EMBEDDED SYSTEM SEBAGAI PENGONTROL KECEPATAN KENDARAAN BERDASARKAN JARAK AMAN MENGGUNAKAN METODE FUZZY SUGENO

SKRIPSI

KEMINATAN TEKNIK KOMPUTER

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun oleh:
Syukri Akbar
115060900111019

PROGRAM STUDI TEKNIK INFORMATIKA
PROGRAM TEKNOLOGI INFORMASI DAN ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG
2016

PENGESAHAN

EMBEDDED SYSTEM SEBAGAI PENGONTROL KECEPATAN KENDARAAN BERDASARKAN JARAK AMAN MENGGUNAKAN METODE FUZZY SUGENO

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun Oleh :
Syukri Akbar
115060900111019

Skripsi ini telah diuji dan dinyatakan lulus pada
21 Januari 2016
Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I	Dosen Pembimbing II
Wijaya Kurniawan, S.T., M.T.	Rekyan Regasari MP, S.T., M.T.
NIP: 19820125 201504 1 002	NIK: 770414 06 1 2 0253

Mengetahui
Ketua Program Studi Teknik Informatika

Drs. Marji, M.T.
NIP: 19670801 199203 1 001

PERNYATAAN ORISINALITAS

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah skripsi ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis disitasi dalam naskah ini dan disebutkan dalam daftar pustaka.

Apabila ternyata didalam naskah skripsi ini dapat dibuktikan terdapat unsur-unsur plagiasi, saya bersedia skripsi ini digugurkan dan gelar akademik yang telah saya peroleh (sarjana) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang, 2016

Syukri Akbar
115060900111019

KATA PENGANTAR

Puji syukur kehadirat Allah SWT yang telah melimpahkan rahmat, taufik dan hidayah-Nya sehingga laporan skripsi yang berjudul **"*EMBEDDED SYSTEM* SEBAGAI PENGONTROL KECEPATAN KENDARAAN BERDASARKAN JARAK AMAN MENGGUNAKAN METODE *FUZZY* SUGENO"** ini dapat terselesaikan. Skripsi ini untuk memenuhi salah satu syarat menyelesaikan studi serta dalam rangka memperoleh gelar Sarjana Pendidikan Strata Satu pada Program Studi Sistem Komputer Fakultas Program Teknologi Informasi dan Ilmu Komputer Universitas Brawijaya. Penghargaan dan terima kasih yang setulus-tulusnya kepada Ayahanda tercinta Fanani, BE dan Ibunda yang kusayangi Ria Fitri R serta kakak tersayang Elisa Trifani S ,S.Kom, semoga Allah SWT melimpahkan Rahmat, Kesehatan, Karunia dan keberkahan di dunia dan di akhirat atas budi baik yang telah diberikan kepada penulis.

Penulis menyadari bahwa skripsi ini tidak akan berhasil tanpa bantuan dari beberapa pihak. Oleh karena itu, penulis ingin menyampaikan rasa hormat dan terima kasih kepada:

1. Bapak Prof. Dr. Ir. Mohammad Bisri, MS selaku Rektor Universitas Brawijaya
2. Bapak Ir Sutrisno, M.T. selaku Dekan Fakultas Program Teknologi Informasi dan Ilmu Komputer Universitas Brawijaya.
3. Bapak Wijaya Kurniawan, S.T., M.T selaku pembimbing I yang telah banyak membantu dalam pengerjaan skripsi ini.
4. Ibu Rekyan Regasari MP, S.T., M.T. selaku pembimbing II yang telah banyak membantu dalam pengerjaan skripsi ini.
5. Rizky Dwi A. P, M. Emirza Alam F, Saputra Yudha W, M. Afif Hamidi, selaku tim sukses *fuzzy*.
6. M. D. A. Labib, Januar Rizky P, Bagus Priyo P ,Alfaviega S P, M Abdul Mujib, dan seluruh teman-teman Teknik Komputer 2011 yang telah menemani selama perkuliahan.

Malang, 2016

Syukri Akbar

115060900111019

ABSTRAK

Kecelakan dari tahun ke tahun semakin bertambah. Kecelakaan tersebut disebabkan oleh kurangnya konsentrasi pengemudi, sehingga menyebabkan kecelakaan dari yang ringan hingga yang sangat berat. Selain itu, kesalahan dalam memprediksi jarak aman berhenti antara kendaraan bermotor dengan kendaraan bermotor yang lain juga menjadi faktor terjadinya kecelakaan. Solusi yang ditawarkan peneliti untuk menyelesaikan masalah tersebut ialah "*embedded system* sebagai pengontrol kecepatan kendaraan berdasarkan jarak aman menggunakan metode *fuzzy* sugeno" .

Sensor yang digunakan sebagai input, terdiri dari dua. Pertama, sensor ultrasonik sebagai pengukur jarak. Kedua, *sensor rotary encoder* sebagai pengukur kecepatan. Kedua data *input* tersebut diproses menggunakan fungsi *fuzzy* yang diletakkan di dalam arduino. *Output*nya adalah hasil fuzzy yang digunakan untuk mengatur PWM motor di dalam *prototype*. Hasil pengujian menunjukkan tingkat akurasi sebesar 88%. Hasil tersebut diperoleh dari 22 pengujian berhasil berdasarkan 25 pengujian yang telah dilakukakan peneliti. Dengan demikian, dapat disimpulkan bahwa prototype ini berhasil.

ABSTRACT

Accidents have increased from time to time. Most of them are caused by lack of concentration of the drivers. This problem has caused mild and severe accidents. Furthermore, incapability of predicting a safe distance of stopping between one vechile to another also becomes a factor of the accident. A solution offered by the researcher to solve the problems is "embedded system as vehicle speed control based on a safe distance using Sugeno fuzzy".

This system used two sensors that are used as input. They are ultrasonic sensor as a measure of distance and rotary encoder sensor as a measure of speed. Both of the input data are processed using the function of fuzzy which is placed in arduino. The output is the use of fuzzy is used to set the motor in the prototype PWM. The result of the test showed that an accuracy level increased up to 88%. The results are obtained from 22 successful tests of 25 tests that have been conducted by the researcher. Thus, it can be concluded that the prototype is successful.

DAFTAR ISI

PENGESAHAN ii
PERNYATAAN ORISINALITAS iii
KATA PENGANTAR iv
ABSTRAK v
ABSTRACT vi
DAFTAR ISI vii
DAFTAR TABEL x
DAFTAR GAMBAR xi
DAFTAR LAMPIRAN xii
BAB 1 PENDAHULUAN 13
1.1 Latar belakang 13
1.2 Rumusan masalah 14
1.3 Tujuan 14
1.4 Manfaat 14
1.5 Batasan masalah 14
1.6 Sistematika penulisan 15
BAB 2 KAJIAN PUSTAKA DAN DASAR TEORI 16
2.1 Kajian pustaka 16
2.2 Dasar teori 16
2.2.1 Dasar logika *fuzzy* 16
2.2.2 Himpunan *fuzzy* 17
2.2.3 Fungsi keanggotaan 18
2.2.4 *Fuzzy* sugeno-takagi 22
2.2.5 Arduino atmega 2560 23
2.2.6 Sensor Rotary Encoder 24
2.2.7 Sensor ultrasonik 26
2.2.8 LCD (*Liquid Crystal Display*) 16 x 2 27
2.2.9 Modul *driver motor DC* 28
BAB 3 METODOLOGI PENELITIAN 29
3.1 Perumusan masalah 30

3.2 Penentuan tujuan penelitian 30
3.3 Studi literatur 30
3.4 Analisis kebutuhan 31
3.4.1 Sistem kerja alat 31
3.4.2 Perangkat keras 31
3.4.3 Perangkat lunak 31
3.5 Perancangan 32
3.5.1 Perancangan perangkat keras 32
3.5.2 Perancangan perangkat lunak 32
3.6 Implementasi 32
3.7 Pengujian dan analisis 32
3.8 Kesimpulan 32
BAB 4 PERANCANGAN DAN IMPLEMENTASI 33
4.1 Blok diagram 33
4.2 Perancangan metode Fuzzy 33
4.2.1 Diagram alir metode *fuzzy* 33
4.2.2 Diagram alir sistem metode *fuzzy* 34
4.2.3 Perancangan fuzzy inference system 35
4.3 Perancangan rangkaian *prototype* 40
4.3.1 Perancangan sensor ultrasonik 40
4.3.2 Perancangan sensor *rotary encoder* 41
4.3.3 Perancangan PWM 42
4.3.4 Perancangan Keseluruhan 43
4.3.5 Perancangan *prototype* 44
4.4 Implementasi 45
4.4.1 Implementasi sensor ultrasonik 45
4.4.2 Implementasi sensor *rotary encoder* 47
4.4.3 Implementasi sistem keseluruhan 48
BAB 5 PENGUJIAN DAN ANALISIS 54
5.1 Pengujian metode *fuzzy* dan analisis 54
5.2 Pengujian sistem dengan implementasi metode *fuzzy* dan analisis ... 56
5.3 Pengujian mode *follow* dan analisis 58

BAB 6 KESIMPULAN DAN SARAN 60
6.1 Kesimpulan..... 60
6.2 Saran 60
DAFTAR PUSTAKA..... 61
LAMPIRAN 1 PROGRAM *FUZZY* 62

DAFTAR TABEL

Tabel 2.1 Spesifikasi Arduino AtMega2560 24
Tabel 2.2 Spesifikasi sensor *Rotary encoder* 25
Tabel 2.3 Spesifikasi sensor ultrasonik 27
Tabel 5.1 Hasil pengujian 1 56
Tabel 5.2 Hasil pengujian 2 56
Tabel 5.3 Hasil pengujian 3 57
Tabel 5.4 Hasil pengujian 4 57
Tabel 5.5 Hasil pengujian 5 58
Tabel 5.6 Hasil pengujian mode *follow* 59

DAFTAR GAMBAR

Gambar 2.1 Repesentasi linear naik 18
Gambar 2.2 Representasi linear turun 19
Gambar 2.3 Representasi linear naik dan turun 19
Gambar 2.4 Representasi Kurva Trapesium 20
Gambar 2.5 Arduino Atmega 2560 23
Gambar 2.6 Sensor *Rotary encoder* 25
Gambar 2.7 Penyusun *rotary encoder* 26
Gambar 2.8 Sensor Ultrasonik 26
Gambar 2.9 LCD 16x2 27
Gambar 2.10 Modul *Driver* Dc 28
Gambar 3.1 Alur pelaksanaan 29
Gambar 4.1 Blok diagram sistem 33
Gambar 4.2 Diagram alir metode *fuzzy* 34
Gambar 4.3 Diagram alir sistem dengan metode *fuzzy* 35
Gambar 4.4 *Fuzzy inference system* 36
Gambar 4.5 Himpunan jarak 36
Gambar 4.6 Himpunan kecepatan 37
Gambar 4.7 *Rules Fuzzy* 38
Gambar 4.8 *Output* kecepatan 39
Gambar 4.9 Simulasi Matlab 40
Gambar 4.10 Rangkaian sensor ultrasonik 41
Gambar 4.11 Rangkaian sensor *rotary encoder* 42
Gambar 4.12 Rangkaian PWM 43
Gambar 4.13 Perencangan rangkaian sistem keseluruhan 44
Gambar 4.14 Perancangan *prototype* 45
Gambar 4.15 Pemasangan sensor ultrasonik 47
Gambar 4.16 Pemasangan sensor *rotary encoder* 48
Gambar 4.17 Implementasi sistem keseluruhan 49
Gambar 5.1 Hasil pengujian metode *fuzzy* 55

DAFTAR LAMPIRAN

LAMPIRAN 1 PROGRAM *FUZZY* 62

BAB 1 PENDAHULUAN

Bab ini menjelaskan latar belakang dari penelitian, rumusan masalah yang harus diselesaikan, batasan masalah yang ada pada penelitian, tujuan dari penelitian, manfaat dari penelitian yang dilakukan, dan sistematika penulisan dari penelitian.

1.1 Latar belakang

Indonesia merupakan negara berkembang dengan tingkat kepadatan penduduk dari tahun ke tahun semakin bertambah. Hal ini menyebabkan banyaknya kecelakaan lalu lintas sebagaimana yang disampaikan oleh *World Health Organization* (WHO) bahwa Indonesia menempati urutan kelima di dunia. Data yang mengejutkan ditunjukkan oleh *Global Status Report on Road Safety* dalam WHO bahwa Indonesia menempati peringkat pertama yang mencapai 120 jiwa perharinya (Republika, 2014).

Faktor yang menyebabkan kecelakaan, salah satunya adalah pengemudi yang melebihi batas kecepatan. Karena lalu lintas mayoritas jarah jauh, maka kecepatan rata-rata tinggi (Simanungkalit, 2014). Semakin cepat pengemudi, maka semakin sedikit waktu yang dimiliki untuk menghentikan laju kendaraan, dengan kata lain tidak memiliki cukup waktu untuk melakukan pengereman.

Hal ini mengindikasikan bahwa kendaraan bermotor wajib memiliki jarak aman, melihat kualitas dan pengereman yang biasanya menggunakan 10 persen antara kecepatan dan jarak pengereman seperti yang berlaku selama ini (Tribunnews, 2011). Seringnya pengendara yang mengabaikan jarak aman mengakibatkan pengurangan kecepatan secara spontan, yang membahayakan keselamatan pengendara tersebut, pengendera yang di depan serta yang di belakang.

Sebelumnya terdapat kendaraan yang sudah memiliki fitur monil pintar pada kendaraan, yaitu mobil bermerek ford. Mobil tersebut memiliki fitur *Actice Stop City,* yang membantu mobil mengerem secara otomatis jika ada mobil di depannya mengerem mendadak, sehingga terhindar tabrakan (Maruli, 2012). Selain itu, terdapat fitur teknologi dari mobil bermerek ford, yaitu mobil tanpa sopir yang dapat berkomunikasi dengan membaca lampu lalu lintas, mendeteksi rambu-rambu lalu lintas serta mendeteksi kendaraan di sekitar mobil (Suhartono Anton, 2015). Fitur tersebut digunakan untuk alat yang dirancang oleh peneliti, sebagai pengatur jarak aman dan pengontrol kecepatan.

Oleh karena itu, perlu adanya sistem yang dapat mengurangi kecelakaan tersebut. Penelitian sebelumnya pernah dilakukan oleh Gunawan, yang berhubungan dengan *Embedded System* sebagai pengontrol kecepatan yang berjudul ”Purwarupa Sistem Kendali Kecepatan Mobil Berdasarkan Jarak dengan

Sistem Inferensi *Fuzzy* Tsukamoto" (Gunawan, 2013). Namun meninjau dari penelitian Gunawan sebelumnya hanya menggunakan satu variabel yaitu jarak, maka menurut peneliti kurang efisien dikarenakan pada sebuah kendaraan memiliki jarak dan kecepatan.

Pada penelitian tersebut, penulis mengusulkan alat pengontrol kecepatan berdasarkan jarak aman untuk pengaplikasian menggunakan *fuzzy* Sugeno lalu mengimplementasikannya pada sistem. Peneliti menggunakan *fuzzy* Sugeno karena lebih mudah, dalam penentuan output menghasilkan sebuah konstanta yang berguna untuk pengurangan kecepatan.

Berdasarkan penelitian di atas, maka peneliti membuat judul "*Embedded System* Sebagai Pengontrol Kecepatan Kendaraan Berdasarkan Jarak Aman Menggunakan Metode *Fuzzy* Sugeno". Sistem ini terdiri dari sensor ultrasonik yang digunakan sebagai sensor jarak, sensor *rotary encoder* yang digunakan sebagai sensor kecepatan, serta arduino sebagai kontroller, lalu membuat sistem tersebut ke dalam *prototype.* Dengan adanya penelitian ini diharapkan dapat mengurangi terjadinya kecelakaan kendaraan bermotor dengan lebih efektif sehingga meminimalkan jumlah korban.

1.2 Rumusan masalah

Berdasarkan pemaparan pada latar belakang, maka dapat dirumuskan permasalahan sebagai berikut:

1. Bagaimana menerapkan *fuzzy* kedalam *prototype*?
2. Seberapa besar keberhasilan sistem menjaga jarak terhadap kendaraan di depan?

1.3 Tujuan

Tujuan dari penelitian ini adalah untuk mengimplementasikan *fuzzy* dalam dalam bentuk *prototype* untuk kendaraan bermotor dan membuat mampu menjaga jarak dengan kendaraan yang ada di depan.

