

BAB II

TINJAUAN UMUM

2.1 Definisi Robot

Kata Robot berasal dari bahasa Ceko Slowakia, yakni *robota*, yang berarti pekerja. Robot diciptakan atas dasar untuk mendukung dan membantu pekerjaan manusia. Istilah ini dikemukakan oleh Karel Čapek pada tahun 1920. Karyanya pada saat itu berjudul "*Rossum's Universal Robots*" yang artinya Robot Dunia milik Rossum.

Robotik adalah ilmu yang mematerikan kecerdasan atau intelegensia terhadap energi, artinya pengendalian secara cerdas terhadap gerakan yang terkoordinasi secara nyata. Kata *Robotics* juga berasal dari novel fiksi sains "*Runaround*" yang ditulis oleh Isaac Asimov pada tahun 1942. Isaac Asimov mengajukan ada 3 hukum dari "*robotics*" dimana berguna untuk melindungi kita dari kecerdasan para robot.

Pengertian robot secara tepat adalah sistem atau alat yang dapat berperilaku atau meniru perilaku manusia dengan tujuan untuk menggantikan dan mempermudah kerja/aktifitas manusia. Untuk dapat diklasifikasikan sebagai robot, mesin harus memiliki dua macam kemampuan yaitu bisa mendapatkan informasi dari sekelilingnya dan bisa melakukan sesuatu secara fisik seperti bergerak atau memanipulasi objek (Christian, 2009).

Berdasarkan Robotics Institute of America (RIA), definisi robot adalah *A robot is a reprogrammable multifunctional manipulator designed to move material, parts, tools, or specialized devices through variable programmed motions for the performance of a variety of tasks*. Bahwa robot adalah sebuah manipulator yang multifungsi dan bekerja untuk sebuah keahlian yang khusus, dimana robot juga harus dapat diprogram ulang.

Dari sudut pandang *engineering*, robot adalah sebuah benda kompleks dimana didalamnya terdapat struktur mekanik, kumpulan sistem sensorik dan sebuah sistem kontrol.

Menurut Japanese Industrial Robot Association (JIRA) robot diklasifikasikan menjadi 5 (lima) bagian (Suwito, 2009), yaitu :

1. *Play-back Robot*, adalah robot yang menjalankan fungsi-fungsi berulang yang telah direkamkan kepadanya, biasanya memiliki sistem kontrol open loop.
2. *Robots controlled by sensors*, adalah robot yang memiliki loopback yang diakibatkan oleh gerakan dan membuat keputusan berdasarkan data yang diperoleh dari sensor.
3. *Robot Vision* adalah robot yang informasi diperoleh dari sistem vision, dimana robot dapat memanipulasi objek yang ditangkap.
4. *Robot controlled adaptably* adalah robot yang dapat secara otomatis memprogram aksinya sendiri berdasarkan data yang diperoleh dari sensor.
5. *Intelligent Robot*, yaitu robot yang menggunakan teknik artificial intelligence untuk membuat keputusan dan mampu memecahkan permasalahannya sendiri.

2.1.1 Robot Sumo

Robot sumo atau pepe sumo adalah olahraga di mana dua robot berusaha untuk mendorong satu sama lain keluar dari lingkaran (dengan cara yang sama dengan olahraga sumo). Robot digunakan dalam kompetisi ini disebut sumobots. Tantangan rekayasa adalah untuk robot untuk menemukan lawan (biasanya dilakukan dengan sensor inframerah atau ultrasonik) dan mendorongnya keluar dari arena datar. Sebuah robot juga harus menghindari meninggalkan arena, biasanya dengan cara sensor yang mendeteksi tepi. Yang paling umum "senjata" yang digunakan dalam kompetisi sumobots adalah pisau miring di depan robot, biasanya miring sekitar sudut 45 derajat ke arah belakang robot. Pisau ini memiliki tinggi disesuaikan untuk taktik yang berbeda.

Dalam hal ini, robot sumo akan mendeteksi keberadaan lawan atau musuh dengan menginisialkan musuh tersebut sebagai benda mati yang akan menggantikan musuh yang dicari. Robot ini akan mendekati musuhnya jika terdeteksi oleh sensor ultrasonik dan berputar sesuai dengan kompas penunjuk arah

yang ada dimana kompas tersebut menemukan musuh. Robot sumo biasanya dipakai dalam pertandingan internasional yang memperagakan layaknya pertandingan sumo di Jepang.

Robot sumo dirancang dengan kemampuan untuk mendeteksi suatu objek kemudian menyerang objek tersebut dengan mendorongnya. Untuk mendeteksi ada atau tidak halangan didepan robot maka pada bagian robot diberikan suatu sensor yang dapat mendeteksi halangan. Robot sumo akan menyerang halangan apabila halangan tersebut benar-benar berada didepan robot sumo. Jika halangan belum berada tepat didepan robot atau belum ada halangan maka robot akan melakukan pencarian. Apabila halangan tersebut berada didepan namun posisinya agak kesamping kiri maupun kekanan maka robot sumo akan mengikutinya terus sampai halangan tersebut tepat berada didepan robot sumo. Jika halangan berada tepat didepan robot maka robot sumo akan mendorongnya hingga keluar arena.

2.2 Kecerdasan Buatan

Kata “*intelligence*” berasal dari bahasa Latin “*intelligo*” yang berarti “saya paham”. Dasar dari *intelligence* ialah kemampuan untuk memahami dan melakukan aksi. Kecerdasan buatan atau kecerdasan yang ditambahkan kepada suatu sistem yang bisa diatur dalam konteks ilmiah didefinisikan sebagai kecerdasan entitas ilmiah. Sistem seperti ini dianggap sebagai komputer. Sebenarnya, area Kecerdasan buatan (*Artificial Intelligence*) atau disingkat dengan AI bermula dari kemunculan komputer sekitar tahun 1940-an, meskipun sejarah perkembangannya dapat dilacak sejak zaman Mesir kuno. Pada masa ini, perhatian difokuskan pada kecerdasan yang diciptakan dan dimasukkan ke dalam suatu mesin (komputer) agar dapat melakukan pekerjaan yang dapat dilakukan oleh manusia. Dalam hal ini, komputer tersebut dapat meniru kemampuan kecerdasan dan perilaku manusia. Beberapa macam bidang yang menggunakan kecerdasan buatan antara lain sistem pakar, permainan komputer (*games*), logika *fuzzy*, jaringan syaraf tiruan dan robotika.

“Kecerdasan buatan (*artificial intelligence*) merupakan kawasan penelitian, aplikasi dan instruksi yang terkait dengan pemrograman komputer untuk

melakukan sesuatu hal yang -dalam pandangan manusia adalah- cerdas”. (**H. A. Simon, 1987**).

“Kecerdasan Buatan (AI) merupakan sebuah studi tentang bagaimana membuat komputer melakukan hal-hal yang pada saat ini dapat dilakukan lebih baik oleh manusia.” (**Rich and Knight, 1991**).

