses fuzzyfikasi. Ada beberapa metode defuzzifikasi antara lain :

1. Metode centroid

metode centroid dilakukan dengan mengambil titik pusat daerah fuzzy. Secara umum dirumuskan dengan :

$$z = \frac{\int z\mu(z)dz}{\int \mu(z)dz} \quad \text{atau} \quad z = \frac{\sum_{j=1}^n z_j\mu(z_j)}{\sum_{j=1}^n \mu(z_j)}$$

2. Metode bisector

metode bisector dilakukan dengan mengambil nilai dari domain fuzzy yang memiliki nilai keanggotaan separo dari nilai keanggotaan fuzzy. Secara dirumuskan dengan :

$$z_p \text{ sedemikian hingga } \int_{R1}^p \mu(z)dz = \int_p^{Rn} \mu(z)dz$$

3. Metode Mean of Maximum (MOM)

metode MOM dilakukan dengan mengambil nilai rata-rata domain yang memiliki nilai keanggotaan maksimum

4. Metode Largest of Minimum (LOM)

metode LOM dilakukan dengan mengambil nilai terbesar dari domain yang memiliki nilai keanggotaan maksimum.

5. Metode Smallest of Maximum (SOM)]

metode LOM dilakukan dengan mengambil nilai terkecil dari domain yang memiliki nilai keanggotaan maksimum.

Contoh :

Pembentukan himpunan *fuzzy*, dimana baik variabel *input* maupun variabel *output* masing-masing dibagi 3 himpunan *fuzzy*. Ada 3 variabel *fuzzy* yang digunakan, yaitu:

- a. Jarak kendaraan adalah jarak antara mobil dengan target, yang terdiri dari 3 himpunan *fuzzy*, yaitu: DEKAT, SEDANG dan JAUH.
- b. Jarak tikungan adalah jarak antara mobil dan tikungan, yang terdiri dari 3 himpunan *fuzzy*, yaitu: DEKAT, SEDANG dan JAUH.

- c. *Output* yang berupa kecepatan ideal suatu mobil, yang terdiri dari 3 himpunan *fuzzy*, yaitu: LAMBAT, SEDANG dan CEPAT.

Data yang akan digunakan untuk membantu penelitian ini adalah data jarak dan kecepatan rata-rata dari suatu mobil yaitu :

- a. Untuk x berupa jarak kendaraan, datanya adalah sebagai berikut:

$$\text{Dekat} : x \leq 5\text{m}$$

$$\text{Jauh} : 15 \text{ m} \leq x \leq 20\text{m}$$

- b. Untuk y berupa jarak tikungan, datanya adalah sebagai berikut:

$$\text{Dekat} : y \leq 5\text{m}$$

$$\text{Jauh} : 10\text{m} \leq y \leq 15\text{m}$$

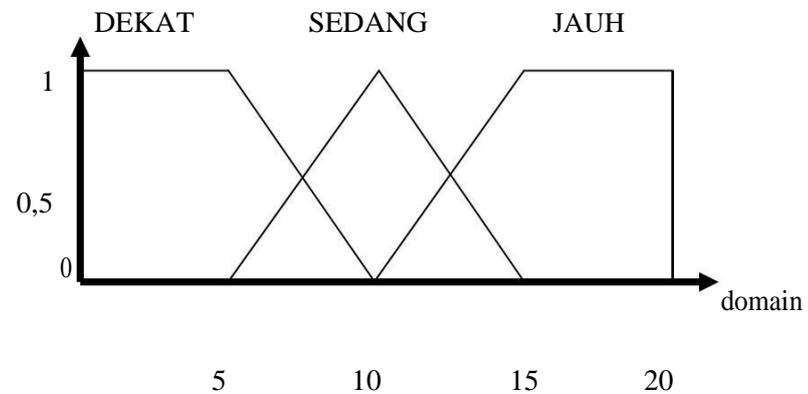
- c. Untuk z berupa kecepatan, datanya adalah sebagai berikut:

$$\text{Lambat} : z \leq 40 \text{ km/jam}$$

$$\text{Cepat} : 80 \text{ km/jam} \leq z \leq 100 \text{ km/jam}$$

Berdasarkan data jarak dan kecepatan, maka dibuatlah sebuah fungsi keanggotaan dari masing-masing variabel yang akan digunakan. Fungsi keanggotaan merupakan suatu kurva yang menunjukkan pemetaan titik-titik input data ke dalam nilai keanggotaannya atau derajat keanggotaan. Adapun fungsi keanggotaan dari masing-masing variabel adalah sebagai berikut:

a. Variabel jarak kendaraan



Gambar 2.7 Fungsi Keanggotaan Variabel Jarak

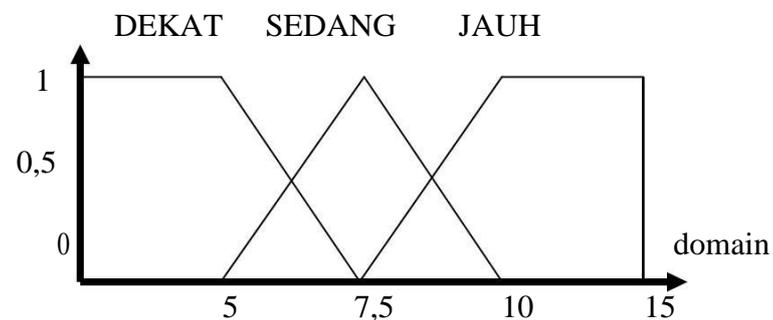
Fungsi keanggotaannya adalah sebagai berikut:

$$\mu_{JKDekat}(x) = \begin{cases} 1 & ; x \leq 5 \\ \frac{(10-x)}{(10-5)} & ; x \leq 5 \leq 10 \\ 0 & ; x \geq 10 \end{cases}$$

$$\mu_{JKSedang}(x) = \begin{cases} 0 & ; x \leq 5 \text{ atau } x \geq 15 \\ \frac{(x-5)}{(10-5)} & ; 5 \leq x \leq 10 \\ \frac{(15-x)}{(15-10)} & ; 10 \leq x \leq 15 \\ 0 & \end{cases}$$