1.4 Manfaat

Adapun manfaat dari *prototype* adalah sebagai berikut:

1. Untuk membantu mengurangi kecelakaan bermotor dengan alat tersebut.
2. Untuk membantu produsen untuk menerapkan alat tersebut untuk keselamatan berkendaraan.

1.5 Batasan masalah

Agar permasalahan yang telah dirumuskan dapat lebih fokus, maka skripsi ini dibatasi dalam lingkup beberapa hal yaitu sebagai berikut:

1. Sistem ini diimplementasikan ke dalam bentuk *prototype*.

2. Metode *fuzzy* yang digunakan adalah *fuzzy* Sugeno.
3. Sensor untuk mendeteksi jarak adalah ultrasonik.
4. Sensor untuk memdeteksi kecelakaan adalah *rotary encoder.*
5. *Hardware* yang digunakan adalah arduino AtMega 2560.

1.6 Sistematika penulisan

Sistematika pembahasan ditunjukan untuk memberikan gambaran dan uraian dari penulisan skripsi ini secara garis besar yang meliputi beberapa bab sebagai berikut:

BAB I : PENDAHULUAN

Pada bab ini berisi tentang penguraian latar belakang masalah yang dikaji, rumusan masalah, batasan masalah pada penelitian, tujuan penelitian, manfaat penelitian, serta sistematika penulisan.

BAB II : KAJIAN PUSTAKA DAN DASAR TEORI

Pada bab ini berisi referensi dan dasar teori yang mendasari pembuatan Metode *Fuzzy* serta sensor-sensor yang akan digunakan kedalam *prototype*.

BAB III : METODOLOGI PENELITIAN

Pada bab ini menguraikan dan membahas langkah kerja yang dilakukan dalam penulisan skripsi yang terdiri dari studi literatur, implementasi, perancangan yang meliputi desain modul sistem secara keseluruhan, perumusan masalah, blok diagram sistem, analisis kebutuhan dan pengujian.

BAB IV : PERANCANGAN DAN IMPLEMENTASI

Bab ini membahas perancangan dari sistem yang akan dibuat dan implementasikannya disertai dengan potongan source code yang penting dalam sistem tersebut.

BAB V : PENGUJIAN DAN ANALISIS

Bab ini berisi hasil uji coba sistem apakah dapat diterima dengan baik oleh user dan analisis hasil pengujian apakah sistem yang dibuat berhasil.

BAB VI : KESIMPULAN DAN SARAN

Pada bab ini berisi kesimpulan atas penelitian yang telah dilakukan, serta memberikan saran untuk pengembangan sistem lebih lanjut.

BAB 2 KAJIAN PUSTAKA DAN DASAR TEORI

Bab ini menjelaskan tinjauan pustaka yang meliputi kajian pustaka dan dasar teori yang diperlukan. Kajian pustaka membahas penelitian-penelitian terkait dengan sensor, medote *fuzzy* yang telah dilakukan sebelumnya. Dasar teori membahas teori yang diperlukan untuk menyusun penelitian yang dilakukan.

2.1 Kajian pustaka

Pada penelitan sebelumnya yang dilakukan oleh Niko Karis Gunawan menggunakan sensor Jarak dan juga Arduino Uno – ATMega 328p untuk melakukan kendali kecepatan prototype dengan menggunakan metode inteferensi *Fuzzy* Tsukamoto. (Gunawan, 2013)

Penggunaan metode *fuzzy* yang dilakukan oleh Niko Karis Gunawan yaitu menggunakan *fuzzy* Tsukamoto yang menggunakan variable-variabel dari jarak (Gunawan, 2013). Variabel *error* jarak dan delta *error* jarak sebagai *input* penelitian Niko Karis Gunawan menghasilkan *output* berupa lambat, cepat, sedang. Dengan mengacu pada penelitian Niko Karis Gunawan, penulis mencoba untuk menerapkan pada variable yang berbeda yaitu dengan menggunakan jarak dan kecepatan dengan mengunakan dua sensor. Pertama, sensor ultrasonik digunakan untuk *input* jarak. Kedua, sensor *rotary encoder* yang digunakan untuk *input* kecepatan. Selanjutnya kedua *input* tersebut diproses dengan menggunakan metode yang berbeda, yaitu menggunakan *fuzzy* Sugeno, sedangkan *output* digunakan untuk mengurangi kecepatan kendaraan.

2.2 Dasar teori

Pada dasar teori ini akan membahas tentang dasar-dasar penulis dalam membuat sistem. Dasar teori yang digunakan oleh penulis antara lain logika *fuzzy*, arduino, sensor ultrasonik, sensor *rotary encoder, Liquid crystal display*, dan *Modul driver motor*.

2.2.1 Dasar logika *fuzzy*

Logika *Fuzzy* adalah suatu cara yang tepat untuk memetakan suatu ruang input kedalam suatu ruang output, mempunyai nilai kontinyu (Nasution, 2012). Oleh karena itu, dapat didefinisikan menjadi sebagian benar dan sebagian waktu salah pada waktu yang sama.

Logika adalah ilmu yang mempelajari secara sistematis aturan-aturan penalaran yang absah (*valid*). Logika yang biasa dipakai dalam penalaran ilmiah yaitu logika yang setiap pernyataan mempunyai dua kemungkinan nilai, 1 dan 0. Logika *fuzzy* merupakan sebuah metode yang tepat untuk memetakan ruang lingkup *input* dan juga *output* (Frans Susilo, 2006).

Variabel numeris dan linguistik adalah suatu lambang atau kata yang menunjukkan kepada sesuatu yang tidak tertentu dalam semesta pembicaraannya (Frans Susilo, 2006). Suatu variabel dapat diganti oleh unsur-

unsur dalam semesta pembicaraannya, ada dua macam variabel dalam logika *fuzzy*, yaitu variabel linguistik dan variabel numeris.

Variabel linguistik merupakan suatu variabel yang semesta pembicaraannya berupa himpunan kata atau istilah yang digunakan sehari-hari. Misalnya: rendah, sedang, banyak, tinggi, pendek, cepat, dan sebagainya.

Variabel numeris merupakan suatu variabel yang semesta pembicaraannya berupa himpunan bilangan-bilangan. Misalnya *x* adalah bilangan yang habis dibagi 2. Variabel "*x*" adalah suatu variabel numeris karena menunjuk sesuatu yang tidak tentu dalam semesta pembicaraanya yaitu himpunan bilangan.

2.2.2 Himpunan *fuzzy*

Pada himpunan tegas (crips),nilai keanggotaan suatu item x dalam suatu himpunan A,yang sering ditulis dengan $\mu_A[X]$,memiliki 2 kemungkinan, yaitu:

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

Himpunan fuzzy memiliki 2 attribut, yaitu:

a. Linguistik, penamaan suatu grub yang mewakili suatu keadaan atau kondisi tertentu dengan bahasa alami, seperti: MUDA,TUA.
b. Numeris, suatu nilai (angka) yang menunjukkan ukuran dari suatu variabel seperti: 10,20,30, dsb.

Ada beberapa hal yang perlu diketahui dalam memahami sistem *fuzzy*, yaitu:

a. Variabel *fuzzy*, merupakan variabel yang hendak dibahas dalam suatu sistem *fuzzy*. Contoh: umur, temperatur, permintaan, dsb.
b. Himpunan *fuzzy,* merupakan suatu grup yang mewakili suatu kondisi atau keadaan tertentu dalam suatu variabel *fuzzy.*
c. Semesta Pembicaraan, 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. Adakalanya nilai semesta pembicaraan ini tidak dibatasi batas atasnya.

Didalam himpunan *fuzzy* terdapat beberapa bagian antara lain variabel fuzzy, semesta pembicaraan, domain, dan fungsi keanggotaan. Variabel *fuzzy* merupakan suatu lambang atau kata yang menunjuk kepada suatu yang tidak tertentu dalam sistem *fuzzy*. Contoh: suhu, kelembaban, penyiraman, dan lainnnya.

Semesta pembicaraan adalah rentang total dari nilai yang diizinkan untuk diolah dalam suatu variabel *fuzzy*. Domain himpunan *fuzzy* adalah rentang nilai yang diijinkan dalam semesta pembicaraan dan boleh diolah dalam suatu himpunan *fuzzy*.

2.2.3 Fungsi keanggotaan

Fungsi keanggotaan (*membership function*) adalah suatu kurva yang menunjukkan pemetaan titik-titik *input* data kedalam nilai keanggotaan yang memiliki interval antara 0 sampai 1. Ada beberapa fungsi yang bisa digunakan diantaranya :

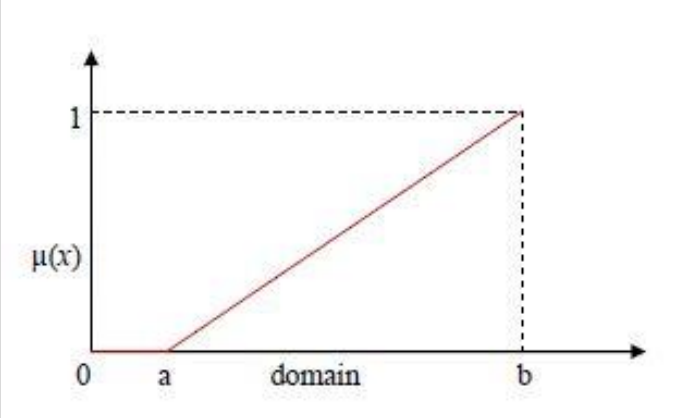
a. Linear.

b. Kurva Segitiga.

c. Kurva Trapesium.

Pada linear, pemetaan *input* ke derajat keanggotaannya dapat digambarkan sebagai suatu garis lurus. Ada 2 keadaan himpunan *fuzzy* yang linear:

1. linear naik, yaitu kenaikan himpunan dimulai dari nilai domain yang memiliki nilai keanggotaan nol [0] menuju arah kanan dan ke nilai domain yang memiliki derajat keanggotaan yang lebih tinggi (Gambar 2.1).



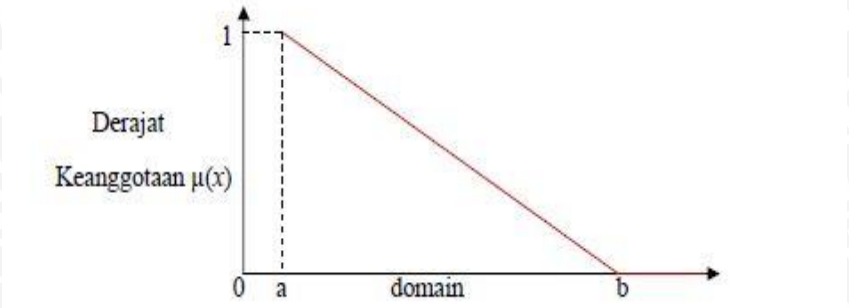
Gambar 2.1 Repesentasi linear naik

Sumber : Aplikasi Logika *Fuzzy* Metode Mamdani dan Sugeno

Fungsi keanggotaan dapat dilihat pada persamaan (2.1).

$$\mu(x)=\begin{cases} 0 & : x \leq a \\ \frac{(x-a)}{(b-a)} & : a < x \leq b \end{cases} \qquad (2.1)$$

2. linear turun, yaitu garis lurus yang dimulai dari nilai domain dengan derajat keanggotaan tertinggi menuju keanggotaan yang lebih rendah.(Gambar 2.2).



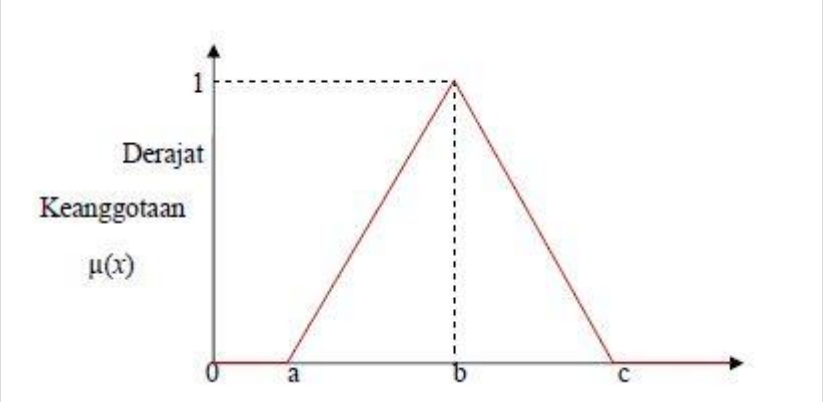
Gambar 2.2 Representasi linear turun

Sumber : Aplikasi Logika *Fuzzy* Metode Mamdani dan Sugeno

Fungsi keanggotaan keanggotaan dapat dilihat pada persamaan (2.2).

$$\mu(x)=\begin{cases}\frac{(x-a)}{(b-a)} & : a \le x < b \\ 0 & : x \ge b\end{cases} \qquad (2.2)$$

Representasi kurva segitiga, adalah gabungan dari linear naik dan linear turun seperti terlihat pada Gambar 2.3.



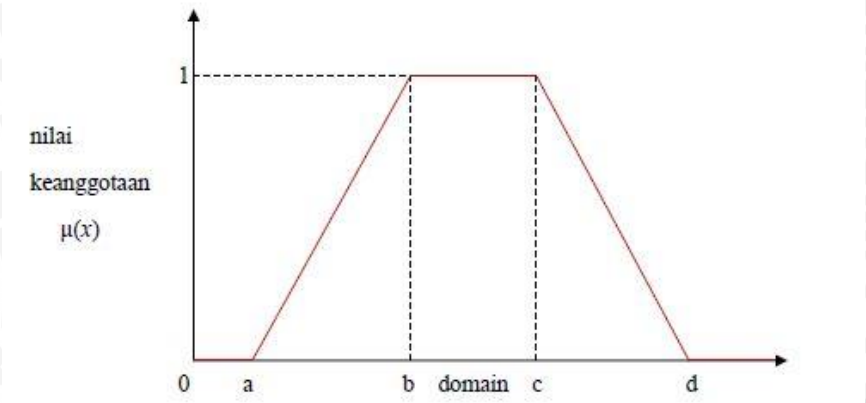
Gambar 2.3 Representasi linear naik dan turun

Sumber : Aplikasi Logika *Fuzzy* Metode Mamdani dan Sugeno

Fungsi keanggotaan keanggotaan dapat dilihat pada persamaan (2.3).

$$\mu(x)=\begin{cases}0 & : x \le a \; dan \; x \ge c \\ \frac{(x-a)}{(b-a)} & : \; a < x \le b \\ \frac{(c-x)}{(c-b)} & : \; b < x < c\end{cases} \qquad (2.3)$$

Kurva trapesium bentuknya mirip seperti kurva segitiga, hanya saja ada beberapa titik yang memiliki nilai keanggotaan 1 (satu), seperti pada Gambar 2.4.



Gambar 2.4 Representasi Kurva Trapesium

Sumber : Aplikasi Logika *Fuzzy* Metode Mamdani dan Sugeno

Fungsi Keanggotaan keanggotaan dapat dilihat pada persamaan (2.4).

$$\mu(x)=\begin{cases}0 & : x \le a \; atau \; x \ge d \\ \frac{(x-a)}{(b-a)} & : a < x \le b \\ \frac{(d-x)}{(d-c)} & : c < x < d \\ 1 & : b < x \le c\end{cases} \qquad (2.4)$$

Salah satu aplikasi logika *fuzzy* yang telah berkembang amat luas dewasa ini adalah sistem inferensi *fuzzy (Fuzzy Inference System/*FIS*)*. FIS adalah sistem komputasi yang bekerja atas dasar prinsip penalaran *fuzzy* seperti halnya manusia melakukan penalaran dengan nalurinya. Dalam subbab ini dibahas salah satu dari proses semacam itu, yaitu penentuan pengurangan kecepatan yang sesuai dengan kebutuhan pengguna. Sistem ini berfungsi untuk mengambil keputusan melalui proses tertentu dengan mempergunakan aturan inferensi berdasarkan logika *fuzzy*. Pada dasarnya sistem inferensi *fuzzy* terdiri dari tiga unit, yaitu:

1. Unit fuzzifikasi *(fuzzification unit)*
2. Unit penalaran logika *fuzzy* (*fuzzy logic reasoning unit)*
3. Unit defuzzifikasi (*defuzzification unit*/unit penegasan)

2.2.3.1 Unit fuzzifikasi

Proses fuzzifikasi merupakan proses mengubah variabel non *fuzzy* (*variable numerik*) menjadi variabel *fuzzy* (*variabel linguistik*) (Frans Susilo, 2006).Karena sistem inferensi *fuzzy* bekerja dengan aturan dan masukan dari fuzzy, maka langkah pertama yaitu mengubah *input* tegas yang diterima, menjadi masukan *fuzzy*, Itulah yang dilakukan oleh unit fuzzifikasi.