“Kecerdasan buatan (AI) merupakan cabang dari ilmu komputer yang dalam merepresentasi pengetahuan lebih banyak menggunakan bentuk simbol-simbol daripada bilangan, dan memproses informasi berdasarkan metode *heuristic* atau dengan berdasarkan sejumlah aturan” (**Encyclopedia Britannica**). (<http://informatika.web.id/category/kecerdasan-buatan/>)

Secara garis besar, AI terbagi ke dalam dua faham pemikiran, yaitu AI Konvensional dan Kecerdasan Komputasional (CI atau *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 atau *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 interatif (misalnya penalaan parameter seperti dalam sistem koneksionis. Pembelajaran ini berdasarkan pada data empiris dan diasosiasikan dengan AI non-simbolis, AI yang tak teratur dan perhitungan lunak.

Metoda-metoda 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.

Kecerdasan buatan tidak hanya dominan di bidang ilmu komputer dan informatika saja tapi bisa membuat irisan dengan ilmu lain. Misal irisan kecerdasan buatan dengan teknik elektro melahirkan berbagai ilmu seperti pengolahan citra, teori kendali, pengenalan pola dan robotika.

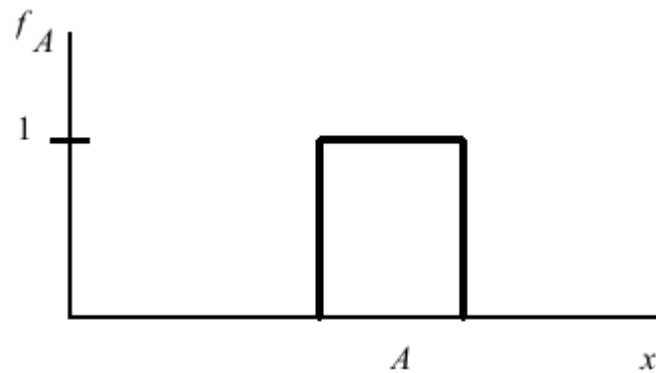
Robot adalah piranti elektromekanik yang dapat diprogram untuk melakukan otomasi terhadap suatu tugas yang biasanya dilakukan manusia. Sebuah robot sebenarnya buta akan bentuk urutan dari aksi bila tanpa usaha untuk mengganti komponennya atau bisa mendeteksi dan memperbaiki kesalahan dalam rencananya akan menjadi sulit bila tanpa kecerdasan.. Dengan sentuhan AI, robot dapat dibuat menjadi cerdas, sehingga bisa melakukan pengambilan keputusan seperti manusia. Kecerdasan Buatan (*Artificial Intelligence*) dalam robotik adalah suatu algoritma (yang dipandang) cerdas yang diprogramkan ke dalam kontroler robot. Pengertian cerdas di sini sangat relatif, karena tergantung dari sisi mana seseorang memandang.

Dengan adanya kecerdasan buatan, diharapkan tidak menutup kemungkinan hanya dengan data pengetahuan yang terbatas, sebuah komputer dapat berpikir seperti manusia dalam menghadapi masalah. **(Brigida Arie Minartiningtyas, 2013)**

2.3 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*

(Sri Kusumadewi, 2008)

2.3.1 Himpunan dan Fungsi Keanggotaan Logika *Fuzzy*

Pada himpunan tegas (*Crisp*), nilai keanggotaan suatu item x dalam suatu himpunan A , yang sering ditulis dengan $\mu_A[x]$, memiliki dua kemungkinan, 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 . (**Brigida Arie Minartiningtyas, 2013**)

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.

Hal yang perlu diketahui dalam memahami sistem *fuzzy*, yaitu :

a. Variabel *fuzzy*

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

b. Himpunan *fuzzy*

Himpunan *fuzzy* merupakan suatu grup yang mewakili suatu kondisi atau keadaan tertentu dalam suatu variabel *fuzzy*.

Contoh :

1. Variabel kecepatan terbagi menjadi tiga himpunan *fuzzy*, yaitu: lambat, sedang, jauh.

2. 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 $[0 \ 50]$

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*:

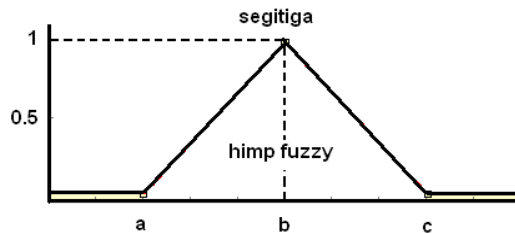
1. DEKAT = $[0, \ 5]$

2. SEDANG = $[7, \ 10]$

3. JAUH = [15, 20]

Fungsi keanggotaan *fuzzy* ada beberapa macam antara lain:

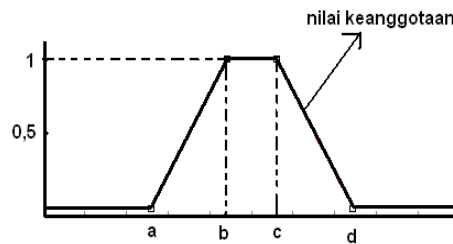
1. fungsi keanggotaan segitiga



Gambar 2.2 Himpunan fungsi keanggotaan

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

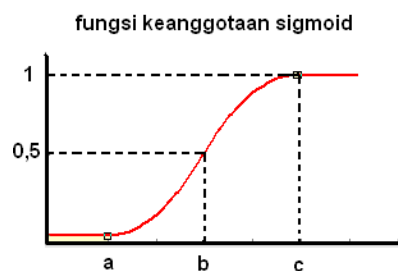
2. fungsi keanggotaan trapesium



Gambar 2.3 Himpunan keanggotaan trapesium

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

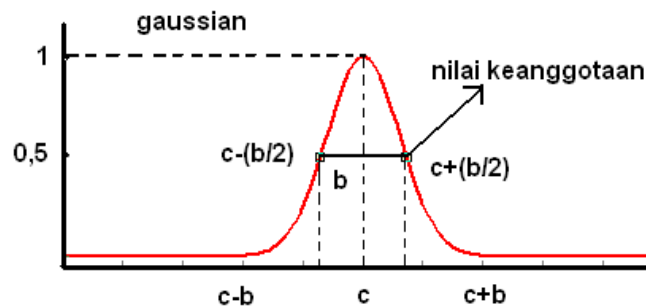
3. fungsi keanggotaan *sigmoid*



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.

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.

2.3.2 Sistem Kontrol *Fuzzy Logic*

Salah satu penerapan kecerdasan buatan pada robotika dapat dilihat dari sebuah metode *fuzzy logic* yang diterapkan pada Robot Sumo. Dalam perancangan sistem kontrol logika *fuzzy*, tentunya harus mengetahui terlebih dahulu basis pengetahuan *fuzzy*, mulai dari *range* semesta pembicaraan (*Universe of Discourse*) yang akan digunakan, *range* nilai yang akan digunakan pada masing-masing anggota himpunan *fuzzy*, dan bentuk pola kurva yang akan digunakan.

Fuzzy logic adalah salah satu sistem kecerdasan buatan tipe *reasoning*. Diantara sistem kontrol cerdas yang berkembang pesat, sistem kontrol *fuzzy* termasuk dalam sistem kontrol cerdas yang semakin populer. Metode pengendalian dengan logika *fuzzy* mempunyai kelebihan dibandingkan dengan jenis pengendalian lainnya, diantaranya adalah tidak diperlukannya model matematik yang eksplisit dari sistem yang dikendali dan algoritma pengendaliannya sangat sederhana. *Fuzzy logic* pertama kali dikenalkan oleh Lotfi Zadeh, seorang profesor di *University of California* di Berkeley.