$$\mu_{JKJauh}(x) = \begin{cases} 0 & ; x \leq 10 \\ \frac{(x-10)}{(15-10)} & ; 10 \leq x \leq 15 \\ 1 & ; x \geq 15 \end{cases}$$

b. Variabel jarak tikungan



Gambar 2.8 Fungsi Keanggotaan Variabel Jarak Terhadap Tikungan

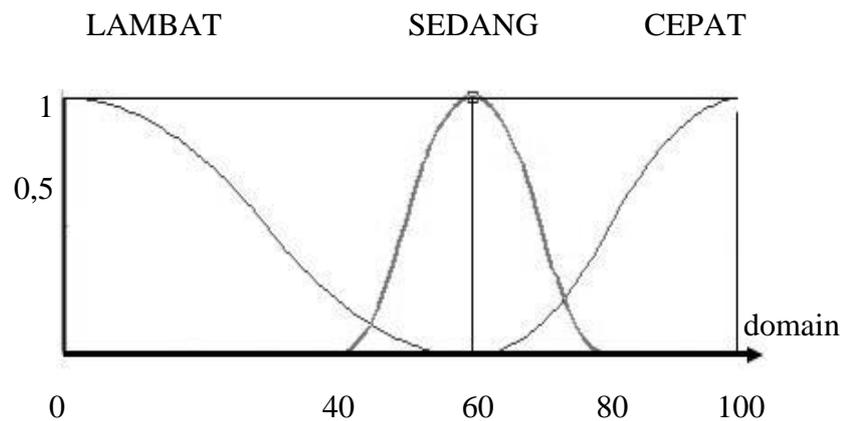
Fungsi keanggotaannya adalah sebagai berikut:

$$\mu_{JKDekat}(x) = \begin{cases} 1 & ; x \leq 5 \\ \frac{(10-x)}{(10-5)} & ; x \leq 5 \leq 10 \\ 0 & ; x \geq 10 \end{cases}$$

$$\mu_{JKSedang}(x) = \begin{cases} 0 & ; x \leq 5 \text{ atau } x \geq 15 \\ \frac{(x-5)}{(10-5)} & ; 5 \leq x \leq 10 \\ \frac{(15-x)}{(15-10)} & ; 10 \leq x \leq 15 \\ 0 & ; x > 15 \end{cases}$$

$$\mu_{JKJauh}(x) = \begin{cases} 0 & ; x \leq 10 \\ \frac{(x-10)}{(15-10)} & ; 10 \leq x \leq 15 \\ 1 & ; x \geq 15 \end{cases}$$

c. Variabel kecepatan



Gambar 2.9. Fungsi Keanggotaan Variabel

Kecepatan Fungsi keanggotaannya adalah sebagai berikut:

$$\mu_{KLambat}(x) = \begin{cases} 1 & ; z \leq 0 \\ 1 - 2 \left(\frac{(z-0)}{(60-0)} \right)^2 & ; 0 \leq z \leq 30 \\ 2 \left(\frac{(60-z)}{(60-0)} \right) & ; 30 \leq z \leq 60 \\ 0 & ; z \geq 60 \end{cases}$$

$$\mu_{KSedang}(x) = \begin{cases} 0 & ; z \leq 40 \text{ atau } z \geq 80 \\ 2 \left(\frac{(z-40)}{(60-40)} \right)^2 & ; 40 \leq z \leq 50 \\ 1 - 2 \left(\frac{(z-0)}{(60-0)} \right)^2 & ; 50 \leq z \leq 60 \\ 1 - 2 \left(\frac{(z-60)}{(80-60)} \right)^2 & ; 60 \leq z \leq 70 \\ 0 & ; 70 \leq z \leq 80 \\ 2 \left(\frac{(80-z)}{(80-60)} \right) & \end{cases}$$

$$\mu_{KCepat}(x) = \begin{cases} 0 & ; z \leq 60 \\ 2 \left(\frac{(z-60)}{(100-60)} \right)^2 & ; 60 \leq z \leq 80 \\ 1 - 2 \left(\frac{(100-z)}{(100-60)} \right)^2 & ; 80 \leq z \leq 100 \\ 1 & ; z \geq 100 \end{cases}$$

Aplikasi fungsi implikasi, dimana pada metode *Mamdani* menggunakan fungsi implikasi *min*. Untuk mendapatkan daerah hasil implikasi diperlukan sejumlah aturan/*rules*. Jumlah aturan yang terbentuk berdasarkan 3 himpunan *fuzzy* adalah sebanyak 27 aturan. Dari 27 aturan hanya 9 aturan yang masuk akal dan layak digunakan, karena jika semua aturan digunakan maka hasilnya jauh dari yang diharapkan. 9 aturan yang digunakan diantaranya adalah sebagai berikut:

- a. Jika (Jarak kendaraan adalah DEKAT) dan (Jarak tikungan adalah DEKAT) maka (Kecepatan adalah LAMBAT)
- b. Jika (Jarak kendaraan adalah DEKAT) dan (Jarak tikungan adalah SEDANG) maka (Kecepatan adalah LAMBAT)
- c. Jika (Jarak kendaraan adalah DEKAT) dan (Jarak tikungan adalah JAUH) maka (Kecepatan adalah LAMBAT)
- d. Jika (Jarak kendaraan adalah SEDANG) dan (Jarak tikungan adalah DEKAT) maka (Kecepatan adalah LAMBAT)
- e. Jika (Jarak kendaraan adalah SEDANG) dan (Jarak tikungan adalah SEDANG) maka (Kecepatan adalah SEDANG)
- f. Jika (Jarak kendaraan adalah SEDANG) dan (Jarak tikungan adalah JAUH) maka (Kecepatan adalah SEDANG)
- g. Jika (Jarak kendaraan adalah JAUH) dan (Jarak tikungan adalah DEKAT) maka (Kecepatan adalah SEDANG)

- h. Jika (Jarak kendaraan adalah JAUH) dan (Jarak tikungan adalah SEDANG) maka (Kecepatan adalah SEDANG)
- i. Jika (Jarak kendaraan adalah JAUH) dan (Jarak tikungan adalah JAUH) maka (Kecepatan adalah CEPAT)

Komposisi antar aturan yang diperoleh dengan cara mengambil nilai minimum aturan, karena menggunakan operator AND. Proses *defuzzifikasi* adalah suatu proses dimana *input* nya adalah suatu himpunan *fuzzy* yang diperoleh dari komposisi aturan-aturan *fuzzy*, sedangkan *output* yang dihasilkan merupakan suatu bilangan pada *domain* himpunan himpunan *fuzzy* tersebut. Sehingga jika diberikan suatu himpunan *fuzzy* dalam *range* tertentu, maka harus dapat diambil suatu nilai *crisp* tertentu sebagai *output*. Pada penelitian ini, proses *defuzzifikasi* menggunakan metode *centroid* dimana nilai tegas diperoleh dengan cara mengambil titik pusat daerah *fuzzy*.

