

RANCANG BANGUN PENGENDALI PINTU GERBANG TOL DENGAN RFID MENGGUNAKAN LOGIKA *FUZZY*

SKRIPSI

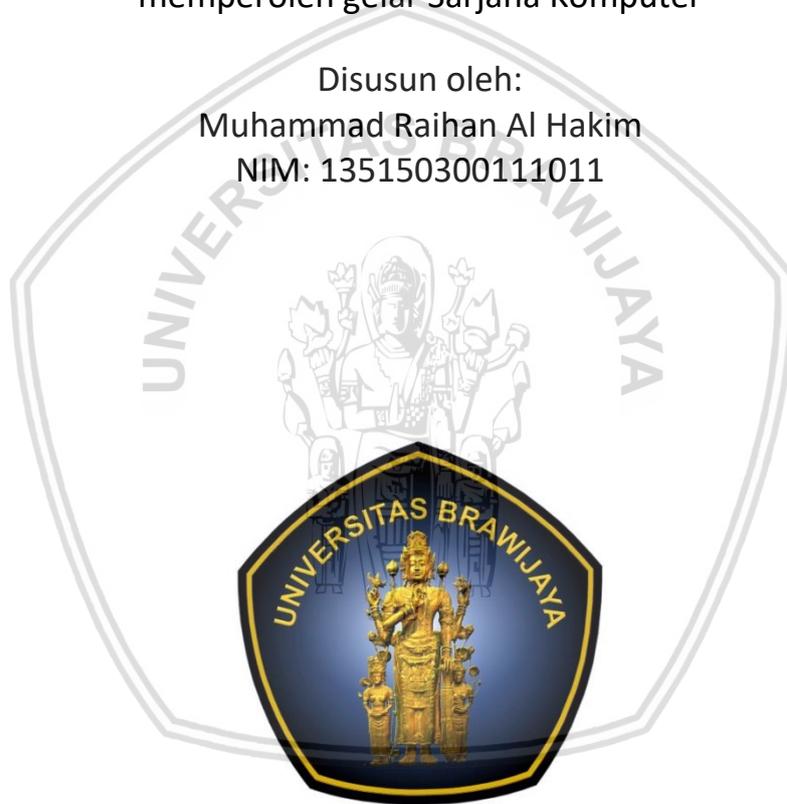
KEMINATAN TEKNIK KOMPUTER

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun oleh:

Muhammad Raihan Al Hakim

NIM: 135150300111011



PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG
2018



PENGESAHAN

RANCANG BANGUN PENGENDALI PINTU GERBANG TOL DENGAN RFID
MENGUNAKAN LOGIKA FUZZY

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun Oleh :
Muhammad Raihan Al Hakim
NIM: 135150300111011

Skripsi ini telah diuji dan dinyatakan lulus pada
1 Agustus 2018
Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I

Dosen Pembimbing II



Rizal Maulana, S.T., M.T., M.Sc
NIK: 201607 891009 1 001



Hurriyatul Fitriyah, S.T., M.Sc
NIP: 198510 01201504 2 003

Mengetahui

Ketua Jurusan Teknik Informatika



Iri Astoto Kurniawan, S.T., M.T., Ph.D
NIP: 19710518 200312 1 001

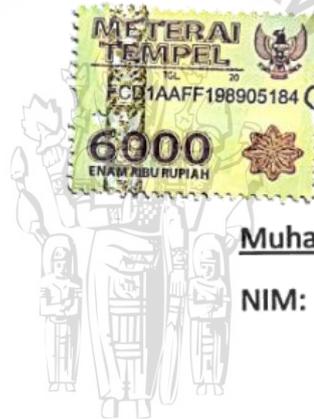


PERNYATAAN ORISINALITAS

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, 1 Januari 2018



Muhammad Raihan Al Hakim

Muhammad Raihan Al Hakim

NIM: 135150300111011



KATA PENGANTAR

Puji syukur penulis panjatkan kehadirat Allah SWT yang telah memberikan rahmat dan karunianya, sehingga penulis dapat menyelesaikan penyusunan laporan skripsi yang berjudul “Rancang Bangun Pengendali Pintu Gerbang Tol Dengan RFID Menggunakan Logika *Fuzzy*” dapat disusun untuk memenuhi sebagian persyaratan memperoleh gelar Sarjana Komputer di Universitas Brawijaya. Shalawat serta salam semoga selalu terlimpahkan kepada Nabi Besar Nabi Muhammad SAW.

Dalam kesempatan ini, penulis menyadari bahwa tidak terlepas dari bantuan berbagai pihak. Oleh sebab itu penulis mengucapkan terimakasih kepada pihak-pihak yang telah membantu dan memberikan dukungan dalam penyusunan laporan skripsi, diantaranya:

1. Bapak Wayan Firdaus Mahmudy, S.Si, M.T, Ph.D. selaku Dekan Fakultas Ilmu Komputer Universitas Brawijaya Malang.
2. Bapak Heru Nurwarsito, Ir., M.Kom. selaku Wakil Ketua I Bidang Akademik Fakultas Ilmu Komputer Universitas Brawijaya Malang.
3. Bapak Tri Astoto Kurniawan, S.T, M.T, Ph.D. selaku Ketua Jurusan Teknik Informatika Universitas Brawijaya Malang.
4. Bapak Sabriansyah Rizqika Akbar, S.T, M.Eng. selaku Ketua Program Studi Teknik Komputer Universitas Brawijaya Malang.
5. Bapak Rizal Maulana, S.T., M.T., M.Sc selaku Dosen Pembimbing satu yang telah memberikan ilmu, saran, penjelasan dan motivasi serta membantu dalam penyusunan laporan penulis.
6. Ibu Hurriyatul Fitriyah, S.T., M.Sc selaku Dosen Pembimbing dua yang telah memberikan ilmu, saran, penjelasan dan motivasi kepada penulis.
7. Alm. Bapak Abdul Jabbar Zulkifli dan Ibu Endah Sri Rahayu selaku orang tua yang penulis cintai serta seluruh keluarga besar yang selalu memberi dukungan dan do’a agar penulis dapat menyelesaikan skripsi ini dengan lancar.
8. Seluruh civitas akademika Informatika Universitas Brawijaya dan terkhusus untuk teman-teman Teknik Komputer Angkatan 2013 yang telah banyak memberi bantuan dan dukungan selama peneliti menempuh studi di Teknik Komputer Universitas Brawijaya dan selama penyelesaian skripsi ini.
9. Pakde Budi Santosa, Bude Ida Restyani beserta keluarga, grup Kedai 999, dan seluruh teman-teman dari Teknik Komputer yang tidak dapat disebutkan namanya satu persatu yang telah memberikan dukungan dan do’a.

10. Seluruh pihak yang tidak dapat diucapkan satu persatu, peneliti mengucapkan banyak terima kasih atas segala bentuk dukungan dan doa sehingga laporan skripsi ini dapat terselesaikan

Peneliti menyadari bahwa tulisan ini masih jauh dari kata sempurna dan masih memiliki berbagai macam kekurangan. Oleh karena itu penulis mengharapkan kritik dan saran yang membangun, agar ke depannya penulis dapat menjadi lebih baik lagi. Semoga isi Laporan Skripsi ini dapat memberi manfaat bagi perkembangan ilmu pengetahuan di kemudian hari.

Malang, 1 Agustus 2018

Muhammad Raihan Al Hakim
Raihan.alhakim@gmail.com



ABSTRAK

Kemacetan yang terjadi di gerbang tol dikarenakan meningkatnya volume kendaraan yang tidak dapat ditampung oleh kapasitas jalan di gerbang tol. Sistem pembayaran di gerbang tol yang saat ini berjalan belum dapat mengatasi kemacetan yang ada melainkan hanya meminimalisir kemacetan tersebut. Berdasarkan hal ini dikembangkan sistem yang dapat membantu upaya meminimalisir kemacetan di gerbang tol menggunakan metode *Fuzzy*. Sebagai pemantau kondisi kepadatan yang terjadi di gerbang tol digunakan sensor Ultrasonik, sebagai alat pembayaran juga sebagai pembaca ukuran kendaraan digunakan RFID RC522, *output* sistem digunakan motor servo sebagai pintu gerbang tol kemudian diproses dalam mikrokontroler Arduino Mega 2560. Dua *input* tersebut diolah menjadi nilai *input* derajat keanggotaannya, kemudian melewati proses implikasi, fungsi implikasi yang digunakan adalah *min*, lalu menuju ke inferensi sistem *Fuzzy* yaitu *max*, dari tahap inferensi dilanjutkan ke proses defuzzifikasi dan akan didapatkan nilai tegas untuk motor servo dalam menentukan durasi pembukaan pintu gerbang tol. Pada pengujian, sensor Ultrasonik memiliki pengukuran yang sama dalam mengukur jarak menggunakan penggaris. RFID sebagai *interface* sistem menghasilkan akurasi pembacaan yang cepat dan memiliki jarak maksimal pembacaan sebesar 1 cm. *Output* sistem berupa motor servo memiliki selisih perbedaan dalam pengukuran derajat posisi motor servo dengan pengukuran busur derajat sebesar 2.4 derajat dan tingkat *error* sebesar 3.766%. Setelah sistem diimplementasikan, dilakukan pengujian dengan skenario *input* yang berbeda, hasil perhitungan sistem kemudian dibandingkan dengan data sampel yang didapat dari beberapa percobaan, secara keseluruhan perhitungan logika *Fuzzy* pada durasi pembukaan pintu gerbang tol memiliki persentase *error* sebesar 4,189%.

Kata kunci: *Fuzzy* Mamdani, Gerbang Tol, Kemacetan, RFID, Ultrasonik

ABSTRACT

The congestion occurring at the toll gate is due to the increasing volume of vehicles that can not be accommodated by road capacity at toll gates. The payment system at the toll road gate that runs has not been able to overcome the congestion but only minimize the congestion. Based on this case developed a system that can helps efforts to minimize congestion at the toll gate using the Fuzzy method. As a monitoring of the density conditions occurring at the toll gate used Ultrasonic sensors, as a means of payment as well as the reader of vehicle size used RFID, the system output is used servo as toll gate then processed in Arduino. The two inputs are processed into the input value of membership degree, then through the implication process, the implication function used is min, then go to the Fuzzy system inference is max, from the inference stage proceed to the defuzzification process and will get the firm value for the servo motor in determining the opening duration toll gate. In the test, Ultrasonic sensors have the same measurements in measuring distances using a ruler. RFID as a system interface produces a fast readout accuracy and has a maximum reading distance of 1 cm. The system output of servo has difference in measurement of servo position with manual measurement 2.4 degree and error rate 3,766%. After the system is implemented, tested with different input scenarios, the result of the calculation of the system is then compared with the sample data obtained from several experiments, overall Fuzzy logic calculation on the toll gate opening duration has an error percentage of 4.189%.

Keywords: Congestion, Fuzzy Mamdani, RFID, Toll Gate, Ultrasonic

DAFTAR ISI

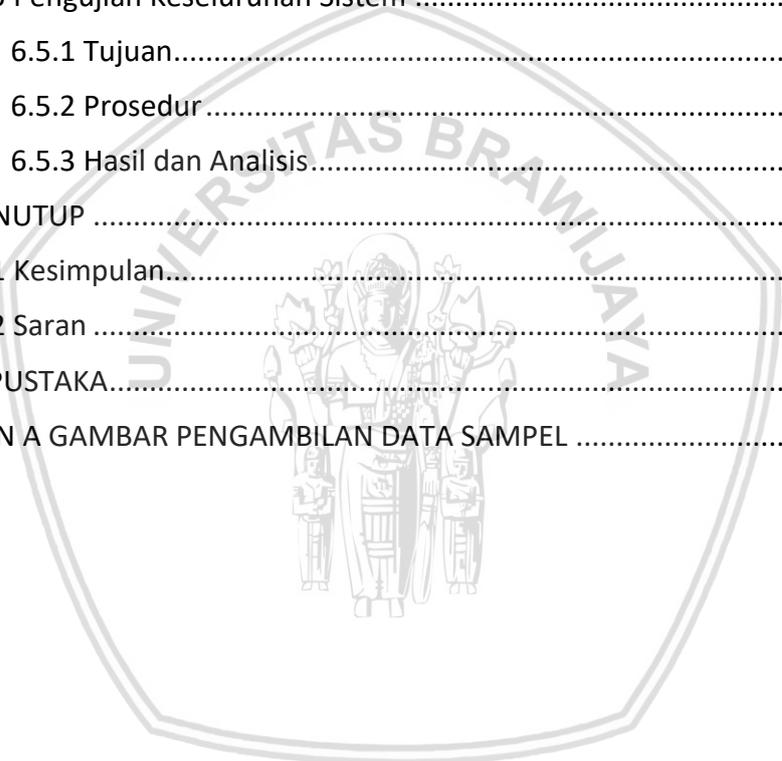
PENGESAHAN	ii
PERNYATAAN ORISINALITAS	iii
KATA PENGANTAR.....	iv
ABSTRAK.....	vi
<i>ABSTRACT</i>	vii
DAFTAR ISI.....	viii
DAFTAR TABEL.....	xi
DAFTAR GAMBAR.....	xii
BAB 1 PENDAHULUAN.....	1
1.1 Latar belakang.....	1
1.2 Rumusan masalah.....	2
1.3 Tujuan	2
1.4 Manfaat.....	2
1.5 Batasan masalah	2
1.6 Sistematika pembahasan.....	3
BAB 2 LANDASAN KEPUSTAKAAN	4
2.1 Tinjauan Pustaka	4
2.2 Landasan Teori.....	5
2.2.1 Jalan tol	5
2.2.2 Gerbang tol.....	6
2.2.3 Pengertian Logika <i>Fuzzy</i>	6
2.2.4 Arduino Mega 2560.....	11
2.2.5 Sensor Ultrasonik	13
2.2.6 <i>Radio Frequency Identification</i> (RFID).....	14
2.2.7 EEPROM Arduino Mega 2560	15
2.2.8 Motor Servo	16
2.2.9 Persentase dan Rata – Rata Kesalahan	17
BAB 3 METODOLOGI	19
3.1 Metodologi Penelitian	19
3.2 Studi Literatur	20



3.3	Rekayasa Kebutuhan.....	20
3.4	Perancangan Sistem.....	20
3.4.1	Perancangan Perangkat Keras	20
3.4.2	Perancangan Perangkat Lunak.....	21
3.5	Implementasi Sistem	21
3.5.1	Implementasi Perangkat Keras	21
3.5.2	Implementasi Perangkat Lunak.....	21
3.6	Pengujian dan Analisis	21
3.7	Kesimpulan.....	22
BAB 4	REKAYASA KEBUTUHAN.....	23
4.1	Deskripsi Umum.....	23
4.2	Analisis Kebutuhan Sistem.....	24
4.2.1	Kebutuhan Antarmuka Pengguna	24
4.2.2	Kebutuhan Fungsional.....	24
4.2.3	Kebutuhan Non Fungsional Sistem	25
BAB 5	PERANCANGAN DAN IMPLEMENTASI	27
5.1	Perancangan Sistem.....	27
5.1.1	Perancangan Perangkat Keras	27
5.1.2	Perancangan Perangkat Lunak.....	30
5.2	Implementasi Sistem	38
5.2.1	Implementasi Perangkat Keras	39
5.2.2	Implementasi Perangkat Lunak.....	41
BAB 6	PENGUJIAN DAN ANALISIS.....	56
6.1	Pengujian RFID	56
6.1.1	Tujuan.....	56
6.1.2	Prosedur	56
6.1.3	Hasil dan Analisis.....	57
6.2	Pengujian Sensor Ultrasonik.....	58
6.2.1	Tujuan.....	58
6.2.2	Prosedur	58
6.2.3	Hasil dan Analisis.....	58
6.3	Pengujian Motor Servo	59



6.3.1 Tujuan.....	59
6.3.2 Prosedur	59
6.3.3 Hasil dan Analisis.....	60
6.4 Pengujian Fungsional Metode <i>Fuzzy</i>	60
6.4.1 Tujuan.....	60
6.4.2 Prosedur	60
6.4.3 Hasil dan Analisis.....	61
6.4.4 Perhitungan Manual.....	62
6.5 Pengujian Keseluruhan Sistem	64
6.5.1 Tujuan.....	64
6.5.2 Prosedur	64
6.5.3 Hasil dan Analisis.....	64
BAB 7 PENUTUP	66
7.1 Kesimpulan.....	66
7.2 Saran	66
DAFTAR PUSTAKA.....	68
LAMPIRAN A GAMBAR PENGAMBILAN DATA SAMPEL	70



DAFTAR TABEL

Tabel 2.1 Tabel Spesifikasi Arduino	12
Tabel 2.2 Spesifikasi Sensor Ultrasonik HC-SR04.....	13
Tabel 2.3 Spesifikasi RFID RC522.....	14
Tabel 2.4 Spesifikasi Motor Servo SG-90.....	16
Tabel 4.1 Kebutuhan Perangkat Keras Sistem	25
Tabel 4.2 Kebutuhan Perangkat Lunak Sistem.....	26
Tabel 5.1 Keterangan Pin Arduino Mega 2560 dan RFID <i>reader</i>	28
Tabel 5.2 Keterangan Pin Sensor Ultrasonik dan Arduino Mega 2560.....	29
Tabel 5.3 Keterangan Pin Motor Servo dan Arduino Mega 2560	29
Tabel 5.4 Variabel Panjang Kendaraan	32
Tabel 5.5 Variabel Kondisi Kepadatan	33
Tabel 5.6 Percobaan Sistem Pada Relawan	35
Tabel 5.7 Aturan <i>Fuzzy</i>	36
Tabel 5.8 Kode Program EEPROM.....	42
Tabel 5.9 Kode Program Ultrasonik	43
Tabel 5.10 Kode Program RFID	48
Tabel 5.11 Kode Program Fungsi Keanggotaan Panjang Kendaraan	51
Tabel 5.12 Kode Program Fungsi Keanggotaan Kondisi Kepadatan	52
Tabel 5.13 Program <i>Rule Based</i> dan Inferensi	54
Tabel 5.14 Kode Program Defuzzifikasi.....	55
Tabel 6.1 Hasil Pengujian RFID.....	57
Tabel 6.2 Hasil Pengujian Jarak Maksimal Pembacaan RFID <i>reader</i>	57
Tabel 6.3 Hasil Pengujian Sensor Ultrasonik.....	58
Tabel 6.4 Hasil Pengujian Motor Servo	60
Tabel 6.5 Hasil Keluaran Logika <i>Fuzzy</i>	61
Tabel 6.6 Pengujian Keseluruhan Sistem	64
Tabel 6.7 Persentase <i>Error</i>	65



DAFTAR GAMBAR

Gambar 2.1 Himpunan <i>Fuzzy</i>	7
Gambar 2.2 Proses Defuzzifikasi	10
Gambar 2.3 Arduino Mega.....	12
Gambar 2.4 Sensor Ultrasonik	14
Gambar 2.5 Prinsip Kerja RFID	15
Gambar 2.6 <i>Tag</i> RFID dan RFID <i>reader</i>	15
Gambar 2.7 Arduino IDE	16
Gambar 2.8 Motor Servo	17
Gambar 3.1 Diagram Alir Penelitian.....	19
Gambar 3.2 Rancangan Sistem	20
Gambar 4.1 Diagram Blok Perancangan Sistem	23
Gambar 4.2 Analisis Kebutuhan Sistem	24
Gambar 5.1 Rangkaian RFID <i>reader</i> RC522 dan Arduino Mega 2560.....	28
Gambar 5.2 Rangkaian Sensor Ultrasonik dan Arduino Mega 2560.....	28
Gambar 5.3 Rangkaian Servo dan Arduino Mega 2560	29
Gambar 5.4 Skema Perancangan Perangkat Keras	30
Gambar 5.5 <i>Flowchart</i> Perancangan Logika <i>Fuzzy</i>	31
Gambar 5.6 <i>Flowchart</i> Proses Fuzzifikasi.....	32
Gambar 5.7 Himpunan <i>Fuzzy</i> Panjang Kendaraan	33
Gambar 5.8 Grafik Himpunan <i>Fuzzy</i> Kondisi Kepadatan	34
Gambar 5.9 <i>Flowchart</i> Proses Inferensi.....	36
Gambar 5.10 <i>Flowchart</i> Proses Defuzzifikasi	37
Gambar 5.11 Grafik <i>Output</i> Durasi Pembukaan Pintu.....	38
Gambar 5.12 Perangkat Keras Arduino dengan Sensor Ultrasonik	39
Gambar 5.13 Perangkat Keras Arduino dengan RFID <i>reader</i>	40
Gambar 5.14 Perangkat Keras Arduino dengan Motor Servo	40
Gambar 5.15 Implementasi Sistem Pada Maket	41
Gambar 6.1 Pohon Pengujian Sistem dan Analisis	56



BAB 1 PENDAHULUAN

1.1 Latar belakang

Pada zaman ini, semua hal dituntut untuk dapat bekerja secara otomatis. Perkembangan teknologi yang berkembang secara cepat juga menuntut manusia untuk dapat berfikir lebih jauh dengan teknologi yang sudah ada. Terjadinya kemacetan yang sering dialami merupakan hal merugikan yang dapat memberikan dampak negatif seperti ruginya waktu yang dihabiskan didalam kemacetan, pemborosan energi karena saat terjadi kemacetan energi terbuang sia – sia, dapat menimbulkan stress bagi pengguna jalan yang mengalami kemacetan dan lain – lain.