Adapun beberapa alasan digunakannya logika *fuzzy* (Kusuma Dewi, 2003) adalah:

1. Konsep logika *fuzzy* mudah dimengerti. Karena logika *fuzzy* menggunakan dasar teori himpunan, maka konsep matematis yang mendasari penalaran *fuzzy* tersebut cukup mudah untuk dimengerti.
2. Logika *fuzzy* sangat fleksibel, artinya mampu beradaptasi dengan perubahan-perubahan, dan ketidakpastian yang menyertai permasalahan.
3. Logika *fuzzy* memiliki toleransi terhadap data yang cukup homogeni, dan kemudian ada beberapa data “eksklusif”, maka logika *fuzzy* memiliki kemampuan untuk menangani data eksklusif tersebut.
4. Logika *fuzzy* dapat membangun dan mengaplikasikan pengalaman-pengalaman para pakar secara langsung tanpa harus melalui proses pelatihan. Dalam hal ini, sering dikenal dengan istilah *fuzzy expert* sistem menjadi bagian terpenting.

5. Logika *fuzzy* dapat bekerjasama dengan teknik-teknik kendali secara konvensional. Hal ini umumnya terjadi pada aplikasi di bidang teknik mesin maupun teknik elektro.
6. Logika *fuzzy* didasari pada bahasa alami. Logika *fuzzy* menggunakan bahasa sehari-hari sehingga mudah dimengerti.

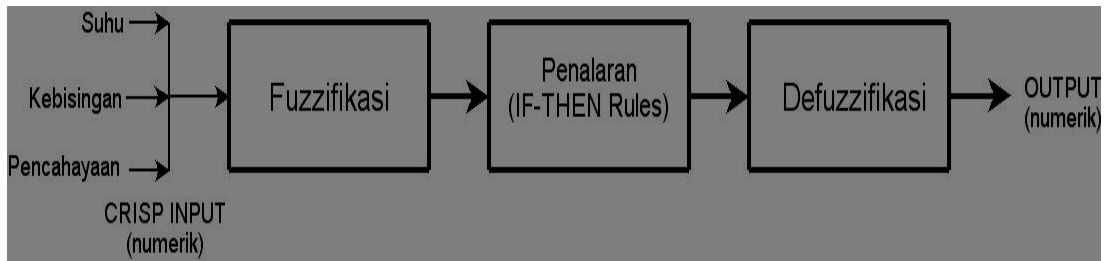
Fuzzy logic sudah banyak diaplikasikan dibidang teknik diantaranya pada mesin cuci, kamera yang bisa memfokuskan secara otomatis, kontrol sistem transmisi pada mobil dengan model terbaru, sistem pendaratan otomatis untuk kapal terbang, kontrol helikopter otomatis, sistem AC otomatis, kontrol motor *sinkron* dan lain-lain.

Logika *fuzzy* dapat dianggap sebagai kotak hitam yang berhubungan antara ruang *input* menuju ruang *output* (**Kusuma Dewi, 2003**). Kotak hitam tersebut berisi cara atau metode yang dapat digunakan untuk mengolah data *input* menjadi *output* dalam bentuk informasi yang baik.

Salah satu kelebihan *fuzzy logic* adalah memiliki kemampuan untuk mengakomodasi informasi linguistik dan numerik dari suatu sistem. Kemampuan ini dapat digunakan untuk mengatasi masalah non-linieritas yang sulit diatasi oleh pengontrol linier biasa, yaitu dengan mendeskripsikannya dalam sejumlah aturan linguistik atau pengetahuan tentang struktur I/O.

Proses-proses dalam *fuzzy logic* adalah fuzzifikasi, penalaran (*reasoning*), dan defuzzifikasi:

- Fuzzifikasi: merupakan proses untuk mendapatkan derajat keanggotaan dari sebuah nilai numerik masukan (*crisp*).
- Penalaran: proses untuk mendapatkan aksi keluaran dari suatu kondisi *input* dengan mengikuti aturan-aturan (*IF-THEN Rules*) yang telah ditetapkan yang disebut sebagai *inference/reasoning*.
- Defuzzifikasi: proses untuk merubah hasil penalaran yang berupa derajat keanggotaan keluaran menjadi variabel numerik kembali.



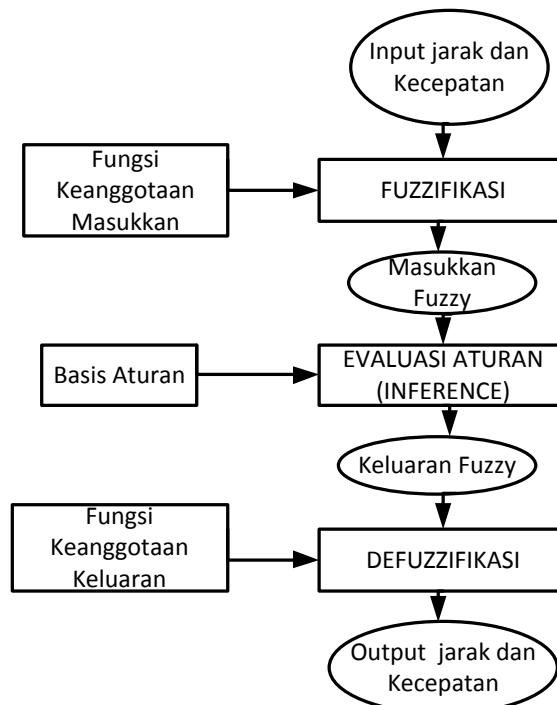
Gambar 2.6 Blok diagram proses dalam *fuzzy logic*

(Wahyu Romadhon, 2013)

Dalam sistem kontrol logika *fuzzy* terdapat beberapa tahapan operasional yang meliputi:

1. Fuzzyfikasi
2. Penalaran (*Inference Machine*)
3. Aturan Dasar (*Rule Baseed*)
4. Defuzzyfikasi

Blok diagram kontrol logika *fuzzy* ditunjukkan pada gambar berikut.



Gambar 2.7 Blok Diagram *Fuzzy Logic Control*

(Sri Kusumadewi, 2003)

a. 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.

b. Aturan Dasar (*Rule Based*)

Aturan dasar terdiri dari sejumlah aturan yang biasanya dinyatakan secara *linguistic*. Aturan *fuzzy* seringkali dinyatakan dengan “IF...THEN....”.

Pada sistem *fuzzy* ini menggunakan aturan-aturan dalam bentuk bahasa alami yang dibatasi oleh istilah linguistik, serta sintaksis yang baku. Sintaksis 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 : 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)

c. Mesin Penalaran Kontrol Logika *Fuzzy*

Mesin Penalaran adalah proses implikasi dalam menalar nilai masukan guna penentuan nilai keluaran sebagai bentuk pengambilan keputusan. Pada umumnya tiap-tiap aturan (proposisi) *fuzzy* dinyatakan dalam bentuk IF..THEN.. dan menyatakan suatu hubungan tertentu. Hubungan *fuzzy* ini sering disebut implikasi. Hubungan *fuzzy* dalam *knowledge base* dapat didefinisikan sebagai himpunan implikasi *fuzzy*.

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.