2.2.3. Operasi Himpunan Fuzzy

Himpunan fuzzy dapat dioperasikan dengan beberapa macam yaitu :

Tabel 2.1 Operasi Himpunan Fuzzy

No.	Operasi	Ekspresi operator	Keterangan
1	Equality	$\mu_A(u) = \mu_B(u)$	$u \in U$
2	Union	$\mu_{a \cup b}(u) = \max\{\mu_A(u), \mu_B(u)\}$	$u \in U$
3	Intersection	$\mu_{a \cap b}(u) = \min\{\mu_A(u), \mu_B(u)\}$	$u \in U$
4	Complement	$\mu_A(u) = 1 - \mu_A(u)$	$u \in U$
5	Normalitation	$\mu_{\text{norm}(A)}(u) = \mu_A(u) / \max(\mu_A(u))$	$u \in U$
6	Concentration	$\mu_{\text{con}A}(u) = (\mu_A(u))^2$	$u \in U$
7	Dilatation	$\mu_{\text{dil}A}(u) = (\mu_A(u))^{0.5}$	$u \in U$
8	Intensification	$\mu_{\text{int}(A)}(u) = \begin{cases} 2(\mu_A(u))^2 \\ 1 - 2(1 - \mu_A(u))^2 \end{cases}$	
9	Algebraic product	$\mu_{A \cdot B}(u) = \mu_A(u) \cdot \mu_B(u)$	$u \in U$
10	Bounded sum	$\mu_{A+B}(u) = \min\{1, \mu_A(u) + \mu_B(u)\}$	$u \in U$
11	Bounded product	$\mu_{A \times B}(u) = \max\{0, \mu_A(u) + \mu_B(u) - 1\}$	$u \in U$
12	Drastic product	$\mu_{A \times B}(u) = \begin{cases} \mu_A(u) \\ \mu_B(u) \\ 0 \end{cases}$ <i>untuk</i> $\mu_B(u) = 1$ <i>untuk</i> $\mu_A(u) = 1$ <i>untuk</i> $\mu_A(u), \mu_B(u) < 1$	

2.3. Robot Wall Follower (Bergerak)

Robot bergerak atau mobile robot adalah robot yang dapat bergerak atau berpindah tempat dengan menggunakan roda atau tiruan bentuk kaki (Suradana dan Sudiarsa, 2013:96). Penggunaan roda atau tiruan bentuk kaki pada robot bergerak, mempunyai keuntungan sendiri-sendiri sesuai dengan tujuannya. Pada penggerak dengan roda akan memberikan kecepatan dalam melintasi bidang yang rata, dan kemudahan dalam desain serta implementasi, sedangkan penggerak pada robot dengan tiruan bentuk kaki akan memudahkan robot untuk bergerak di daerah yang halus atau kasar, memanjat tangga, menghindar, dan melangkah di atas halangan.

Robot Wall Follower termasuk dalam robot yang bergerak, yaitu robot dengan roda sebagai alat geraknya dan sensor ultrasonik sebagai indra penunjuk jalan dalam

menelusuri dinding. Robot berjalan dengan pembacaan jarak antara dinding ke sensor ultrasonik pada robot, oleh sebab itulah dinamakan Robot Wall Follower atau robot pengikut dinding.

2.3.1. Penyusun Robot Wall Follower

Robot *Wall Follower* ini komponen utamanya yaitu sensor ultrasonik, Arduino Mega 2560 dan aktuator. Penyusun lengkap robot *wall follower* dapat dilihat sebagai berikut.

2.3.1.1. Liquid Cristal Display (LCD)

LCD adalah suatu display dari bahan cairan kristal yang pengoperasiannya menggunakan sistem dot matriks. LCD banyak digunakan sebagai display dari alat-alat elektronika seperti kalkulator, multimeter digital, jam digital dan sebagainya. LCD menggunakan *Inter Integrated Circuit* atau sering disebut I2C. I2C adalah standar komunikasi serial dua arah menggunakan dua saluran yang didesain khusus untuk mengirim maupun menerima data. Adapun Karakteristik I2C LCD yaitu :

1. Serial Bus
Data dikirim serial secara per-bit.
2. Menggunakan dua Penghantar Koneksi dengan ground bersama
I2C terdiri dari dua penghantar:
 - a. SCL (*Serial Clock Line*) untuk menghantarkan sinyal clock.
 - b. SDA (*Serial Data*) untuk mentransaksikan data
3. Jumlah Peserta Bus maximal 127
Peserta dialamatkan melalui 7-bit-alamat. Alamat ditetapkan kebanyakan secara hardware dan hanya sebagian kecil dapat dirubah.
4. Pengirim dan Penerima
Setiap transaksi data terjadi antara pengirim (*Transmitter*) dan penerima (*Receiver*). Pengirim dan penerima adalah peserta bus.

5. *Master dan Slave*

Device yang mengendalikan operasi transfer disebut *Master*, sementara device yang di kendalikan oleh master di sebut *Slave*. Device yang mengendalikan operasi transfer data disebut *master*, sedangkan device lainnya yang dikendalikan oleh master disebut *slave*. *Master device* harus menghasilkan serial clock melalui pin SCL, mengendalikan akses ke BUS serial dan menghasilkan sinyal kendali START dan STOP.

Adapun definisi-definisi Kondisi Bus, yaitu :

1. *Bus not busy*

Pada saat ini Bus tidak sibuk, SCL dan SDA dua-duanya dalam keadaan HIGH.