Kemacetan merupakan hal yang sering dapat ditemukan dikarenakan peningkatan volume kendaraan yang pesat tidak dapat diatasi oleh kapasitas jalan yang tersedia. Pemerintah berupaya mengatasi hal ini dengan membangun akses jalan tol atau jalan bebas hambatan yang menjadi terobosan untuk mengantisipasi atau meminimalisir terjadinya kemacetan di ruas – ruas jalan yang mengalami kemacetan. Jalan tol adalah jalan umum yang kepada pemakainya diberi kewajiban untuk membayar tol sebagai bentuk hubungan timbal balik dengan pelayanan jalan tol yang mengkedepankan kelancaran, keamanan dan kenyamanan selama berkendara di jalan tol. Pembayaran tol dilakukan di gerbang – gerbang tol yang telah disediakan saat pertama masuk tol, ruas – ruas kota dan saat terakhir keluar tol (UU No. 38/2004 tentang Jalan Pasal 44).

Namun, pembayaran tarif tol di gerbang - gerbang tol ini seringkali mengakibatkan kemacetan dikarenakan hal – hal yang sering terjadi seperti uang yang diberikan pengguna jalan tol terlalu besar sehingga untuk mengembalikan kembalian dari tarif tol membuat petugas membutuhkan waktu yang lebih lama dari yang seharusnya. Hal ini dikarenakan pengguna yang tidak menyiapkan uang pas atau karena tarif tol yang nominalnya tidak sama dengan mata uang yang diedarkan.

Mulai Oktober 2017, seluruh gerbang tol tidak lagi melayani pembayaran secara konvensional tapi sudah menggunakan *e-toll* sebagai metode pembayaran tol. "Pemberlakuan 100 persen pada Oktober 2017 itu tidak berarti 100 persen, yang sekarang itu 47 persen gerbang tol otomatis (GTO). Pemberlakuannya akan secara bertahap dari gerbang tol manual ke GTO," ujar AVP Corporate Communication PT Jasa Marga Dwimawan Heru kepada detikcom, Jumat (19/5/2017). Pemberlakuan ini untuk mengurangi kepadatan antrian di gerbang tol serta mengurangi kebiasaan penggunaan transaksi tunai dikalangan masyarakat atau disebut *cashless society* (R, 2017).

Penggunaan OBU (*On Board Unit*) sebagai pembayaran tol sudah sering kali dibicarakan. Kepala Badan Pengatur Jalan Tol (BPJT) Herry Trisaputra Zuna menuturkan, ada dua jenis teknologi OBU yang sedang diuji coba yaitu teknologi *Dedicated Short Range Communication* (DRSC) dan *Radio Frequency Identification* (RFID), Menurut Herry, sistem pembayaran melalui OBU ini memiliki beragam keuntungan. Dari sisi waktu yang hanya membutuhkan 1 – 2 detik, juga sisi praktis karena pengendara tidak perlu membuka kaca untuk melakukan transaksi pembayaran (Prabowo, 2018).

Hal ini yang melatarbelakangi penelitian ini dengan menggunakan RFID sebagai bentuk komunikasi antara kendaraan dengan otomatisasi sistem pada gerbang tol menggunakan logika

Fuzzy agar lebih meminimalisir dan mengefisiensi waktu yang digunakan untuk pembayaran tol di gerbang tol.

1.2 Rumusan masalah

Berdasarkan latar belakang yang telah dijelaskan di atas maka dapat dirumuskan menjadi rumusan masalah sebagai berikut:

1. Bagaimana membuat perancangan sistem pintu gerbang tol dengan RFID menggunakan mikrokontroller Arduino Mega?
2. Bagaimana implementasi logika *Fuzzy* ke dalam sistem rancang bangun pengendali pintu gerbang tol dengan RFID?
3. Bagaimana analisis hasil logika *Fuzzy* terhadap sistem rancang bangun pengendali pintu gerbang tol dengan RFID?

1.3 Tujuan

Berdasarkan rumusan masalah didapatkan tujuan sebagai berikut:

1. Membuat perancangan sistem pengendali pintu gerbang tol dengan RFID menggunakan mikrokontroller Arduino Mega
2. Mengimplementasikan logika *Fuzzy* ke dalam sistem rancang bangun pengendali pintu gerbang tol dengan RFID
3. Melakukan pengujian dan analisis hasil logika *Fuzzy* terhadap sistem rancang bangun pengendali pintu gerbang tol dengan RFID

1.4 Manfaat

Manfaat yang dapat diperoleh dari rancang bangun sistem pengendalian pintu gerbang tol dengan menggunakan logika *Fuzzy* adalah:

1. Membuat purwarupa sistem pengendalian pintu gerbang tol dengan RFID berbasis sistem embedded
2. Menjadi salah satu solusi dalam meminimalisir terjadinya kemacetan di jalan tol
3. Menjadi salah satu rujukan penelitian yang terkait

1.5 Batasan masalah

Agar penelitian yang dilakukan dapat terfokus pada tujuan penelitian dan tidak menyimpang, maka dilakukan pembatasan beberapa hal yaitu:

1. Jumlah pintu gerbang tol yang digunakan sejumlah 1 gerbang
2. Bagian yang dikontrol adalah pintu gerbang tol dengan menggunakan motor servo
3. Kecepatan kendaraan pada penelitian digunakan simulasi/prototipe

1.6 Sistematika pembahasan

Penjelasan singkat mengenai struktur dan isi dari masing-masing bab pada skripsi ini adalah:

BAB 1 : PENDAHULUAN

Pada bab ini akan dijelaskan mengenai latar belakang penelitian, rumusan masalah, tujuan penelitian, manfaat penelitian, batasan masalah, dan sistematika pembahasan.

BAB 2 : LANDASAN KEPUSTAKAAN

Pada bab ini akan dijelaskan mengenai tinjauan pustaka dan dasar teori

BAB 3 : METODOLOGI

Pada bab ini akan dijelaskan mengenai Studi literatur, analisis kebutuhan sistem, perancangan sistem, implementasi, analisis, dan pengambilan kesimpulan

BAB 4 : REKAYASA KEBUTUHAN

Pada bab ini akan dijelaskan mengenai kebutuhan perangkat keras dan perangkat lunak yang memiliki spesifikasi yang sesuai dengan penelitian yang akan dilakukan.

BAB 5 : PERANCANGAN DAN IMPLEMENTASI

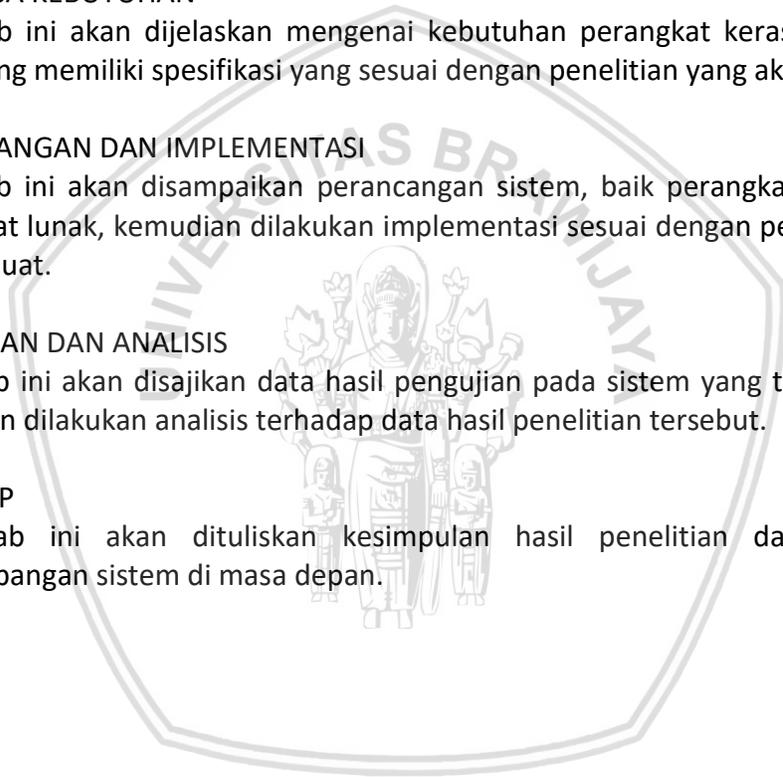
Pada bab ini akan disampaikan perancangan sistem, baik perangkat keras maupun perangkat lunak, kemudian dilakukan implementasi sesuai dengan perancangan yang telah dibuat.

BAB 6 : PENGUJIAN DAN ANALISIS

Pada bab ini akan disajikan data hasil pengujian pada sistem yang telah dibuat, dan kemudian dilakukan analisis terhadap data hasil penelitian tersebut.

BAB 7 : PENUTUP

Pada bab ini akan dituliskan kesimpulan hasil penelitian dan saran untuk pengembangan sistem di masa depan.



BAB 1 LANDASAN KEPUSTAKAAN

Pada bab 2 yaitu Landasan Kepustakaan ini berisi uraian dan pembahasan tentang tinjauan pustaka dan dasar teori yang dibutuhkan dan mendukung penelitian ini

1.1 Tinjauan Pustaka

Pada tinjauan pustaka ini akan dijelaskan tentang penelitian yang sudah pernah dilakukan yang berkaitan dengan penelitian yang telah dilakukan sebelumnya. Tinjauan pustaka yang dijadikan penulis sebagai sumber adalah penelitian yang dilakukan oleh (Sighlia, 2016) yang menerapkan RFID sebagai otomatisasi sistem dalam pembukaan pintu gerbang menggunakan *IoT*. Pada penelitian ini RFID digunakan didalam lingkungan komplek perumahan untuk membuka pintu gerbang agar kendaraan pemilik dapat masuk ataupun keluar. Jika ada kendaraan yang tidak terdaftar didalam database, maka pengguna kendaraan tersebut diharuskan menginformasikan kepada orang yang ingin ditemui didalam komplek perumahan tersebut sehingga orang yang ingin ditemui tersebut dapat masuk ke situs web dari sistem gerbang tersebut dan membukakan gerbang secara manual dengan mengklik tombol *open*.

Tinjauan pustaka selanjutnya adalah penelitian yang dilakukan oleh (Prasetya, 2017), penelitian ini bertujuan untuk mengatur dan menjaga debit air sungai agar tidak meluap dan menyebabkan banjir, pada aliran sungai digunakan pintu air. Pada penelitian ini penulis akan melakukan perancangan sistem pengendali pintu air dengan menggunakan logika *Fuzzy* dan *Simple Additive Weighting*. Logika *Fuzzy* pada penelitian ini akan digunakan untuk menentukan besaran pembukaan pintu air yang didasarkan pada data ketinggian permukaan air yang didapatkan dari hasil akuisisi sensor *water level*, dan data curah hujan, yang didapatkan dari hasil akuisisi sensor *rain module*. Sebagai *actuator* untuk menggerakkan pintu air digunakan motor servo. Data hasil akuisisi sensor akan dikirimkan ke Arduino Mega 2560, kemudian pada mikrokontroler tersebut, dilakukan proses perhitungan Logika *Fuzzy* dan metode *Simple Additive Weighting*. Secara keseluruhan perhitungan logika *Fuzzy* pada Sungai Utama dan seluruh Anak Sungai memiliki persentase *error* sebesar 0,09875%, sedangkan pada perhitungan metode *SAW* yang digunakan untuk menentukan anak sungai yang menjadi alternatif terbaik, tidak terdapat perbedaan antara perhitungan manual dan perhitungan yang dilakukan oleh sistem

Berdasarkan penelitian pertama yang telah dijelaskan, penulis bermaksud untuk mengembangkan penelitian pertama yang memanfaatkan RFID untuk diaplikasikan di gerbang tol dengan membuat sistem embedded yang disimulasikan dengan *prototype*. Pada penelitian kedua, penerapan kedua metode menghasilkan persentase *error* yang sangat kecil yaitu sebesar 0,09875% sehingga sangat memungkinkan untuk melakukan penerapan sistem yang lain dengan menggunakan metode ini. Atas kedua penelitian inilah penulis ingin membuat sistem yang saling berkaitan yaitu sistem jalan tol yang berbasis RFID dengan menggunakan metode logika *Fuzzy*. Dengan pembuatan sistem ini diharap dapat memberikan salah satu solusi meminimalisir permasalahan kemacetan yang terjadi di gerbang tol saat ini.

1.2 Landasan Teori

Dasar teori merupakan teori yang dibutuhkan dan relevan mengenai unsur pendukung baik perangkat keras, perangkat lunak maupun objek penelitian digunakan untuk membuat sistem ini.

1.2.1 Jalan tol

Jalan tol (di Indonesia disebut juga sebagai jalan bebas hambatan) adalah suatu jalan yang dikhususkan untuk kendaraan yang memiliki roda lebih dari dua (mobil, bus, truk) dan bertujuan untuk mempersingkat jarak dan waktu tempuh dari satu tempat ke tempat lain.

Untuk menikmatinya, para pengguna jalan tol harus membayar sesuai tarif yang berlaku. Penetapan tarif didasarkan pada golongan kendaraan. Bangunan atau fasilitas di mana tol dikumpulkan dapat disebut pintu gerbang tol, rumah tol, plaza tol atau di Indonesia lebih dikenal sebagai gerbang tol. Bangunan ini biasanya ditemukan di dekat pintu keluar, di awal atau akhir jembatan (misal: Jembatan Suramadu), dan ketika memasuki suatu jalan layang (*flyover*).

Di Indonesia, jalan tol sering dianggap sinonim untuk jalan bebas hambatan, meskipun hal ini sebenarnya salah. Di dunia secara keseluruhan, tidak semua jalan bebas hambatan memerlukan bayaran. Jalan bebas hambatan seperti ini dinamakan *freeway* atau *expressway* (*free* berarti "gratis", dibedakan dari jalan-jalan bebas hambatan yang memerlukan bayaran yang dinamakan *tollway* atau *tollroad* (kata *toll* berarti "biaya").

Jalan tol merupakan bagian sistem jaringan dan sebagai jalan nasional yang penggunaannya diwajibkan membayar tol (sejumlah uang tertentu yang dibayarkan untuk penggunaan jalan tol). Untuk dapat menggunakan fasilitas jalan tol ini, setiap kendaraan harus masuk melalui pintu gerbang tol. Di pintu ini setiap pengemudi harus membayar pembayaran tol sesuai dengan golongan kendaraannya. Di Indonesia terdapat lima golongan kendaraan yang mempunyai tarif pembayaran tol berbeda sesuai dengan golongan kendaraan. Jalan tol menjadi sarana yang sangat membantu melancarkan akses transportasi di area perkotaan yang jalan rayanya selalu padat dengan arus kendaraan.

Tidak hanya bagi masyarakat perkotaan, jalan raya juga dimanfaatkan oleh masyarakat yang melakukan perjalanan antar kota. Dengan semakin banyaknya jumlah kendaraan yang menggunakan jalan tol tiap harinya, masalah kemacetan pun kembali dijumpai di jalan ini. Kebanyakan kemacetan terjadi di bagian pintu masuk tol. Penyebabnya, tiap kendaraan harus berhenti untuk melakukan pembayaran tol. Meskipun ada beberapa pintu gerbang tol yang dibuka, namun kemacetan di pintu gerbang tol masih saja terlihat, terutama pada saat menjelang libur panjang atau hari besar. Hal tersebut tentunya akan sangat menyita waktu dan energi (BBM). (Vicky Primandani, 2012).

1.2.2 Gerbang tol

Gerbang tol atau pintu gerbang tol adalah tempat pelayanan transaksi tol bagi pemakai tol yang terdiri dari beberapa gardu dan sarana perlengkapan lainnya. Berdasarkan PP No. 15 Th 2005 Pasal 25 ayat 4 penggunaan gerbang tol diatur sebagai berikut:

- a. Bangunan gerbang tol dipergunakan untuk pelaksanaan transaksi tol

- b. Di gerbang tol, pengguna wajib menghentikan kendaraannya untuk mengambil atau menyerahkan karcis masuk atau membayar tol
- c. Dilarang menaikkan atau menurunkan penumpang, barang dan hewan di gerbang tol

Besarnya waktu pelayanan sangat dipengaruhi oleh pengumpulan tol, kemampuan peralatan tol serta keterampilan dan kesiapan petugas gerbang tol juga pengguna jalan. Berdasarkan data Departemen Pekerjaan Umum Direktorat Jenderal Bina Marga, pembayaran tarif tol di gerbang tol terbagi memiliki beberapa jenis diantaranya:

- Gerbang tol terbuka, pada sistem ini pengguna jalan tol membayar tarif tol yang tertera sesuai golongan kendaraan masing-masing. Waktu pelayanan untuk gerbang tol terbuka ini memiliki durasi 6 detik.
- Gerbang tol tertutup, pada sistem ini pengguna jalan tol mengambil tiket (kartu) di gerbang tol pada saat akan memasuki jalan tol, lalu membayar pada saat akan keluar dari jalan tol tersebut. Pengguna yang kehilangan tiket biasanya dikenakan denda untuk menebus tiket yang dihilangkan. Waktu pelayanan untuk gerbang tol tertutup ini memiliki durasi 4 detik saat akan mengambil kartu di gerbang tol saat pertama masuk dan 10 detik di gerbang tol keluar untuk mengembalikan tiket serta membayar tarif tol.