Jika dalam sistem *fuzzy* terdapat beberapa aturan, maka ada 3 metode yang dipakai dalam menentukan inferensi yaitu : *max-min*, *additive* dan *probabilistic OR* (*probor*)

1. Metode *Max-Min*

Max dapat dianalogikan dengan operasi logika OR sedangkan Min dianalogikan dengan operasi logika AND.

2. Metode *additive*

metode *additive* dilakukan dengan melakukan *bounded-sum* terhadap semua *output* daerah *fuzzy*. Secara umum dapat dituliskan dengan L:

$$\mu_{sf}[x_i] \leftarrow \min (1, \mu_{sf}[x_i] + \mu_{kf}[x_i])$$

$\mu_{sf}[x_i]$ = nilai keanggotaan solusi *fuzzy* sampai aturan ke-i

$\mu_{kf}[x_i]$ = nilai keanggotaan konsekuen (*output*) *fuzzy* sampai aturan ke-i

3. Metode *Probor*

Metode *probor* diperoleh dengan melakukan *product* (perkalian) terhadap semua *output* daerah *fuzzy*. Secara umum dituliskan dengan :

$$\mu_{sf}[x_i] \leftarrow (\mu_{sf}[x_i] + \mu_{kf}[x_i]) - \leftarrow (\mu_{sf}[x_i] * \mu_{kf}[x_i])$$

$\mu_{sf}[x_i]$ = nilai keanggotaan solusi *fuzzy* sampai aturan ke-i

$\mu_{kf}[x_i]$ = nilai keanggotaan konsekuen (*output*) *fuzzy* sampai aturan ke-i

Salah satu model yang banyak dipakai adalah penalaran *Max-Min*. Dalam penalaran *max-min* proses pertama yang dilakukan adalah melakukan operasi *min* sinyal keluaran lapisan *fuzzyfikasi*, yang diteruskan dengan operasi *max* untuk mencari nilai keluaran yang selanjutnya akan difuzzifikasikan sebagai bentuk keluaran pengontrol Defuzzifikasi.

d. Defuzzifikasi

Merupakan proses pemetaan himpunan *fuzzy* ke himpunan tegas (*crisp*).

Proses ini merupakan kebalikan dari proses fuzzyfikasi.

Proses defuzzyfikasi diekspresikan sebagai berikut:

Z^* = defuzzyfikasi (Z)

Dimana :

Z = Hasil penalaran fuzzy

Z^* = Keluaran *control* FL

Defuzzyfikasi = Operasi defuzzier

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 separuh 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 Rata-rata (*Average*)

Metode ini digunakan untuk fungsi keanggotaan keluaran yang simetris.

Persamaan dan metode ini adalah:

$$Z^* = \frac{\int \mu_{C_m}(Z) \cdot z dz}{\int \mu_{C_m}(Z) dz}$$

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

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

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

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

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

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

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

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

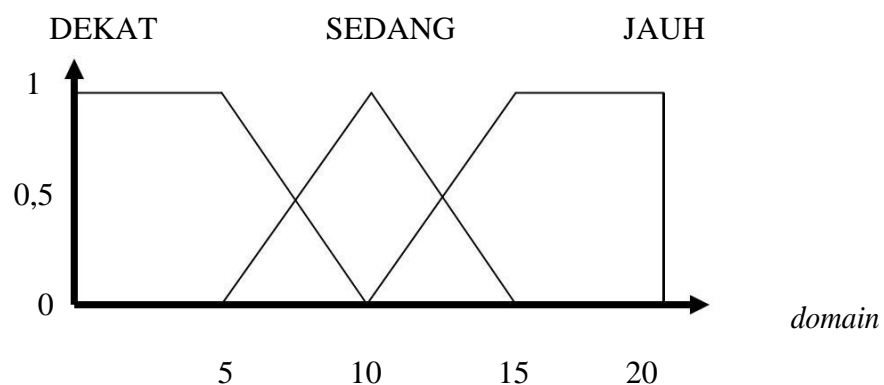
- Untuk z berupa kecepatan, datanya adalah sebagai berikut:

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

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

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.

- Variabel Jarak Kendaraan



Gambar 2.8 Fungsi Keanggotaan Variabel Jarak (Sri Kusumadewi, 2003)

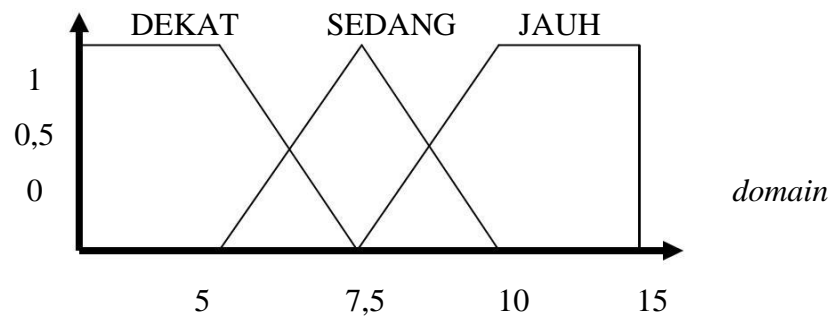
Fungsi keanggotaannya adalah sebagai berikut:

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

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

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

b. Variabel Jarak Tikungan



Gambar 2.9 Fungsi Keanggotaan Variabel Jarak terhadap Tikungan

(Sri Kusumadewi, 2003)

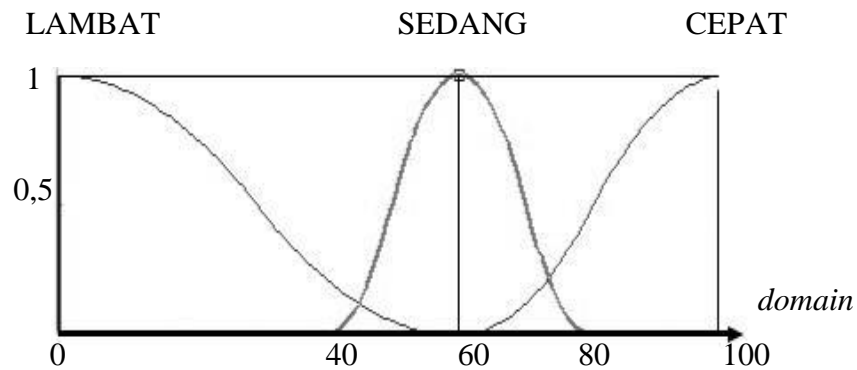
Fungsi keanggotaannya adalah sebagai berikut:

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

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

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

c. Variabel Kecepatan



Gambar 2.10 Fungsi Keanggotaan Variabel
(Sri Kusumadewi, 2003)

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 maksimum aturan, kemudian mengaplikasikannya ke *output* dengan menggunakan operator OR. 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*.