2. *Start data transfer*

Ditandai dengan perubahan kondisi SDA dari HIGH ke LOW ketika SCL HIGH.

3. *Stop data transfer*

Ditandai dengan perubahan kondisi SDA dari LOW ke HIGH ketika SCL HIGH.

4. *Data valid*

Data yang dikirim bit demi bit dianggap valid jika setelah START, kondisi SDA tidak berubah selama SCL HIGH, baik SDA HIGH maupun SDA LOW tergantung dari bit yang ingin ditransfer. Setiap siklus HIGH SCL baru menandakan pengiriman bit baru. Duty cycle untuk SCL tidak mesti 50%, tetapi frekuensi kemunculannya hanya ada 2 macam, yaitu mode standar 100kHz dan fast mode atau mode cepat 400kHz. Setelah SCL mengirimkan sinyal HIGH yang kedelapan, arah transfer SDA berubah, sinyal kesembilan pada SDA ini dianggap sebagai acknowledge dari receiver ke transmitter.

5. *Acknowledge*

Setiap receiver wajib mengirimkan sinyal acknowledge atau sinyal balasan setiap selesai pengiriman 1-byte atau 8-bit data. Master harus memberikan ekstra clock pada SCL, yaitu clock kesembilan untuk memberikan kesempatan receiver mengirimkan sinyal acknowledge ke transmitter berupa keadaan LOW pada SDA selama SCL HIGH. Meskipun master berperan sebagai receiver, ia tetap sebagai penentu sinyal STOP. Pada bit-akhir penerimaan byte terakhir,

master tidak mengirimkan sinyal acknowledge, SDA dibiarkan HIGH oleh receiver dalam hal ini master, kemudian master mengubah SDA dari LOW menjadi HIGH yang berarti sinyal STOP.

Mode Pengoperasian Transfer Data tergantung kondisi bit R/W, 2 jenis transfer dimungkinkan, yaitu :

1. *Data transfer from a master transmitter to a slave receiver*

Byte pertama yang dikirimkan oleh *master* adalah alamat *slave*, setelah itu *master* mengirimkan sejumlah byte data. *Slave* atau receiver mengirimkan sinyal acknowledge setiap kali menerima 1-byte data. Pada tiap byte, bit pertama yang dikirim adalah MSB.

2. *Data transfer from a slave transmitter to a master receiver*

Meskipun *master* berperan sebagai receiver, byte pertama dikirimkan oleh *master* berupa alamat *slave*. Setelah itu *slave* mengirimkan bit *acknowledge*, dilanjutkan dengan pengiriman sejumlah byte dari *slave* ke *master*. *Master* mengirimkan bit acknowledge untuk setiap byte yang diterimanya, kecuali byte terakhir. Pada akhir byte, *master* mengirimkan sinyal 'not acknowledge', setelah itu *master* mengirimkan sinyal STOP.



Gambar 2.10. LCD dan I2C

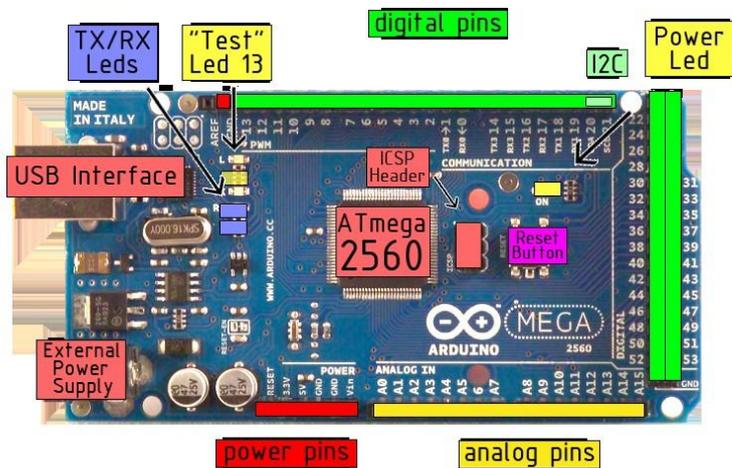
2.3.1.2. Arduino Mega2560

Arduino Mega2560 adalah papan mikrokontroler berbasis ATmega2560 (datasheet). Arduino Mega2560 memiliki 54 pin digital input/output, dimana 15 pin dapat digunakan sebagai output PWM, 16 pin sebagai input analog, dan 4 pin sebagai UART (port serial hardware), 16 MHz kristal osilator, koneksi USB, jack power, header ICSP, dan tombol reset. Ini semua yang diperlukan untuk mendukung mikrokontroler. Cukup dengan menghubungkannya ke komputer melalui kabel USB atau power dihubungkan dengan adaptor AC-DC atau baterai untuk mulai mengaktifkannya. Arduino Mega2560 kompatibel dengan sebagian besar shield yang dirancang untuk Arduino Duemilanove atau Arduino Diecimila. Arduino Mega2560 adalah versi terbaru yang menggantikan versi Arduino Mega.

Arduino Mega2560 berbeda dari papan sebelumnya, karena versi terbaru sudah tidak menggunakan chip driver FTDI USB-to-serial. Tapi, menggunakan chip ATmega16U2 (ATmega8U2 pada papan Revisi 1 dan Revisi 2) yang diprogram sebagai konverter USB-to-serial. Arduino Mega2560 Revisi 2 memiliki resistor penarik jalur HWB 8U2 ke Ground, sehingga lebih mudah untuk dimasukkan ke dalam mode DFU.