1.2.3 Pengertian Logika Fuzzy

Konsep logika *Fuzzy* diperkenalkan oleh Prof. Lotfi Astor Zadeh pada 1962. Logika *Fuzzy* adalah metodologi sistem control pemecahan masalah, yang cocok diimplementasikan pada sistem, embedded sistem, jaringan PC dan sistem kontrol. Metodologi ini dapat diterapkan pada perangkat keras, perangkat lunak atau kombinasi keduanya. Dalam logika klasik dinyatakan bahwa segala sesuatu bersifat biner, yang artinya adalah hanya mempunyai dua kemungkinan. "Ya atau Tidak", "Benar atau Salah", "Baik atau Buruk", dan lain-lain. Akan tetapi, dalam logika *Fuzzy* memungkinkan nilai keanggotaan berada diantara 0 atau 1. Akan tetapi, dalam logika *Fuzzy* memungkinkan nilai keanggotaan berada di antara 0 dan 1. Artinya, bisa saja suatu keadaan mempunyai dua nilai "Ya dan Tidak" secara bersamaan, namun besar nilainya tergantung pada bobot keanggotaan yang dimilikinya (T.Sutojo, 2011).

1.2.3.1 Himpunan Fuzzy

Pada himpunan tegas (*crisp*), nilai keanggotaan suatu item x dalam suatu himpunan A , yang sering ditulis dengan (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.

Contoh:

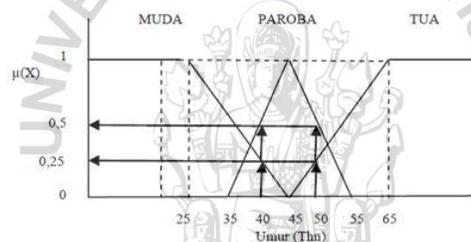
Misalkan variabel umur dibagi menjadi tiga kategori, yaitu:

1. MUDA umur < 35 tahun
2. PAROBAYA $35 \leq \text{umur} \leq 55$ tahun
3. TUA Umur > 55 tahun

Dari kategori di atas dapat dijelaskan bahwa:

1. Apabila seseorang berusia 34 tahun, maka ia dikatakan MUDA ($\mu(34)=1$)
2. Apabila seseorang berusia 35 tahun, maka ia dikatakan TIDAK MUDA ($\mu(35)=0$)
3. Apabila seseorang berusia 35 tahun kurang 1 hari, maka ia dikatakan TIDAK MUDA ($\mu(35-1\text{hari})=1$)
4. Apabila seseorang berusia 35 tahun, maka ia dikatakan PAROBAYA ($\mu(35)=1$)
5. Apabila seseorang berusia 34 tahun, maka ia dikatakan TIDAK PAROBAYA ($\mu(34)=0$)
6. Apabila seseorang berusia 55 tahun, maka ia dikatakan PAROBAYA ($\mu(55)=1$)
7. Apabila seseorang berusia 35 tahun kurang 1 hari, maka ia dikatakan TIDAK PAROBAYA ($\mu(35-1\text{hari})=0$)

Dari sini dapat dikatakan bahwa pemakaian himpunan crisp untuk menyatakan umur sangat tidak adil. Adanya perubahan kecil saja pada suatu nilai mengakibatkan perbedaan kategori yang cukup signifikan. Himpunan *Fuzzy* digunakan untuk mengantisipasi hal tersebut. Seseorang dapat masuk dalam dua himpunan berbeda, MUDA dan PAROBAYA, PAROBAYA dan TUA dan sebagainya. Seberapa besar eksistensinya dalam himpunan tersebut dapat dilihat pada nilai keanggotaannya. Gambar berikut menunjukkan himpunan *Fuzzy* untuk variabel umur.



Gambar 1.1 Himpunan *Fuzzy*

(Sumber: docplayer.info, 2016)

Pada Gambar 2.1 di atas, dapat dilihat bahwa:

1. Seseorang yang berumur 40 tahun, termasuk dalam himpunan MUDA dengan ($\mu(40)=0,25$); namun juga termasuk dalam himpunan Parobaya ($\mu(40)=0,5$).
2. Seseorang yang berumur 50 tahun, termasuk dalam himpunan TUA dengan ($\mu(50)=0,25$); namun dia juga termasuk juga dalam himpunan PAROBAYA ($\mu(50)=0,5$).

Terkadang kemiripan antara keanggotaan *Fuzzy* dengan probabilitas menimbulkan kerancuan. Keduanya memiliki nilai interval $[0,1]$ namun interpretasi nilainya sangat berbeda antara kedua kasus tersebut. Keanggotaan *Fuzzy* memberikan suatu ukuran terhadap pendapat dan keputusan, sedangkan probabilitas mengindikasikan proporsi terhadap keseringan suatu hasil bernilai benar dalam jangka panjang. Misalnya, jika nilai keanggotaan bernilai suatu himpunan *Fuzzy* USIA adalah 0,9; maka tidak perlu dipermasalahkan berapa seringnya nilai itu diulang secara individual untuk mengharapkan suatu hasil yang hampir muda. Dilain pihak, nilai probabilitas 0,9 usia berarti 10% dari himpunan tersebut diharapkan tidak muda (Kusumadewi, 2010).

Himpunan *Fuzzy* memiliki 2 atribut (Kusumadewi, 2010) yaitu:

1. Linguistik, yaitu penamaan suatu grup yang mewakili suatu keadaan atau kondisi tertentu dengan menggunakan Bahasa alami, seperti: MUDA, PAROBAYA, TUA.
2. Numeris, yaitu suatu nilai (angka) yang menunjukkan ukuran dari suatu variabel seperti: 40,25,50, dan sebagainya.

Ada beberapa hal yang perlu diketahui dalam memahami sistem *Fuzzy* (Kusumadewi, 2010), yaitu:

1. Variabel *Fuzzy* yaitu variabel yang hendak dibahas dalam suatu sistem *Fuzzy*. Contohnya: umur, temperature, permintaan, dsb.
2. Himpunan *Fuzzy* yaitu grup yang mewakili suatu kondisi atau keadaan tertentu dalam suatu variabel *Fuzzy*. Contohnya : variabel temperatur, terbagi menjadi 5 himpunan *Fuzzy*, yaitu: Dingin, Sejuk, Normal, Hangat, dan Panas.
3. Semesta Pembicaraan yaitu keseluruhan nilai yang diperoleh untuk dioperasikan dalam suatu variabel *Fuzzy*, semesta pembicaraan merupakan himpunan bilangan *real* yang senantiasa naik (bertambah) secara monoton dari kiri kekanan. Nilai semesta pembicaraan dapat berupa bilangan positif maupun negatif. Adakalanya nilai semesta pembicaraan ini tidak dibatasi atasnya. Contoh semesta pembicaraan untuk variabel umur: $[0 +\infty]$.

Domain Himpunan *Fuzzy* yaitu keseluruhan nilai yang diijinkan 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*. DINGIN= $[0,20]$, SEJUK= $[15,20]$, NORMAL= $[20,30]$, HANGAT= $[25,35]$ dan PANAS = $[30,40]$.

Logika *Fuzzy* menyerupai pembuatan keputusan pada manusia dengan kemampuannya untuk bekerja dari data yang ditafsirkan dan mencari solusi yang tepat. Keadaan - keadaan konvensional seperti ya atau tidak, benar atau salah, hitam atau putih, dan sebagainya. Beberapa alasan orang memakai logika *Fuzzy* yaitu fleksibel, toleransi adanya data-data yang tidak tepat, mampu memodelkan fungsi-sungsi non-linear yang sangat kompleks, dapat membangun dan mengaplikasikan pengalaman-pengalaman para pakar secara langsung tanpa harus melalui proses pelatihan.

Alasan penggunaan logika *Fuzzy* pada penelitian ini adalah karena logika *Fuzzy* merupakan penggambaran cara sederhana dalam pengambilan kesimpulan pasti dari informasi yang ambigu, samar - samar, atau tidak tepat. Dalam penelitian ini variabel panjang kendaraan yang diwakili oleh golongan kendaraan memiliki nilai yang tidak jelas dimana panjang kendaraan pada setiap kendaraan yang melewati gerbang tol itu bervariasi. Dalam variabel kondisi kepadatan juga ditemui hal yang serupa dimana kondisi kepadatan yang ada di gerbang tol memiliki nilai – nilai yang ambigu. Juga pembukaan pintu gerbang tol (palang) dimana pembukaannya terpacu pada durasi pelayanan kendaraan yang melewati pintu gerbang tol memiliki nilai yang samar – samar.

1.2.3.2 Metode Fuzzy Mamdani

Metode Mamdani sering juga dikenal sebagai metode *Max-Min*. Metode ini diperkenalkan oleh Ebrahim Mamdani pada tahun 1975 (Kusumadewi, 2010). Untuk mendapatkan *output* diperlukan 4 tahapan, antara lain:

1. Pembentukan himpunan *Fuzzy*

Pada Metode Mamdani, baik variabel *input* maupun *output* dibagi menjadi satu atau lebih himpunan *Fuzzy*.

2. Aplikasi fungsi implikasi

Pada Metode Mamdani, fungsi implikasi yang digunakan adalah Min.

3. Komposisi Aturan

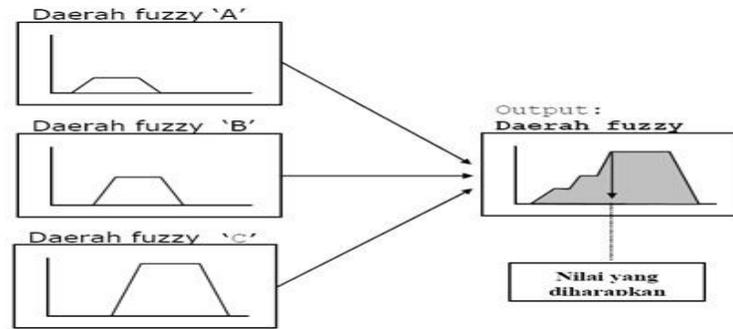
Tidak seperti penalaran monoton, apabila sistem terdiri dari beberapa aturan, maka inferensi diperoleh dari kumpulan dan kolerasi antar aturan. Ada 3 metode yang digunakan dalam melakukan *inferensi* sistem *Fuzzy*, yaitu *max*, *addictive* dan *probabilistic OR (probor)*.

- a. Metode *Maximum*

Pada metode ini solusi himpunan *Fuzzy* diperoleh dengan cara mengambil nilai maksimum aturan, kemudian menggunakannya untuk memodifikasi daerah *Fuzzy*, dan mengaplikasikannya ke *output* dengan menggunakan operator OR (union). Jika semua proposisi telah dievaluasi maka, *output* akan berisi suatu himpunan *Fuzzy* yang merefleksikan kontribusi dari tiap-tiap proporsi

4. Penegasan (*Defuzzify*)

Input dari proses deFuzzifikasi adalah suatu himpunan yang diperoleh dari komposisi aturan-aturan *Fuzzy*, sedangkan *output* yang dihasilkan merupakan suatu bilangan pada domain himpunan *Fuzzy* tersebut. Sehingga jika diberikan suatu himpunan *Fuzzy* dalam *range* tertentu sebagai *output* seperti terlihat pada gambar berikut:



Gambar 1.2 Proses Defuzzifikasi

(Sumber: Kusumadewi, 2010)

Ada beberapa metode defuzzifikasi pada komposisi aturan Mamdani antara lain:

a. Metode *Centroid (Composite Moment)*

Pada metode *centroid* solusi *crisp* diperoleh dengan cara mengambil titik pusat daerah *Fuzzy*.

b. Metode *Bisektor*

Pada metode *bisector* solusi *crisp* dengan cara mengambil nilai pada domain yang memiliki nilai keanggotaan setengah dari jumlah total keanggotaan pada daerah *Fuzzy*.

c. Metode *Mean of Maximum (MOM)*

Pada metode *mean of maximum* solusi *crisp* diperoleh dengan cara mengambil nilai rata-rata domain yang memiliki nilai keanggotaan maksimum.

d. Metode *Largest of Maksimum (LOM)*

Pada metode *largest of maksimum* solusi *crisp* diperoleh dengan cara mengambil nilai terbesar dari domain yang memiliki nilai keanggotaan maksimum.

e. Metode *Smallest of Maximum (SOM)*

Pada metode *smallest of maximum* solusi *crisp* diperoleh dengan cara mengambil nilai terkecil dari domain yang memiliki nilai keanggotaan maksimum.

Ada beberapa cara metode penegasan yang biasa dipakai pada komposisi aturan Mamdani, dalam skripsi ini metode yang akan dipakai adalah metode centroid, pada metode ini solusi tegas diperoleh dengan cara mengambil titik pusat daerah *Fuzzy*. Secara umum dituliskan persamaan **2.1** dan persamaan **2.2** berikut:

$$z^* = \frac{\text{Momen}}{\text{Luas}} = \frac{\int z Z \mu(Z) dZ}{\int z \mu(Z) dZ} \quad \text{untuk variabel kontinu} \quad (2.1)$$

$$z^* = \frac{\sum_{j=1}^n Z_j \mu(Z_j)}{\sum_{i=1}^n \mu(Z_j)} \quad \text{untuk variabel diskrit} \quad (2.2)$$

Keterangan:

Z* = nilai centroid (titik pusat daerah Fuzzy)

Z = fungsi untuk daerah hasil fuzzifikasi

μ = daerah hasil fuzzifikasi/daerah komposisi

Alasan digunakannya metode Fuzzy Mamdani menurut jurnal yang ditulis oleh Laras Purwati Ayuningtias tentang “Analisa Perbandingan Logic Fuzzy Metode Tsukamoto, Sugeno, dan Mamdani (Studi Kasus: Prediksi Jumlah Pendaftar Mahasiswa Baru Fakultas Sains dan Teknologi Universitas Islam Negeri Sunan Gunung Djati Bandung), bahwa hasil penelitian yang telah dilakukan menunjukkan metode Fuzzy Mamdani mempunyai tingkat error yang jauh lebih kecil sebesar 19.76 % dibandingkan dengan metode Tsukamoto sebesar 39.03 % dan sugeno sebesar 86.41 % pada prediksi jumlah pendaftar mahasiswa baru (Ayuningtias, 2017).

1.2.4 Arduino Mega 2560

Arduino merupakan salah satu vendor penyedia mikrontroller yang banyak digunakan untuk melakukan purwarupa suatu sistem. Arduino memiliki banyak tipe, mulai dari yang paling sederhana, dengan kemampuan dan spesifikasi sederhana, hingga yang menggunakan prosesor dan fungsi yang kompleks. Sebuah mikrokontroler Arduino terdiri dari IC utama yang berfungsi sebagai pengendali utama dari jalannya sistem mikrokontroler tersebut. Pada mikrontroller ini juga dilengkapi dengan pin *input / output* yang jumlahnya berbeda bergantung pada tipe Arduino tersebut. Pin tersebut memiliki fungsi khusus, seperti PWM, pin digital, dan pin analog.



Sumber : **Gambar 1.3** Arduino Mega (Arduino, 2017)

Tabel 1.1 Tabel Spesifikasi Arduino

Chip Mikrokontroler	Pin Arduino Mega 2560
Tegangan Operasi	5V
Rekomendasi Tegangan <i>Input</i>	7V – 12V
Tegangan <i>Input</i> Maksimal	6V – 20V
Pin I/O Digital	54 pin, 15 diantaranya PWM
Pin <i>Input</i> Analog	16 pin

Arus DC Pin I/O	20mA
Arus DC Pin 3.3 V	50mA
Flash Memori	256 KB, 8KB untuk bootloader
SRAM	8KB
EEPROM	4KB
Clockspeed	16Mhz
Dimensi	101.52 mm x 53.3 mm
Berat	37gr

Namun dari semua tipe Arduino, Arduino Mega 2560 merupakan tipe yang paling banyak digunakan. Arduino mega banyak digunakan karena bersifat open source, memiliki banyak pin digital maupun analog yang dapat dimanfaatkan untuk membuat sistem yang lebih kompleks, serta memiliki harga yang terjangkau. Arduino mega mampu mengakomodir kebutuhan komputasi dan menangani lebih dari satu *input* dari sensor maupun perangkat lain serta mampu menghasilkan *output* dalam bentuk aktuator maupun dalam bentuk lain.

Arduino juga menyediakan aplikasi untuk memasukkan kode program yaitu Arduino IDE. Setelah kode program selesai dibuat, maka *board* Arduino mega dapat dihubungkan ke komputer yang memiliki aplikasi tersebut untuk melakukan *upload* program yang telah dibuat ke *board* Arduino Mega. *Upload* program ini dapat dilakukan berulang, sehingga apabila terjadi kesalahan, program dapat di *upload* kembali. Untuk melakukan *upload* program membutuhkan bantuan kabel yang menghubungkan Arduino dengan komputer. Arduino dapat dihubungkan dengan sensor maupun aktuator yang mendukung (Geddes, 2016).

1.2.5 Sensor Ultrasonik

Sensor ultrasonik merupakan sensor yang diaplikasikan untuk melakukan pengukuran jarak tertentu dari sensor ke obyek terdekat yang berada tegak lurus di depan sensor tersebut. Sensor ultrasonik terdiri dari rangkaian *piezoelektrik* yang berfungsi untuk mengubah energi listrik mejadi energi mekanik, *transmitter* yang berfungsi memancarkan gelombang dengan frekuensi tertentu, receiver yang terdiri dari tranduser ultrasonik menggunakan bahan *piezoelektrik* yang berfungsi sebagai penerima gelombang pantulan yang dipancarkan oleh transmiteer.

Tabel 1.2 Spesifikasi Sensor Ultrasonik HC-SR04

Tegangan Operasi	5V
Konsumsi Arus	15Ma
Frekuensi Operasi	40KHz
Minimum Pendeteksi Jarak	0.02 m (2 cm)
Maksimum Pendeteksi Jarak	4 m

Sudut Pantul Gelombang Pengukuran	15 derajat
Dimensi	45 x 20 x 15 mm

Sumber : (Elec freaks, 2005)

Cara kerja sensor ultrasonik adalah dengan memanfaatkan gelombang suara yang dipantulkan. Frekuensi sinyal yang biasa digunakan untuk mengukur jarak adalah di atas 20 kHz, pada umumnya sensor ultrasonik menggunakan frekuensi 40 kHz. Sinyal tersebut akan merambat dengan kecepatan 340 m/s, dan akan terpantul saat sinyal menumbuk suatu benda (Tjahyadi, 2017).



Gambar 1.4 Sensor Ultrasonik

Sumber : (Santoso, 2015)

1.2.6 Radio Frequency Identification (RFID)

RFID RC522 (Radio Frequency Identification) merupakan suatu teknologi yang memanfaatkan frekuensi radio sebagai peng-identifikasi-an terhadap suatu objek. RFID mempunyai 2 bagian komponen utama yang tak dapat dipisahkan, yaitu *Tag* RFID dan RFID Reader.

RFID (*Radio Frequency Identification*) merupakan sebuah alat yang bekerja dengan memanfaatkan gelombang frekuensi transmisi radio untuk menyampaikan data yang berisi nomor unik. Teknologi ini memiliki kelebihan karena cara penyampaian datanya yang tanpa menggunakan kontak tertentu dan mampu bekerja di setiap kondisi lingkungan. Untuk menggunakan komunikasi dengan RFID diperlukan adanya *RFID reader*, *tag* RFID, dan komputer

- *Tag* RFID : alat yang berfungsi sebagai penyimpan data untuk identifikasi
- *RFID reader* : alat yang berinteraksi dengan *tag* RFID dan berfungsi untuk membaca data dari *tag* RFID dengan menggunakan gelombang frekuensi.
- Komputer : komputer yang berisikan perangkat lunak yang mampu menunjukkan data hasil pembacaan *RFID reader* terhadap *tag* RFID.