Untuk masing–masing variabel masukan, ditentukan suatu fungsi fuzzifikasi yang akan mengubah variabel masukan yang tegas (yang biasa dinyatakan dalam

bilangan real) menjadi nilai pendekatan *fuzzy*. Fungsi fuzzifikasi ditentukan berdasarkan kriteria di bawah ini :

1. Fungsi fuzzifikasi diharapkan mengubah suatu nilai tegas, misalnya *a* elemen R, ke suatu himpunan *fuzzy Ã* dengan nilai keanggotaan *a* terletak pada selang tertutup [0,1].
2. Bila nilai masukannya cacat karena gangguan (derau), diharapkan fungsi fuzzifikasi dapat menekan sejauh mungkin gangguan itu.
3. Fungsi fuzzifikasi diharapkan dapat membantu menyederhanakan komputasi yang harus dilakukan oleh sistem tersebut dalam proses inferensinya.

2.2.3.2 Unit penalaran

Penalaran *fuzzy* atau yang sering disebut juga dengan penalaran hampiran adalah suatu cara penarikan kesimpulan berdasarkan seperangkat implikasi *fuzzy* dan suatu fakta yang diketahui (sering disebut premis).

Pada unit penalaran terdapat 3 operator yang digunakan dalam melakukan inferensi *fuzzy*, yaitu operator AND, operator OR, dan operator NOT.

a. Operator AND berhubungan dengan operasi interseksi pada himpunan. *a*-predikat sebagai hasil operasi dengan operator AND diperoleh dengan mengambil nilai keanggotaan terkecil (min) antar elemen pada himpunan-himpunan yang bersangkutan.

$$\mu A \cap B = \min(\mu A[x], \mu B[y]) \quad (2.5)$$

b. Operator OR berhubungan dengan operasi union pada himpunan. *a*-predikat sebagai hasil operasi dengan operator OR diperoleh dengan mengambil nilai keanggotaan terbesar (max) antar elemen pada himpunan-himpunan yang bersangkutan.

$$\mu A \cup B = \max(\mu A[x], \mu B[y]) \quad (2.6)$$

c. Operator NOT berhubungan dengan operasi komplemen himpunan. Predikat sebagai hasil operasi dengan operator NOT diperoleh dengan mengambil nilai keanggotaan komplemen antar elemen pada himpunan-himpunan yang bersangkutan.

$$\mu A' = | - \mu A[x] \quad (2.7)$$

2.2.3.3 Unit defuzzifikasi

Unit defuzzifikasi digunakan untuk menghasilkan nilai variabel solusi yang diinginkan dari suatu daerah keluaran *fuzzy*. Karena sistem inferensi hanya dapat membaca nilai logika tegas, maka diperlukan suatu mekanisme untuk mengubah nilai keluaran *fuzzy* itu menjadi nilai logika tegas. Itulah yang dilakukan unit defuzzifikasi yang memuat fungsi-fungsi penegasan dalam sistem itu.

2.2.4 *Fuzzy* sugeno-takagi

Dalam penelitian ini digunakannya metode *fuzzy* sugeno karena dengan hasil keluaran sugeno yang bernilai konstan dengan komputasi yang efisien, maka bisa diambil sebuah keputusan yang berdasarkan pada hasil pengujian. (Frans Susilo, 2006)

Penalaran metode Sugeno ini hampir sama dengan penalaran Mamdani, hanya saja *output* sistem tidak berupa himpunan *fuzzy*, melainkan berupa konstanta atau persamaan linear. Metode ini diperkenalkan oleh Takagi-Sugeno Kang pada tahun 1985. Perbedaan antara Metode Mamdani dan Metode Sugeno ada pada konsekuen. Metode Sugeno menggunakan konstanta atau fungsi matematika dari variabel *input*: *Jika a adalah Ai dan b adalah Bi maka c adalah Ci = f(a,b).*

Dengan *a, b* dan *c* adalah variabel linguistik; *Ai* dan *Bi* himpunan *fuzzy* ke-*I* untuk *a* dan *b*, dan *f(a,b)* adalah fungsi matematik. Untuk mendapatkan *output* (hasil), maka terdapat 4 langkah / tahapan sebagai berikut:

1. Pembentukan himpunan *fuzzy*

   Langkah ini merupakan langkah pertama *fuzzy* yaitu membentuk himpunan keanggotaan masing-masing *input fuzzy*.

2. Aplikasi fungsi implikasi

   Menyusun basis aturan, yaitu aturan-aturan berupa implikasi *fuzzy* yang menyatakan relasi antara variabel *input* dengan variabel *output*. Pada metode Sugeno, fungsi implikasi yang digunakan adalah *Min*. Bentuk umumnya adalah sebagai berikut:*Jika a adalah Ai dan b adalah Bi maka c adalah Ci = f(a,b)*

3. Komposisi aturan

   Apabila sistem terdiri dari beberapa aturan, maka inferensi diperoleh dari kumpulan dan korelasi antar aturan. Metode yang digunakan dalam melakukan inferensi sistem *fuzzy* adalah metode *centroid*. Pada metode ini, solusi himpunan *fuzzy* diperoleh dengan cara mengambil nilai minimum aturan, kemudian menggunakan nilai tersebut untuk memodifikasi daerah *fuzzy* dan mengaplikasikannya ke *output* dengan menggunakan operator AND (irisan). Jika semua proporsi telah dievaluasi, maka *output* akan berisi suatu himpunan *fuzzy* yang merefleksikan kontribusi dari tiap-tiap proporsi. Secara umum dapat dituliskan pada persamaan 2.8.

   $$\mu(xi) = max(\mu sf(xi), \mu kf(xi)) \quad (2.8)$$

   dengan:
   µsf (*x*i) = nilai keanggotaan solusi *fuzzy* sampai aturan ke-I µkf (*x*i) = nilai keanggotaan konsekuen *fuzzy* aturan ke-i.

4. Defuzzifikasi dengan metode centroid

   Masukan dari proses penegasan adalah suatu himpunan *fuzzy* yang diperoleh dari komposisi aturan-aturan *fuzzy*, sedangkan *output* yang

dihasilkan merupakan suatu bilangan *real* yang tegas. Sehingga jika diberikan suatu himpunan *fuzzy* dalam range tertentu, maka dapat diambil suatu nilai tegas tertentu sebagai *output.* Apabila komposisi aturan menggunakan metode Sugeno maka defuzzifikasi (Z*) dilakukan dengan cara mencari nilai rata-rata terpusatnya. Seperti yang dituliskan pada persamaan 2.9.

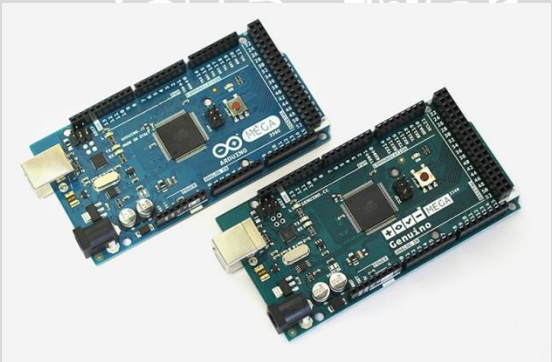
$$Z^* = \frac{\sum_{i=1}^{n} d_i U_{\widetilde{A}_i}(d_i)}{\sum_{i=1}^{n} U_{\widetilde{A}_i}(d_i)} \quad (2.9)$$

dengan *di* adalah nilai keluaran pada aturan ke-*i* adalah derajat keanggotaan nilai keluaran pada aturan ke-*i* sedangkan *n* adalah banyaknya aturan yang digunakan.

2.2.5 Arduino atmega 2560

Arduino Atmega 2560 memiliki 54 pin *digital input* dan *output*, 15 dari 54 pin tersebut dapat digunakan sebagai *output* PWM. 16 *input* analog, sebuah osilator Kristal 16 MHz, sebuah *port* USB, sebuah power jack DC 12 volt, sebuah ICSP header, dan sebuat tombol reset. (Arduino, 2015)

Arduino Atmega 2560 memiliki semua yang dibutuhkan untuk melakukan sebuah kontrol, Arduino Atmega 2560 juga mudah dihubungkan ke sebuah komputer dengan sebuah kabel USB atau mensuplainya dengan sebuah adaptor AC ke DC atau menggunakan baterai.Gambar Arduino UNO dapat dilihat pada gambar 2.5.



Gambar 2.5 Arduino Atmega 2560

Sumber :http://arduino.cc/en/main/arduinoBoardMega2560

Tabel 2.1 Spesifikasi Arduino AtMega2560

Mikrokontroler	ATmega328
Tegangan Operasi	5 Volt
Input Voltage (disarankan)	7 – 12 Volt
Input Voltage (batas akhir)	6 – 20 Volt
Digital I/O Pin	54 (15 pin sebagai *output* PWM)
Analog *Input* Pin	16
Arus DC per pin I/O	20 mA
Arus DC untuk pin 3.3V	50 mA
Flash Memory	256 KB, 8 KB untuk bootloader
SRAM	8 KB
EEPROM	4 KB
Clock Speed	16 MHz

(Sumber: www.arduino.cc)

2.2.6 Sensor Rotary Encoder

Pada perancangan sensor kecepatan, sensor kecepatan yang akan digunakan adalah *rotary encoder*. *Rotary encoder* umumnya menggunakan sensor optik untuk menghasilkan serial pulsa yang dapat diartikan menjadi gerakan, posisi, dan arah. Sehingga posisi sudut suatu poros benda berputar dapat diolah menjadi informasi berupa kode digital oleh *rotary encoder* untuk diteruskan oleh rangkaian kendali. Gambar *rotary encoder* dapat dilihat pada gambar 2.6.

Gambar 2.6 Sensor *Rotary encoder*

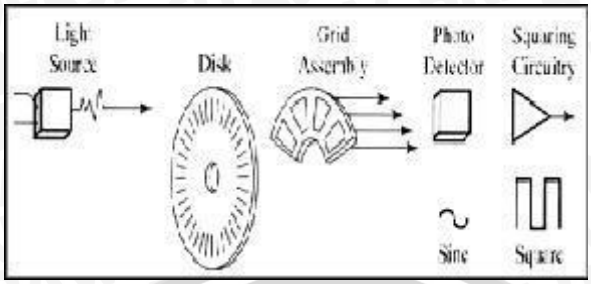
Sumber: indo-ware.com

Tabel 2.2 Spesifikasi sensor *Rotary encoder*

Sensor	*Rotary encoder*
Tegangan Suplai	DC 5 Volt
Output	*Digital*
Logika output "1"	saat tidak mendeteksi lubang piringan
Logika output "0"	saat mendeteksi lubang piringan
Banyak Lubang	36

(Sumber: http:// indo-ware.com)

Rotary encoder tersusun dari suatu piringan tipis yang memiliki lubang-lubang padabagian lingkaran piringan. LED ditempatkan pada salah satu sisi piringan sehingga cahaya akan menuju ke piringan. Di sisi yang lain suatu photo-transistor diletakkan sehingga photo-transistor ini dapat mendeteksi cahaya dari LED yang berseberangan. Piringan tipis tadi dikopel dengan poros motor, atau divais berputar lainnya yang ingin kita ketahui posisinya, sehingga ketika motor berputar piringan juga akan ikut berputar. Apabila posisi piringan mengakibatkan cahaya dari LED dapat mencapai photo-transistor melalui lubang-lubang yang ada, maka photo-transistor akan mengalami saturasi dan akan menghasilkan suatu pulsa gelombang persegi. Gambar 2.7 menunjukkan bagan skematik sederhana dari *rotary encoder*. Semakin banyak deretan pulsa yang dihasilkan pada satu putaran menentukan akurasi rotary encoder tersebut, akibatnya semakin banyak jumlah lubang yang dapat dibuat pada piringan menentukan akurasi rotary encoder tersebut.



Gambar 2.7 Penyusun *rotary encoder*

Sumber: http://www.automationdirect.com/static/catalog/21-sensor-encoder-rotary.pdf

2.2.7 Sensor ultrasonik

Sensor ultrasonik PING adalah sensor 40 KHz produksi parallax yang banyak digunakan untuk aplikasi atau kontes robot cerdas untuk mendeteksi jarak suatu objek.Sensor PING mendeteksi jarak objek dengan cara memancarkan gelombang ultrasonik (40 KHz) selama t = 200 us kemudian mendeteksi pantulannya. Sensor PING memancarkan gelombang ultrasonik sesuai dengan kontrol dari mikrokontroller pengendali (pulsa trigger dengan tout min 2 us). Gambar sensor ultrasonik PING dapat dilihat pada gambar 2.8.

Gambar 2.8 Sensor Ultrasonik

Sumber : electroschematics.com

Tabel 2.3 Spesifikasi sensor ultrasonik

Sensor	ultrasonik hc-sr04
Tegangan Suplai	DC 5 Volt
Jarak terjauh	400 cm
Jarak terdekat	2 cm
Frekuensi sinyal	40 Hz
Jenis Pin	VCC,*Ground,Trigger,Echo*

(Sumber: http://www.electroschematics.com/8902/hc-sr04-datasheet/)

2.2.8 LCD (*Liquid Crystal Display*) 16 x 2

LCD adalah salah satu komponen elektronika yang berfungsi sebagai tampilan suatu data, baik karakter, huruf ataupun grafik. LCD (*Liquid Crystal Display*) adalah salah satu jenis display elektronik yang dibuat dengan teknologi CMOS logic yang bekerja dengan tidak menghasilkan cahaya tetapi memantulkan cahaya yang ada di sekelilingnya terhadap front-lit atau mentransmisikan cahaya dari back-lit. (engineersgarage, 2012)

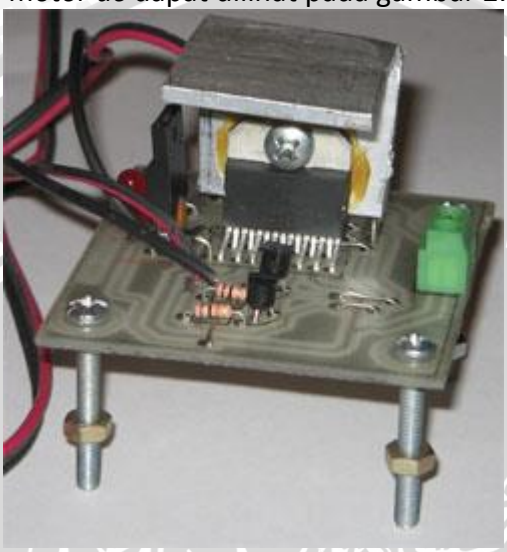
LCD (*Liquid Crystal Display*) berfungsi sebagai penampil data baik dalam bentuk karakter, huruf, angka ataupun grafik.Gambar LCD 16x2 dapat dilihat pada gambar 2.9.

Gambar 2.9 LCD 16x2

Sumber : instructables.com

2.2.9 Modul *driver motor DC*

Modul *Driver Motor* ini adalah sebagai PWM yang digunakan untuk mengontrol kecepatan pada *prototype,* peneliti mengunakan *driver motor* L298N.Untuk mengontrol *driver* L298N ini dibutuhkan 6 buah pin mikrokontroler.Pada prinsipnya rangkaian driver motor L298N ini dapat mengatur tegangan dan arus sehingga kecepatan dan arah motor dapat diatur. Gambar modul *driver motor dc* dapat dilihat pada gambar 2.10.