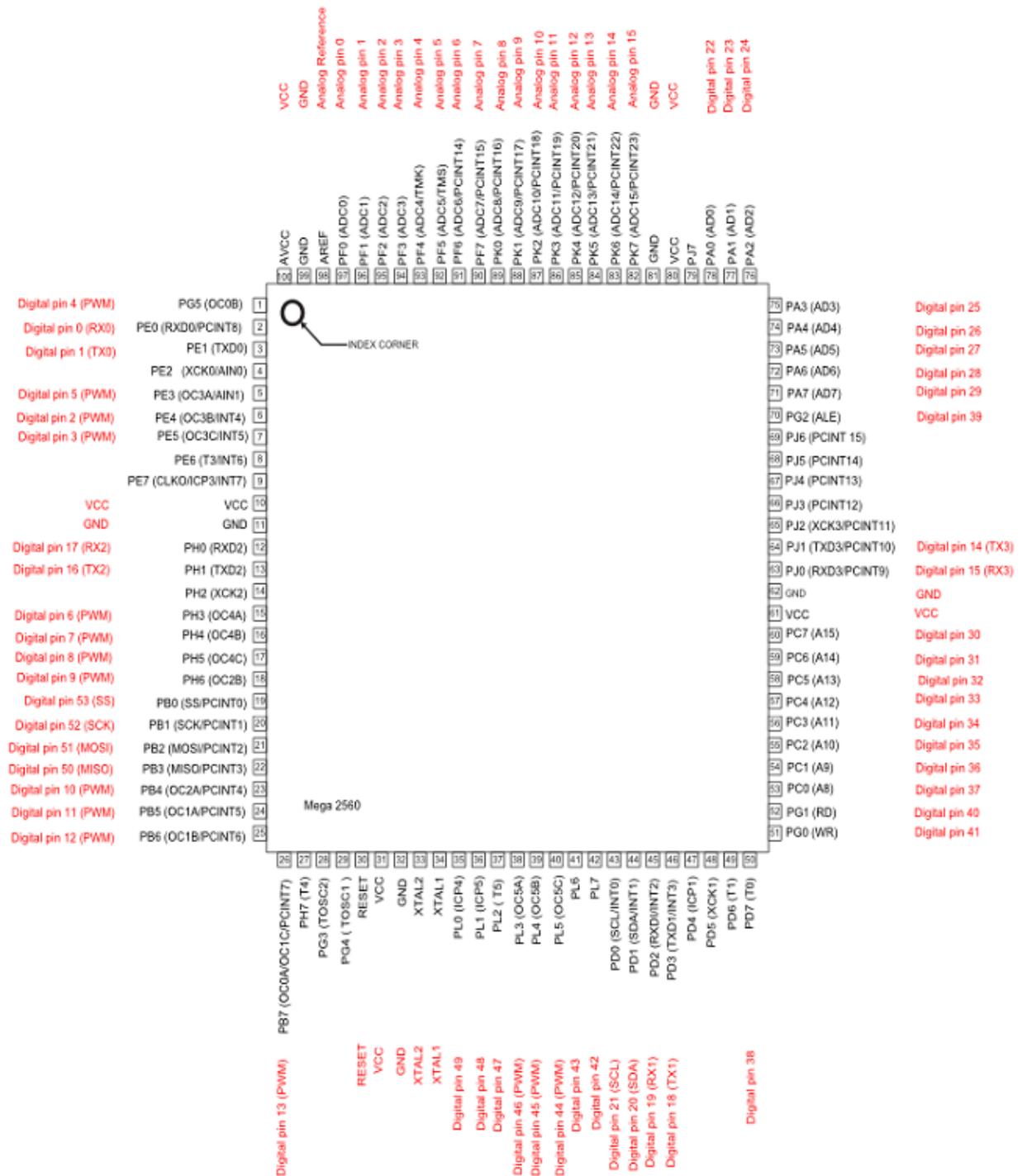
Arduino Mega2560 Revisi 3 memiliki fitur-fitur baru berikut:

1. 1.0 pinout : Ditambahkan pin SDA dan pin SCL yang dekat dengan pin AREF dan dua pin baru lainnya ditempatkan dekat dengan pin RESET, IOREF memungkinkan shield untuk beradaptasi dengan tegangan yang tersedia pada papan. Di masa depan, shield akan kompatibel baik dengan papan yang menggunakan AVR yang beroperasi dengan 5 Volt dan dengan Arduino Due yang beroperasi dengan tegangan 3.3 Volt. Dan ada dua pin yang tidak terhubung, yang disediakan untuk tujuan masa depan.
2. Sirkuit RESET.
3. Chip ATmega16U2 menggantikan chip ATmega8U2.



Gambar 2.11 Arduino Mega2560

Adapun dibawah ini pemetaan dari pin-pin ATmega2560 dengan Arduino Mega2560.



Gambar 2.12 Pemetaan Pin ATmega2560

Dibawah ini spesifikasi sederhana dari Arduino Mega2560 dapat dilihat pada tabel 2.2.

Tabel 2.2 spesifikasi sederhana arduino

Mikrokontroler	ATmega 2560
Tegangan Operasi	5V
Input Voltage (disarankan)	7-12V
Input Voltage (Limit)	6-20V
Pin Digital I/O	54 (15 Pin digunakan sebagai output PWM)
Pins Input Analog	16
Arus DC per pin I/O	40 mA
Arus DC per pin 3.3 V	50 mA
Flash Memory	256 KB (8 KB digunakan untuk bootloade)
SRAM	8 KB
EEPROM	4 KB
Clock Speed	16 MHz

2.2.5.2.1. Sumber Daya Arduino Mega2560

Arduino Mega dapat diaktifkan melalui koneksi USB atau dengan catu daya eksternal. Sumber daya dipilih secara otomatis. Sumber daya eksternal (non-USB) dapat berasal baik dari adaptor AC-DC atau baterai. Adaptor dapat dihubungkan dengan mencolokkan steker 2,1 mm yang bagian tengahnya terminal positif ke ke jack sumber tegangan pada papan. Jika tegangan berasal dari baterai dapat langsung dihubungkan melalui header pin Gnd dan pin Vin dari konektor POWER.

Papan Arduino ATmega2560 dapat beroperasi dengan pasokan daya eksternal 6 Volt sampai 20 volt. Jika diberi tegangan kurang dari 7 Volt, maka, pin 5 Volt mungkin akan menghasilkan tegangan kurang dari 5 Volt dan ini akan membuat papan menjadi tidak stabil. Jika sumber tegangan menggunakan lebih dari 12 Volt,

regulator tegangan akan mengalami panas berlebihan dan bisa merusak papan. Rentang sumber tegangan yang dianjurkan adalah 7 Volt sampai 12 Volt.