2.3.3 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	$\mu_{A \times B}(u) = \begin{cases} \mu_A(u) \\ \mu_B(u) \\ 0 \end{cases}$ <p><i>untuk</i> $\mu_B(u) = 1$ <i>untuk</i> $\mu_A(u) = 1$ <i>untuk</i> $\mu_A(u), \mu_B(u) < 1$</p>	

2.4 Arduino Mega2560

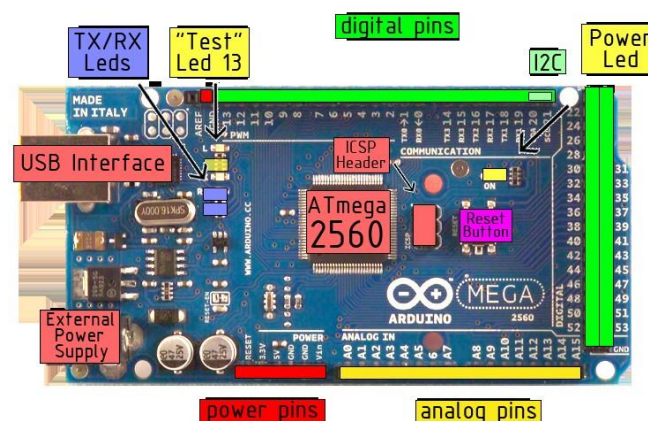
Arduino Mega2560 adalah papan mikrokontroler berbasis ATmega2560 (*datasheet*). Arduino Mega2560 memiliki 54 pin digital *input/output*, dimana 15 pin dapat digunakan sebagai output PWM, 16 pin sebagai input analog, dan 4 pin sebagai UART (*port serial hardware*), 16 MHz kristal osilator, koneksi USB, *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:

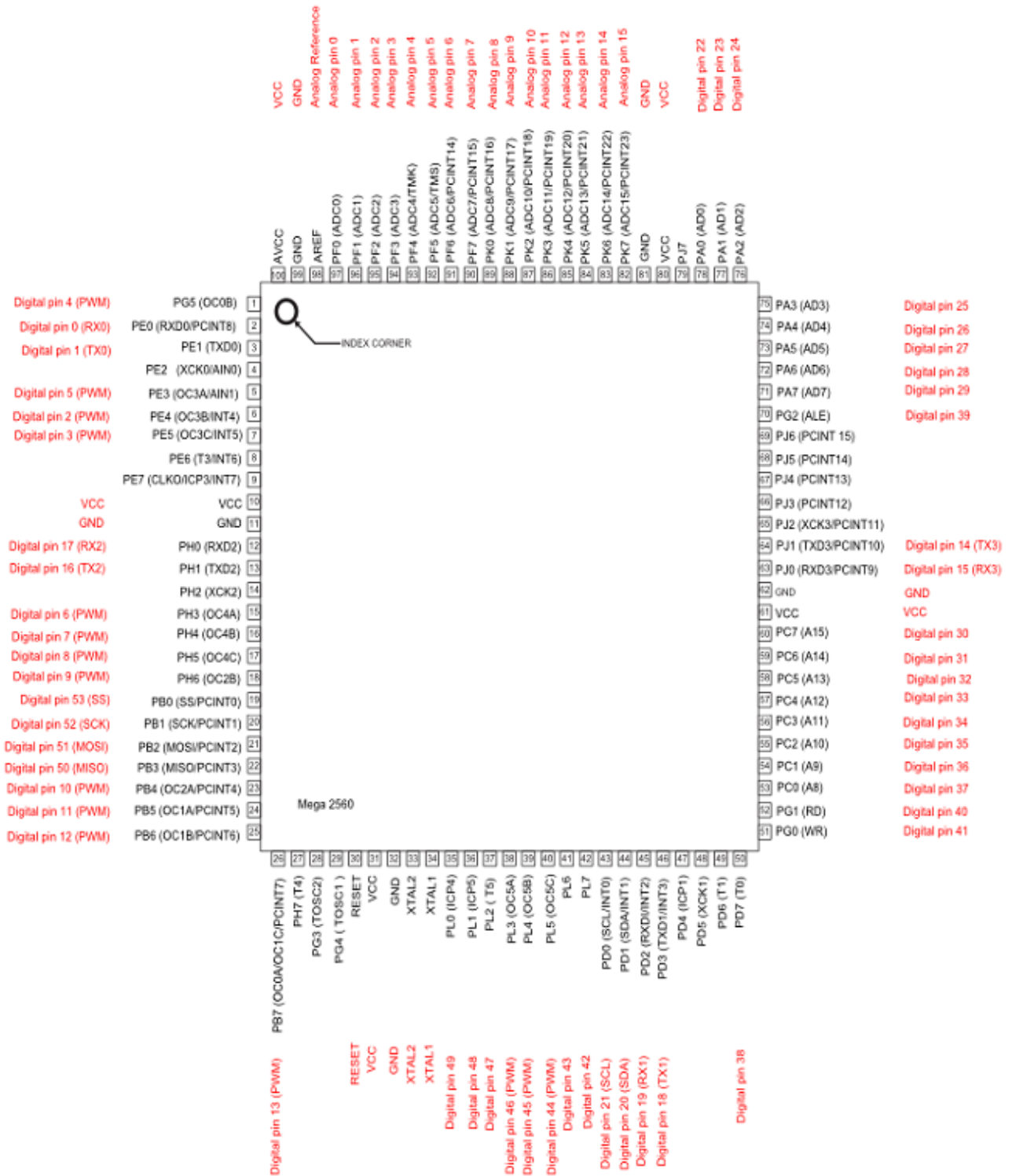
1. 1.0 pin *out* : Ditambahkan pin SDA dan pin SCL yang dekat dengan pin AREF dan dua pin baru lainnya ditempatkan dekat dengan pin RESET, IOREF memungkinkan *shield* untuk beradaptasi dengan tegangan yang tersedia pada papan. Di masa depan, *shield* akan kompatibel baik dengan papan yang menggunakan 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

(Kadir, Abdul. 2013. *Panduan Praktis Mempelajari Aplikasi Mikrokontroler dan Pemrogramannya menggunakan Arduino*. Yogyakarta: Andi Offset)

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



Gambar 2.12 Pemetaan Pin ATmega2560

(www.arduino.cc, diakses 16 Mei 2016)

Berikut merupakan spesifikasi sederhana dari Arduino Mega2560.

Tabel 2.2 Spesifikasi Sederhana Arduino Mega2560

Mikrokontroler	ATmega 2560
Tegangan Operasi	5V
Input Voltage (disarankan)	7-12V
Input Voltage (Limit)	6-20V
Pin Digital I/O	54 (yangn 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

2.4.1 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. **3V3** : 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)

2.4.2 *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)

2.4.3 Komunikasi Arduino Mega2560

Arduino Mega2560 memiliki sejumlah fasilitas untuk berkomunikasi dengan komputer, dengan Arduino lain, atau dengan mikrokontroler lainnya. Arduino ATmega328 menyediakan 4 *hardware* komunikasi serial UART TTL (5

Volt). Sebuah *chip* ATmega16U2 (ATmega8U2 pada papan Revisi 1 dan Revisi 2) yang terdapat pada papan digunakan sebagai media komunikasi serial melalui USB dan muncul sebagai *COM Port Virtual* (pada peralatan komputer) untuk berkomunikasi dengan perangkat lunak pada komputer, untuk sistem operasi Windows masih tetap memerlukan file inf, tetapi untuk sistem operasi OS X dan Linux akan mengenali papan sebagai *port* COM secara otomatis.