Gambar 2.10 Modul *Driver* Dc

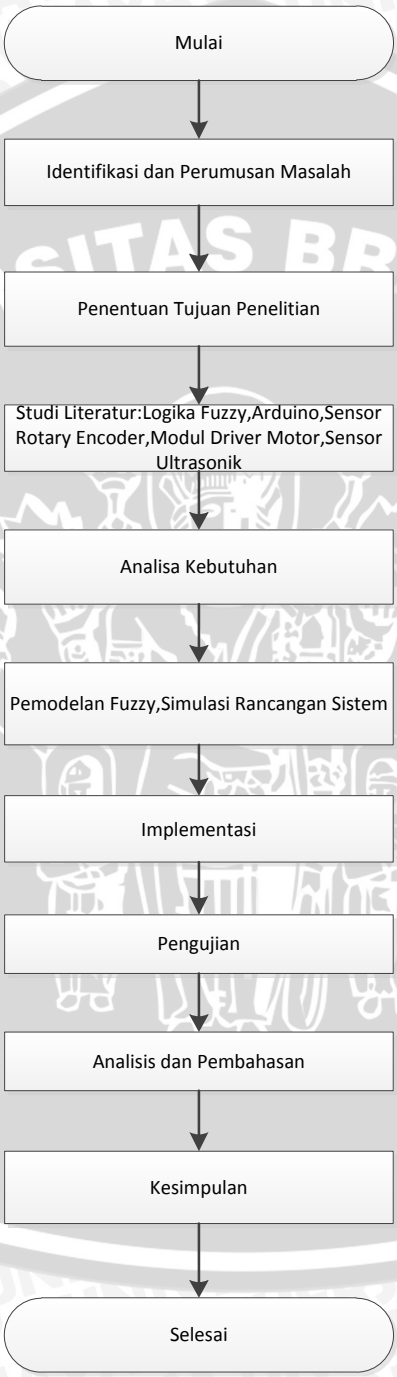
Sumber : ikalogic.com

Sinyal PWM pada umumnya memiliki amplitude dan frekuensi dasar yang tetap, namun memiliki lebar pulsa yang bervariasi. Lebar pulsa PWM berbanding lurus dengan amplitude sinyal asli yang belum termodulasi. Artinya, sinyal PWM memiliki frekuensi gelombang yang tetap namun duty cycle bervariasi antara 0% hingga 100%.PWM merupakan salah satu teknik untuk mendapatkan sinyal analog dari sebuah piranti digital.

Secara analog setiap perubahan PWM-nya sangat halus, sedangkan secara digital setiap perubahan PWM dipengaruhi oleh resolusi PWM itu sendiri. Resolusi adalah jumlah variasi perubahan nilai dalam PWM tersebut. Misalkan suatu PWM memiliki resolusi 8 bit, berarti PWM ini memiliki variasi perubahan nilai sebanyak 256 variasi mulai dari 0 – 255 perubahan nilai yang mewakili *duty cycle* 0% – 100% dari keluaran PWM tersebut.

BAB 3 METODOLOGI PENELITIAN

Pada bab ini akan menjelaskan langkah-langkah yang dilakukan dalam penyusunan skripsi. Metode penelitian yang digunakan pada skripsi ini ditunjukan pada diagram alir pada Gambar 3.1.



Gambar 3.1 Alur pelaksanaan

3.1 Perumusan masalah

Langkah ini merupakan awal dari penelitian, yaitu dengan mencari masukan terhadap masalah yang akan diteli melalui observasi. Peneliti merumuskan masalah bagaimana sistem bisa memberikan keputusan dalam mengurangi kecepatan.

3.2 Penentuan tujuan penelitian

Dalam penelitian ini ditetapkan beberapa tujuan untuk memfokuskan permasalahan dengan hasil akhir berupa laporan. Adapun tujuan dari penelitian ini adalah yang sudah dijelaskan pada bab 1. Hasil dari tujuan penelitian ini diharapkan bisa mengurai tingkat kecelakaan.

3.3 Studi literatur

Studi literatur digunakan sebagai teori penguat dan landasan dasar dalam penelitian. Teori pendukung tersebut diperoleh dari buku, jurnal, paper dan sumber lain yang dapat dipertanggung jawabkan. Literatur yang digunakan meliputi:

1. Logika *Fuzzy*
    a. Fuzzifikasi
    b. Penalaran
    c. Defuzzifikasi
2. Arduino
    a. Pengenalan Arduino
    b. Arduino IDE
    c. Pemrograman Arduino
3. Sensor Ultrasonik
4. Sensor Rotary Encoder
5. Modul Driver Motor
6. LCD 16x2
7. I2C LCD

3.4 Analisis kebutuhan

Analisis kebutuhan bertujuan untuk mendapatkan semua kebutuhan yang dierlukan oleh sistem yang akan dibangun. Analisa kebutuhan ini disesuaikan dengan kebutuhan yang digunakan, kebutuhan tersebut meliputi :

3.4.1 Sistem kerja alat

Sistem kerja alat ini yaitu menggunakan dua buah sensor yaitu sensor Rotary Encoder dan sensor Ultrasonik yang digunakan untuk mendeteksi kendaraan yang ada pada depan *prototype*. Dan sensor Rotary Encoder digunakan untuk mengetahui kecepatan pada *prototype*.

Ketika kedua sensor tersebut mendeteksi jarak yang melewati batas jarak aman memberikan keputusan mikrokontroller akan mengurangi kecepatan melalui modul driver motor.

Pada sistem kerja alat tersebut bekerja secara fungsional atau berjalan sesuai dengan fungsinya, perbedaaan nilai dari sensor dan juga alat ukur masih ditoleransi yang penting sistem masih bisa berjalan sesuai dengan fungsi awalnya.

3.4.2 Perangkat keras

Perangkat keras pada sistem ini digunakan sebagai pengolah data dari *input* sehingga menghasilkan *output.* Beberapa perangkat keras yang dibutuhkan oleh sistem adalah sebagai berikut:

1. Mikrokontroler Arduino Atmega 2560
2. Sensor Ultrasonik
3. Sensor Rotary Encoder
4. LCD
5. Kabel Jumper
6. Modul Driver Motor
7. I2C LCD

3.4.3 Perangkat lunak

Perangkat lunak yang dibutuhkan untuk mendukung kerja pada sistem adalah sebagai berikut:

1. Sistem Operasi, sistem operasi yang digunakan untuk menjalankan perangkat lunak adalah Windows 7.
2. Arduino 1.0.4, sebuah perangkat lunak yang digunakan untuk membuat program yang dibutuhkan pada sistem yang akan dibuat,
3. Matlab 7.6.0 (R2008a), sebuah perangkat lunak yang digunakan untuk membuat simulasi program *fuzzy*.
4. Proteus 8, sebuah perangkat lunak yang digunakan untuk membuat simulasi komponen elektronik.

5. Microsoft Visio, sebuah perangkat lunak yang digunakan untuk membuat gambar flowchart.

3.5 Perancangan

Perancangan dalam pembuatan sistem ini meliputi dua bagian yaitu perancangan perangkat keras dan perancangan perangkat lunak.

3.5.1 Perancangan perangkat keras

Pada perancangan perangkat keras ini yaitu berupa skematik dari sistem yang akan dibuat. Pada perancangan perangkat keras ini juga berisi diagram alir sistem.

3.5.2 Perancangan perangkat lunak

Pada perancangan perangkat lunak ini berupa simulasi *fuzzy* Sugeno yang disimulasikan pada aplikasi matlab untuk melihat apakah aturan-aturang yang dibuat berjalan dengan baik atau tidak.

3.6 Implementasi

Implementasi sistem dilakukan dengan mengacu pada perancangan sistem yang sudah dibuat sebelumnya.Pada implementasi juga mencantumkan gambar sistem yang sudah dibuat serta potongan-potongan bahasa pemrograman yang digunakan.

3.7 Pengujian dan analisis

Pengujian sistem pada penelitian ini dilakukan agar dapat menunjukkan bahwa sistem telah mampu bekerja sesuai dengan spesifikasi dari keutuhan yang melandasinya. Pengujian yang dilakukan meliputi:

1. Pengujian semua sensor yang dibutuhkan
2. Pengujian logika *fuzzy*
3. Pengujian *output*
4. Pengujian secara keseluruhan.

3.8 Kesimpulan

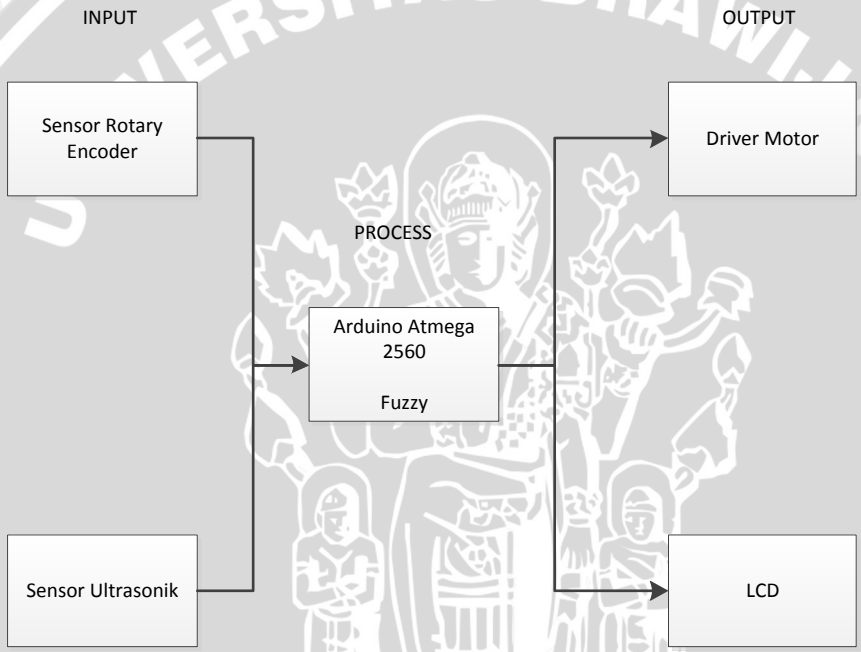
Kesimpulan didapatkan setelah melakukan perancangan, implementasi, pengujian dan analisis kepada sistem.Kesimpulan diambil berdasarkan hasil pengujian dan analisis sistem yang telah dibuat. Isi dari kesimpulan diharapkan dapat menjadi acuan pada penelitian lain untuk mengembangkan teknologi untuk membantu mengatur kecepatan kendaraan. Selain itu, pada akhir penulisan terdapat saran yang bertujuan untuk memperbaiki kesalahan-kesalahan yang terjadi dan menyempurnakan penulisan serta untuk memberikan pertimbangan atas pengembangan sistem selanjutnya.

BAB 4 PERANCANGAN DAN IMPLEMENTASI

Pada bab ini akan dijelaskan perancangan dari sistem yang dibuat dan kemudian implementasi dari perancangan. Bab ini akan memudahkan proses pengujian dan analisis karena didasarkan pada perancangan dan implementasi.

4.1 Blok diagram

Pada blok diagram ini akan menjelaskan tentang sistem kerja alat yang akan dibuat secara keseluruhan. Blok diagram ini gunakan untuk mempermudah pembaca dalam memahami alur kerja sistem. Blok diagram sistem dapat dilihat pada gambar 4.1.

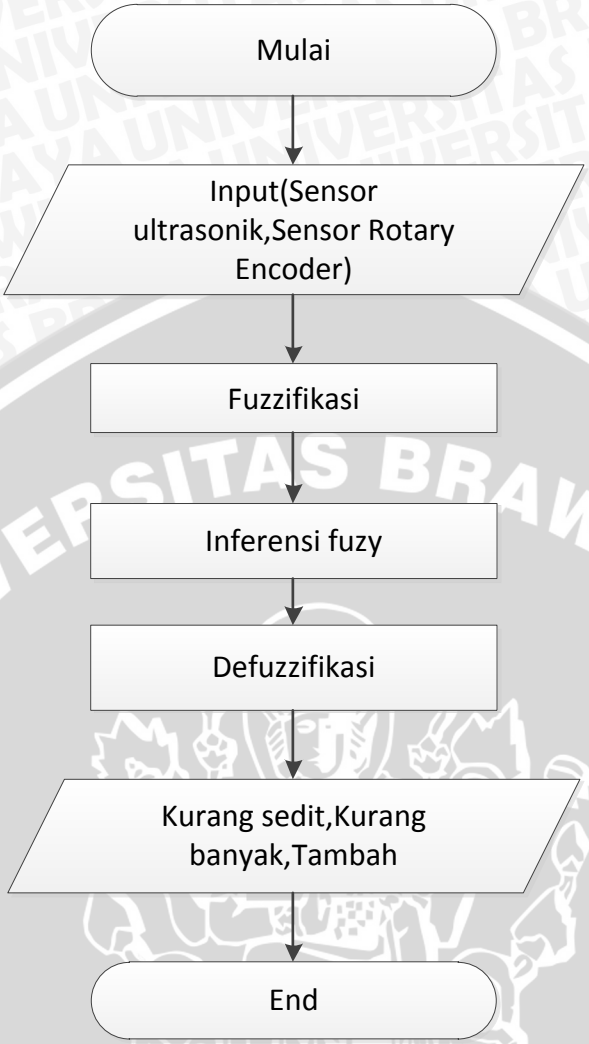


Gambar 4.1 Blok diagram sistem

4.2 Perancangan metode Fuzzy

4.2.1 Diagram alir metode *fuzzy*

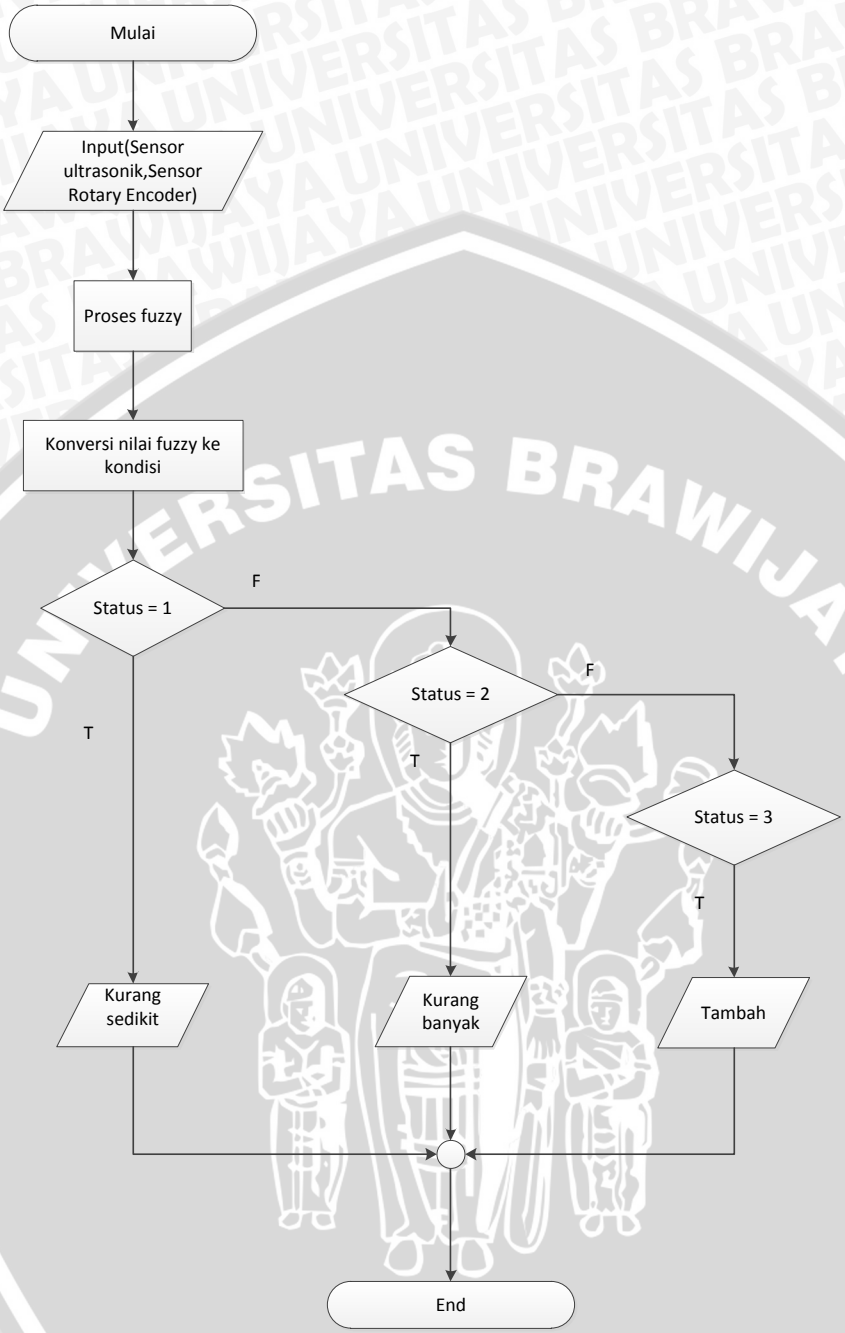
Pada diagram *fuzzy* dijelaskan urutan proses *fuzzy* dengan data yang berasal dari hasil uji sensor ultrasonik, dan sensor *rotary encoder*. Langkah-langkah pemrosesan metode *fuzzy* meliputi fuzzifikasi yaitu pembentukan variabel input, inferensi *fuzzy* yaitu pembentukan *rule base*, dan defuzzifikasi yaitu penentuan hasil. Alur proses *fuzzy* ditunjukan pada Gambar 4.2.