Pin tegangan yang tersedia pada papan Arduino adalah sebagai berikut:

1. **VIN** : Adalah input tegangan untuk papan Arduino ketika menggunakan sumber daya eksternal (sebagai 'saingan' tegangan 5 Volt dari koneksi USB atau sumber daya ter-regulator lainnya). Anda dapat memberikan tegangan melalui pin ini, atau jika memasok tegangan untuk papan melalui jack power, kita bisa mengakses/mengambil tegangan melalui pin ini.
2. **5V** : Sebuah pin yang mengeluarkan tegangan ter-regulator 5 Volt, dari pin ini tegangan sudah diatur (ter-regulator) dari regulator yang tersedia (built-in) pada papan. Arduino dapat diaktifkan dengan sumber daya baik berasal dari jack power DC (7-12 Volt), konektor USB (5 Volt), atau pin VIN pada board (7-12 Volt). Memberikan tegangan melalui pin 5V atau 3.3V secara langsung tanpa melewati regulator dapat merusak papan Arduino.
3. **3.3 V** : Sebuah pin yang menghasilkan tegangan 3,3 Volt. Tegangan ini dihasilkan oleh regulator yang terdapat pada papan (on-board). Arus maksimum yang dihasilkan adalah 50 mA.
4. **GND** : Pin Ground atau Massa.
5. **IOREF** : Pin ini pada papan Arduino berfungsi untuk memberikan referensi tegangan yang beroperasi pada mikrokontroler. Sebuah perisai (shield) dikonfigurasi dengan benar untuk dapat membaca pin tegangan IOREF dan memilih sumber daya yang tepat atau mengaktifkan penerjemah tegangan (voltage translator) pada output untuk bekerja pada tegangan 5 Volt atau 3,3 Volt. (www.arduino.cc)

2.2.5.2.2. Input dan Output Arduino Mega2560

Masing-masing dari 54 digital pin pada Arduino Mega dapat digunakan sebagai input atau output, menggunakan fungsi `pinMode()` , `digitalWrite()` , dan `digitalRead()`. Arduino Mega beroperasi pada tegangan 5 volt. Setiap pin dapat memberikan atau menerima arus maksimum 40 mA dan memiliki resistor pull-up

internal (yang terputus secara default) sebesar 20-50 kOhms. Selain itu, beberapa pin memiliki fungsi khusus, antara lain:

1. **Serial** : 0 (RX) dan 1 (TX); **Serial 1** : 19 (RX) dan 18 (TX); **Serial 2** : 17 (RX) dan 16 (TX); **Serial 3** : 15 (RX) dan 14 (TX). Digunakan untuk menerima (RX) dan mengirimkan (TX) data serial TTL. Pins 0 dan 1 juga terhubung ke pin chip ATmega16U2 Serial USB-to-TTL.
2. **Eksternal Interupsi** : Pin 2 (interrupt 0), pin 3 (interrupt 1), pin 18 (interrupt 5), pin 19 (interrupt 4), pin 20 (interrupt 3), dan pin 21 (interrupt 2). Pin ini dapat dikonfigurasi untuk memicu sebuah interupsi pada nilai yang rendah, meningkat atau menurun, atau berubah nilai.
3. **SPI** : Pin 50 (MISO), pin 51 (MOSI), pin 52 (SCK), pin 53 (SS). Pin ini mendukung komunikasi SPI menggunakan perpustakaan SPI. Pin SPI juga terhubung dengan header ICSP, yang secara fisik kompatibel dengan Arduino Uno, Arduino Duemilanove dan Arduino Diecimila.
4. **LED** : Pin 13. Tersedia secara built-in pada papan Arduino ATmega2560. LED terhubung ke pin digital 13. Ketika pin diset bernilai HIGH, maka LED menyala (ON), dan ketika pin diset bernilai LOW, maka LED padam (OFF).
5. **TWI** : Pin 20 (SDA) dan pin 21 (SCL). Yang mendukung komunikasi TWI menggunakan perpustakaan Wire. Perhatikan bahwa pin ini tidak di lokasi yang sama dengan pin TWI pada Arduino Duemilanove atau Arduino Diecimila.

Arduino Mega2560 memiliki 16 pin sebagai analog input, yang masing-masing menyediakan resolusi 10 bit (yaitu 1024 nilai yang berbeda). Secara default pin ini dapat diukur/diatur dari mulai Ground sampai dengan 5 Volt, juga memungkinkan untuk mengubah titik jangkauan tertinggi atau terendah mereka menggunakan pin AREF dan fungsi `analogReference()`.

Ada beberapa pin lainnya yang tersedia, antara lain:

1. **AREF** : Referensi tegangan untuk input analog. Digunakan dengan fungsi `analogReference()`.

2. **RESET** : Jalur LOW ini digunakan untuk me-reset (menghidupkan ulang) mikrokontroler. Jalur ini biasanya digunakan untuk menambahkan tombol reset pada shield yang menghalangi papan utama Arduino. (www.arduino.cc)

2.2.5.2.3. Pemrograman Arduino Mega2560

Arduino Mega dapat diprogram dengan software Arduino. ATmega2560 pada Arduino Mega sudah tersedia preburned dengan bootloader yang memungkinkan Anda untuk meng-upload kode baru tanpa menggunakan programmer hardware eksternal. Hal ini karena komunikasi yang terjadi menggunakan protokol asli STK500. Anda juga dapat melewati (bypass) bootloader dan program mikrokontroler melalui pin header ICSP (In-Circuit Serial Programming).

Chip ATmega16U2 (atau 8U2 pada board Rev. 1 dan Rev. 2) source code firmware tersedia pada repositori Arduino. ATmega16U2/8U2 dapat dimuat dengan bootloader DFU, yang dapat diaktifkan melalui:

1. **Pada papan Revisi 1** : Menghubungkan jumper solder di bagian belakang papan (dekat dengan peta Italia) dan kemudian akan me-reset 8U2.
2. **Pada papan Revisi 2** : Ada resistor yang menghubungkan jalur HWB 8U2/16U2 ke ground, sehingga lebih mudah untuk dimasukkan ke dalam mode DFU.