Perangkat lunak Arduino termasuk didalamnya serial monitor memungkinkan data tekstual sederhana dikirim ke dan dari papan Arduino. LED RX dan TX yang tersedia pada papan akan berkedip ketika data sedang dikirim atau diterima melalui *chip USB-to-serial* yang terhubung melalui USB komputer (tetapi tidak untuk komunikasi *serial* seperti pada pin 0 dan 1).

Sebuah perpustakaan *SoftwareSerial* memungkinkan untuk komunikasi *serial* pada salah satu pin digital Mega2560. ATmega2560 juga mendukung komunikasi TWI dan SPI. Perangkat lunak Arduino termasuk perpustakaan *Wire* digunakan untuk menyederhanakan penggunaan bus TWI. Untuk komunikasi SPI, menggunakan perpustakaan SPI. (www.arduino.cc)

2.4.4 Reset (*Software*) Otomatis

Daripada menekan tombol reset sebelum *upload*, Arduino Mega2560 didesain dengan cara yang memungkinkan Anda untuk *me-reset* melalui perangkat lunak yang berjalan pada komputer yang terhubung. Salah satu jalur kontrol *hardware* (DTR) mengalir dari ATmega8U2/16U2 dan terhubung ke jalur reset dari ATmega2560 melalui kapasitor 100 nanofarad. Bila jalur ini di-set rendah/*low*, jalur *reset drop* cukup lama untuk *me-reset chip*. Perangkat lunak Arduino menggunakan kemampuan ini untuk memungkinkan kita meng-*upload* kode dengan hanya menekan tombol *upload* pada perangkat lunak Arduino. Ini berarti bahwa *bootloader* memiliki rentang waktu yang lebih pendek, seperti menurunkan DTR dapat terkoordinasi (berjalan beriringan) dengan dimulainya *upload*.

Pengaturan ini juga memiliki implikasi lain. Ketika Mega2560 terhubung dengan komputer yang menggunakan sistem operasi Mac OS X atau Linux, papan Arduino akan di-*reset* setiap kali dihubungkan dengan *software* komputer (melalui USB). Dan setengah detik kemudian atau lebih, *bootloader* berjalan pada papan

Mega2560. Proses *reset* melalui program ini digunakan untuk mengabaikan data yang cacat (yaitu apapun selain meng-*upload* kode baru), ia akan memotong dan membuang beberapa *byte* pertama dari data yang dikirim ke papan setelah sambungan dibuka. Jika sebuah sketsa dijalankan pada papan untuk menerima satu kali konfigurasi atau menerima data lain ketika pertama kali dijalankan, pastikan bahwa perangkat lunak diberikan waktu untuk berkomunikasi dengan menunggu satu detik setelah terkoneksi dan sebelum mengirim data.

Mega2560 memiliki trek jalur yang dapat dipotong untuk menonaktifkan fungsi *auto-reset*. Pad di kedua sisi jalur dapat dihubungkan dengan disolder untuk mengaktifkan kembali fungsi *auto-reset*. Pad berlabel “RESET-EN”. kita juga dapat menonaktifkan *auto-reset* dengan menghubungkan resistor 110 ohm dari 5V ke jalur reset. (www.arduino.cc)

2.4.5 Pemrograman Arduino Mega2560

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

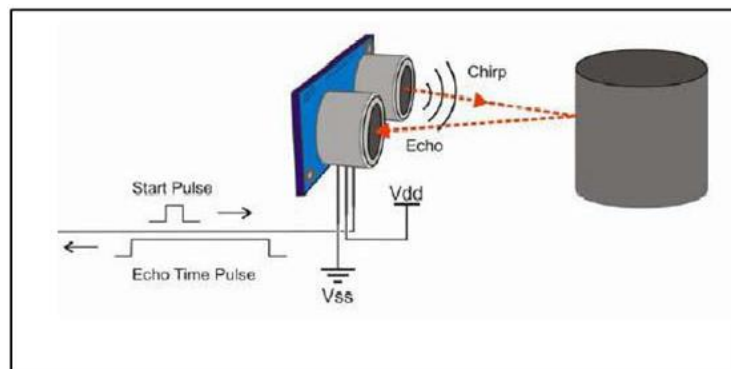
Chip ATmega16U2 (atau 8U2 pada *board* Rev. 1 dan Rev. 2) *source code firmware* tersedia pada repositori Arduino. ATmega16U2/8U2 dapat dimuat dengan *bootloader* DFU, yang dapat diaktifkan melalui:

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

2.5 Sensor Ultrasonik

Sensor ultrasonik adalah sebuah sensor yang mengubah besaran *fisis* (bunyi) menjadi besaran listrik. Pada sensor ini gelombang ultrasonik dibangkitkan melalui sebuah benda yang disebut *piezoelektrik*. *Piezoelektrik* ini akan menghasilkan gelombang ultrasonik dengan frekuensi 40 kHz ketika sebuah osilator diterapkan pada benda tersebut.



Gambar 2.13 Sensor Ultrasonik

(Sumber: <http://komponenelektronika.biz/sensor-ultrasonik.html>, diakses tanggal 4 April 2016)

Sensor ultrasonik secara umum digunakan pada aplikasi pengukuran jarak. Alat ini secara umum memancarkan gelombang suara ultrasonik menuju suatu target yang memantulkan balik gelombang kearah sensor. Kemudian mengukur waktu yang diperlukan untuk pemancaran gelombang sampai kembali ke sensor dan menghitung jarak dengan menggunakan kecepatan suara.

Rangkaian penyusun sensor ultrasonik terdiri dari *transmitter*, *receiver*, dan komparator. Selain itu, gelombang ultrasonik dibangkitkan oleh sebuah kristal tipis bersifat *piezoelektrik*. Bagian – bagian dari sensor ultrasonik adalah sebagai berikut:

1. *Piezoelektrik*

Peralatan *piezoelektrik* secara langsung mengubah energi listrik menjadi energi mekanik. Tegangan *input* yang digunakan menyebabkan bagian keramik meregang dan memancarkan gelombang ultrasonik. Tipe operasi transmisi elemen *piezoelektrik* sekitar frekuensi 32 kHz. Efisiensi lebih

baik, jika frekuensi osilator diatur pada frekuensi resonansi *piezoelektrik* dengan sensitivitas dan efisiensi paling baik. Jika rangkaian pengukur beroperasi pada mode pulsa elemen *piezoelektrik* yang sama dapat digunakan sebagai *transmitter* dan *receiver*.

2. *Transmitter*

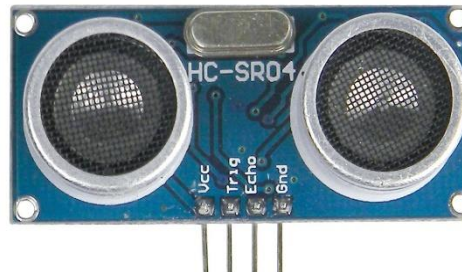
Transmitter adalah sebuah alat yang berfungsi sebagai pemancar gelombang ultrasonik dengan frekuensi sebesar 40 kHz yang dibangkitkan dari sebuah osilator. Untuk menghasilkan frekuensi 40 kHz, harus dibuat sebuah rangkaian osilator dan keluaran dari osilator dilanjutkan menuju penguat sinyal. Besarnya frekuensi ditentukan oleh komponen kalang RLC/Kristal tergantung dari *desain* osilator yang digunakan. Penguat sinyal akan memberikan sebuah sinyal listrik yang diumpankan ke *piezoelektrik* dan terjadi reaksi mekanik sehingga bergetar dan memancarkan gelombang yang sesuai dengan besar frekuensi pada osilator.