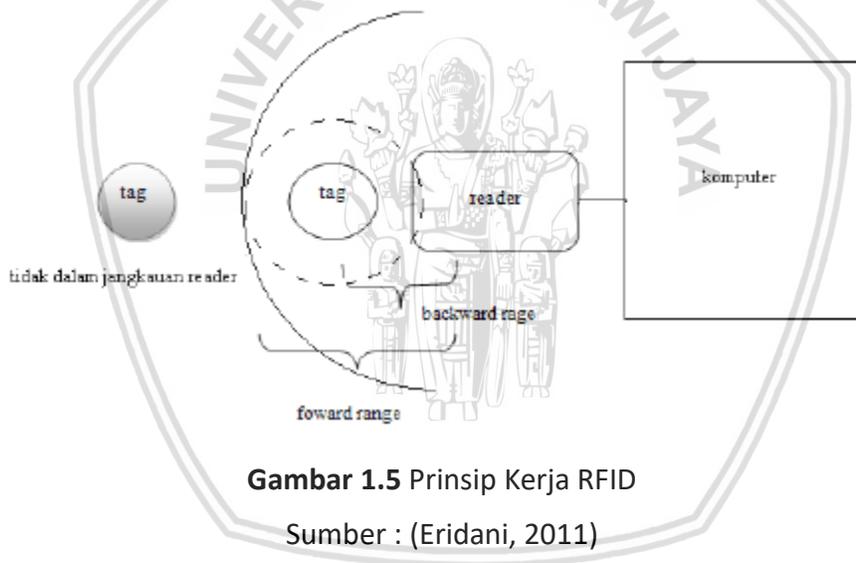
Tabel 1.3 Spesifikasi RFID RC522

Tegangan Operasi	3,3V
Frekuensi	13,56 MHz
Konsumsi Arus	15mA

Jarak Pembacaan	< 50 mm
Protokol Akses	SPI (<i>Serial Peripheral Interface</i>)
Kecepatan Transmisi	10 Mbit/s
Berat	8 gram
Suhu Operasional	-20 - +80 derajat celcius
Dimensi	60 x 40 mm

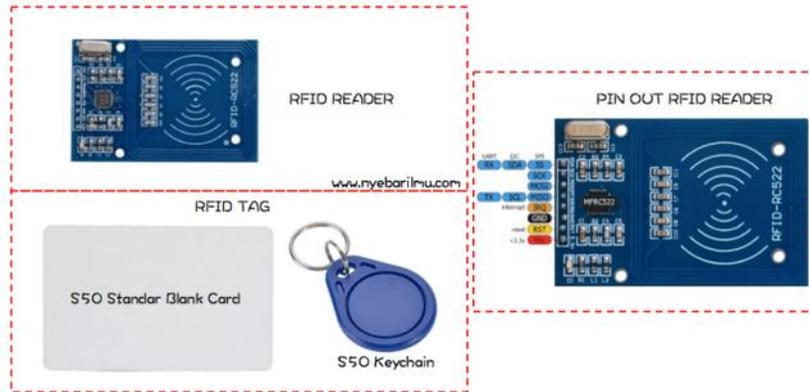
Sumber : (e-Gizmo, 2017)

Di dalam proses kerja RFID, *tag* RFID yang memiliki *chip* yang berisi nomor unik cukup didekatkan kepada *reader* RFID yang terhubung dengan komputer dan nomor unik tersebut akan dikirimkan dari *tag* melalui gelombang frekuensi yang sesuai hingga terbaca dan mampu diidentifikasi oleh *reader*. Jarak antara *reader* dan *tag* RFID untuk dapat membaca nomor unik sangat bervariasi tergantung dari frekuensi yang dimiliki oleh *reader* ataupun *tag*.



Gambar 1.5 Prinsip Kerja RFID

Sumber : (Eridani, 2011)



Gambar 1.6 Tag RFID dan RFID reader

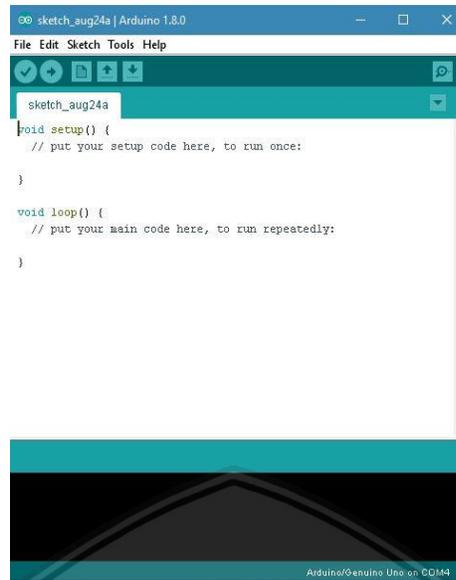
Sumber : (Kece, 2017)

1.2.7 EEPROM Arduino Mega 2560

EEPROM (*Electrically Erasable Programmable Read-Only Memory*) Arduino adalah sebuah *memory* yang jika terdapat data yang disimpan didalamnya tidak akan terhapus walaupun tanpa dialiri listrik (*non-volatile*) berbeda dengan cara kerja RAM (*Random Access Memory*) dimana tanpa dialiri listrik maka *memory* kembali menjadi kosong (*volatile*).

EEPROM Arduino ini memiliki fungsi yang sangat banyak pada *control system* dan kendali. Sebagai contoh *input* password, dengan EEPROM ini password – password default dapat diubah menjadi password baru kemudian password baru akan disimpan di EEPROM. Intinya adalah *memory* EEPROM dapat digunakan untuk menyimpan variabel yang nilainya dapat diubah tanpa merubah program pada *chip* Arduino.

Pada Arduino Mega 2560 kapasitas EEPROM tidak terlalu besar yaitu hanya 4 Kb yang apabila ingin menyimpan data lebih dari itu maka harus menambahkan media penyimpanan yang lain (Ajie, 2015).



Gambar 1.7 Arduino IDE

1.2.8 Motor Servo

Motor servo merupakan salah satu perangkat elektronika yang biasa digunakan sebagai aktuator. Motor servo menggunakan sistem kontrol umpan balik dengan *loop* tertutup, sehingga dapat diatur untuk memastikan posisi sudut dari poros *output* motor. Motor servo terdiri dari penggerak berupa motor DC, serangkaian gir, rangkaian kontrol, dan petensiometer. Rangkaian gir yang terdapat pada poros motor DC berfungsi untuk meningkatkan torsi, sedangkan petensiometer berfungsi untuk menentukan batas posisi putaran poros motor servo sesuai dengan perubahan resistansi pada petensiometer.

Tabel 1.4 Spesifikasi Motor Servo SG-90

Tegangan Operasi	5V
Bahan <i>Gear</i>	Plastik
Kecepatan Operasi	0.1s/60 derajat
Rotasi	0 – 180 derajat
Berat	9 gram

Sumber: (Components, 2017)

Motor servo terbagi menjadi dua jenis, yaitu motor servo standar dengan putaran maksimal 180°, masing – masing 90° ke kanan, dan 90° ke kiri. Motor servo jenis ini paling umum digunakan, karena besaran sudut dapat ditentukan pada rentang jarak 0° sampai dengan 180°. Jenis kedua dari motor servo adalah motor servo *continuous*, yaitu motor servo yang dapat bergerak ke arah kanan dan kiri secara terus menerus. Motor servo dikendalikan dengan memberikan sinyal modulasi lebar pulsa (PWM) melalui kabel kontrol, yang akan menentukan besaran sudut putaran dari poros motor servo (Elektro, 2014).



Gambar 1.8 Motor Servo

(Monk, 2016)

1.2.9 Persentase dan Rata – Rata Kesalahan

Persentase kesalahan digunakan untuk membandingkan nilai pembacaan dengan nilai pasti. Perhitungan kesalahan dapat membantu untuk melihat seberapa dekat estimasi terhadap nilai riil. Rumus persentase kesalahan berguna untuk menentukan ketepatan dari perhitungan dalam melakukan analisis hasil pengujian yang telah dilakukan, untuk melakukan perhitungan persentase *error* digunakan persamaan berikut (Anne Marie Helmenstine, 2018).

Untuk menentukan selisih dari hasil pengukuran sensor dan alat ukur manual dengan **Persamaan (2.3)** berikut.

$$\text{Selisih} = (\text{pembacaan alat ukur} - \text{pembacaan sensor}) \quad (2.3)$$

Untuk menghitung nilai selisih pembacaan nilai sensor dan alat ukur pembanding menggunakan **Persamaan (2.4)** berikut.

$$\text{Presentase Error} = \frac{\text{Selisih nilai pembacaan}}{\text{Pembacaan alat ukur manual}} \times 100\% \quad (2.4)$$

Untuk menghitung rata – rata selisih hasil pembacaan sensor dengan alat ukur manual, digunakan **Persamaan (2.5)** berikut.

$$\text{Rata – rata selisih} = \frac{\text{Selisih nilai pembacaan total}}{\text{jumlah data total}} \quad (2.5)$$

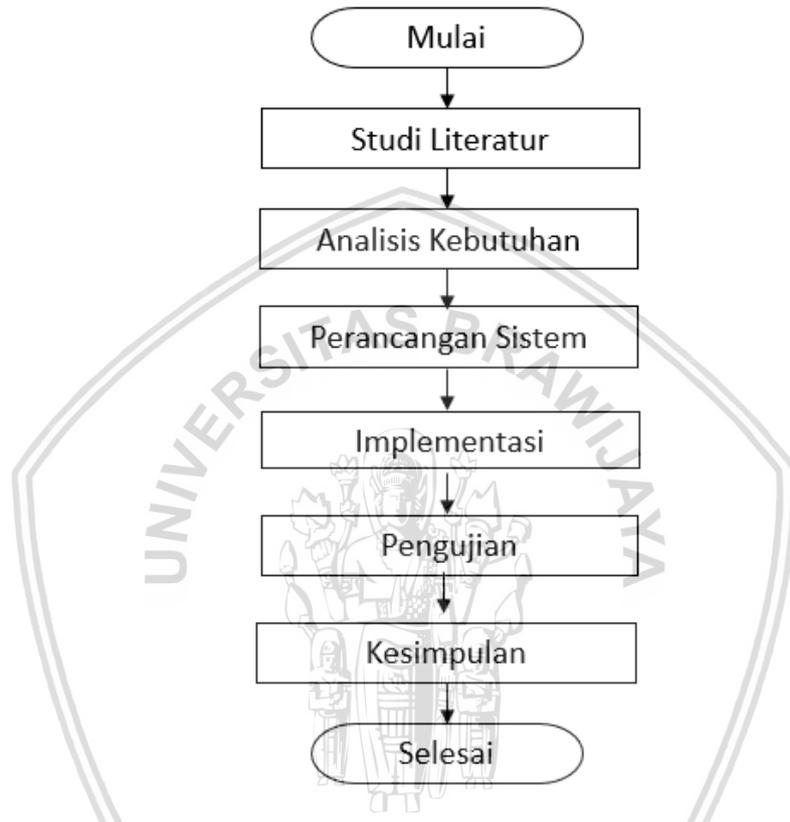
Untuk menghitung rata – rata persentase *error* hasil pembacaan sensor dengan alat ukur manual, digunakan **Persamaan (2.6)** berikut.

$$\text{Rata – rata persentase error} = \frac{\text{persentase error}}{\text{jumlah data total}} \quad (2.6)$$

BAB 3 METODOLOGI

3.1 Metodologi Penelitian

Untuk melaksanakan penelitian, dan agar penelitian dapat berjalan dengan baik dan dapat mencapai tujuan maka diperlukan perencanaan urutan kegiatan dari awal dalam diagram alir sebagai berikut :



Gambar 3.1 Diagram Alir Penelitian

Pada Gambar 3.1 dijelaskan diagram alir dalam pelaksanaan penelitian. Penelitian ini diawali dengan studi literatur untuk mencari dasar teori yang relevan dengan topik penelitian. Proses selanjutnya adalah melakukan analisis kebutuhan dari sistem untuk menentukan spesifikasi yang tepat dari sistem yang akan dibuat. Setelah ditentukan spesifikasi dari sistem proses dilanjutkan ketahap perancangan sistem. Setelah proses perancangan selesai proses dilanjutkan dengan implementasi sistem. Setelah proses selesai di implementasikan, maka dilakukan analisis terhadap kinerja sistem dan kemudian ditarik kesimpulan berdasarkan keseluruhan proses penelitian. Apabila setelah pengambilan kesimpulan terdapat hal – hal yang tidak sesuai dengan perancangan awal sistem maka kembali ke perancangan awal sistem untuk menelusuri ada tidaknya kesalahan selama melakukan penelitian.

3.2 Studi Literatur

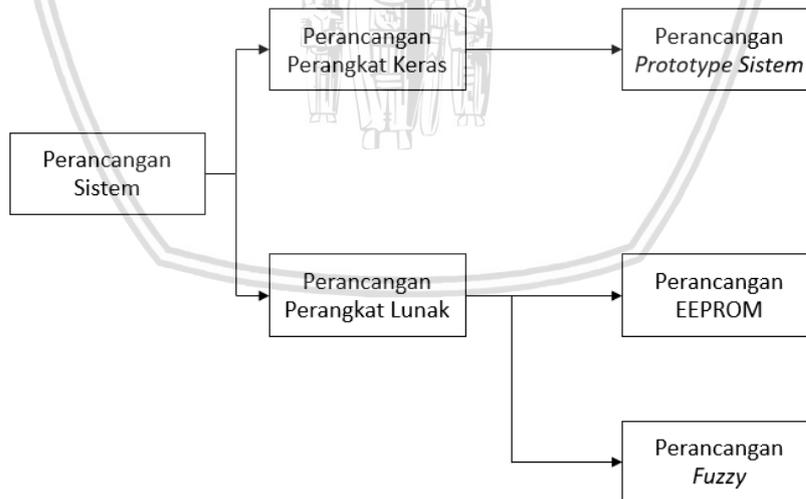
Pada studi literatur dibutuhkan informasi dan referensi yang dilakukan dari penelitian – penelitian sebelumnya. Dari proses ini studi literatur ini kemudian diperoleh dasar – dasar teori yang akan digunakan untuk penelitian ini. Beberapa dasar teori yang digunakan dalam penelitian ini diantaranya parameter kondisi kepadatan, panjang kendaraan yang ada di jalan tol, sensor ultrasonic, RFID, logika *Fuzzy*, motor servo, mikrokontroler Arduino Mega 2560, EEPROM Arduino.

3.3 Rekayasa Kebutuhan

Rekayasa kebutuhan dilakukan untuk melakukan analisis kebutuhan yang terdiri dari dua aspek yaitu kebutuhan pengguna (*user*) dan kebutuhan sistem itu sendiri. Pada analisis kebutuhan pengguna (*user*), dijelaskan tentang apa saja yang dapat dilakukan pengguna (*user*) terhadap sistem, termasuk fitur – fitur yang ada.

3.4 Perancangan Sistem

Perancangan sistem merupakan tahap yang dilakukan setelah semua kebutuhan pengguna sistem dianalisis dan terpenuhi. Pada Gambar 3.2 dapat diketahui bahwa perancangan sistem dibagi menjadi beberapa tahap, yaitu perancangan perangkat keras dimana perangkat prototipe sistem akan dibuat dan perancangan perangkat lunak dimana algoritma dan metode sistem akan dibuat.



Gambar 3.2 Rancangan Sistem

3.4.1 Perancangan Perangkat Keras

Perancangan perangkat keras pada pembuatan rancang bangun sistem ini meliputi sensor - sensor, yaitu sensor ultrasonik, motor servo dan *reader* RFID yang dihubungkan ke pin Arduino Mega 2560.

3.4.2 Perancangan Perangkat Lunak

Perancangan perangkat lunak pada pembuatan rancang bangun sistem ini menggunakan Arduino IDE sebagai tempat pengolahan data sensor sekaligus untuk memprogram sistem. Pada pemrograman sistem dibagi menjadi beberapa bagian yaitu pemrograman untuk mengambil data sensor, pemrograman untuk membuat metode logika *Fuzzy*, pemrograman untuk mengontrol *output*, dan pemrograman EEPROM sebagai penyimpanan data.

3.5 Implementasi Sistem

Implementasi merupakan tahapan untuk merealisasikan sistem yang telah dirancangan. Implementasi perancangan dilakukan setelah analisis kebutuhan sistem selesai dilakukan. Pada tahap ini dilakukan pengimplementasian semua gagasan dan ide baik desain maupun perhitungan menjadi sebuah satu kesatuan. Implementasi dibagi menjadi dua bagian yaitu implementasi perangkat keras dan implementasi perangkat lunak.

3.5.1 Implementasi Perangkat Keras

Tahap ini implementasi dilakukan pada perangkat keras yang telah dianalisis. Perangkat yang telah dianalisis kemudian dihubungkan satu sama lain, seperti pemasangan sensor ultrasonik, *reader* RFID, motor servo sebagai aktuator dan menghubungkannya pada Arduino Mega 2560. Kemudian *tag* RFID yang dipasangkan pada kendaraan sesuai panjangnya.

3.5.2 Implementasi Perangkat Lunak

Tahap ini implementasi dilakukan pada perangkat lunak yang telah dianalisis. Implementasi perangkat lunak pada penelitian ini menggunakan aplikasi Arduino IDE. Pada Arduino IDE program – program sistem akan dibuat. Program – program yang dibuat antara lain, program pembacaan *input* sensor – sensor, program metode *Fuzzy*, program *output* untuk mengontrol pembukaan pintu gerbang tol dengan aktuator motor servo, dan program baca tulis data menggunakan EEPROM yang ada pada Arduino. Selanjutnya program tersebut digabungkan menjadi satu kesatuan.

3.6 Pengujian dan Analisis

Tahap pengujian dan analisis sistem untuk menguji apakah sistem yang dibuat sudah sesuai seperti yang diharapkan oleh peneliti. Pengujian dan analisis menggunakan beberapa parameter, Berikut ini adalah parameter yang digunakan dalam pengujian sistem :

1. Pegujian fungsional dari sistem. Hal ini ditujukan untuk mengetahui apakah sistem sudah berjalan sesuai dengan keinginan peneliti. Tolak ukur dalam hal ini dapat dilihat melalui serial monitor pada Arduino IDE

2. Pengujian EEPROM Arduino terhadap sinkronisasi data. Pengujian ini ditujukan untuk melihat akurasi EEPROM Arduino dalam memberikan data panjang kendaraan terhadap sistem sesuai *tag* RFID yang dibaca oleh *reader* RFID
3. Pengujian logika *Fuzzy* terhadap sistem. Pengujian ini ditujukan untuk mengatur buka tutup pintu gerbang tol yang menggunakan aktuator yaitu motor servo sebagai *output*, apakah sistem mampu mengolah masukan dan menghasilkan keluaran sesuai dengan harapan peneliti

3.7 Kesimpulan

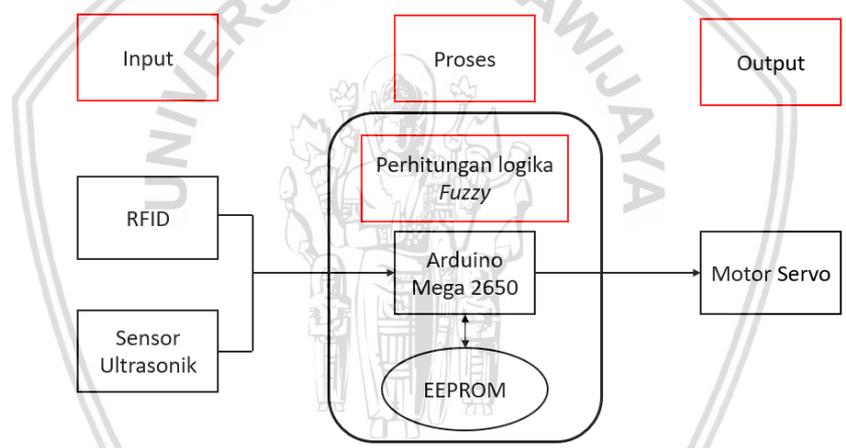
Penarikan kesimpulan merupakan tahap yang dilakukan setelah melakukan seluruh kegiatan pengujian sistem yang telah dirancang sebelumnya. Tujuan penarikan kesimpulan, agar penelitian ini dapat digunakan sebagai tolak ukur dan dapat dilanjutkan menjadi penelitian yang lebih baik serta tidak berhenti sampai kegiatan penulis selesai. Pengambilan saran bertujuan agar penelitian ini dapat dikembangkan menjadi penelitian yang lebih baik ke depannya.