Kemudian Anda dapat menggunakan Atmel FLIP software (sistem operasi Windows) atau DFU programmer (sistem operasi Mac OS X dan Linux) untuk memuat firmware baru. Atau Anda dapat menggunakan pin header ISP dengan programmer eksternal (overwrite DFU bootloader). (www.arduino.cc)

2.3.1.3. Sensor Flame

Jenis sensor yang digunakan untuk deteksi jarak dekat api dan dapat digunakan untuk memantau proyek-proyek atau sebagai tindakan pencegahan keamanan untuk memotong perangkat off / on. Sensor nyala api sangat sensitif terhadap IR panjang gelombang pada 760 nm ~ 1100 nm cahaya.



Gambar 2.13 Sensor Flame

Tabel 2.3 Pin dan Kegunaan Sensor Flame

No	Pin	Kegunaan
1	VCC	Tegangan Input Positif
2	DO (Digital Out)	Bila suhu mencapai batas tertentu, output tinggi dan rendah ambang sinyal diatur melalui potensiometer.
3	GND	Tegangan Negatif

2.3.1.4. Sensor Ultrasonik HC-SR04

Sensor HC-SR04 adalah sensor pengukur jarak berbasis gelombang ultrasonik. Prinsip kerja sensor ini mirip dengan radar ultrasonik. Gelombang ultrasonik dipancarkan kemudian diterima balik oleh receiver ultrasonik. Jarak antara waktu pancar dan waktu terima adalah representasi dari jarak objek. Sensor HC-SR04 adalah versi low cost dari sensor ultrasonik PING buatan parallax. Perbedaannya terletak pada pin yang digunakan. HC-SR04 menggunakan 4 pin sedangkan PING buatan parallax menggunakan 3 pin, seperti pada gambar 2.13. berikut.



Gambar 2.14. sensor ultrasonik HC-SR04

Cara kerja sensor ultrasonik adalah sebagai berikut:

1. Sinyal dipancarkan oleh pemancar ultrasonik dengan frekuensi tertentu dan dengan durasi waktu tertentu. Sinyal tersebut berfrekuensi diatas 20kHz. Untuk mengukur jarak benda (sensor jarak), frekuensi yang umum digunakan adalah 40kHz.
2. Sinyal yang dipancarkan akan merambat sebagai gelombang bunyi dengan kecepatan sekitar 340 m/s. Ketika menumbuk suatu benda, maka sinyal tersebut akan dipantulkan oleh benda tersebut.
3. Setelah gelombang pantulan sampai di alat penerima, maka sinyal tersebut akan diproses untuk menghitung jarak benda tersebut. Jarak benda dihitung berdasarkan rumus :

$$S = \frac{340 \times t}{2} \dots\dots\dots(2.1)$$

dimana S merupakan jarak antara sensor ultrasonik dengan benda (bidang pantul), dan t adalah selisih antara waktu pemancaran gelombang oleh transmitter dan waktu ketika gelombang pantul diterima receiver.

Gelombang ultrasonik adalah gelombang bunyi yang mempunyai frekuensi sangat tinggi yaitu 20.000 Hz. Bunyi ultrasonik tidak dapat di dengar oleh telinga manusia. Bunyi ultrasonik dapat didengar oleh anjing, kucing, kelelawar, dan lumba-lumba. Bunyi ultrasonik nisa merambat melalui zat padat, cair dan gas. Reflektivitas bunyi ultrasonik di permukaan zat padat hampir sama dengan reflektivitas bunyi ultrasonik di permukaan zat cair. Akan tetapi, gelombang bunyi ultrasonik akan diserap oleh tekstil dan busa.

Sensor ini merupakan sensor ultrasonik siap pakai, satu alat yang berfungsi sebagai pengirim, penerima, dan pengontrol gelombang ultrasonik. Alat ini bisa digunakan untuk mengukur jarak benda dari 2cm - 4m dengan akurasi 3mm. Alat ini memiliki 4 pin yaitu pin Vcc, Gnd, Trigger, dan Echo. Pin Vcc untuk listrik positif dan Gnd untuk ground-nya. Pin Trigger untuk trigger keluarnya sinyal dari sensor dan pin Echo untuk menangkap sinyal pantul dari benda.

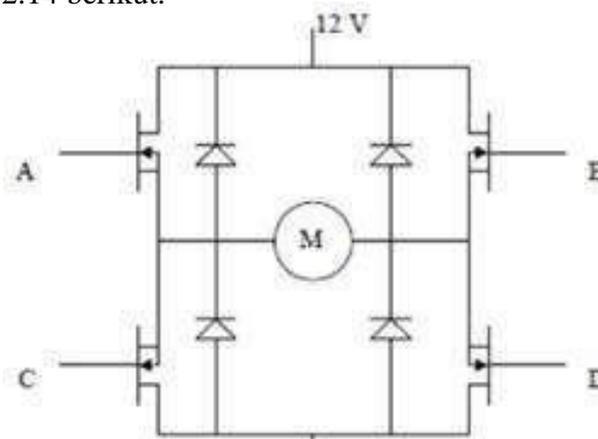
Cara menggunakan alat ini yaitu: ketika kita memberikan tegangan positif pada pin Trigger selama 10uS, maka sensor akan mengirimkan 8 step sinyal

ultrasonik dengan frekuensi 40kHz. Selanjutnya, sinyal akan diterima pada pin Echo. Untuk mengukur jarak benda yang memantulkan sinyal tersebut, maka selisih waktu ketika mengirim dan menerima sinyal digunakan untuk menentukan jarak benda tersebut. Rumus untuk menghitungnya sudah saya sampaikan di atas.

2.3.1.5. L298N Dual H-Bridge Motor Controller Module

H-Bridge ini biasanya digunakan dalam mengendalikan kecepatan dan arah motor, tetapi dapat digunakan untuk proyek-proyek lainnya seperti mengemudi kecerahan proyek pencahayaan tertentu seperti tinggi array LED bertenaga.

. Disebut *H-bridge* karena bentuk rangkaiannya yang menyerupai huruf H seperti pada Gambar 2.14 berikut.