3. *Receiver*

Receiver terdiri dari transduser ultrasonik menggunakan bahan *piezoelektrik*, yang berfungsi sebagai penerima gelombang pantulan yang berasal dari *transmitter* yang dikenakan pada permukaan suatu benda atau gelombang langsung LOS (*Line Of Sight*) dari *transmitter*. Oleh karena bahan *piezoelektrik* memiliki reaksi yang *reversible*, elemen keramik akan membangkitkan tegangan listrik pada saat gelombang datang dengan frekuensi yang resonan dan akan menggetarkan bahan *piezoelektrik* tersebut.

2.5.1 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.



Gambar 2.14 Sensor Ultrasonik HC-SR04

(<http://komponenelektronika.biz/sensor-ultrasonik.html>, diakses 4 April 2016)

Lamanya waktu pancar dan terima sebanding dengan dua kali jarak sensor dengan objek, sehingga jarak sensor dengan objek dapat ditentukan persamaan :

$$s = \frac{v \times t}{2}$$

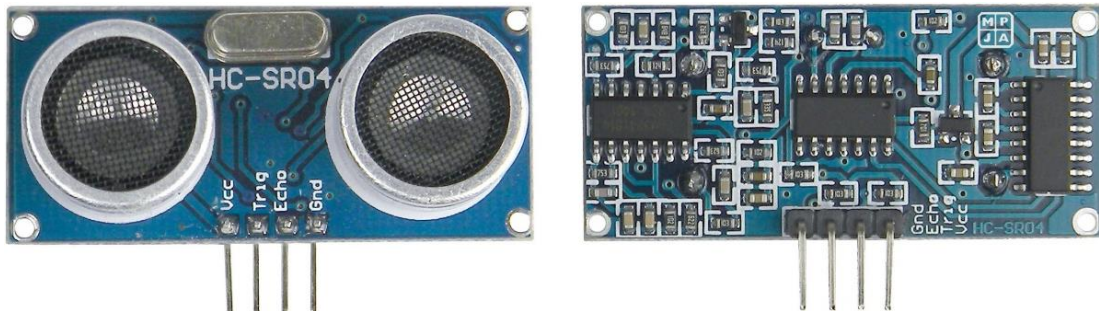
Keterangan:

s = jarak (meter)

v = kecepatan suara diudara (340 m/detik)

t = waktu tempuh (detik)

HC-SR04 dapat mengukur jarak dalam rentang antara 2 cm – 4 m dengan *output* panjang pulsa yang sebanding dengan jarak objek. Sensor ini hanya memerlukan 2 pin I/O untuk berkomunikasi dengan mikrokontroler, yaitu *trigger* dan *echo*. Untuk mengaktifkan HC-SR04 mikrokontroler pada arduino mengirimkan pulsa positif melalui pin *trigger* minimal 10 μ s untuk *trigger*, selanjutnya modul secara otomatis akan mengirimkan frekuensi suara sebesar 40 KHz melalui pin *echo*. HC-SR04 akan mengirimkan pulsa positif melalui pin *echo*. Penerima ultrasonik akan mendeteksi gelombang yang dipantulkan.



Gambar 2.15 Sensor Ultrasonik HC-SR04 Tampak Depan dan Belakang

(https://www.mpja.com/download/hc-sr04_ultrasonic_module_user_guidejohn.pdf, diakses pada 25 Juli 2016)

Dari gambar diatas dapat dijelaskan fungsi dari masing – masing koneksi sensor ultrasonik HC-SR04 sebagai berikut:

- a. *VCC (5V supply)* : sebagai tegangan *supply* yang nantinya dihubungkan ke *power supply* 5V.
- b. *Trigger Pulse Input* : sebagai pin *input* yang nantinya dihubungkan ke mikrokontroler untuk mendapatkan pulsa dari mikrokontroler.
- c. *Echo Pulse Output* : sebagai pin *output* yang nantinya dihubungkan ke mikrokontroler sehingga mikrokontroler dapat membaca pulsa yang dihasilkan sensor.
- d. *GND (0V Ground)* : sebagai pertanahan atau *grounding*.

Adapun spesifikasi dari sensor ultrasonik HC-SR04 adalah sebagai berikut:

- a. Dimensi : 1-13/16" X 13/16" X 5/8"
- b. Tegangan : 5 VDC
- c. Konsumsi Arus : 15 mA
- d. Frekuensi Suara : 40 kHz
- e. Jangkauan : 2 cm – 4 m
- f. Sensitivitas : Mampu mendeteksi objek dengan diameter 2 cm pada jarak > 3 m
- g. Input Trigger : 10 mS min. Pulsa Level TTL
- h. Pulsa Echo : Sinyal level TTL, sebanding dengan jarak yang dideteksi

2.6 Motor DC

Motor DC adalah jenis motor listrik yang bekerja menggunakan sumber tegangan DC. Motor DC atau motor arus searah sebagaimana namanya, menggunakan arus langsung dan tidak langsung/*direct-unidirectional*. Motor DC digunakan pada penggunaan khusus dimana diperlukan penyalaan torque yang tinggi atau percepatan yang tetap untuk kisaran kecepatan yang luas. Motor DC merupakan suatu mesin yang berfungsi untuk mengubah tenaga listrik arus searah (DC) menjadi tenaga mekanik (putaran). Motor bekerja berdasarkan prinsip induksi elektromagnetik.



Gambar 2.16 Motor DC

(<http://www.img.hisupplier.com>, diakses pada 14 Juli 2016)

Bagian utama paling penting motor DC adalah stator dan rotor dimana kumparan medan pada motor DC disebut stator (bagian yang tidak berputar). Bagian rotor adalah bagian yang berputar dari suatu motor DC seperti lilitan jangkar, jangkar, koutator, tali, isolator, poros, bantalan dan kipas. Bagian stator adalah badan motor, sikat-sikat dan inti kutub magnet. Motor DC memiliki 3 bagian atau komponen utama untuk dapat berputar sebagai berikut.

1. Kutub medan, motor DC sederhana memiliki dua kutub medan: kutub utara dan kutub selatan. Garis magnetik energi membesar melintasi ruang terbuka diantara kutub-kutub dari utara ke selatan. Untuk motor yang lebih besar atau lebih kompleks terdapat satu atau lebih elektromagnet.
2. *Current Elektromagnet* atau Dinamo, dinamo yang berbentuk silinder, dihubungkan ke as penggerak untuk menggerakkan beban. Untuk kasus

motor DC yang kecil, dinamo berputar dalam medan magnet yang dibentuk oleh kutub-kutub, sampai kutub utara dan selatan magnet berganti lokasi.

3. Komutator, komponen ini terutama ditemukan dalam motor DC. Kegunaannya adalah untuk transmisi arus antara dinamo dan sumber daya.

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. Motor DC mempunyai bagian yang mantap (stator) yang berupa magnet permanen dan bagian yang bergerak (rotor) yang berupa *koil* atau gulungan kawat tembaga. Dimana setiap ujungnya tersambung dengan komutator. Komutator dihubungkan dengan kutub positif (+) dan kutub negatif (-) dari catu daya.