Gambar 4.2 Diagram alir metode *fuzzy*

4.2.2 Diagram alir sistem metode *fuzzy*

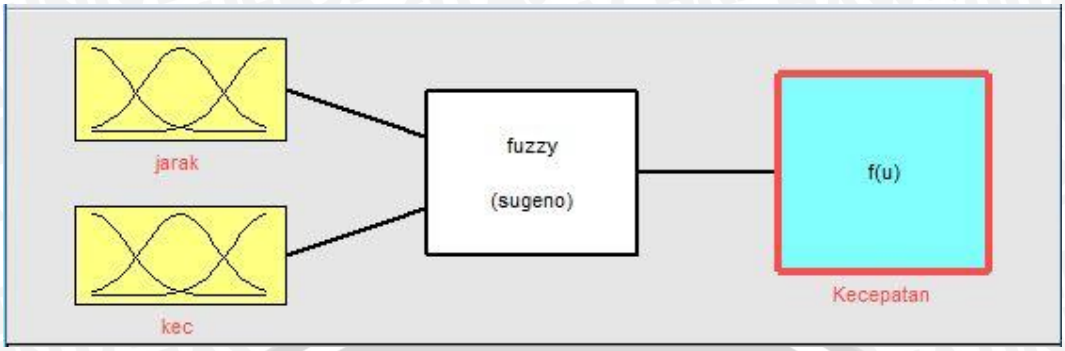
Pada diagram alir sistem menggunakan metode *fuzzy* dijelaskan urutan proses *fuzzy* dengan data yang berasal dari hasil baca sensor ultrasonik, dan sensor *rotary encoder*. Data sensor kemudian diproses dan dikonversi menggunakan metode *fuzzy* sehingga didapatkan *output* dari proses tersebut. Alur proses sistem dengan metode *fuzzy* ditunjukan pada Gambar 4.3.



Gambar 4.3 Diagram alir sistem dengan metode *fuzzy*

4.2.3 Perancangan fuzzy inference system

Pada perancangan *fuzzy inference system* langkah-langkah yang dilakukan meliputi perancangan *fuzzy input*, *fuzzy rule*, dan defuzzifikasi. Perancangan *fuzzy input* dilakukan dengan melakukan pemodelan fuzzifikasi dari dua variabel masukan yang diperlukan dalam sistem, yaitu jarak, dan kecepatan. Pemodelan variabel masukan tersebut meliputi: himpunan jarak, dan himpunan kecepatan. *Fuzzy Inference System* dapat dilihat pada Gambar 4.4.



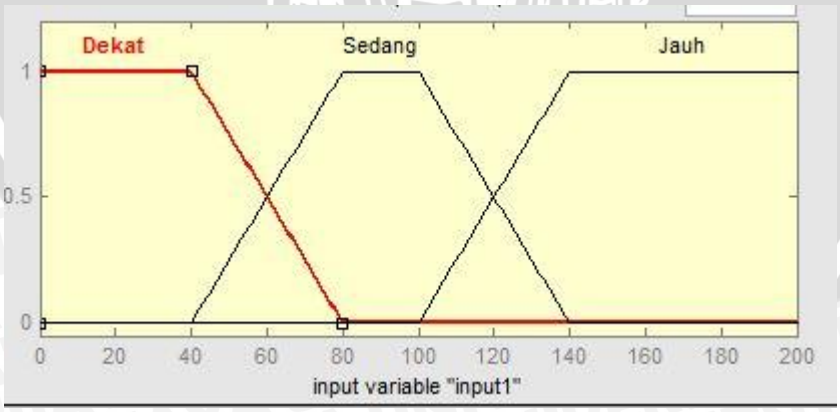
Gambar 4.4 *Fuzzy inference system*

4.2.3.1 Pemodelan *Fuzzy* Matlab

Pemodelan fuzzy dimatlab ini memodelkan fuzzifikasi dari beberapa variabel masukan yang diperlukan dalam sistem penulis yaitu jarak, dan kecepatan. Pemodelan variabel masukan tersebut meliputi himpunan jarak dan himpunan kecepatan.

4.2.3.2 Fuzzifikasi *input* jarak (variabel masukan)

Pada himpunan jarak berisi batas-batas nilai jarak pada sensor ultrasonik. Himpunan jarak memiliki tiga keanggotaan, yaitu dekat, sedang dan jauh. Penentuan keanggotaan dekat, sedang dan, jauh dilakukan dengan *range* tertentu dan menggunakan skala. Skala yang digunakan adalah 0 sampai 200 cm, rentang ini diperoleh dari nilai terkecil dan terbesar pada hasil pengujian. Himpunan keanggotaan dekat berada pada skala 0 sampai 60 cm, himpunan keanggotaan sedang berada pada skala 61 sampai 120 cm, dan himpunan keanggotaan jauh berada pada skala 121 cm ke atas. Pemodelan himpunan jarak ditunjukan pada Gambar 4.5.



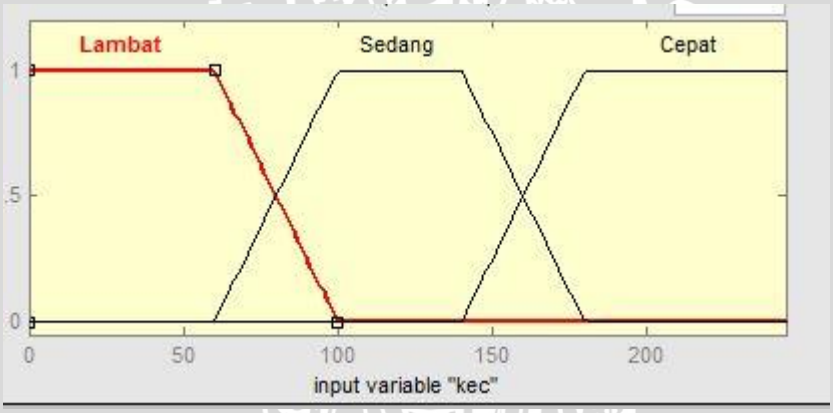
Gambar 4.5 Himpunan jarak

Berdasarkan gambar di atas dapat dilihat bahwa domain untuk fungsi keanggotaan dekat adalah [0 0 40 80]. Hal ini dikarenakan rata-rata jarak paling

dekat yang diperoleh ketika pengujian berada pada nilai 30 cm. Untuk domain dari fungsi keanggotaan sedang adalah [40 80 100 140]. Hal ini dikarenakan rata-rata jarak sedang yang diperoleh ketika pengujian berada pada nilai 90 cm. Terakhir, untuk domain dari fungsi keanggotaan jauh adalah [100 140 200 200]. Hal ini dikarenakan rata-rata jarak paling jauh yang diperoleh ketika pengujian berada pada nilai 160 cm.

4.2.3.2 Fuzzifikasi *input* kecepatan (variabel masukan)

Pada himpunan kecepatan berisi batas-batas nilai jarak pada sensor *rotary encoder*. Himpunan kecepatan memiliki 3 keanggotaan yaitu lambat, sedang dan cepat. Penentuan keanggotaan lambat, sedang dan, cepat dilakukan dengan range tertentu dan menggunakan skala. Skala yang digunakan adalah 0 sampai 245 rpm, rentang ini diperoleh dari nilai terkecil dan terbesar pada hasil pengujian. Himpunan keanggotaan lambat berada pada skala 0 sampai 75 rpm, himpunan keanggotaan sedang berada pada skala 76 sampai 160 rpm, dan himpunan keanggotaan cepat berada pada skala 161 rpm keatas. Pemodelan himpunan kecepatan ditunjukan pada Gambar 4.6.



Gambar 4.6 Himpunan kecepatan

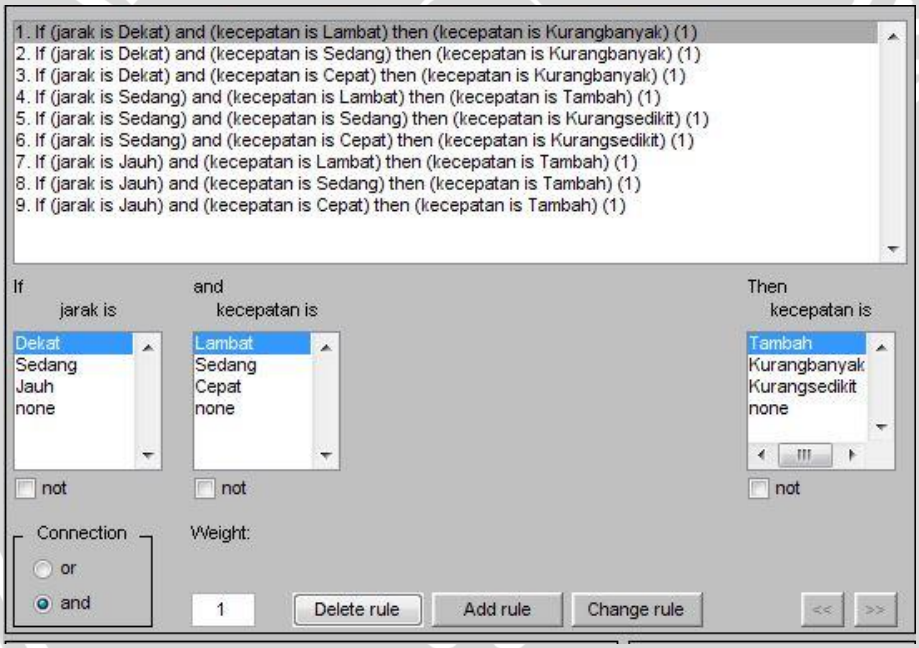
Berdasarkan gambar di atas dapat dilihat bahwa domain untuk fungsi keanggotaan lambat adalah [0 0 60 100]. Hal ini dikarenakan rata-rata kecepatan paling lambat yang diperoleh ketika pengujian berada pada nilai 50 rpm. Domain dari fungsi keanggotaan sedang adalah [60 100 140 180]. Hal ini dikarenakan rata-rata kecepatan sedang yang diperoleh ketika pengujian berada pada nilai 120 rpm. Selanjutnya, untuk domain dari fungsi keanggotaan cepat adalah [140 180 245 245]. Hal ini dikarenakan rata-rata kecepatan paling cepat yang diperoleh ketika pengujian berada pada nilai 200 cm.

4.2.3.3 Rule Fuzzy

Berdasarkan kedua variabel input di atas menghasilkan Inferensi *fuzzy* yang ditunjukan pada *rule* berikut:

1. IF (input1 is Dekat) AND (input2 is Lambat) then (output1 is Kurang_Banyak)
2. IF (input1 is Dekat) AND (input2 is Sedang) then (output1 is Kurang_Banyak)
3. IF (input1 is Dekat) AND (input2 is Cepat) then (output1 is Kurang_Banyak)
4. IF (input1 is Sedang) AND (input2 is Lambat) then (output1 is Tambah)
5. IF (input1 is Sedang) AND (input2 is Sedang) then (output1 is Kurang_sedikit)
6. IF (input1 is Sedang) AND (input2 is Cepat) then (output1 is Kurang_sedikit)
7. IF (input1 is Jauh) AND (input2 is Lambat) then (output1 is Tambah)
8. IF (input1 is Jauh) AND (input2 is Sedang) then (output1 is Tambah)
9. IF (input1 is Jauh) AND (input2 is Cepat) then (output1 is Tambah)

Berikut tampilan *rule fuzzy* pada aplikasi matlab yang dapat dilihat pada gambar 4.7:



Gambar 4.7 *Rules Fuzzy*

4.4.3.5 Defuzzifikasi *output* kecepatan (variabel keluaran)

Pada himpunan kecepatan ini memiliki 3 *output* yang di gunakan sebagai keluaran dari fuzzy yaitu kurang sedikit, kurang banyak, tambah. Kurang sedikit akan diwakili PWM bernilai 80, Kurang banyak diwakili PWM bernilai 5, sementara Tambah diwakili PWM bernilai 255. Berikut tampilan fuzzifikasi pada aplikasi matlab yang dapat dilihat pada gambar 4.8:

Gambar 4.8 *Output* kecepatan

Berdasarkan hasil pembuatan matlab, maka bisa di tampilkan dalam bentuk simulasi sebagai berikut:

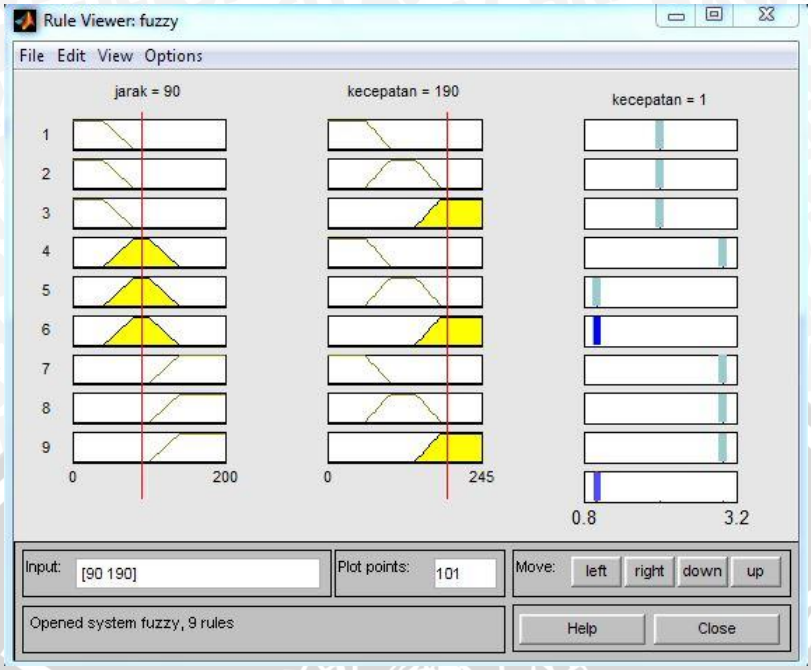
- Nilai sensor ultrasonik 90 cm
- Nilai sensor kecepatan 190 rpm

Langkah pertama yang dilakukan untuk menentukan *output* adalah menentukan derajat keanggotaan.

Sesuai range *fuzzy* jarak = 90 cm masuk ke dalam keanggotaan sedang, dan kecepatan = 190 rpm masuk kedalam keanggotaan cepat. Setelah diketahui derajat keanggotaan masing-masing *input* kemudian dilakukan pencocokan dengan *fuzzy rule* yang telah dibuat.

IF jarak sedang *AND kecepatan cepat THEN output=1*.

Maka hasil status yang sesuai dengan *input* sensor adalah kurang sedikit.
Contoh matlab dapat dilihat pada gambar 4.9.



Gambar 4.9 Simulasi Matlab

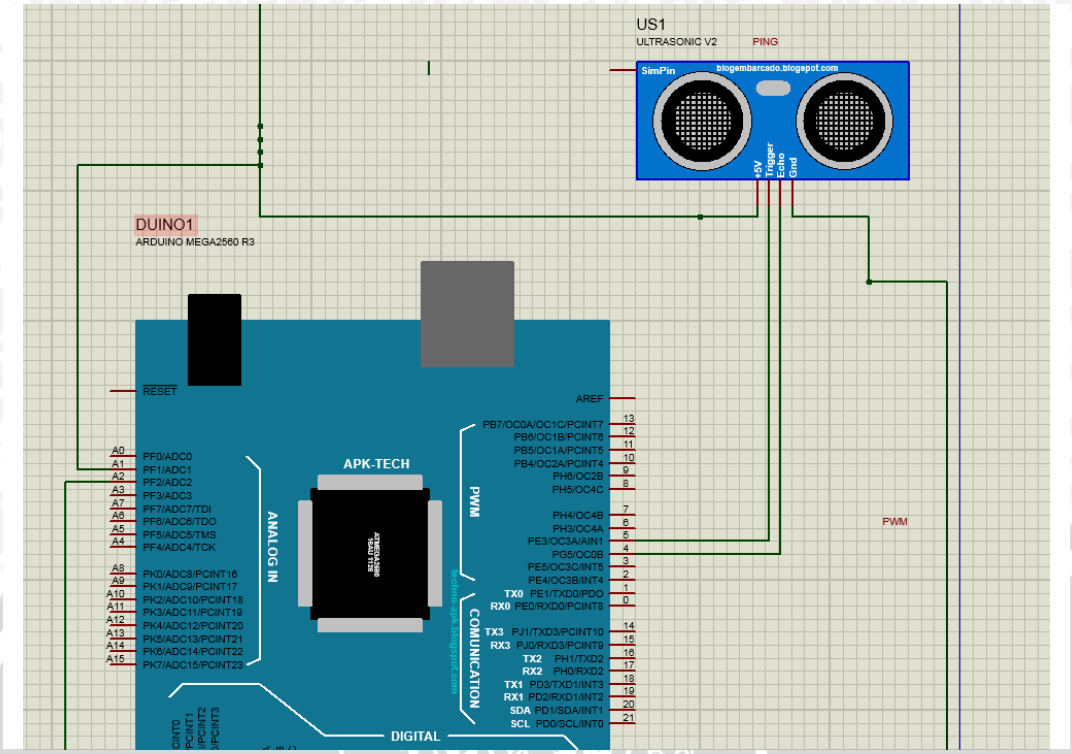
Setelah melakukan pencarian manual dan simulasi matlab, maka dapat disimpulkan bahwa *fuzzy* yang dibuat pada matlab dapat digunakan pada sistem *prototype*.