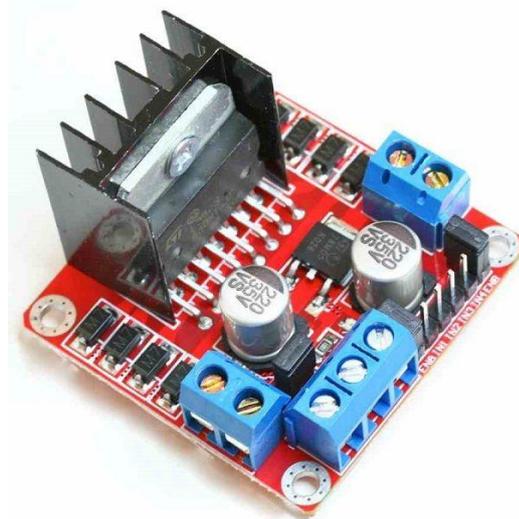
Gambar 2.15. Konfigurasi *H-Bridge* MOSFET

Rangkaian ini terdiri dari dua buah MOSFET kanal P dan dua buah MOSFET kanal N. Prinsip kerja rangkaian ini adalah dengan mengatur mati hidupnya ke empat MOSFET tersebut. Huruf M pada gambar adalah motor DC yang akan dikendalikan. Bagian atas rangkaian akan dihubungkan dengan sumber daya kutub positif, sedangkan bagian bawah rangkaian akan dihubungkan dengan sumber daya kutub negatif.

H-Bridge adalah sirkuit yang dapat mendorong arus baik polaritas dan dikendalikan oleh * Pulse Width Modulation (PWM). Daftar pin PWM untuk dua jenis utama Arduino digunakan:

1. AT MEGA - PWM: 2-13 dan 44 untuk 46. Memberikan output PWM 8-bit dengan fungsi `analogWrite ()`.

2. UNO - PWM: 3, 5, 6, 9, 10, dan 11. Memberikan output PWM 8-bit dengan fungsi `analogWrite ()`.



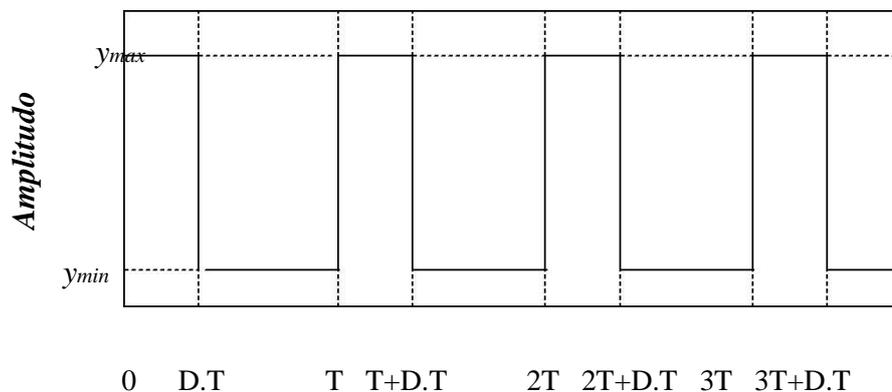
Gambar 2.16 Motor Driver L298N Module

Tabel 2.4 Pin dan fungsi Motor Driver L298N Module

No.	Pin	Fungsi
1	Out 1	Motor A mengarah
2	Out 2	Motor A mengarah
3	Out 3	Motor B memimpin keluar
4	Out 4	Motor B memimpin keluar (Bisa benar-benar berasal dari 5V-35v, hanya ditandai sebagai 12v)
5	GND	Ground
6	5V	Masukan 5V (tidak perlu jika sumber daya Anda 7V-35v, jika sumber daya 7V- 35v maka dapat bertindak sebagai 5V keluar)
7	Ena	Memungkinkan sinyal PWM untuk motor A
8	In1	Aktifkan motor A
9	In2	Aktifkan motor A
10	In3	Aktifkan motor B
11	In4	Aktifkan motor B
12	ENB	Memungkinkan sinyal PWM untuk motor B

2.3.1.6. PWM (Pulse Width Modulation)

Pulse Width Modulation merupakan sarana dalam mengendalikan durasi pulsa elektronik. Dalam motor coba bayangkan sikat sebagai roda air dan elektron sebagai tetesan mengalir air. tegangan akan menjadi air mengalir di atas roda dengan laju yang konstan, semakin banyak air yang mengalir semakin tinggi tegangan. Motor yang dinilai pada tegangan tertentu dan dapat rusak jika tegangan diterapkan untuk berat atau jika terjatuh cepat untuk memperlambat motor turun. Jadi PWM. Mengambil analogi roda air dan memikirkan air memukul di pulsa tetapi pada aliran konstan. Semakin lama pulsa yang cepat roda akan berubah, semakin pendek pulsa, semakin lambat roda air akan berubah. Motor akan bertahan lebih lama dan lebih dapat diandalkan jika dikontrol melalui PWM seperti dapat dilihat pada gambar 2.15.



Gambar 2.17. Gelombang kotak yang memiliki y_{min} , y_{max} dan D

2.3.1.7. DC-DC Step Down Module

DC to DC Converter ada 2 jenis, yaitu Step down DC Converter dan boost (step up) DC Converter. Pada step down DC Converter berfungsi seperti regulator biasa, menurunkan tegangan menjadi level tertentu dengan kestabilan tinggi, biasanya ripple maximum 1%. Dan input tegangan dengan range yg lebar. Pada 12A Adjustable DC-DC Step Down Module input voltage 4,5 - 30 VDC, output voltage 0.8-28 VDC 12A.



Gambar 2.18. 12A Adjustable DC-DC Step Down Module

Dan sebaliknya boost (Step up) DC Converter, berfungsi untuk menaikkan atau menstabilkan tegangan pada level tertentu dengan output yg bisa lebih tinggi dari pada input, dan tetap stabil dengan ripple max 1%. Misalnya DC-DC Converter Input 10VDC sd 60VDC output 24VDC 10A atau Input 3VDC sd 15VDC output dual 15VDC 500mA.

2.3.1.8. Motor DC

Motor DC memerlukan suplai tegangan yang searah pada kumparan medan untuk diubah menjadi energi mekanik. Bagian utama motor DC adalah stator dan rotor dimana kumparan medan pada motor dc disebut stator (bagian yang tidak berputar).



Gambar 2.19. Motor DC

Prinsip dasar cara kerja motor DC yaitu jika arus lewat pada suatu konduktor, timbul medan magnet disekitar konduktor. Arah medan magnet ditentukan oleh aliran arus pada konduktor.