BAB 4 REKAYASA KEBUTUHAN

4.1 Deskripsi Umum

Sistem yang akan dibuat yaitu sistem pengendalian pintu gerbang tol dengan menggunakan logika *Fuzzy* yang akan digunakan untuk menentukan durasi pembukaan pintu gerbang tol. Metode tersebut akan digunakan sehingga menghasilkan keluaran. Sistem akan terdiri dari perangkat keras berupa *mikrokontroller* yang berfungsi untuk mengendalikan keseluruhan sistem. Perangkat keras lainnya berupa sensor, pada sistem ini sensor yang digunakan adalah RFID (*Radio Frequency Identification*) yang digunakan sebagai *interface* kendaraan dengan sistem dengan membaca *tag* yang ada pada kendaraan menggunakan *reader* RFID lalu data dari *tag* RFID akan disinkronisasi didalam EEPROM dari Arduino, sensor ultrasonik yang digunakan untuk mengetahui kondisi kepadatan yang terjadi. Pada bagian aktuator digunakan motor servo sebagai penggerak pintu gerbang tol. Pada gambar 4.1 dapat dilihat gambaran umum diagram blok perancangan sistem.



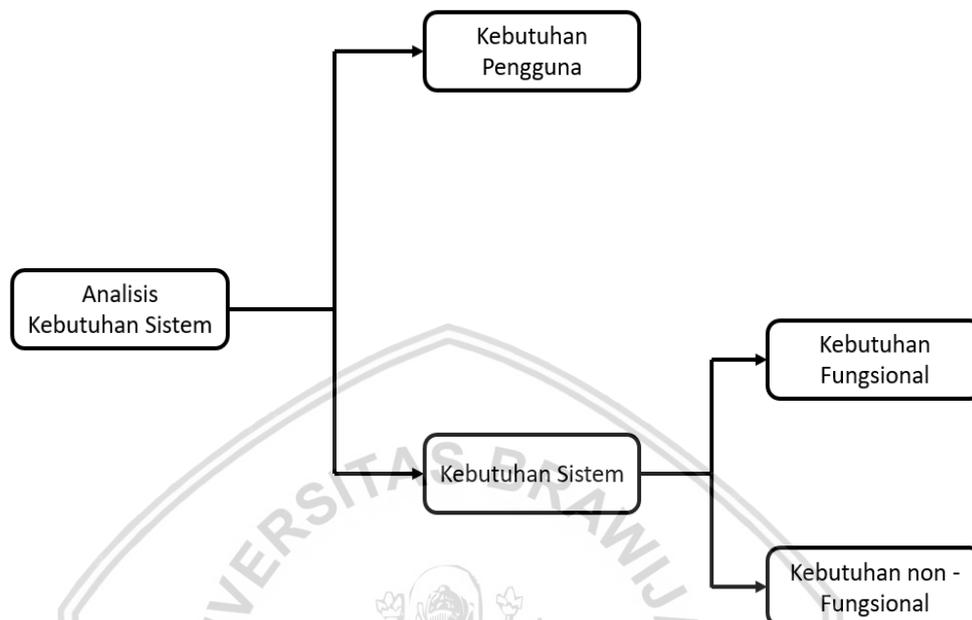
Gambar 4.1 Diagram Blok Perancangan Sistem

Pada gambar 4.1 di atas merupakan blok diagram dari sistem secara umum. Terdapat 3 bagian utama yaitu blok sensor, mikrokontroller dan juga aktuator. Pada blok *input* terdapat RFID yang akan melakukan akuisisi data berupa pembacaan *tag* yang ada pada kendaraan, dan sensor ultrasonik yang akan melakukan akuisisi data berupa kondisi kepadatan, data hasil akuisisi kedua sensor tersebut akan dikirimkan ke mikrokontroller sebagai *input* untuk melakukan perhitungan logika *Fuzzy*.

Pada blok mikrokontroller akan dilakukan perhitungan logika *Fuzzy* yang mengacu pada data panjang kendaraan dimana ditentukan dalam EEPROM Arduino sesuai kartu *tag* yang mempunyai panjang kendaraan masing - masing dan kondisi kepadatan untuk menentukan besaran pembukaan pintu gerbang tol menggunakan logika *Fuzzy*. Kemudian setelah hasil perhitungan selesai, mikrokontroller akan mengirimkan sinyal ke motor servo untuk bergerak

membuka pintu gerbang tol dengan besaran yang sesuai dengan hasil perhitungan logika *Fuzzy* yang telah dilakukan.

4.2 Analisis Kebutuhan Sistem



Gambar 4.2 Analisis Kebutuhan Sistem

Pada bagian ini akan dibahas lebih lanjut mengenai kebutuhan – kebutuhan yang digunakan untuk sistem ini, seperti kebutuhan pengguna, kebutuhan perangkat keras dan kebutuhan perangkat lunak. Mengacu pada Gambar 4.2 semua komponen dalam analisis kebutuhan saling berkaitan satu sama lain, hal ini dikarenakan sistem ini membutuhkan semua komponen pada Gambar 4.2 agar berjalan sesuai keinginan.

4.2.1 Kebutuhan Antarmuka Pengguna

Kebutuhan antarmuka (*user*) merupakan kebutuhan untuk menghubungkan pengguna sistem dan sistem itu sendiri. Dalam sistem ini kebutuhan antarmuka menggunakan *tag* RFID sebagai interface kendaraan dengan sistem dimana *tag* RFID tersebut akan tertempel pada kendaraan masing – masing pengguna dan memiliki nomor unik yang berbeda pada setiap *tag* RFID

4.2.2 Kebutuhan Fungsional

Kebutuhan fungsional dari sistem adalah kebutuhan yang harus terpenuhi agar sistem dapat berjalan dengan baik sesuai tujuan. Jika salah satu fungsi sistem tidak dapat bekerja maka sistem dapat dikatakan sebagai gagal. Pada sistem rancang bangun pengendali pintu gerbang tol dengan RFID fungsional yang harus dipenuhi adalah sebagai berikut :

4.2.2.1 Fungsi Mengambil dan Menampilkan Data Sensor

Pada fungsi ini sistem dapat mengambil data – data sensor agar dapat mendeteksi panjang kendaraan dan kondisi kepadatan. Kemudian data tersebut digunakan sebagai *input* sistem untuk menentukan durasi buka tutup pintu gerbang tol.

4.2.2.2 Fungsi Kontrol Logika *Fuzzy*

Fungsi ini mengharuskan sistem dapat melakukan pembukaan maupun penutupan pintu gerbang tol sesuai perhitungan logika *Fuzzy*.

4.2.2.3 Fungsi Menyimpan Data

Fungsi ini mengharuskan sistem untuk dapat menyimpan data panjang kendaraan didalam EEPROM Arduino. Data tersebut dapat ditambah, diubah ataupun dihapus sesuai keinginan peneliti.

4.2.3 Kebutuhan Non Fungsional Sistem

Kebutuhan nonfungsional dibagi menjadi dua bagian yaitu kebutuhan perangkat keras dan perangkat lunak

4.2.3.1 Kebutuhan Perangkat Keras

Kebutuhan perangkat keras yang digunakan untuk menyusun sistem ini terdiri dari beberapa komponen yang dijelaskan melalui Tabel 4.1 berikut ini

Tabel 4.1 Kebutuhan Perangkat Keras Sistem

Perangkat Keras (<i>Hardware</i>)	Fungsi
Arduino Mega 2560	Arduino Mega 2560 berfungsi sebagai mikrokontroller dan mengatur/mengolah dari hasil I/O dari keseluruhan sistem
RFID	Sebuah teknologi yang berfungsi membaca reader yang ada pada kendaraan yang akan melewati gerbang tol.
Sensor <i>Ultrasonik</i>	Sebuah sensor yang berfungsi untuk mengetahui panjang kemacetan jika ada mobil yang menghalangi pantulan sinar ultrasonik yang dipancarkan sesuai waktu yang telah ditentukan.
Motor Servo	Aktuator yang digunakan untuk membuka dan menutup pintu gerbang tol pada maket.

4.2.3.2 Kebutuhan Perangkat Lunak

Kebutuhan perangkat lunak pada sistem ini dijelaskan melalui Tabel 4.2 berikut.

Tabel 4.2 Kebutuhan Perangkat Lunak Sistem

Perangkat Lunak (Software)	Fungsi
Arduino IDE	Pada sistem ini berfungsi untuk menulis dan melakukan <i>upload</i> program pada mikrokontroller Arduino agar dapat berfungsi sebagai kontroler I/O sebuah sistem.
EEPROM Arduino	Sebagai tempat penyimpanan data panjang kendaraan yang akan berinteraksi dengan sistem
Logika <i>Fuzzy</i>	Melakukan perhitungan dengan metode <i>Fuzzy</i> Mamdani untuk menentukan besaran pembukaan pintu gerbang tol pada pemodelan sistem



BAB 5 PERANCANGAN DAN IMPLEMENTASI

Pada bab ini membahas mengenai perancangan dan implementasi yang dilakukan pada penelitian ini. perancangan membahas mengenai persiapan dan inialisasi sistem sehingga siap untuk diimplementasikan. Sedangkan implementasi berisi mengenai penerapan perancangan sistem sesuai dengan perancangan yang sudah dilakukan sebelumnya.

5.1 Perancangan Sistem

Pada bagian ini berisi mengenai perancangan dan implementasi pada penelitian “Rancang bangun pengendali pintu gerbang tol dengan RFID menggunakan logika *Fuzzy*”. Pada perancangan sistem akan dijelaskan bagaimana sistem dibangun agar mampu terkoneksi dengan baik antar perangkat satu dengan perangkat yang lain. Pada bagian perancangan terdapat dua perancangan sistem, yaitu perancangan perangkat lunak dan perancangan perangkat keras. Pada bagian implementasi sistem terdiri dari implementasi perangkat keras dan implementasi perangkat lunak.

5.1.1 Perancangan Perangkat Keras

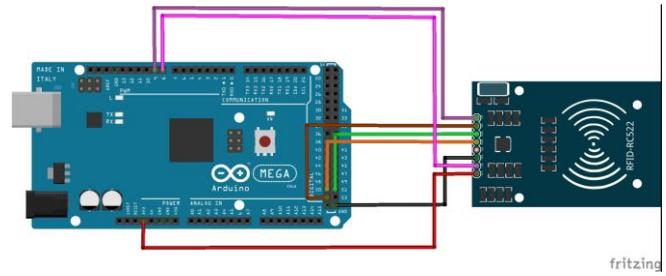
Perangkat keras disusun menggunakan beberapa komponen diantaranya Arduino Mega 2560 sebagai mikrokontroller, RFID *reader*, Sensor Ultrasonik dan servo. Rancangan sistem ini dibangun secara keseluruhan untuk menggambarkan konsep dan cara kerja sistem dalam mengatur *output* sistem berupa pengkontrollan servo.

Arduino Mega 2560 terhubung dengan sensor – sensor yang akan digunakan sistem yaitu RFID *reader* RC522, Sensor Ultrasonik dan servo.

5.1.1.1 Perancangan rangkaian RFID *reader* RC522

RFID *reader* RC522 digunakan sebagai bentuk *interface* pengguna dengan sistem dimana RFID *reader* RC522 akan membaca *Tag* RFID yang dibawa pengguna untuk membayar tol. *Tag* RFID yang dibawa pengguna berisi tentang identitas kendaraan berupa panjang kendaraan mobil yang dibawa oleh pengguna tersebut.

Pin RFID *reader* RC522 akan terintegrasi dengan pin Arduino Mega 2560 dengan penempatan pin sesuai dengan gambar 5.1 berikut.



Gambar 5.1 Rangkaian RFID reader RC522 dan Arduino Mega 2560

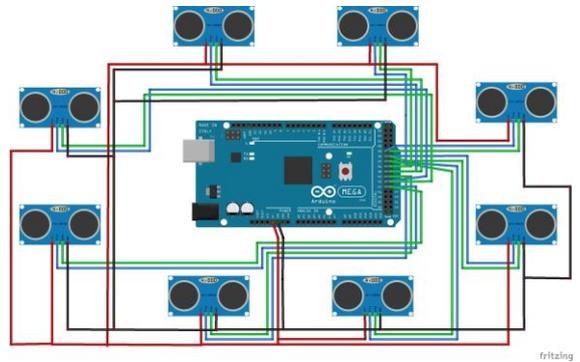
RFID reader RC522 digunakan untuk mengakuisisi data berupa panjang kendaraan yang ada pada Tag RFID yang memiliki nomor unik tersendiri sehingga Tag RFID yang satu berbeda dengan yang lainnya.

Tabel 5.1 Keterangan Pin Arduino Mega 2560 dan RFID reader

Pin RFID reader	Pin Arduino Mega 2560
VCC	3,3 V
RST	8
MISO	50
MOSI	51
SCK	52
SDA	9
GND	GND

5.1.1.2 Perancangan rangkaian Sensor Ultrasonik

Sensor Ultrasonik merupakan salah satu sensor yang digunakan dalam penelitian ini. Sensor Ultrasonik yang digunakan sebanyak 8 buah, dengan perancangan sebagai berikut.



Gambar 5.2 Rangkaian Sensor Ultrasonik dan Arduino Mega 2560

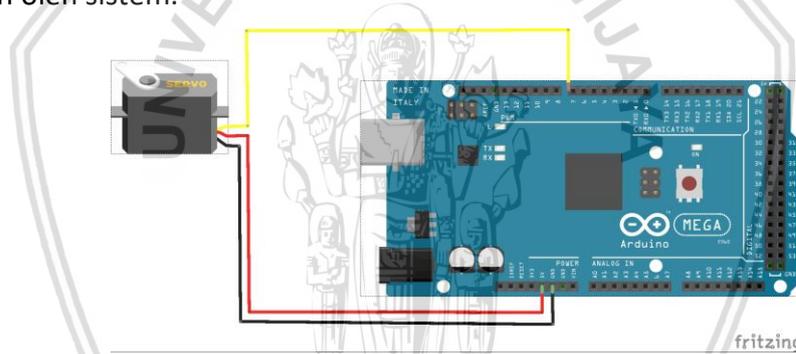
Pada Gambar 5.2 terdapat delapan sensor Ultrasonik yang akan digunakan pada sistem ini. setiap sensor akan mengakuisisi data berupa panjang jalan secara horizontal.

Tabel 5.2 Keterangan Pin Sensor Ultrasonik dan Arduino Mega 2560

Pin Sensor Ultrasonik	Pin Arduino Mega 2560
VCC	5V
Trig	30, 32, 34, 36, 38, 40, 42, 44
Echo	31, 33, 35, 37, 39, 41, 43, 45
GND	GND

5.1.1.3 Perancangan rangkaian servo

Pada sistem ini aktuator yang digunakan adalah motor servo yang akan menggerakkan pintu gerbang tol buka dan tutup. Servo pada sistem ini juga sebagai *output* dimana servo akan terbuka sesuai dengan perhitungan yang dilakukan oleh sistem.



Gambar 5.3 Rangkaian Servo dan Arduino Mega 2560

Servo terdiri dari 3 pin, yaitu GND, VCC, dan kontrol (PWM). Pin kontrol penggunaannya disesuaikan dengan pin PWM yang terdapat pada Arduino Mega 2560.

Tabel 5.3 Keterangan Pin Motor Servo dan Arduino Mega 2560

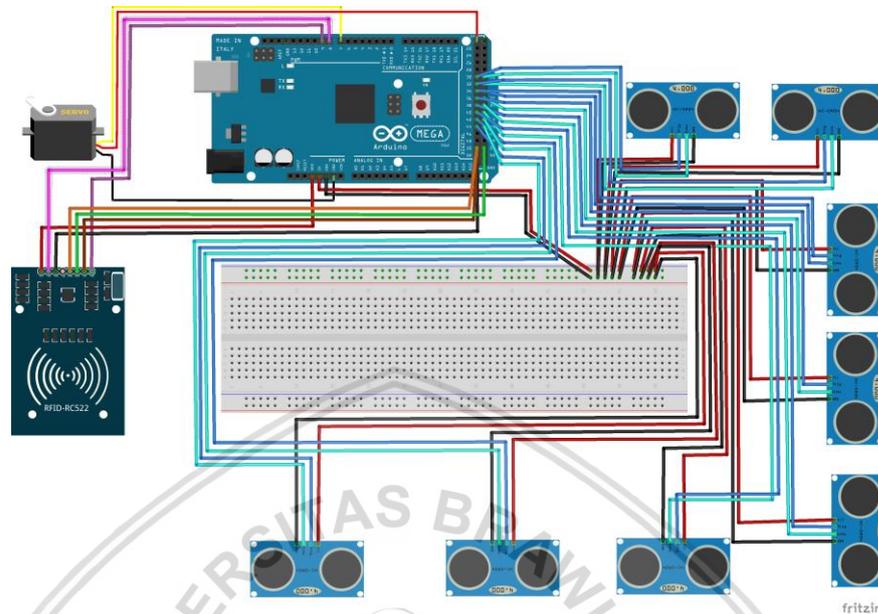
Pin Motor Servo	Pin Arduino Mega 2560
VCC	5V
Kontrol	7
GND	GND

5.1.1.4 Perancangan perangkat keras keseluruhan

Masing – masing perangkat keras yang telah dirancang dihubungkan dan disatukan menjadi kesatuan sehingga membentuk rancangan perangkat keras

repository.ub.ac.id

sistem yang diinginkan. Untuk sensor ultrasonik yang pada sistem ini digunakan 8 buah, bagian VCC dan GND pada masing masing sensor dijadikan satu pada *project board* agar sensor dapat dijalankan.



Gambar 5.4 Skema Perancangan Perangkat Keras

5.1.2 Perancangan Perangkat Lunak

Perancangan perangkat lunak pendukung yang digunakan dalam pembuatan sistem ini adalah Arduino IDE. Program yang dibuat dengan menggunakan metode *Fuzzy Mamdani* sebagai pengambilan keputusan untuk menentukan *rule outputnya* berdasarkan *input* yang didapat dari sensor Ultrasonik dan RFID. Perancangan perangkat lunak dimulai dari *inputan* masing – masing sensor yaitu sensor Ultrasonik dan RFID *reader* yang akan membaca *Tag* RFID yang memuat nilai panjang kendaraan. Dari 2 *input* sensor tersebut akan diproses menggunakan metode *Fuzzy Mamdani* untuk menentukan himpunan *Fuzzy*.

5.1.2.1 Perancangan penyimpanan data menggunakan EEPROM Arduino

Perancangan ini bertujuan untuk menyimpan data panjang kendaraan pada masing – masing *Tag* RFID yang digunakan dalam penelitian. Perancangan ini memanfaatkan Arduino IDE sebagai media untuk membaca dan menulis data yang memuat panjang kendaraan tiap kendaraan yang akan melewati sistem.