4.3 Perancangan rangkaian *prototype*

Pada perancangan komponen elektronik ini yaitu membuat perancangan pada aplikasi proteus sehingga dapat mempermudah saat merancang kealat yang sebenarnya.Pada perancangan komponen elektronik ini dibagi menjadi 3 yaitu perancangan sensor ultrasonik, perancangan sensor *rotary encoder*,dan perancangan keseluruhan

4.3.1 Perancangan sensor ultrasonik

Pada perancangan sensor ultrasonik ini penempatan pin input diletakkan pada pin PWM pada arduino, kemudian untuk daya diletakkan pada pin 5v dan untuk *ground* diletakkan di pin gnd. Berikut gambar perancangan sensor ultrasonik dapat dilihat pada gambar 4.10.

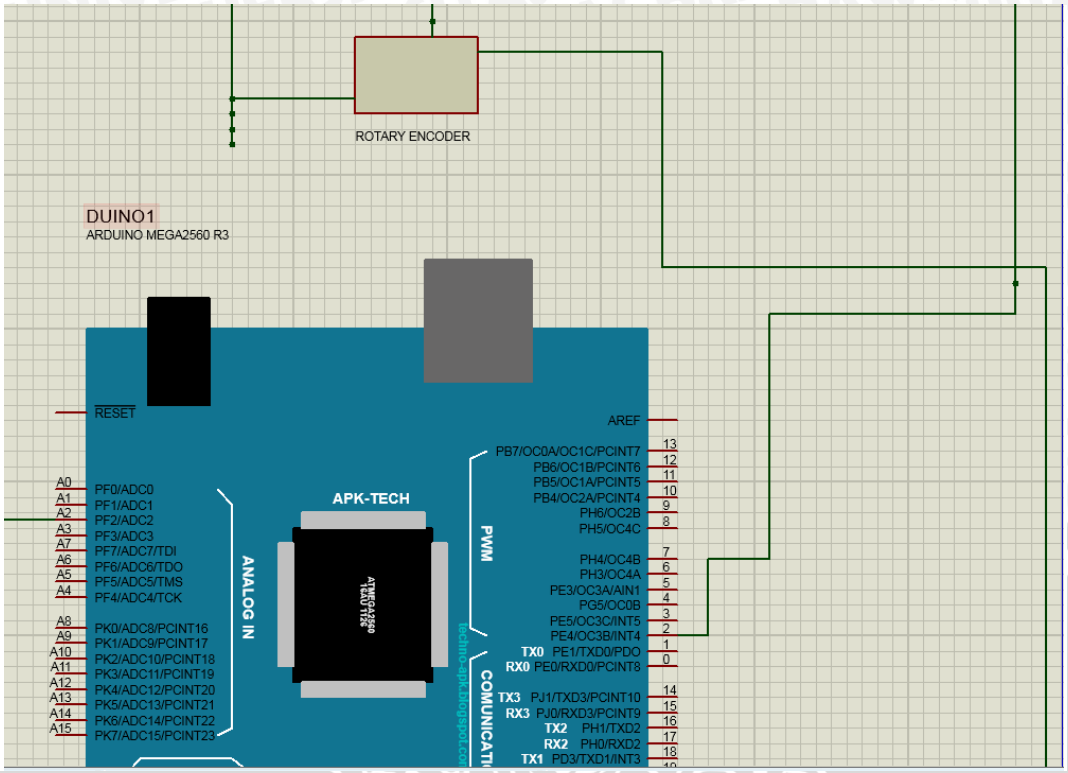


Gambar 4.10 Rangkaian sensor ultrasonik

Hasil output yang dikeluarkan oleh sensor tersebut akan diproses di dalam arduino dengan sensor *rotary encoder.*

4.3.2 Perancangan sensor *rotary encoder*

Pada perancangan sensor *rotary encoder* ini, penempatan pin input diletakkan pada pin PWM pada arduino. Kemudian, untuk daya diletakkan pada pin 5v dan untuk ground diletakkan di pin gnd. Berikut gambar perancangan sensor ultrasonik dapat dilihat pada gambarl 4.11.

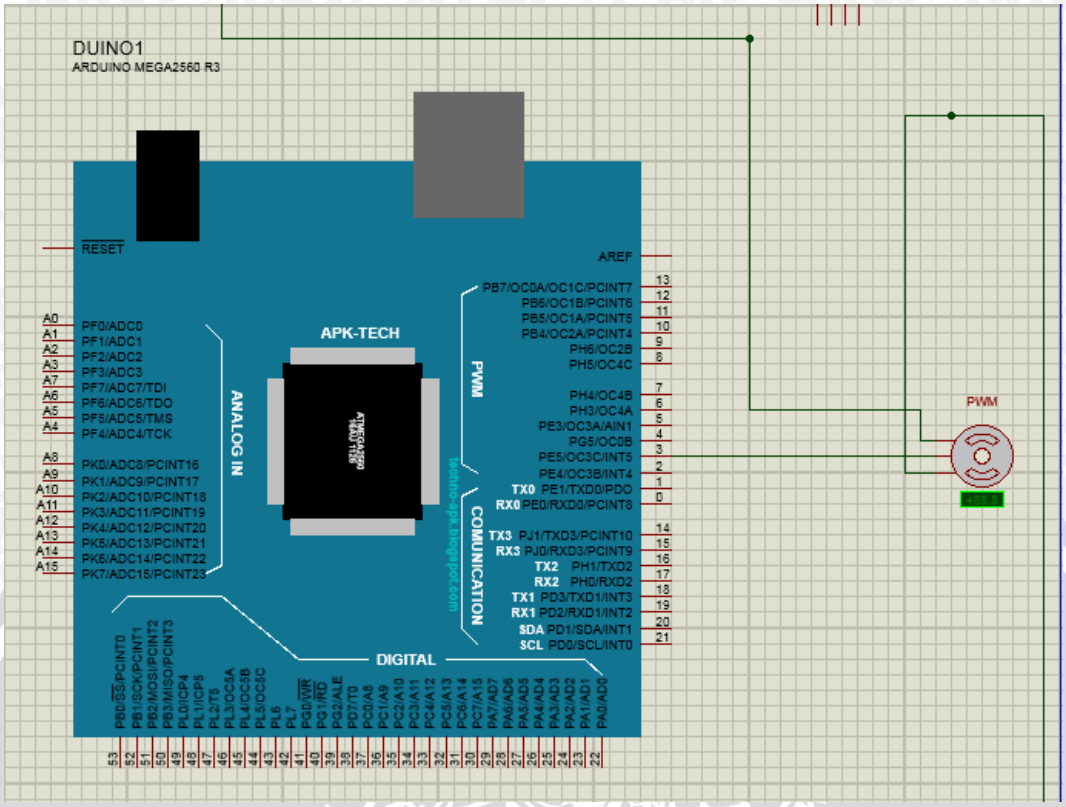


Gambar 4.11 Rangkaian sensor *rotary encoder*

Hasil output yang dikeluarkan oleh sensor tersebut akan diproses di dalam arduino dengan sensor *rotary encoder*.

4.3.3 Perancangan PWM

Pada perancangan PWM ini penempatan pin input diletakkan pada pin PWM pada arduino, kemudian untuk daya diletakkan pada pin 5v dan untuk *ground* diletakkan di pin gnd. *Output* dari fungsi *fuzzy* digunakan *input* sebagai PWM. Berikut gambar perancangan sensor ultrasonik dapat dilihat pada gambar 4.12.

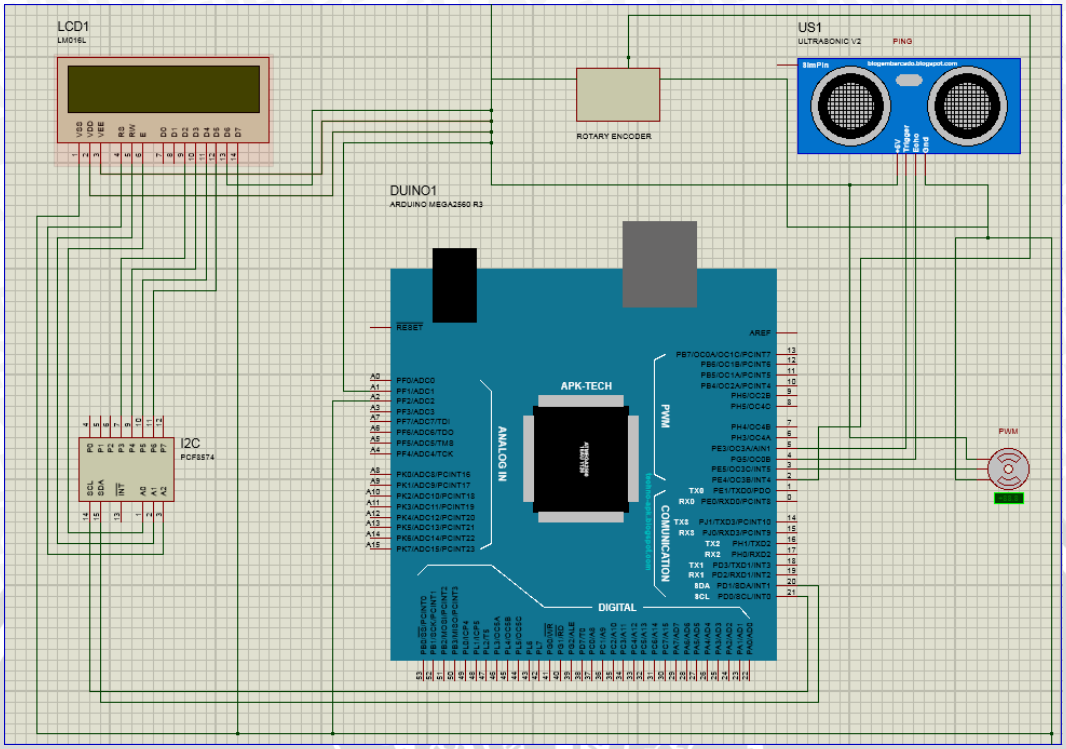


Gambar 4.12 Rangkaian PWM

Hasil *output* yang dikeluarkan PWM digunakan utuk mengatur kecepatan prototype. Variasi perubahan nilai PWM sebanyak 256 variasi mulai dari 0 – 255 perubahan nilai yang mewakili *duty cycle* 0% - 100% dari keluaran PWM tersebut. Outputnya PWM adalah nilai 80 yang mewakili kurang sedikit yang di ambil *range* tengah *duty cycle* sekitar 40% - 60%, nilai 5 bit yang mewakili nilai kurang banyak yang di ambil nilai bawah *duty cycle* 5% serta, nilai 255 yang mewakili dari tambah yang di ambil nilai atas *duty cycle* sekitar 100%.

4.3.4 Perancangan Keseluruhan

Pada perancangan keseluruhan ini, akan dibuat gambaran sistem secara keseluruhan dengan menggunakan aplikasi proteus, sehingga dapat mempermudah untuk mengimplementasikan kedalam alat yang sesungguhnya. Gambar perancangan keseluruhan dapat dilihat pada gambar 4.13.

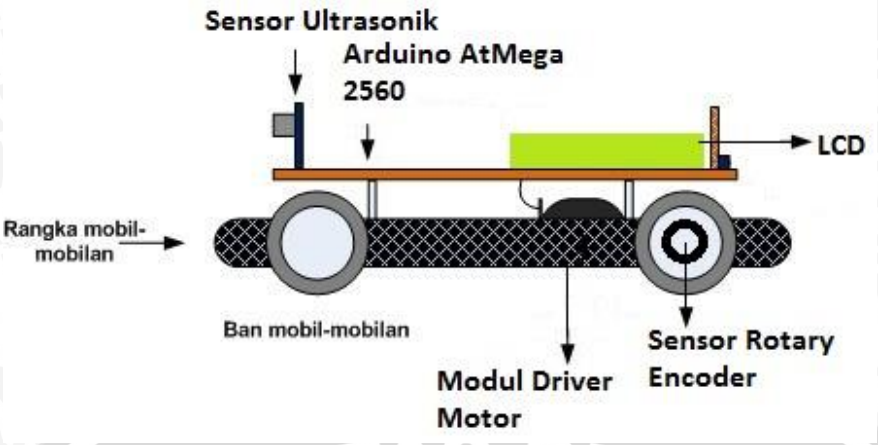


Gambar 4.13 Perencangan rangkaian sistem keseluruhan

Pada gambar 4.13 dapat dilihat bahwa sistem memiliki dua buah sensor yaitu sensor ultrasonik dan sensor *rotary encoder* , serta modul *driver motor* yang pasang ke mikrokontroller dan satu buah IC yang digunakan untuk komunikasi serial dari LCD ke mikrokontroller. Terdapat satu buah LCD yang digunakan untuk menampilkan data dari sensor dan juga penanda yang akan diberikan kepada pengguna.

4.3.5 Perancangan *prototype*

Pada perancangan *prototype* ini yaitu melakukan perancangan untuk meletakkan sensor pada *prototype* sehingga sensor bisa bekerja dengan baik. Perancangan *prototype* dapat dilihat pada gambar 4.14.



Gambar 4.14 Perancangan *prototype*

.

4.4 Implementasi

Pada implementasi akan mengimplementasikan perancangan hardware yang mencangkup tentang sensor uiltrasonik, sensor *rotary encoder* yang akan digunakan serta perancangan logika *fuzzy* pada Arduino AtMega 2560.

4.4.1 Implementasi sensor ultrasonik

Pada implementasi sensor ultrasonik yaitu dengan menghubungkan pin *trigger* ultrasonik serta *echo* ke pin PWM pada mikrokontroller. Program untuk membaca sensor ultrasonik tersebut adalah:

```
void readUltrasonicSensor()
{
    digitalWrite(ultrasonicTrigPin,
HIGH);//trigger ultrasonic
    delayMicroseconds(10);
    digitalWrite(ultrasonicTrigPin,
LOW);

    ultrasonicDuration =
pulseIn(ultrasonicEchoPin,
HIGH);//get the ultrasonic pulse
    ultrasonicDistance =
(ultrasonicDuration/2)/distance_calib
ration;//ultrasonic calibration

    lcd.setCursor(0,0);
    lcd.print("Jarak:");
    lcd.print(ultrasonicDistance);
    lcd.print("   ");
}
```

Pada sensor ultrasonik, yang dilakukan adalah mengatur 10 *microsecond* yang fungsinya untuk mengaktifkan triger lalu memancarkan gelombang ultrasonik. Secara umum, alat ini akan menembakkan gelombang ultrasonik menuju suatu area atau suatu target. Setelah gelombang menyentuh permukaan target, maka target akan memantulkan kembali gelombang tersebut. Gelombang pantulan dari target akan ditangkap oleh echo pada sensor, kemudian sensor menghitung selisih antara waktu pengiriman gelombang dan waktu gelombang pantul diterima.

Pemasangan sensor ultrasonik dapat dilihat pada gambar 4.15.