Arus listrik dari kutub positif akan masuk melalui komutator, kemudian berjalan mengikuti gulungan kawat sebelumnya, akhirnya masuk ke kutub negatif catu daya. Karena adanya medan elektromagnetik maka motor akan berputar.

Karena putaran rotor, arus listrik di dalam kawat akan berjalan bolak-balik karena Jika arus lewat pada suatu konduktor, timbul medan magnet di sekitar konduktor. Arah medan magnet ditentukan oleh arah aliran arus pada konduktor, maka rotor akan selalu berputar terus menerus selama arus listrik tetap mengalir di dalam kawat.

2.7 *Driver Motor DC*

Driver motor merupakan suatu rangkaian khusus yang memiliki fungsi untuk mengatur arah ataupun kecepatan pada motor DC. Adapun macam-macam jenis driver yang dapat digunakan untuk mengatur arah dan kecepatan motor DC yaitu IC L298N.

2.7.1 IC L298N

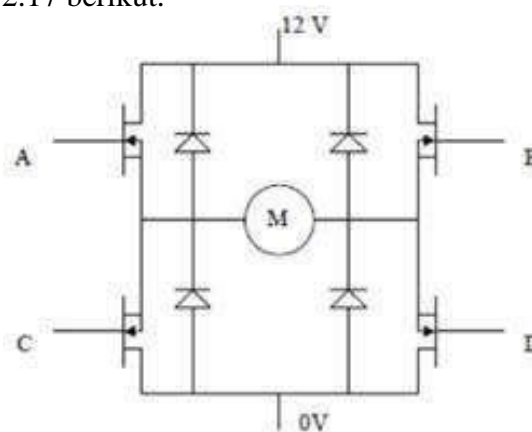
IC L298N adalah IC yang didesain khusus sebagai *driver* motor DC yang digunakan untuk mengendalikan motor DC dengan arus maksimum 4A. IC ini dapat dikendalikan dengan rangkaian TTL maupun mikrokontroler. Motor DC yang dikontrol dengan *driver* IC L298N dapat dihubungkan ke *ground* maupun ke

sumber tegangan positif karena di dalam *driver* L298N sistem *driver* yang digunakan adalah *totem pool*. Dengan IC *driver* motor DC L298N dapat digunakan untuk mengendalikan 2 buah motor DC sekaligus secara *independent*. Kemampuan tiap *driver* motor DC dalam IC L298N ini adalah 4A untuk masing-masing *drivernya*. IC L298 adalah *driver* motor DC *H-Bridge* dengan 2 unit *driver* di dalam 1 *chip* IC.

2.7.2 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 2.17 berikut.



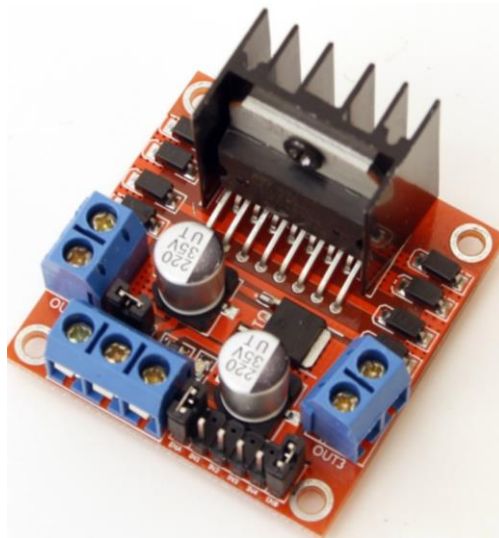
Gambar 2.17 Konfigurasi *H-Bridge* MOSFET

(<http://www.instructables.com/id/Arduino-Modules-L298N-Dual-H-Bridge-Motor-Controll/>, diakses 15 Mei 2016)

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 2.18 Motor Driver L298N Module (<http://www.idealcreations.net/>, diakses pada 25 Juli 2016)

Berikut adalah tabel dari pin-pin *driver* motor L298N.

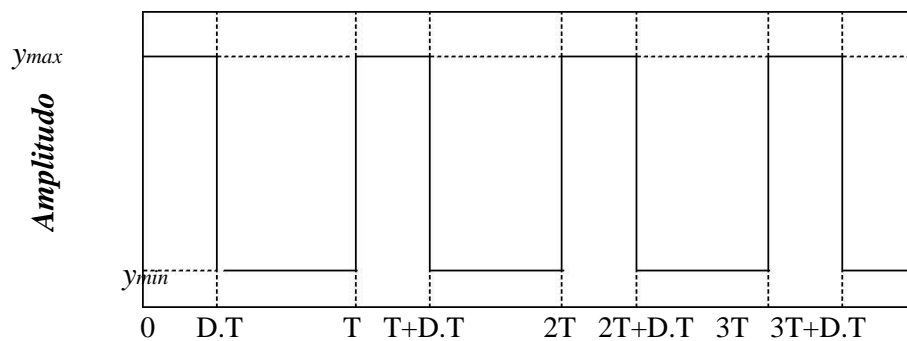
Tabel 2.3 Pin dan Fungsi Motor Driver L298N Module

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

2.7.3 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 2.19.



Gambar 2.19 Gelombang kotak PWM

2.8 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*, untuk *input voltage* (tegangan masukan) 4,5 - 30 VDC, sedangkan *output voltage* (tegangan keluaran) adalah 0.8-28 VDC 12A.



Gambar 2.20 12A Adjustable DC-DC Step Down Module

(<https://id.wikipedia.org/wiki/dc-dc-step-down-module>, diakses 14 April 2016)

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

2.9 Modul Kompas GY-273 HMC5883L

Modul Kompas GY-273 adalah sebuah modul yang digunakan untuk menunjukkan arah mata angin digital, atau juga disebut kompas digital. Modul ini menggunakan komponen utama berupa IC HMC5883 yang merupakan IC kompas digital 3 axis yang memiliki interface berupa 2 pin I2C.

HMC5883 memiliki sensor magneto-resistive HMC118X series ber-resolusi tinggi, ditambah ASIC dengan konten amplification, automatic degaussing strap driver, offset cancellation dan 12 bit ADC yang memungkinkan keakuratan kompas mencapai 1 sampai 2 derajat. Modul ini biasa digunakan untuk keperluan sistem navigasi otomatis, mobile phone, netbook dan perangkat navigasi personal



Gambar 2.19 Modul Kompas HMC5883L

(<https://splashtronic.wordpress.com/2013/10/29/modul-kompas-gy-273-hmc5883l/>, diakses 26 Mei 2016)

Modul ini memiliki 5 pin, diantaranya :

1. VCC (5V)
2. GND
3. SCL
4. SDA
5. DRDY

Berikut adalah beberapa fitur dari Modul Kompas GY-273:

1. Berbasis sensor magnetoresistive 3 axis.
2. 12-Bit ADC terkopling dengan Low Noise AMR Sensor yang memiliki 2 mili-gauss Field dengan resolusi.kurang lebih 8 Gauss Fields.
3. Tegangan kerja 5V DC.
4. Menggunakan antarmuka I2C.
5. Keluaran rata-rata maksimum 160 Hz.

(<https://splashtronic.wordpress.com/2013/10/29/modul-kompas-gy-273-hmc5883l/>)