Pada Arduino IDE, terdapat banyak *library* yang dapat digunakan untuk memudahkan operasi suatu sistem. Fungsi dari *library* pada Arduino sendiri merupakan sebuah kumpulan kode – kode yang dapat mengeksekusi perintah – perintah tertentu berdasarkan fungsi dan prosedur yang terbangun oleh file headernya.

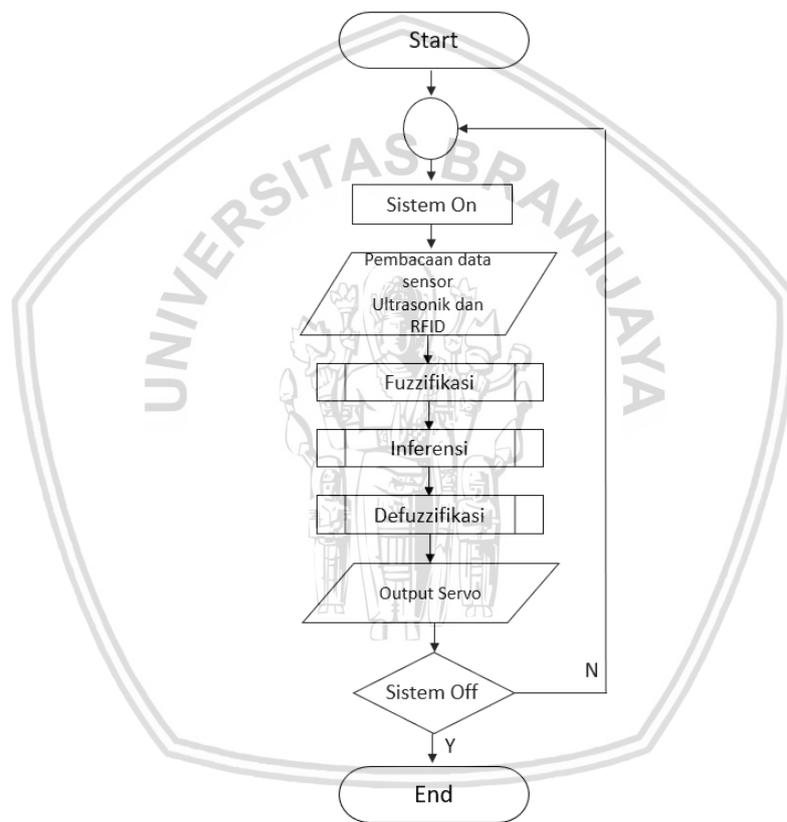
Kapasitas EEPROM yang ada di Arduino Mega 2560 seperti yang telah dijelaskan pada dasar teori memiliki kapasitas sebesar 4 KB. Dimana kapasitas



tersebut tidak terlalu besar jika ingin menyimpan data dengan ukuran file yang lebih dari 4 KB.

5.1.2.2 Perancangan Logika Fuzzy

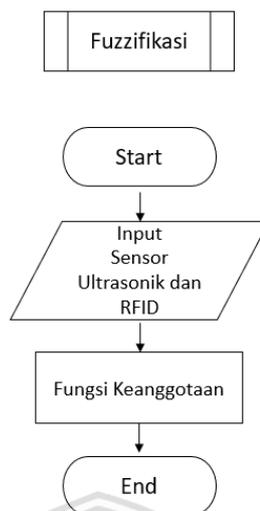
Perancangan sistem yang menggunakan metode *Fuzzy Mamdani* ini memerlukan beberapa tahap agar keputusan yang menjadi *output* dari sistem sesuai dengan beberapa perhitungan metode *Fuzzy*. Tahap tersebut adalah *Fuzzifikasi*, inferensi *Fuzzy* dan defuzzifikasi. Pada gambar berikut menunjukkan tahapan pada implementasi metode *Fuzzy* pada kontrol pintu gerbang tol memiliki fungsi yang saling berhubungan dengan tahapan yang lain sehingga tahapan yang dihasilkan akan menjadi *input* dari tahapan berikutnya sampai menjadi *output* akhir dari sistem.



Gambar 5.5 Flowchart Perancangan Logika Fuzzy

5.1.2.2.1 Fuzzifikasi

Fuzzifikasi adalah suatu perubahan nilai tegas dalam fungsi keanggotaan *Fuzzy*. Pada Gambar 5.6, tahapan fuzzifikasi akan menerima proses *input* dari sensor Ultrasonik yang memiliki fungsi untuk *input* data kondisi kepadatan dan RFID untuk *input* data panjang kendaraan. Berikut ini adalah *flowchart* tahap fuzzifikasi.



Gambar 5.6 Flowchart Proses Fuzzifikasi

Sistem kontrol pintu gerbang tol menggunakan metode *Fuzzy Mamdani* memiliki 2 *input* dari sensor yaitu panjang kendaraan dan kondisi kepadatan. Pada variabel panjang kendaraan dibagi menjadi 3 himpunan keanggotaan yaitu golongan 1, golongan 2 dan golongan 3. Sedangkan untuk kondisi kepadatan dibagi menjadi 3 himpunan keanggotaan yaitu sepi, sedang dan padat

5.1.2.2.1.1 Variabel Panjang Kendaraan

Pada variabel panjang kendaraan, menggunakan 3 variabel linguistik yang diperoleh dari golongan – golongan kendaraan yang melewati jalan tol. Proses perhitungan *Fuzzy* diawali dengan pendefinisian variabel *input* dan *output* yang akan digunakan. Variabel *input* pertama yang digunakan pada penelitian ini adalah panjang kendaraan. Pengkategorian variabel panjang kendaraan didefinisikan pada Tabel 5.4 berikut.

Tabel 5.4 Variabel Panjang Kendaraan

Linguistik	Numerik
Gol. 1	< 25 cm
Gol. 2	20 – 40 cm
Gol. 3	> 35 cm

Pada Tabel 5.4 tersebut didapatkan data panjang kendaraan yang dikategorikan dalam 3 kategori. Data tersebut didapatkan dari panjang mobil asli diskalakan 1 banding 20. Berdasarkan data tersebut maka dapat diolah panjang kendaraan yang menjadi salah satu variabel *input* dalam penelitian ini. Berikut adalah himpunan pada variabel panjang kendaraan.

- Gol. 1 [20]

Fungsi keanggotaan dapat dirumuskan sebagai berikut.



$$\mu[\text{Gol. 1}] = \begin{cases} 1, & x \leq 20 \\ \frac{25-x}{25-20}, & 20 < x < 25 \\ 0, & x \geq 25 \end{cases} \quad (5.1)$$

- Gol. 2 [20 40]

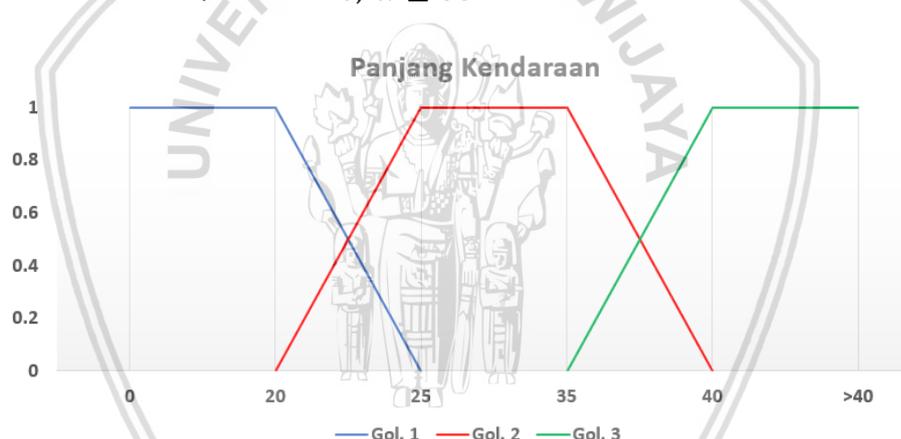
Fungsi keanggotaan dapat dirumuskan sebagai berikut.

$$\mu[\text{Gol.2}] = \begin{cases} 0, & x \leq 20 \text{ atau } \geq 40 \\ \frac{x-20}{25-20}, & 20 < x < 25 \\ 1, & 25 \leq x \leq 35 \\ \frac{40-x}{40-35}, & 35 < x < 40 \end{cases} \quad (5.2)$$

- Gol. 3 [35]

Fungsi keanggotaan dapat dirumuskan sebagai berikut.

$$\mu[\text{Gol.3}] = \begin{cases} 1, & x \geq 40 \\ \frac{x-35}{40-35}, & 35 < x < 40 \\ 0, & x \leq 35 \end{cases} \quad (5.3)$$



Gambar 5.7 Himpunan Fuzzy Panjang Kendaraan

5.1.2.2.1.2 Variabel Kondisi Kepadatan

Variabel *input* kedua kondisi kepadatan yang ada pada gerbang tol. Pendefinisian variabel kondisi kepadatan dalam bentuk linguistik maupun numerik yang diperoleh dari panjang kepadatan sesuai sensor Ultrasonik yang telah terpasang akan didefinisikan pada Tabel 5.5 berikut.

Tabel 5.5 Variabel Kondisi Kepadatan

Linguistik	Numerik
Sepi	< 42 cm
Sedang	37,5 – 60 cm
Padat	> 55,5 cm

Nilai normal diperoleh dari panjang jalan yang dibuat pada maket yaitu 76,5 cm. Kemudian panjang jalan total di maket dikurangi jarak gerbang tol dengan sensor Ultrasonik yang pertama kali membaca kondisi jalan yaitu 33 cm. Berikut adalah himpunan pada variabel kondisi kepadatan.

- Sepi [42]

Fungsi keanggotaan dapat dirumuskan sebagai berikut.

$$\mu[\text{Sepi}] = \begin{cases} 1, & x \leq 37,5 \\ \frac{42-x}{42-37,5}, & 37,5 < x < 42 \\ 0, & x \geq 42 \end{cases} \quad (5.4)$$

- Sedang [37,5 60]

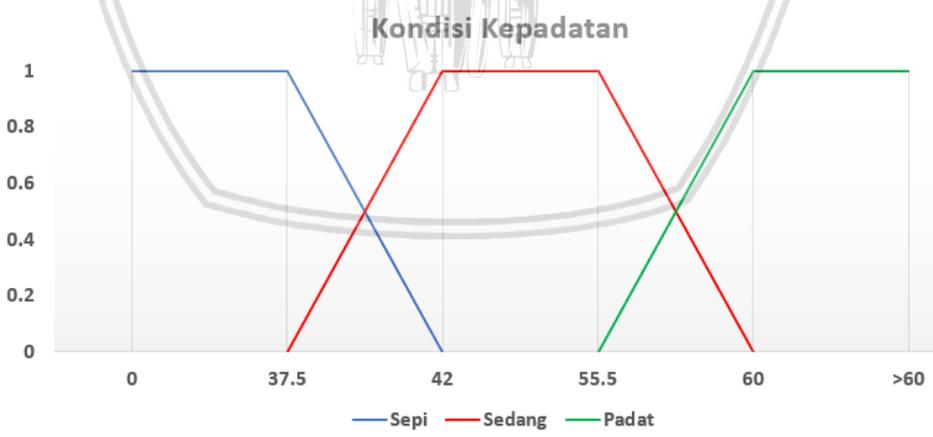
Fungsi keanggotaan dapat dirumuskan sebagai berikut.

$$\mu[\text{Sedang}] = \begin{cases} 0, & x \leq 37,5 \text{ atau } \geq 60 \\ \frac{x-37,5}{40-37,5}, & 37,5 < x < 42 \\ 1, & 42 \leq x \leq 55,5 \\ \frac{60-x}{60-55,5}, & 55,5 < x \leq 60 \end{cases} \quad (5.5)$$

- Padat [55,5]

Fungsi keanggotaan dapat dirumuskan sebagai berikut.

$$\mu[\text{Padat}] = \begin{cases} 1, & x \geq 60 \\ \frac{x-55,5}{60-55,5}, & 55,5 < x < 60 \\ 0, & x \leq 55,5 \end{cases} \quad (5.6)$$



Gambar 5.8 Grafik Himpunan Fuzzy Kondisi Kepadatan

5.1.2.2.2 Pembuatan Aturan Dasar (rule) Fuzzy

Aturan dasar (rule based) merupakan bentuk dari aturan implikasi if – then. Fungsi implikasi merupakan proses penyusunan aturan yang menyatakan relasi antara variabel input dan output. Dalam metode Fuzzy Mamdani, fungsi implikasi

yang dipakai adalah fungsi min. Dalam pembuatan aturan, penulis membuat dengan “IF” dan “AND” dan menghasilkan perintah “THAN”.

Nilai *output* didapatkan dari data pembukaan pintu gerbang tol oleh Departemen Pekerjaan Umum Direktorat Jenderal Bina Marga dan melalui percobaan 30 orang relawan dalam melakukan simulasi pembayaran tol dengan kondisi tertentu pada maket yang telah dibuat. Berikut merupakan hasil percobaan yang ditunjukkan pada Tabel 5.6 berikut.

Tabel 5.6 Percobaan Sistem Pada Relawan

No.	Relawan ke -	Panjang Kendaraan	Kondisi Kepadatan	Durasi Pembayaran Tol
1	1	16	Sepi	4.2 detik
2	2	21	Sedang	4.55 detik
3	3	31	Sedang	7.9 detik
4	4	16	Padat	5.51 detik
5	5	31	Sepi	7.25 detik
6	6	25	Sepi	4.79 detik
7	7	16	Padat	5.13 detik
8	8	21	Padat	5.89 detik
9	9	31	Padat	5.87 detik
10	10	16	Sepi	4.6 detik
11	11	21	Sedang	5.14 detik
12	12	25	Sedang	5.27 detik
13	13	21	Sedang	5.74 detik
14	14	21	Sedang	5.3 detik
15	15	25	Sepi	5.26 detik
16	16	31	Padat	7.14 detik
17	17	25	Sedang	4.67 detik
18	18	25	Sepi	4.32 detik
19	19	31	Sepi	6.9 detik
20	20	16	Padat	5.88 detik

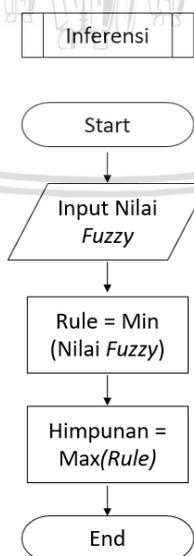
Aturan *Fuzzy* yang dipakai untuk menentukan kondisi *output* yang berupa durasi buka tutup pintu gerbang tol menggunakan motor servo berdasarkan data yang dijabarkan pada tabel di atas dijelaskan pada Tabel 5.7 di bawah ini.

Tabel 5.7 Aturan Fuzzy

Input		Output
Panjang Kendaraan	Kondisi Kepadatan	Durasi buka tutup pintu gerbang tol
Gol. 1	Sepi	Cepat
Gol. 1	Sedang	Cepat
Gol. 1	Padat	Sedang
Gol. 2	Sepi	Cepat
Gol. 2	Sedang	Sedang
Gol. 2	Padat	Lambat
Gol. 3	Sepi	Sedang
Gol. 3	Sedang	Lambat
Gol. 3	Padat	Lambat

5.1.2.2.3 Penalaran (inferensi) Fuzzy

Penalaran (inferensi) adalah tahap implikasi untuk menguraikan nilai *input* untuk menentukan nilai *output* sebagai bentuk pengambilan keputusan. Berdasarkan pembuatan aturan pada Tabel 5.7 akan ditentukan nilai α pada setiap aturan. α adalah nilai keanggotaan baru hasil operasi 2 himpunan atau lebih. Fungsi yang digunakan adalah fungsi min dan *max* pada setiap fungsi implikasi. Berikut *flowchart* tahap inferensi untuk mendapatkan nilai α untuk memperoleh nilai keanggotaan baru.



Gambar 5.9 Flowchart Proses Inferensi

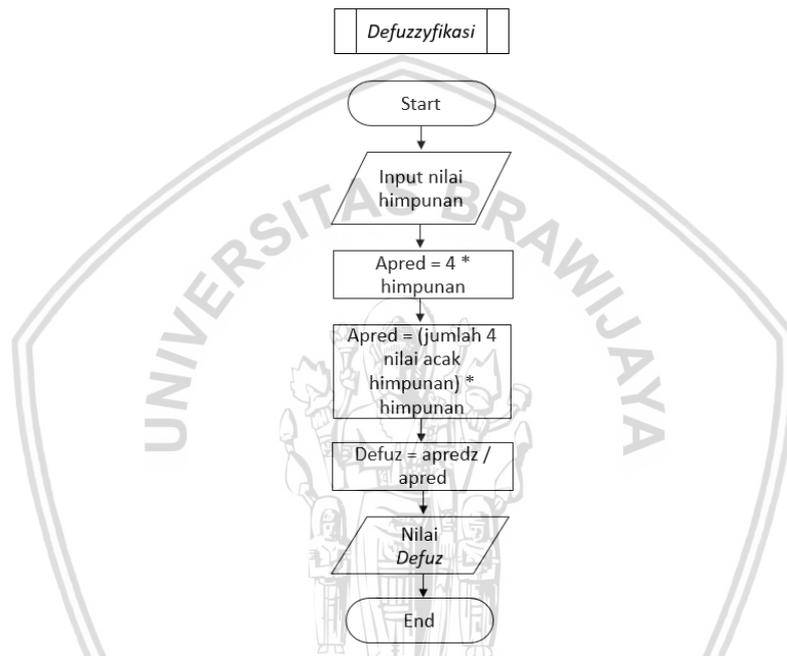
Pada Gambar 5.9 digambarkan bahwa tahap inferensi membutuhkan *input* dari nilai fungsi keanggotaan yang didapatkan pada tahap fuzzifikasi. Nilai dari



fungsi keanggotaan akan dicocokkan dengan aturan yang telah dibuat. Ketika didapatkan aturan yang cocok, akan diterapkan metode min, *max* untuk mencari nilai yang terkecil dan terbesar dari setiap aturan sehingga didapatkan nilai α yang merupakan nilai keanggotaan baru.

5.1.2.2.4 Proses Defuzzifikasi

Defuzzifikasi merupakan tahap mengubah *Fuzzy output* menjadi *crisp value* berdasarkan fungsi keanggotaan yang telah ditentukan. Setelah memperoleh nilai α berdasarkan min, *max method*, setiap variabel akan menilai setiap *rule* untuk memberi hasil inferensi min, *max*. di bawah ini adalah *flowchart* tahapan defuzzifikasi.



Gambar 5.10 Flowchart Proses Defuzzifikasi

Pada Gambar 5.10 dijelaskan dalam tahap defuzzifikasi membutuhkan *input* dari nilai α , nilai dari keanggotaan baru yang diperoleh dari tahap inferensi. Jika nilai α telah diperoleh, digunakan *Centroid Method* untuk proses defuzzifikasi. Titik – titik area abu – abu ditentukan secara acak sehingga akan didapatkan satu titik pusat (*Center of area* atau *center of gravity*).

5.1.2.2.4.1 Defuzzifikasi

Pada *output* durasi pembukaan pintu gerbang tol berupa motor servo terdapat 3 fungsi keanggotaan yaitu cepat, sedang, dan lambat. Masing – masing *domain range* didapat dari percobaan data sampel dan juga mengacu pada waktu pelayanan di gerbang tol. Fungsi keanggotaan *output* durasi pembukaan pintu gerbang tol dibentuk menggunakan representasi kurva trapesium dan kurva bahu.