Gambar 4.15 Pemasangan sensor ultrasonik

4.4.2 Implementasi sensor *rotary encoder*

Pada implementasi sensor *rotary encoder* yaitu dengan menghubungkan pin *output* dari sensor *rotary encoder* ke pin pwm pada mikrokontroller. Program untuk membaca sensor ultrasonik tersebut adalah:

```
void readTachoSensor()
{
    rpm = (1000/(millis() -
timeold)*(revolutions))*rpm_calibrati
on;
    timeold = millis();
    revolutions = 0;

    lcd.setCursor(0,1);
    lcd.print("Rpm:");
    lcd.print(rpm);
    lcd.print("    ");
}
```

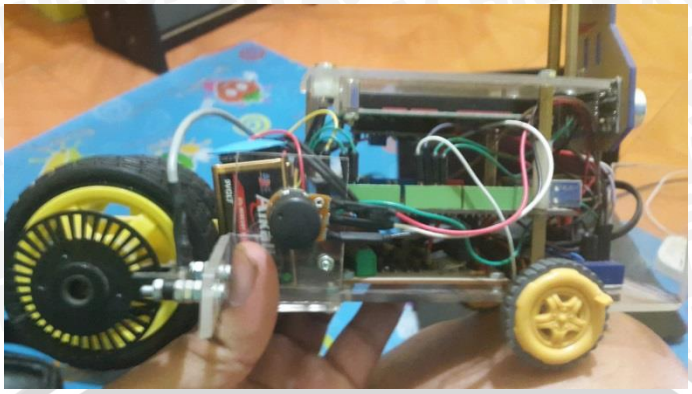
Pada sensor *rotary encoder*, yang dilakukan adalah membaca dari tiap titik apabila lubang maka dibaca 0 dan apabila tidak kena lubang maka dibaca 1. Dalam sebuah sensor terdapat dua komponen yaitu photo-transistor dan piringan lubang pada piringan tersebut terdapat 36 lubang yang terbentuk satu putaran. Semakin cepat putaran maka semakin cepat pergeseran pembacaan piringan sehingga nilainya selalu meningkat.

Pemasangan sensor *rotary encoder* dapat dilihat pada gambar 4.16.

Gambar 4.16 Pemasangan sensor *rotary encoder*

4.4.3 Implementasi sistem keseluruhan

Implementasi rangkaian sistem merupakan rangkaian keseluruhan dari *hardware* yang digunakan menjadi satu sistem yang utuh. Implementasi rangkaian keseluruhan sistem dapat dilihat pada Gambar 4.17.



Gambar 4.17 Implementasi sistem keseluruhan

4.4.3.1 Implementasi *fuzzy*

Implementasi *fuzzy* dilakukan dengan cara membuat program Arduino yang telah dirancang sebelumnya. Program *fuzzy* terdiri dari tiga proses yaitu fuzzifikasi, inferensi *fuzzy*, dan defuzzifikasi. Program *fuzzy* ditunjukan sebagai berikut:

1. Program fuzzifikasi untuk mencari nilai keanggotaan variabel *input*.

a. Fungsi Keanggotaan jarak

Berdasarkan fuzzifikasi yang telah disimulasikan pada gambar 4.5. Berikut potongan *source code* keanggotaan jarak tersebut :

```
float *keanggotaan (float m)
{
  if (m>40 && m<80){
    hk[0]=(80-m)/(80-40); //keanggotaan turun
range (c-d)
    hk[1]=(m-40)/(80-40); //keanggotaan turun
range (a-b)
    hk[2]=0;
  } // Perpotongan jarak dekat dan sedang
  else if(m>100 && m<140){
    hk[0]=0;
    hk[1]=(m-100)/(140-100);
    hk[2]=(140-m)/(140-100);
  }// Perpotongan jarak sedang dan jauh
  else if (m<40){
    hk[0] =1;
    hk[1] =0;
    hk[2] =0;
  } // Himpunan jarak dekat
  else if (m>=80 && m<100){
    hk[0] =0;
    hk[1] =1;
    hk[2] =0;
  } // Himpunan jarak sedang
  else if (m>=140 && m<200){
    hk[0] =0;
    hk[1] =0;
    hk[2] =1;
  } // Himpunan jarak jauh
}
```

Program di atas digunakan untuk mencari nilai keanggotaan jarak. Keanggotaan jarak terdiri dari tiga kondisi, yaitu kondisi dekat dan kondisi sedang, serta kondisi jauh. Nilai sensor jarak disimpan ke dalam variabel “m” yang kemudian digunakan untuk menentukan kondisinya.

b. Fungsi Keanggotaan kecepatan

Berdasarkan fuzzifikasi yang telah disimulasilakan pada gambar 4.6. Berikut potongan *source code* keanggotaan kecepatan tersebut :

```
float *keanggotaan2 (float m)
{
  if (m>60 && m<100){
    hk2[0]=(100-m)/(100-60); //keanggotaan turun
range (c-d)
    hk2[1]=(m-60)/(100-60); //keanggotaan turun
range (a-b)
    hk2[2]=0;
  }// Perpotongan kecepatan lambat dan sedang
  else if(m>140 && m<180){
    hk2[0]=0;
    hk2[1]=(m-140)/(180-140);
    hk2[2]=(180-m)/(180-140);
  }// Perpotongan kecepatan sedang dan cepat
  else if (m<60){
    hk2[0] =1;
    hk2[1] =0;
    hk2[2] =0;
  } // Himpunan kecepatan lambat
  else if (m>=100 && m<140){
    hk2[0] =0;
    hk2[1] =1;
    hk2[2] =0;
  } // Himpunan kecepatan sedang
  else if (m>180){
    hk2[0] =0;
    hk2[1] =0;
    hk2[2] =1;
  } // Himpunan kecepatan cepat
}
```

Program di atas digunakan untuk mencari nilai keanggotaan kecepatan. Keanggotaan kecepatan terdiri dari tiga kondisi yaitu kondisi lambat, kondisi

kecepatan sedang, dan kondisi cepat. Nilai sensor *rotary encoder* disimpan kedalam variabel "m" yang kemudian digunakan untuk menentukan kondisinya.

2. Menentukan inferensi dari variabel *input*

Berdasarkan inferensi yang telah disimulasilakan pada gambar 4.7. Berikut potongan *source code* inferensi tersebut :

```
float rule1(float n1,float n2,float n3,float
x1,float x2,float x3){
pred1 = min(n1,x1);
z1=1;
pred2 = min(n1,x2);
z2=2;
pred3 = min(n1,x3);
z3=2;
pred4 = min(n2,x1);
z4=3;
pred5 = min(n2,x2);
z5=1;
pred6 = min(n2,x3);
z6=1;
pred7 = min(n3,x1);
z7=3;
pred8 = min(n3,x2);
z8=3;
pred9 = min(n3,x3);
z9=3;
}
```

Program di atas digunakan untuk menentukan nilai terkecil dari kedua sensor, sesuai operator yang digunakan pada *fuzzy rule* yaitu operator AND. Pada program terdapat variabel "n" yang berisi nilai jarak, variabel "x" yang berisi nilai kecepatan, variabel "z" yang berisi nilai *output*, variabel "pred1" sampai "pred9" yang merupakan *fuzzy rule*.

3. Defuzzifikasi

Berdasarkan defuzifikasi yang telah disimulasilakan pada gambar 4.8. Berikut potongan *source code* untuk menentukan defuzifikasi tersebut :

```
float hasil()
{
  float Za1 =((pred1*z1) + (pred2*z2) + (pred3*z3) +
(pred4*z4) + (pred5*z5) + (pred6*z6) + (pred7*z7) +
(pred8*z8) + (pred9*z9)) /
  ((pred1) + (pred2) + (pred3) + (pred4) + (pred5) +
(pred6) + (pred7) + (pred8) + (pred9));

  if (Za1 >1 && Za1 <2){
  pwm_out=pwm_dikit;
  kondisi= "mandek minimal";
  }
  else if (Za1 >=2 && Za1 <3){
  pwm_out=pwm_stop;
  kondisi= "mandek banyak";
  }
  else if (Za1 >=3 && Za1 <4){
  pwm_out=pwm_banyak;
  kondisi= "gass maximal";
  }
  Serial.print(Za1);
  Serial.print(kondisi);

}
```

Program di atas berisi proses defuzzifikasi, yaitu penentuan hasil berdasarkan *fuzzy rule* sesuai perhitungan. Pada program hasil perhitungan disimpan dalam variabel “hasil”. Nilai hasil diperoleh dengan memproses nilai variabel “pred” dan “z”.

Nilai hasil yang menentukan *output* pengurangan kecepatan untuk kurang sedikit, kurang banyak, tambah, lalu output disesuaikan dengan nilai PMW yang terhubung dengan *driver motor* dan ditampilkan ke LCD nilainya.

BAB 5 PENGUJIAN DAN ANALISIS

Bab ini membahas mengenai tahapan pengujian dan analisis dari sistem yang sudah dibuat. Proses pengujian terdapat tiga tahap pengujian, yaitu pengujian metode *fuzzy*, pengujian sistem dengan implementasi metode fuzzy dan pengujian mode *follow*.

Pengujian metode *fuzzy* digunakan untuk mengetahui apakah perhitungan *fuzzy* sudah berjalan dengan baik dan benar dalam memproses data *input*. Pengujian sistem dengan implementasi metode fuzzy digunakan untuk mengetahui apakah *prototype* berjalan dengan baik dan benar. Lalu pengujian mode *follow* digunakan untuk mengetahui apakah *prototype* dapat menggontrol kecepatan. Setelah itu menganalisis hasil pengujian yang telah dilakukan pada, pengujian metode *fuzzy*, sistem dengan implementasi metode fuzzy, dan pengujian mode *follow*.

5.1 Pengujian metode *fuzzy* dan analisis

Pengujian metode *fuzzy* dilakukan untuk mengetahui apakah metode *fuzzy* yang digunakan dapat memberikan keluaran yang benar antara *rule* pada percobaan. Metode *fuzzy* yang digunakan adalah *fuzzy* sugeno. Operator yang digunakan adalah operator AND, *predikat* merupakan hasil operasi dengan menggunakan operator AND yang diperoleh dengan mengambil nilai keanggotaan terkecil antar elemen pada himpunan-himpunan yang bersangkutan.

Dalam menentukan output atau hasil pada *fuzzy* sugeno jika nilai yang dimasukan berada pada nilai atas (1) kurva maka penentuan hasil dilakukan dengan melakukan pencocokan dengan *fuzzy rule*.

Pada pengujian diberikan nilai *input* perpotongan kurva, contoh sebagai berikut:

- Nilai sensor ultrasonik 49 cm
- Nilai sensor kecepatan 234 rpm

Penyelesaiaan:
Mencari nilai keanggotaan menggunakan rumus 5.1:

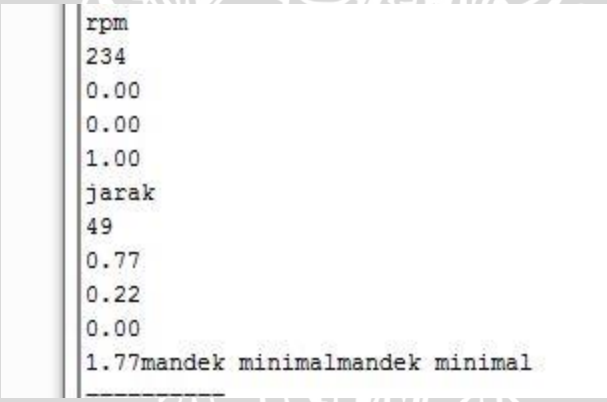
$$(x)= \begin{cases} 0 : x \leq a \; atau \; x \geq c \\ \frac{(x-a)}{(b-a)} : \; a < x \; \leq b \\ \frac{(c-x)}{(c-b)} : \; b < x < c \end{cases} \qquad (5.1)$$

Mencari nilai keanggotaan variabel jarak:
μ jarak dekat [70]: (80 - 49) / (80 - 40) = 0.775
μ jarak sedang [70]: (49 - 40) / (80 - 40) = 0.225
μ jarak jauh [70]: 0

Mencari nilai keanggotaan variabel kecepatan:
μ kecepatan *lambat* [234]: 0
μ kecepatan *sedang* [234]: 0
μ kecepatan cepat [234]: 1

Setelah diperoleh nilai keanggotan dari setiap variabel input, maka dilakukan proses inferensi *fuzzy*, sebagai berikut:
[R3] IF (jarak dekat) AND (kecepatan cepat) THEN output = 2
*pred*3 = min(0.775, 1) = 0.775
Z3 = 2
[R6] IF (jarak sedang) AND (kecepatan cepat) THEN output = 1
*pred*6 = min(0.225, 1) = 0.225
Z6 = 1
Langkah berikutnya adalah defuzzifikasi menggunakan rumus berikut:
Z =((*pred*3*Z3) + (*pred*6*Z6))/(0.775 + 0.225)
Z = ((0.775*2) + (0.225 * 1)) / 1
Z = 1.775

Maka statusnya adalah kurang sedikit. Ketika program *fuzzy* dijalankan menggunakan *input* yang sama dengan nilai pada perhitungan manual maka hasilnya ditunjukkan pada Gambar 5.1.



Gambar 5.1 Hasil pengujian metode *fuzzy*

Pada Gambar 5.1 dijelaskan bahwa hasil program *fuzzy* memiliki nilai yang sama dengan perhitungan manual yaitu 1.775. Pada nilai keanggotaan masing-masing *input* juga bernilai sama yaitu himpunan jarak μ dekat adalah 0.775 dan μ sedang adalah 0.225, himpunan kecepatan μ rendah adalah 0, μ sedang adalah 0, dan μ cepat adalah 1.

Analisis hasil pengujian yang dilakukan dengan metode *fuzzy* menunjukan bahwa program *fuzzy* telah dapat memberikan keluaran dengan akurasi 100% sesuai dengan perhitungan manual metode *fuzzy*. Pengujian menggunakan Arduino IDE menujukan bahwa program *fuzzy* dapat menampilkan nilai masing-masing sensor dan nilai keanggotaan setiap sensor beserta hasil perhitungan.

5.2 Pengujian sistem dengan implementasi metode *fuzzy* dan analisis

Pengujian sistem dengan implementasi metode *fuzzy* dilakukan untuk mengetahui apakah *prototype* yang diimplementasikan dengan metode fuzzy dapat memberikan keluaran yang benar antara *rule* pada percobaan dengan keluaran yang diperoleh dari implementasi menggunakan *hardware*.

Pengujian dilakukan sebanyak 25 kali dengan 5 kali pengujian untuk jarak yang sama memberikan halangan di depan mobil *prototype*. Berikut adalah hasil percobaan yang pertama dapat dilihat pada tabel 5.1.

Tabel 5.1 Hasil pengujian 1

Pengujian Ke-	Jarak 0-100 cm
1.	0 cm
2.	4 cm
3.	5 cm
4.	6 cm
5.	4 cm

Analisi pengujian 1 pada tabel 5.1 dilakukan dengan *prototype* dijalankan sejauh 200 cm, lalu diberi halangan pada jarak 100 cm. Nilai pada tabel diatas adalah nilai antara halangan dengan mobil prototype yang telah berhenti setelah dijalankan dari jarak 0 cm. Perbedaan hasil nilai dipengaruhi dari sensor serta lamanya *fuzzy* melakukan proses, sehingga berbeda antara nilai jarak satu dengan nilai jarak yang lainnya.

Selanjutnya adalah hasil percobaan yang kedua dapat dilihat pada tabel 5.2.

Tabel 5.2 Hasil pengujian 2

Pengujian Ke-	Jarak0-120 cm
1.	4 cm
2.	4 cm
3.	5 cm
4.	3 cm
5.	5 cm

Analisis pengujian 2 pada tabel 5.2 dilakukan dengan *prototype* dijalankan sejauh 200 cm, lalu diberi halangan pada jarak 120 cm. Nilai pada tabel diatas adalah nilai antara halangan dengan mobil prototype yang telah berhenti setelah dijalankan dari jarak 0 cm. Perbedaan hasil nilai dipengaruhi dari sensor serta lamanya *fuzzy* melakukan proses sehingga berbeda antara nilai jarak satu dengan nilai jarak yang lainnya.

Selanjutnya adalah hasil percobaan yang ketiga dapat dilihat pada tabel 5.3.

Tabel 5.3 Hasil pengujian 3

Pengujian Ke-	Jarak 0-140 cm
1.	Error
2.	5 cm
3.	3 cm
4.	4 cm
5.	4 cm

Analisis pengujian 3 pada tabel 5.3 dilakukan dengan *prototype* dijalankan sejauh 200 cm, lalu diberi halangan pada jarak 140 cm. Nilai pada tabel diatas adalah nilai antara halangan dengan mobil prototype yang telah berhenti setelah dijalankan dari jarak 0 cm. Perbedaan hasil nilai dipengaruhi dari sensor serta lamanya *fuzzy* melakukan proses sehingga berbeda antara nilai jarak satu dengan nilai jarak yang lainnya. Pada pengujian terdapat *error* dikarenakan pembacaan sensor yang kurang tepat atau akurasi yang kurang tepat sehingga membuat nilai *fuzzy* menjadi salah.

Selanjutnya adalah hasil percobaan yang keempat dapat dilihat pada tabel 5.4.