Gambar 5.11 Grafik *Output* Durasi Pembukaan Pintu

Berikut ini adalah himpunan pada variabel *output* durasi pembukaan pintu gerbang tol serta fungsi keanggotaan yang dapat dirumuskan seperti di bawah ini:

- Cepat [5]

Fungsi keanggotaan dapat dirumuskan sebagai berikut.

$$\mu[\text{Cepat}] = \begin{cases} 1, & x \leq 4 \\ \frac{5-x}{5-4}, & 4 < x < 5 \\ 0, & x \geq 5 \end{cases} \quad (5.7)$$

- Sedang [4 7]

Fungsi keanggotaan dapat dirumuskan sebagai berikut.

$$\mu[\text{Sedang}] = \begin{cases} 0, & x \leq 4 \text{ atau } \geq 8 \\ \frac{x-4}{5-4}, & 4 < x < 5 \\ 1, & 5 \leq x \leq 6 \\ \frac{7-x}{7-6}, & 6 < x \leq 7 \end{cases} \quad (5.8)$$

- Lambat [6 8]

Fungsi keanggotaan dapat dirumuskan sebagai berikut.

$$\mu[\text{Lambat}] = \begin{cases} 1, & 7 \geq x \geq 8 \\ \frac{x-6}{7-6}, & 6 < x < 7 \\ 0, & x \leq 6 \end{cases} \quad (5.9)$$

5.2 Implementasi Sistem

Pada sub bab ini menjelaskan tentang implementasi sistem pada perangkat keras dan perangkat lunak.

5.2.1 Implementasi Perangkat Keras

Pada tahapan implementasi perangkat keras ini, dilakukan pembuatan sistem, dengan menggunakan perangkat keras yang telah ditentukan sebelumnya. Sesuai dengan perancangan yang telah ditentukan, sensor yang digunakan untuk melakukan akuisisi data adalah sensor ultrasonik dan RFID, dengan jumlah sensor Ultrasonik sebanyak 8 buah, RFID *reader* RC522 dan *Tag* RFID. Data hasil akuisisi oleh sensor kemudian dikirimkan ke mikrontroller, pada sistem ini mikrokontroller yang digunakan pada sistem ini adalah Arduino Mega 2560.

5.2.1.1 Implementasi Perangkat Arduino dengan Sensor Ultrasonik

Implementasi perangkat keras ini bertujuan untuk merancang perangkat keras Arduino Mega 2560 dengan sensor Ultrasonik yang telah dirancang. Penggunaan sensor Ultrasonik yang akan digunakan sistem yaitu sebanyak 8 buah sensor. Pengimplementasian Arduino Mega 2560 dengan sensor Ultrasonik ditunjukkan pada Gambar 5.11 berikut.

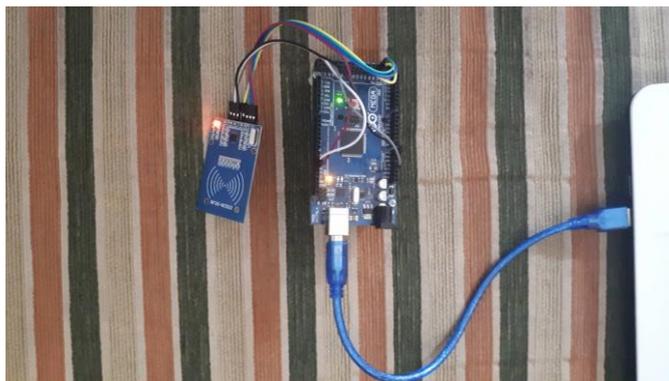


Gambar 5.12 Perangkat Keras Arduino dengan Sensor Ultrasonik

Pada Gambar 5.12 perangkat keras sensor Ultrasonik telah terintegrasi dengan Arduino Mega 2560 sesuai dengan perancangan. Sensor Ultrasonik terhubung pada pin VCC, GND, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45 Arduino Mega 2560.

5.2.1.2 Implementasi Perangkat Arduino dengan RFID

Implementasi perangkat keras ini bertujuan untuk merancang perangkat keras Arduino Mega 2560 dengan RFID *reader* RC522 yang telah dirancang. RFID *reader* yang akan digunakan sistem yaitu sebanyak 1 buah. Pengimplementasian Arduino Mega 2560 dengan RFID *reader* RC522 ditunjukkan pada Gambar 5.12 berikut.



Gambar 5.13 Perangkat Keras Arduino dengan RFID reader

Pada Gambar 5.13 perangkat keras RFID reader telah terintegrasi dengan Arduino Mega 2560 sesuai dengan perancangan. RFID reader terhubung pada pin VCC, GND, 8, 9, 50, 51, 52 Arduino Mega 2560.

5.2.1.3 Implementasi Perangkat Arduino dengan Motor Servo

Implementasi perangkat keras ini bertujuan untuk merancang perangkat keras Arduino Mega 2560 dengan motor servo yang telah dirancang. Motor servo yang akan digunakan sistem yaitu sebanyak 1 buah. Pengimplementasian Arduino Mega 2560 dengan motor servo ditunjukkan pada Gambar 5.12 berikut.

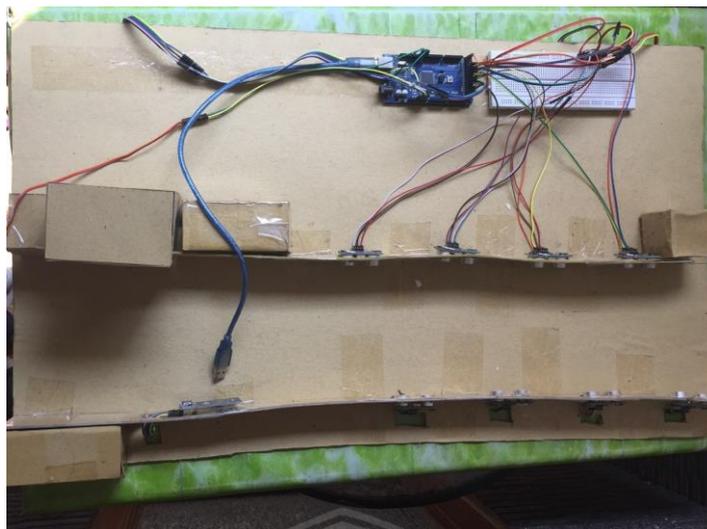


Gambar 5.14 Perangkat Keras Arduino dengan Motor Servo

Pada Gambar 5.14 perangkat keras motor servo telah terintegrasi dengan Arduino Mega 2560 sesuai dengan perancangan. RFID reader terhubung pada pin VCC, GND, 7 Arduino Mega 2560.

5.2.1.4 Implementasi Perangkat Keras Sistem

Implementasi ini merupakan tahap akhir dari implementasi perangkat keras dimana perangkat keras yang telah dirancang dibuat menjadi kesatuan agar dapat bekerja sesuai dengan sistem yang telah dirancang pula. Penggabungan seluruh perangkat keras sistem ditunjukkan pada Gambar 5.15 berikut



Gambar 5.15 Implementasi Sistem Pada Maket

Pada Gambar 5.15 di atas merupakan implementasi sistem yang telah dirancang pada maket gerbang tol dengan skala 1 : 20. Pada Gambar 5.14 di atas, sensor yang ditandai dengan lingkaran berwarna merah merupakan RFID *reader* RC522, sensor yang ditandai dengan lingkaran yang berwarna hijau adalah sensor Ultrasonik. Bagian yang diberi lingkaran berwarna kuning adalah aktuator berupa motor servo, dan yang diberi tanda lingkaran berwarna biru adalah Arduino Mega 2560

5.2.2 Implementasi Perangkat Lunak

Implementasi perangkat lunak yaitu hasil dari perancangan sistem yang telah dibuat. Pada bagian implementasi ini terdapat beberapa bagian yaitu program pembacaan sensor, program penyimpanan data panjang kendaraan menggunakan EEPROM Arduino, pembentukan fuzzifikasi, pembuatan *rule base* sebagai hasil kombinasi himpunan, melakukan inferensi penalaran dan defuzzifikasi sebagai fungsi untuk menentukan *output* sistem.

Pada tahapan ini juga dilakukan implementasi metode yang digunakan dalam bentuk kode program yang akan *download* ke dalam Arduino Mega 2560, untuk memasukkan kode program ke dalam Arduino Mega 2560 digunakan aplikasi Arduino IDE yang diinstall pada laptop sebagai media untuk memasukkan kode program ke dalam *board* Arduino.

5.2.2.1 Proses Penyimpanan Data Menggunakan EEPROM

Untuk membaca dan menulis data menggunakan EEPROM menggunakan *library* khusus yang pada Arduino IDE dimana tidak harus mendownload *library* tersebut karena sudah terpasang pada Arduino IDE. Nama *library* tersebut adalah <EEPROM.h> dan dalam *library* ini terdapat dua fungsi utama yang digunakan untuk membaca dan menulis data ke EEPROM yaitu:

- EEPROM.write (addr, val) digunakan untuk menulis data ke EEPROM. Parameter 'addr' adalah alamat EEPROM mulai dari 0 sampai 1023,

27	EEPROM.write(7, mobilpanjang8); //menulis data mobil ke 8
28	EEPROM.write(8, mobilpanjang9); //menulis data mobil ke 9
29	EEPROM.write(9, mobilpanjang10; //menulis data mobil ke 10
30	Serial.println("Done");
31	delay(1000000000);
32	}

Tabel 5.8 di atas menunjukkan bahwa dalam perancangannya data panjang kendaraan yang disimpan adalah 10 mobil. Walaupun tipe data yang digunakan adalah float, tipe data yang hanya bisa digunakan adalah tipe data int.

5.2.2.2 Proses Pembacaan Kondisi Kepadatan Menggunakan Sensor Ultrasonik

Implementasi yang diterapkan pada sensor Ultrasonik ini bertujuan agar sensor dapat membaca data adanya kendaraan yang menghalangi sensor yang bertujuan untuk membaca kondisi kepadatan. Kode program untuk sensor Ultrasonik dapat bekerja sesuai dengan fungsi yang diharapkan sistem ditunjukkan pada Tabel 5.9 di bawah ini.

Tabel 5.9 Kode Program Ultrasonik

No	Kode Program
1	const int trigPin1 = 44; //inisialisasi pin trig Ultrasonik
2	const int echoPin1 = 45; //inisialisasi pin echo Ultrasonik
3	const int trigPin2 = 36; //inisialisasi pin trig Ultrasonik
4	const int echoPin2 = 37; //inisialisasi pin echo Ultrasonik
5	const int trigPin3 = 42; //inisialisasi pin trig Ultrasonik
6	const int echoPin3 = 43; //inisialisasi pin echo Ultrasonik
7	const int trigPin4 = 34; //inisialisasi pin trig Ultrasonik
8	const int echoPin4 = 35; //inisialisasi pin echo Ultrasonik
9	const int trigPin5 = 40; //inisialisasi pin trig Ultrasonik
10	const int echoPin5 = 41; //inisialisasi pin echo Ultrasonik
11	const int trigPin6 = 32; //inisialisasi pin trig Ultrasonik
12	const int echoPin6 = 33; //inisialisasi pin echo Ultrasonik
13	const int trigPin7 = 38; //inisialisasi pin trig Ultrasonik
14	const int echoPin7 = 39; //inisialisasi pin echo Ultrasonik
15	const int trigPin8 = 30; //inisialisasi pin trig Ultrasonik
16	const int echoPin8 = 31; //inisialisasi pin echo Ultrasonik
17	void setup() { // fungsi setup sensor Ultrasonik
18	Serial.begin(9600);
19	pinMode(echoPin1, OUTPUT);
20	pinMode(trigPin1, INPUT);
21	pinMode(echoPin2, OUTPUT);



```
22   pinMode(trigPin2, INPUT);
23   pinMode(echoPin3, OUTPUT);
24   pinMode(trigPin3, INPUT);
25   pinMode(echoPin4, OUTPUT);
26   pinMode(trigPin4, INPUT);
27   pinMode(echoPin5, OUTPUT);
28   pinMode(trigPin5, INPUT);
29   pinMode(echoPin6, OUTPUT);
30   pinMode(trigPin6, INPUT);
31   pinMode(echoPin7, OUTPUT);
32   pinMode(trigPin7, INPUT);
33   pinMode(echoPin8, OUTPUT);
34   pinMode(trigPin8, INPUT);
35 }
36 void ultrasonik1() { //fungsi Ultrasonik pertama
37 long duration1, cm1; //inisialisai variabel duration1, cm1
38   pinMode(trigPin1, OUTPUT);
39   digitalWrite(trigPin1, LOW);
40   delayMicroseconds(2);
41   digitalWrite(trigPin1, HIGH);
42   delayMicroseconds(5);
43   digitalWrite(trigPin1, LOW);
44   pinMode(echoPin1, INPUT);
45   duration1 = pulseIn(echoPin1, HIGH);
46   cm1 = (duration1 / 2) / 29.4; //inisialisasi variabel cm1
47           =(duration1/2) / 29.4
48   if (cm1 <= 7) { //inisialisasi fungsi if jika cm1 <=7
49     jarakUtama = 33;
50     Serial.println("Ada MObil di ping 1");
51   }
52 }
53 void ultrasonik2() { // fungsi Ultrasonik kedua
54 long duration2, cm2; //inisialisasi variabel duration2, cm2
55   pinMode(trigPin2, OUTPUT);
56   digitalWrite(trigPin2, LOW);
57   delayMicroseconds(2);
58   digitalWrite(trigPin2, HIGH);
59   delayMicroseconds(5);
```

```
60 digitalWrite(trigPin2, LOW);
61 pinMode(echoPin2, INPUT);
62 duration2 = pulseIn(echoPin2, HIGH);
63 cm2 = (duration2 / 2) / 29.4; //inisialisasi variabel cm2
64         =(duration1/2) / 29.4
65 if (cm2 <= 7) { //inisialisasi fungsi if jika cm2 <=7
66     jarakUtama = 37.5;
67     Serial.println("Ada MObil di ping 2");
68 }
69 }
70 void ultrasonik3() { //fungsi Ultrasonik ketiga
71 long duration3, cm3; //inisialisasi variabel duration3, cm3
72 pinMode(trigPin3, OUTPUT);
73 digitalWrite(trigPin3, LOW);
74 delayMicroseconds(2);
75 digitalWrite(trigPin3, HIGH);
76 delayMicroseconds(5);
77 digitalWrite(trigPin3, LOW);
78 pinMode(echoPin3, INPUT);
79 duration3 = pulseIn(echoPin3, HIGH);
80 cm3 = (duration3 / 2) / 29.4; //inisialisasi variabel cm3
81         =(duration1/2) / 29.4
82 if (cm3 <= 7) { // inisialisasi fungsi if jika cm3 <=7
83     jarakUtama = 42;
84     Serial.println("Ada MObil di ping 3");
85 }
86 }
87 void ultrasonik4() { //fungsi Ultrasonik keempat
88 long duration4, cm4; //inisialisasi variabel duration4, cm4
89 pinMode(trigPin4, OUTPUT);
90 digitalWrite(trigPin4, LOW);
91 delayMicroseconds(2);
92 digitalWrite(trigPin4, HIGH);
93 delayMicroseconds(5);
94 digitalWrite(trigPin4, LOW);
95 pinMode(echoPin4, INPUT);
96 duration4 = pulseIn(echoPin4, HIGH);
97 cm4 = (duration4 / 2) / 29.4; //inisialisasi variabel cm4
```

```
98         =(duration1/2) / 29.4
99     if (cm4 <= 7) { //inisialisasi fungsi if jika cm4 <=7
100         jarakUtama = 46.5;
101         Serial.println("Ada MObil di ping 4");
102     }
103 }
104 void ultrasonik5() { // fungsi Ultrasonik kelima
105 long duration5, cm5; //inisialisasi variabel duration5, cm5
106     pinMode(trigPin5, OUTPUT);
107     digitalWrite(trigPin5, LOW);
108     delayMicroseconds(2);
109     digitalWrite(trigPin5, HIGH);
110     delayMicroseconds(5);
111     digitalWrite(trigPin5, LOW);
112     pinMode(echoPin5, INPUT);
113     duration5 = pulseIn(echoPin5, HIGH);
114     cm5 = (duration5 / 2) / 29.4; //inisialisasi variabel cm5
115         =(duration1/2) / 29.4
116     if (cm5 <= 7) { //inisialisasi fungsi if jika cm5 <=7
117         jarakUtama = 51;
118         Serial.println("Ada MObil di ping 5");
119     }
120 }
121 }
122 void ultrasonik6() { //fungsi Ultrasonik keenam
123 long duration6, cm6; //inisialisasi variabel duration6, cm6
124     pinMode(trigPin6, OUTPUT);
125     digitalWrite(trigPin6, LOW);
126     delayMicroseconds(2);
127     digitalWrite(trigPin6, HIGH);
128     delayMicroseconds(5);
129     digitalWrite(trigPin6, LOW);
130     pinMode(echoPin6, INPUT);
131     duration6 = pulseIn(echoPin6, HIGH);
132     cm6 = (duration6 / 2) / 29.4; //inisialisasi variabel cm6
133         =(duration1/2) / 29.4
134     if (cm6 <= 7) { //inisialisasi fungsi if jika cm6 <=7
135         jarakUtama = 55.5;
136         Serial.println("Ada MObil di ping 6");
```

```

137 }
138 }
139 void ultrasonik7() { //fungsi Ultrasonik ketujuh
140 long duration7, cm7; //inisialisasi variabel duration7, cm7
141   pinMode(trigPin7, OUTPUT);
142   digitalWrite(trigPin7, LOW);
143   delayMicroseconds(2);
144   digitalWrite(trigPin7, HIGH);
145   delayMicroseconds(5);
146   digitalWrite(trigPin7, LOW);
147   pinMode(echoPin7, INPUT);
148   duration7 = pulseIn(echoPin7, HIGH);
149   cm7 = (duration7 / 2) / 29.4; //inisialisasi variabel cm7
150                                     =(duration1/2) / 29.4
151   if (cm7 <= 7) { //inisialisasi fungsi if jika cm7 <=7
152     jarakUtama = 60;
153     Serial.println("Ada MObil di ping 7");
154   }
155 }
156 void ultrasonik8() { //fungsi Ultrasonik kedepalan
157 long duration8, cm8; //inisialisasi variabel duration8, cm8
158   pinMode(trigPin8, OUTPUT);
159   digitalWrite(trigPin8, LOW);
160   delayMicroseconds(2);
161   digitalWrite(trigPin8, HIGH);
162   delayMicroseconds(5);
163   digitalWrite(trigPin8, LOW);
164   pinMode(echoPin8, INPUT);
165   duration8 = pulseIn(echoPin8, HIGH);
166   cm8 = (duration8 / 2) / 29.4; //inisialisasi variabel cm8
167                                     =(duration1/2) / 29.4
168   if (cm8 <= 7) { //inisialisasi fungsi if jika cm8 <=7
169     jarakUtama = 64.5;
170     Serial.println("Ada MObil di ping 8");
171   }
172 }

```

Tabel 5.9 di atas menunjukkan dalam perancangannya sensor Ultrasonik membaca data lebar jalan pada maket, kemudian jika pembacaan data kurang dari 7 cm yang dapat disimpulkan bahwa ada kendaraan yang menghalangi,

setiap sensor yang terhalangi akan mengirimkan nilai *input* untuk perhitungan *Fuzzy*.

5.2.2.3 Proses Pembacaan Panjang Kendaraan Menggunakan RFID reader

RFID *reader* pada sistem bertujuan agar dapat membaca *Tag* RFID yang berisikan data panjang kendaraan sesuai nomor serial yang ada pada *Tag* RFID. Kode program untuk RFID *reader* agar dapat bekerja dengan fungsi yang diharapkan sistem ditunjukkan pada Tabel 5.10 di bawah ini.