Tabel 5.4 Hasil pengujian 4

Pengujian Ke-	Jarak 0-160 cm
1.	7cm
2.	5 cm
3.	Error
4.	4 cm
5.	5 cm

Hasil pengujian 4 pada tabel 5.4 dilakukan dengan *prototype* dijalankan sejauh 200 cm, lalu diberi halangan pada jarak 160 cm. Nilai pada tabel diatas adalah nilai antara halangan dengan mobil prototype yang telah berhenti setelah dijalankan dari jarak 0 cm. Perbedaan hasil nilai dipengaruhi dari sensor serta lamanya *fuzzy* melakukan proses sehingga berbeda antara nilai jarak satu dengan nilai jarak yang lainnya. Pada pengujian terdapat *error* dikarenakan pembacaan sensor yang kurang tepat atau akurasi yang kurang tepat sehingga membuat nilai *fuzzy* menjadi salah.

Dan yang terakhir adalah hasil percobaan yang kelima dapat dilihat pada tabel 5.5.

Tabel 5.5 Hasil pengujian 5

Pengujian Ke-	Jarak 0-180 cm
1.	6 cm
2.	4 cm
3.	3 cm
4.	Error
5.	5 cm

Hasil pengujian 5 pada tabel 5.5 dilakukan dengan *prototype* dijalankan sejauh 200 cm, lalu diberi halangan pada jarak 180 cm. Nilai pada tabel diatas adalah nilai antara halangan dengan mobil prototype yang telah berhenti setelah dijalankan dari jarak 0 cm. Perbedaan hasil nilai dipengaruhi dari sensor serta lamanya *fuzzy* melakukan proses sehingga berbeda antara nilai jarak satu dengan nilai jarak yang lainnya. Pada pengujian terdapat *error* dikarenakan pembacaan sensor yang kurang tepat atau akurasi yang kurang tepat sehingga membuat nilai *fuzzy* menjadi salah.

Nilai yang tertera pada tabel adalah jarak berhentinya prototype dengan halangan. Peneliti menggunakan perbandingan prototype dengan jarak sesungguhnya yaitu 1:10. Dari hasil pengujian metode fuzzy yang diimplementasikan pada mikrokontroller dihasilkan 12 % *error* dari 25 kali percobaan dengan perhitungan jarak aman kecepatan prototype.

5.3 Pengujian mode *follow* dan analisis

Pengujian mode follow dilakukan dengan meletakan *prototype* tanpa sistem yang digerakkan dengan kecepatan yang sama dengan mobil *prototype* yang di Implementasikan sistem. Tujuan dari penelitian ini adalah untuk mengetahui apakah sistem dapat menjaga kecepatan antara *prototype* tanpa sistem dengan *prototype* yang di Implementasikan sistem. Lalu dilihat jarak antara mobil *prototype* tanpa sistem dan mobil *prototype* yang dilakukan 5 kali hasilnya dapat dilihat pada tabel 5.6.

Tabel 5.6 Hasil pengujian mode *follow*

Pengujian mode follow	Jarak
1	62 cm
2	59 cm
3	61 cm
4	58 cm
5	55 cm

Dapat dilihat dari tabel di atas bawah sistem dapat menjaga jarak antara *prototype* tanpa sistem dengan *prototype* yang diimplementasikan sistem. Terdapat perbedaan nilai dikarenakan untuk mendapat mode follow harus melawati dua fase yaitu, "tambah" dimana kecepatan bertambah, lalu memasuki fase "kurang sedikit" untuk mencapai mode follow, yang membuat nilai *input* sensor berubah-ubah sehingga jaraknya berbeda.

BAB 6 KESIMPULAN DAN SARAN

Bab ini berisi kesimpulan dari hasil perancangan, implementasi, dan analisis pengujian yang telah dilakukan, beserta saran yang dapat meningkatkan dan mengembangkan sistem yang telah ada.

6.1 Kesimpulan

Kesimpulan yang dapat diambil setelah melakukan penelitian ini adalah sebagai berikut:

1. *Prototype* dengan dua sensor sebagai input. Input pertama yaitu sensor ultrasonik sebagai pengukur jarak. Input kedua yaitu *sensor rotary encoder* sebagai pengukur kecepatan.Kedua data *input* diproses menggunakan fungsi *fuzzy* yang ada di dalam arduino. *Output*nya adalah hasil nilai *fuzzy* yang digunakan untuk mengatur kecepatan PWM motor di dalam *prototype*.
2. Hasil pengujian memiliki tingkat keberhasilan sebesar 88% yang diperoleh dari 22 kali pengujian berhasil berhenti dalam 25 kali pengujian. Nilai tersebut menunjukan bahwa *prototype* ini berhasil menjaga jarak aman serta menggontrol kecepatan.

6.2 Saran

1. Untuk meningkatkan akurasi dapat menggunakan sensor yang memiliki sensitifitas tinggi dan *response time* yang cepat.
2. Untuk pengembangan lebih lanjut dari *prototype* bisa menambahkan fitur pengereman yang lebih akurat.

DAFTAR PUSTAKA

Arduino AtMega 2560, 2015. Tersedia di <http://arduino.cc/en/main/arduinoBoardMega2560> [Diakses 20 September 2015]

Cleo. ”5 Penyebab Kecelakaan Mobil”.http://www.cleo.co.id/life.career/travel/5.penyebab.kecelakaan.mobil/001/012/178 (Diakses pada 15 Juni 2015)

Gunawan, Niko K dan Abdul Rouf. Oktober 2013. ”Purwarupa Sistem Kendali Kecepatan Mobil Berdasarkan Jarak dengan Sistem Inferensi Fuzzy Tsukamoto ”. Program Studi Elektronika dan Instrumentasi dan Jurusan Ilmu Komputer dan Elektronika. Universitas GajahMada. Yogyakarta. 2013. [Diakses 11 November 2015]

Heriawan, Rusman., 2011. Statistik Transportasi 2010. Tersedia di < http://www.bps.go.id/> (Diakses pada 20 Oktober 2015)

Passino K.Y., 2008. *Fuzzy Control.* California: Addison-Wesley Longman Inc.

Prawiroredjo, Kiki dan Nyssa Asteria. Februari 2008. ”Detektor Jarak dengan Sensor Ultrasonik Berbasis Mikrokontroler”. Jurusan Teknik Elektro. Universitas Trisakti. Jakarta. 2008 [Diakses 18 September 2015]

Republika. 07 November 2014. ”Survei Kecelakaan Lalu Lintas di Seluruh Dunia: Orang-Orang yang Mati dalam Diam”. http://www.republika.co.id/berita/koran/halaman-1/14/11/07/nenhso57-survei-kecelakaan-lalu-lintas-di-seluruh-dunia-orangorang-yang-mati-dalam-diam (Diakses pada 15 Juni 2015)

Sensor rotary encoder Datasheet. <http://www.automationdirect.com/encoders> (Diakses 28 September 2015)

Solikhin, F., 2011. *Aplikasi Logika Fuzzy Dalam Optimasi Produksi Barang Menggunakan Metode Mamdani dan Metode Sugeno.* Yogyakarta: Universitas Negeri Yogyakarta.

Sensor ultrasonik Datasheet. <http://www.electroschematics.com/8902/hc-sr04-datasheet/> [Diakses 24 September 2015]

Susilo, F., 2006. *Himpunan Logika Kabur serta Aplikasinya.* Yogyakarta: Graha Ilmu.

Tempo. 06 Januari 2011. “Enam Penyebab Utama Kecelakaan di Jalan Raya”. http://otomotif.tempo.co/read/news/2011/01/06/122304141/Enam-Penyebab-Utama-Kecelakaan-di-Jalan-Raya (Diakses pada 16 Juni 2015)

Tribunnews. 26 Agustus 2011. “Tips Mudik:Berapa Jarak Aman Anda dengan Kendaraan di Depan?”. http://www.tribunnews.com/otomotif/2011/08/26/tips-mudik-berapa-jarak-aman-anda-dengan-kendaraan-di-depan (Diakses pada 16 Juni 2015)

LAMPIRAN 1 PROGRAM *FUZZY*

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>

const int tachoPin = 2;//pin tacho sensor
const int ultrasonicTrigPin = 4;//pin Trigger
ultrasonic
const int ultrasonicEchoPin = 5;//pin Echo ultrasonic

const int EN1 = 3;//pin pwm
const int LED = 22;//pin LED

unsigned int revolutions;
unsigned int
rpm,rpm_calibration=2,tachoState,fl_tacho=0;
unsigned char ramp_up,fl_stop=1;

unsigned int
ultrasonicDistance,distance_calibration=24;
unsigned int ultrasonicDuration;

unsigned long timeold;
unsigned int
pwm_out=5,pwm_banyak=255,pwm_dikit=80,pwm_stop=5;
unsigned int distance_min=5,distance_max=200;
unsigned char a=0,b=0,fl_led_blink=0;



int z1,z2,z3,z4,z5,z6,z7,z8,z9;
float
pred1,pred2,pred3,pred4,pred5,pred6,pred7,pred8,pred9;
float hk[3];
float hk2[3];
float jarak[3],kecepatan[3];
char* kondisi;

LiquidCrystal_I2C lcd(0x27,16,2);// set the LCD address
to 0x27 for a 16 chars and 2 line display
void setup()
{
  Serial.begin(9600);
  lcd.init();// initialize the lcd
  lcd.backlight();
  lcd.clear();
  pinMode(tachoPin, INPUT_PULLUP);

  pinMode(ultrasonicTrigPin, OUTPUT);
  pinMode(ultrasonicEchoPin, INPUT);
```

```
  pinMode(LED, OUTPUT);
  pinMode(EN1, OUTPUT);

  digitalWrite(LED, LOW);
  digitalWrite(EN1, LOW);

  revolutions = 0;
  timeold = 0;

  tachoState = digitalRead(tachoPin);
  if (tachoState == HIGH && fl_tacho==0)
  {revolutions++; fl_tacho=1;}
  else if(tachoState == LOW && fl_tacho==1)
  {fl_tacho=0;}

  analogWrite(EN1,pwm_out);//firing the PWM into EN1

        if(fl_led_blink==1)
        {
        a++;
        if(a>10){a=0;b++;}
        if(b<128){digitalWrite(LED,HIGH);}
        else{digitalWrite(LED,LOW);}
        }
        else {digitalWrite(LED,LOW);}
}
cli();//stop interrupts

  //set timer2 interrupt at 8kHz
  TCCR2A = 0;// set entire TCCR2A register to 0
  TCCR2B = 0;// same for TCCR2B
  TCNT2  = 0;//initialize counter value to 0
  // set compare match register for 8khz increments
  OCR2A = 249;// = (16*10^6) / (8000*8) - 1 (must be
<256)
  // turn on CTC mode
  TCCR2A |= (1 << WGM21);
  // Set CS21 bit for 8 prescaler
  TCCR2B |= (1 << CS21);
  // enable timer compare interrupt
  TIMSK2 |= (1 << OCIE2A);

  sei();//allow interrupts
}
ISR(TIMER2_COMPA_vect)
{
//timer2 interrupt 8kHz
//generates pulse wave of frequency 8kHz/2 = 4kHz (takes
two cycles for full wave- toggle high then toggle low)
tachoState = digitalRead(tachoPin);
        if (tachoState == HIGH && fl_tacho==0)
{revolutions++; fl_tacho=1;}
```

```
else if(tachoState == LOW && fl_tacho==1){fl_tacho=0;}

        analogWrite(EN1,pwm_out);//firing the PWM into
EN1

        if(fl_led_blink==1)
        {
        a++;
        if(a>10){a=0;b++;}
        if(b<128){digitalWrite(LED,HIGH);}
        else{digitalWrite(LED,LOW);}
        }
        else {digitalWrite(LED,LOW);}
}
//////////main

void loop()
{
      readTachoSensor();
      readUltrasonicSensor();
       Serial.println("rpm");
      Serial.println(rpm);
      keanggotaan2(rpm);
      kecepatan[0]=hk2[0];
      kecepatan[1]=hk2[1];
      kecepatan[2]=hk2[2];
      Serial.println(kecepatan[0]);
      Serial.println(kecepatan[1]);
      Serial.println(kecepatan[2]);

      Serial.println("jarak");
      Serial.println(ultrasonicDistance);
      keanggotaan(ultrasonicDistance);
      jarak[0]=hk[0];
      jarak[1]=hk[1];
      jarak[2]=hk[2];

      Serial.println(jarak[0]);
      Serial.println(jarak[1]);
      Serial.println(jarak[2]);
      rule1(jarak[0],jarak[1],jarak[2],kecepatan[0],
       kecepatan[1],kecepatan[2]);
      hasil();

      lcd.setCursor(8,1);
      lcd.print("PWM:");
      Serial.println(kondisi);
      lcd.print("   ");
      Serial.println("==========");
      Serial.println(pwm_out);
      Serial.println("==========");

}
```

```
//////////sub

float *keanggotaan (float m)
{
  if (m>40 && m<80){
    hk[0]=(80-m)/(80-40); //
keanggotaan turun range (c-d)
    hk[1]=(m-40)/(80-40); //
keanggotaan turun range (a-b)
    hk[2]=0;
  }
  else if(m>100 && m<140){
    hk[0]=0;
    hk[1]=(m-100)/(140-100);
    hk[2]=(140-m)/(140-100);
  }
  else if (m<40){
    hk[0] =1;
    hk[1] =0;
    hk[2] =0;
  }
  else if (m>=80 && m<100){
    hk[0] =0;
    hk[1] =1;
    hk[2] =0;
  }
  else if (m>=140 && m<200){
    hk[0] =0;
    hk[1] =0;
    hk[2] =1;
  }
}
```

```
float *keanggotaan2 (float m)
{
  if (m>60 && m<100){
    hk2[0]=(100-m)/(100-60); //keanggotaan turun range
(c-d)
    hk2[1]=(m-60)/(100-60); //keanggotaan turun range

(a-b)
    hk2[2]=0;
  }
  else if(m>140 && m<180){
    hk2[0]=0;
    hk2[1]=(m-140)/(180-140);
    hk2[2]=(180-m)/(180-140);
  }
  else if (m<60){
    hk2[0] =1;
    hk2[1] =0;
    hk2[2] =0;
  }
  else if (m>=100 && m<140){
    hk2[0] =0;
    hk2[1] =1;
    hk2[2] =0;
  }
  else if (m>180){
    hk2[0] =0;
    hk2[1] =0;
    hk2[2] =1;
  }
}
```

```
float rule1(float n1,float n2,float n3,float x1,float
x2,float x3){
pred1 = min(n1,x1);
z1=2;
pred2 = min(n1,x2);
z2=2;
pred3 = min(n1,x3);
z3=2;
pred4 = min(n2,x1);
z4=3;
pred5 = min(n2,x2);
z5=1;
pred6 = min(n2,x3);
z6=1;
pred7 = min(n3,x1);
z7=3;
pred8 = min(n3,x2);
z8=3;
pred9 = min(n3,x3);
z9=3;
}
float hasil()
{


  float Za1 =((pred1*z1) + (pred2*z2) + (pred3*z3) +
(pred4*z4) + (pred5*z5) + (pred6*z6) + (pred7*z7) +
(pred8*z8) + (pred9*z9)) /
  ((pred1) + (pred2) + (pred3) + (pred4) + (pred5) +
(pred6) + (pred7) + (pred8) + (pred9));

  if (Za1 >1 && Za1 <2){
  pwm_out=pwm_dikit;
  kondisi= "mandek minimal";
  }
  else if (Za1 >=2 && Za1 <3){
  pwm_out=pwm_stop;
  kondisi= "mandek banyak";
  }
  else if (Za1 >=3 && Za1 <4){
  pwm_out=pwm_banyak;
  kondisi= "gass maximal";
  }
  Serial.print(Za1);
  Serial.print(kondisi);

}
```

```
void readUltrasonicSensor()
{
    digitalWrite(ultrasonicTrigPin,
 HIGH);//trigger ultrasonic
    delayMicroseconds(10);
    digitalWrite(ultrasonicTrigPin, LOW);

    ultrasonicDuration = pulseIn(ultrasonicEchoPin,
HIGH);//get the ultrasonic pulse
    ultrasonicDistance =
(ultrasonicDuration/2)/distance_calibration;//ultrasonic
calibration

    lcd.setCursor(0,0);
    lcd.print("Jarak:");
    lcd.print(ultrasonicDistance);
    lcd.print("   ");
}
void readTachoSensor()
{
    rpm = (1000/(millis() -
timeold)*(revolutions))*rpm_calibration;
    timeold = millis();
    revolutions = 0;

    lcd.setCursor(0,1);
    lcd.print("Rpm:");
    lcd.print(rpm);
    //Serial.println(rpm);
    //Serial.println("========");
    lcd.print("   ");
}
```