Tabel 5.10 Kode Program RFID

No	Kode Program
1	<code>#include <SPI.h> //library serial parallel interface</code>
2	<code>#include <RFID.h> //library RFID</code>
3	<code>#define sda 9 //Pin Serialdata (SDA)</code>
4	<code>#define rst 9 //pin Reset</code>
5	<code>RFID rfid(sda, rst);</code>
6	<code>void setup() {</code>
7	<code> Serial.begin(9600); //baud komunikasi serial monitor</code>
8	<code> SPI.begin(); //Prosedur antarmuka SPI</code>
9	<code> rfid.init(); //Memulai inialisasi module RFID</code>
10	<code> delay (4000);</code>
11	<code>}</code>
12	<code>void loop() {</code>
13	<code> if (rfid.isCard()) { //inisialisasi fungsi if jika ada tag</code>
14	<code> RFID yang terbaca pada RFID reader</code>
15	<code> if (rfid.readCardSerial()) {</code>
16	<code> Serial.print(rfid.serNum[0]); //serial no.1</code>
17	<code> Serial.print(" ");</code>
18	<code> Serial.print(rfid.serNum[1]); //serial no.2</code>
19	<code> Serial.print(" ");</code>
20	<code> Serial.print(rfid.serNum[2]); //serial no.3</code>
21	<code> Serial.print(" ");</code>
22	<code> Serial.print(rfid.serNum[3]); //serial no.4</code>
23	<code> Serial.print(" ");</code>
24	<code> Serial.print(rfid.serNum[4]); //serial no.5</code>
25	<code> Serial.println("");</code>
26	<code> Serial.print(rfid.serNum[5]); //serial no.6</code>
27	<code> Serial.println("");</code>
28	<code> Serial.print(rfid.serNum[6]); //serial no.7</code>
29	<code> Serial.println("");</code>

```
30 Serial.print(rfid.serNum[7]); //serial no.8
31 Serial.println("");
32 Serial.print(rfid.serNum[8]); //serial no.9
33 Serial.println("");
34 Serial.print(rfid.serNum[9]); //serial no.10
35 Serial.println("");
36 if (rfid.serNum[0] == 81 && rfid.serNum[1] == 181 &&
37     rfid.serNum[2] == 43 && rfid.serNum[3] == 230 &&
38     rfid.serNum[4] == 41) {
39     hitungFuzzy = true; //kondisi nilai boolean true
40     panjangkendaraan = mobil[0]; //memasukkan nilai
41                             panjang kendaraan 1
42 }
43 if (rfid.serNum[0] == 209 && rfid.serNum[1] == 221 &&
44     rfid.serNum[2] == 169 && rfid.serNum[3] == 137 &&
45     rfid.serNum[4] == 44) {
46     hitungFuzzy = true; //kondisi nilai boolean true
47     panjangkendaraan = mobil[1]; //memasukkan nilai
48                             panjang kendaraan 2
49 }
50 if (rfid.serNum[0] == 170 && rfid.serNum[1] == 69 &&
51     rfid.serNum[2] == 234 && rfid.serNum[3] == 182 &&
52     rfid.serNum[4] == 179) {
53     hitungFuzzy = true; //kondisi nilai boolean true
54     panjangkendaraan = mobil[2]; //memasukkan nilai
55                             panjang kendaraan 3
56 }
57 if (rfid.serNum[0] == 138 && rfid.serNum[1] == 202 &&
58     rfid.serNum[2] == 232 && rfid.serNum[3] == 182 &&
59     rfid.serNum[4] == 30) {
60     hitungFuzzy = true; //kondisi nilai boolean true
61     panjangkendaraan = mobil[3]; //memasukkan nilai
62                             panjang kendaraan 4
63 }
64 if (rfid.serNum[0] == 42 && rfid.serNum[1] == 151 &&
65     rfid.serNum[2] == 223 && rfid.serNum[3] == 182 &&
66     rfid.serNum[4] == 212)
67     hitungFuzzy = true; //kondisi nilai boolean true
```

```
68     panjangkendaraan = mobil[4]; //memasukkan nilai
69         panjang kendaraan 5
70     }
71     if (rfid.serNum[0] == 218 && rfid.serNum[1] == 59 &&
72         rfid.serNum[2] == 233 && rfid.serNum[3] == 182 &&
73         rfid.serNum[4] == 190) {
74         hitungFuzzy = true; //kondisi nilai boolean true
75         panjangkendaraan = mobil[5]; //memasukkan nilai
76             panjang kendaraan 6
77     }
78     if (rfid.serNum[0] == 10 && rfid.serNum[1] == 230 &&
79         rfid.serNum[2] == 205 && rfid.serNum[3] == 182 &&
80         rfid.serNum[4] == 151) {
81         hitungFuzzy = true; //kondisi nilai boolean true
82         panjangkendaraan = mobil[6]; //memasukkan nilai
83             panjang kendaraan 7
84     }
85     if (rfid.serNum[0] == 202 && rfid.serNum[1] == 31 &&
86         rfid.serNum[2] == 220 && rfid.serNum[3] == 182 &&
87         rfid.serNum[4] == 191) {
88         hitungFuzzy = true; //kondisi nilai boolean true
89         panjangkendaraan = mobil[7]; //memasukkan nilai
90             panjang kendaraan 8
91     }
92     if (rfid.serNum[0] == 218 && rfid.serNum[1] == 210 &&
93         rfid.serNum[2] == 207 && rfid.serNum[3] == 182 &&
94         rfid.serNum[4] == 113) {
95         hitungFuzzy = true; //kondisi nilai boolean true
96         panjangkendaraan = mobil[8]; //memasukkan nilai
97             panjang kendaraan 9
98     }
99     if (rfid.serNum[0] == 170 && rfid.serNum[1] == 254 &&
100         rfid.serNum[2] == 215 && rfid.serNum[3] == 182 &&
101         rfid.serNum[4] == 53) {
102         hitungFuzzy = true; //kondisi nilai boolean true
103         panjangkendaraan = mobil[9]; //memasukkan nilai
104             panjang kendaraan 10
105     }
```



```

18 else if (panjangkendaraan > 20 && panjangkendaraan < 25)
19 { miuPanjangKendaraan[2] = (panjangkendaraan - 20) / (25 -
20 20);
21 }
22 else if (panjangkendaraan >= 25 && panjangkendaraan <= 35)
23 { // fungsi else if untuk panjang kendaraan >=25 dan <=35
24     miuPanjangKendaraan[2] = 1; //statement fungsi else if
25 }
26 else if (panjangkendaraan > 35 && panjangkendaraan < 40) {
27 miuPanjangKendaraan[2] = (40 - panjangkendaraan) / (40 -
28 35);
29 }
30 if (panjangkendaraan >= 40) { //fungsi if jika panjang
31     kendaraan >=40
32     miuPanjangKendaraan[3] = 1; //statement fungsi if
33 }
34 else if (panjangkendaraan > 35 && panjangkendaraan < 40)
35 { miuPanjangKendaraan[3] = (panjangkendaraan - 35) / (40
36 - 35);
37 }
38 else if (panjangkendaraan <= 35) { //fungsi else if jika
39     panjang kendaraan <=35
40     miuPanjangKendaraan[3] = 0; //statement fungsi else if
41 }
42 Serial.println("Miu Panjang");
43 Serial.println(miuPanjangKendaraan[1]);
44 Serial.println(miuPanjangKendaraan[2]);
45 Serial.println(miuPanjangKendaraan[3]);
46 }

```

Pada Tabel 5.11 di atas adalah kode program untuk menentukan fungsi keanggotaan panjang kendaraan. Pada variabel himpunan keanggotaan panjang kendaraan masing – masing memanggil fungsi yang berisi rumus keanggotaan sesuai bentuk kurva yang digunakan.

Tabel 5.12 Kode Program Fungsi Keanggotaan Kondisi Kepadatan

No	Kode Program
1	//fuzzifikasi KondisiKepadatan
2	void KondisiKepadatan() { //fungsi kondisi kepadatan
3	if (ultrasonic <= 37.5) { //inisialisasi fungsi if jika

```
4         ultrasonic <=37,5
5         miuKondisiKepadatan[1] = 1; //statement fungsi if
6     }
7     else if (ultrasonic > 37.5 && ultrasonic < 42) {
8     miuKondisiKepadatan[1] = (42 - ultrasonic) / (42 - 37.5);
9     }
10    else if (ultrasonic >= 42) { //inisialisasi fungsi else if
11        miuKondisiKepadatan[1] = 0; //statement fungsi else if
12    }
13    if (ultrasonic <= 37.5 || ultrasonic >= 60) {
14        miuKondisiKepadatan[2] = 0; //statement fungsi else if
15    }
16    else if (ultrasonic > 37.5 && ultrasonic < 42) {
17        miuKondisiKepadatan[2] = (ultrasonic - 37.5) / (42 -
18 37.5); // statement fungsi else if
19    }
20    else if (ultrasonic >= 42 && ultrasonic <= 55.5) {
21        miuKondisiKepadatan[2] = 1; //statement fungsi else if
22    }
23    else if (ultrasonic > 55.5 && ultrasonic < 60) {
24        miuKondisiKepadatan[2] = (60 - ultrasonic) / (60 - 55.5);
25    }
26    if (ultrasonic >= 60) { //fungsi if jika ultrasonic >= 60
27        miuKondisiKepadatan[3] = 1; //statement fungsi if
28    }
29    else if (ultrasonic > 55.5 && ultrasonic < 60) {
30        miuKondisiKepadatan[3] = (ultrasonic - 55.5) / (60 -
31 \55.5); //statement fungsi else if
32    }
33    else if (ultrasonic <= 55.5) { //fungsi else if jika
34        ultrasonic <=55,5
35        miuKondisiKepadatan[3] = 0; //statement fungsi else if
36    }
37    Serial.println("Miu miuKondisiKepadatan");
38    Serial.print("Miu Kondisi Kepadatan Sepi:");
39    Serial.println(miuKondisiKepadatan[1]);
40    Serial.print("Miu Kondisi Kepadatan Sedang:");
41    Serial.println(miuKondisiKepadatan[2]);
```

```

42 Serial.print("Miu Kondisi Kepadatan Padat:");
43 Serial.println(miuKondisiKepadatan[3]);
44 }
    
```

2. Inferensi dan Aturan dasar (*rule based*)

Aturan *Fuzzy* dibuat dari gabungan antara himpunan-himpunan keanggotaan yang ada dan dalam sistem ini terdapat 9 *rule Fuzzy*. *Rule* tersebut dibuat menggunakan fungsi implikasi yang dalam metode Mamdani digunakan fungsi *min*, *max*. dari *rule* yang ada terdapat kemungkinan bahwa *output* yang dihasilkan sama. Seperti yang ditunjukkan pada Tabel 5.13 di bawah ini.

Tabel 5.13 Program *Rule Based* dan Inferensi

No	Kode Program
1	//Inferensi MinMax
2	void implikasiMin() {
3	rule[1][1] = min(miuPanjangKendaraan[1],
4	miuKondisiKepadatan[1]);
5	rule[1][2] = min(miuPanjangKendaraan[1],
6	miuKondisiKepadatan[2]);
7	rule[1][3] = min(miuPanjangKendaraan[1],
8	miuKondisiKepadatan[3]);
9	rule[2][1] = min(miuPanjangKendaraan[2],
10	miuKondisiKepadatan[1]);
11	rule[2][2] = min(miuPanjangKendaraan[2],
12	miuKondisiKepadatan[2]);
13	rule[2][3] = min(miuPanjangKendaraan[2],
14	miuKondisiKepadatan[3]);
15	rule[3][1] = min(miuPanjangKendaraan[3],
16	miuKondisiKepadatan[1]);
17	rule[3][2] = min(miuPanjangKendaraan[3],
18	miuKondisiKepadatan[2]);
19	rule[3][3] = min(miuPanjangKendaraan[3],
20	miuKondisiKepadatan[3]);
21	Serial.println("Nilai RULE:");
22	for (int i = 1; i < 4; i ++) {
23	for (int j = 1; j < 4; j ++) {
24	Serial.println(rule[i][j]);
25	}
26	}



27	}
----	---

3. Defuzzifikasi

Tabel 5.14 Kode Program Defuzzifikasi

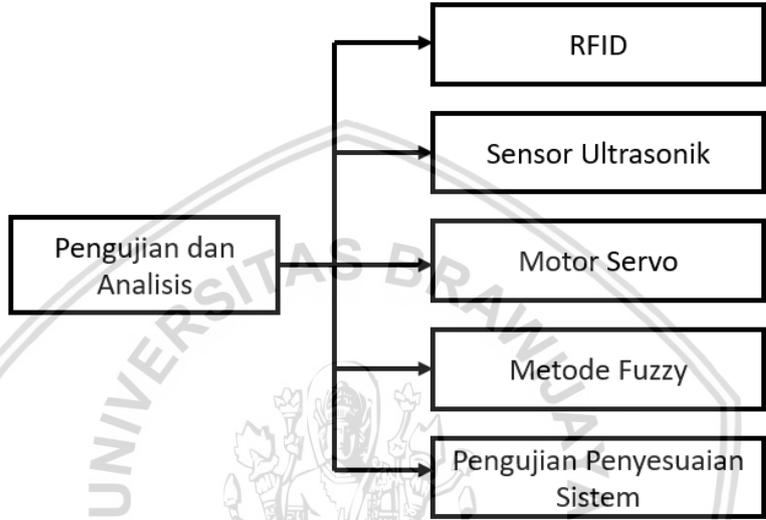
No	Kode Program
1	//Defuzzifikasi
2	void Defuz() {
3	z = ((3.3 + 3.7 + 4.3 + 4.7) * maximal[1]) + ((5 + 5.5 + 6 +
4	6.5) * maximal[2]) + ((6.5 + 7 + 7.2 + 7.3) * maximal[3]);
5	z = z / ((4 * maximal[1]) + (4 * maximal[2]) + (4 *
6	maximal[3]));
7	Serial.print("z = ");
8	Serial.println(z);
9	}

Pada Table 5.14 di atas berisi kode program defuzzifikasi. Dimana fungsi ini akan membandingkan *rule* yang berada dalam *output* tersebut, kemudian mengambil *rule* dengan pengambilan hasil inferensi min, *max* dan menggunakan metode Centroid untuk proses defuzzifikasi.



BAB 6 PENGUJIAN DAN ANALISIS

Pada bab ini akan membahas tentang tahap pengujian pada penelitian yang telah dilakukan yaitu tentang “Rancang Bangun Pengendali Pintu Gerbang Tol Dengan RFID Menggunakan Logika *Fuzzy*”. Dalam penelitian ini dilakukan 5 pengujian yaitu, pengujian akuisisi data dari RFID, Sensor Ultrasonik, pengujian fuzzifikasi, dan pengujian keseluruhan sistem. Tahapan pengujian tersebut digambarkan pada Gambar 6.1.



Gambar 6.1 Pohon Pengujian Sistem dan Analisis

6.1 Pengujian RFID

6.1.1 Tujuan

Tujuan pengujian RFID dilakukan untuk mengetahui apakah RFID mampu membaca dengan akurat *Tag* RFID yang didekatkan dengan RFID *reader*. Setiap *Tag* RFID mempunyai serial number masing – masing yang berbeda dan pada penelitian, *Tag* RFID yang ditentukan adalah sebanyak 10 buah yang diinisialisasikan sebagai 10 macam kendaraan yang memiliki panjang kendaraan yang berbeda – beda.

6.1.2 Prosedur

Prosedur yang harus dilakukan dalam pengujian RFID antara lain adalah :

1. Merancang RFID dengan Arduino Mega menggunakan kabel penghubung agar sensor dan mikrokontroler dapat terhubung dengan pin yang digunakan
2. Buka Arduino IDE kemudian tuliskan kode program agar RFID dapat membaca serial number yang terdapat pada *Tag* RFID yang akan dibaca
3. *Compile* dan *upload* program yang sudah ditulis



4. Mengamati hasil *output* dan mencatat setiap data serial number yang terbaca
5. Mengukur jarak maksimal RFID *reader* dapat membaca *Tag* RFID yang didekatkan untuk dibaca menggunakan penggaris secara manual
6. Kesimpulan

6.1.3 Hasil dan Analisis

Berikut adalah Tabel 6.1 yang menggunakan hasil dari pembacaan *Tag* RFID oleh RFID *reader* dalam membaca serial number yang dimiliki *Tag* RFID. Setelah melalui berbagai prosedur didapatkan data sebagai berikut.

Tabel 6.1 Hasil Pengujian RFID

No.	Pengujian <i>Tag</i> RFID ke-	Pengecekan Kondisi	Serial Number
1	1	Terbaca	81 181 43 230 41
2	2	Terbaca	209 221 169 137 44
3	3	Terbaca	170 69 234 182 179
4	4	Terbaca	138 202 232 182 30
5	5	Terbaca	42 151 223 182 212
6	6	Terbaca	218 59 233 182 190
7	7	Terbaca	10 230 205 182 151
8	8	Terbaca	202 32 220 182 191
9	9	Terbaca	218 210 207 182 113
10	10	Terbaca	170 254 215 182 53

Pada tabel 6.1 di atas menunjukkan bahwa pengujian RFID *reader* telah berhasil membaca 10 *Tag* RFID yang memiliki serial number yang berbeda. Dilanjutkan pengujian selanjutnya yaitu menguji jarak maksimal RFID *reader* dapat membaca *Tag* RFID ketika didekatkan. Pengujian dilakukan dengan mendekatkan *Tag* RFID dengan RFID *reader* lalu menambahkan jarak pendekatan dengan maksimal jarak 5 cm.

Tabel 6.2 Hasil Pengujian Jarak Maksimal Pembacaan RFID *reader*

No.	Jarak RFID <i>reader</i> dengan <i>Tag</i> RFID	Pengecekan Kondisi
1	0 cm	Terbaca
2	1 cm	Terbaca
3	2 cm	Tidak Terbaca
4	3 cm	Tidak Terbaca
5	4 cm	Tidak Terbaca

6	5 cm	Tidak Terbaca
---	------	---------------

Pada Tabel 6.2 di atas menunjukkan bahwa pengujian jarak maksimal RFID reader dalam membaca Tag RFID yaitu berada pada jarak 0 – 1 cm sedangkan untuk jarak yang melebihi batas tersebut sudah tidak dapat terbaca.

6.2 Pengujian Sensor Ultrasonik

6.2.1 Tujuan

Tujuan pengujian sensor Ultrasonik dilakukan untuk mengetahui apakah sensor ultrasonik mampu mengukur jarak jalan yang ada pada penelitian dengan akurat. *Output* dari sensor ultrasonik telah dikalibrasi agar dapat mengukur jarak jalan yang ada maket lalu berkurang seiring dengan adanya kendaraan yang menghalangi. Pengujian dilakukan dengan menambahkan objek didepan sensor secara berkala dan membandingkan dengan pengukuran manual menggunakan penggaris.

6.2.2 Prosedur

Prosedur yang harus dilakukan dalam pengujian sensor Ultrasonik antara lain adalah:

1. Merancang sensor Ultrasonik dengan Arduino Mega menggunakan kabel penghubung agar sensor dan mikrokontroler dapat terhubung dengan pin yang digunakan
2. Buka Arduino IDE kemudian tuliskan kode program agar sensor ultrasonik dapat mengukur jarak jalan yang ada pada maket dan mengkalibrasi *output* dari sensor agar berupa nilai jarak (cm)
3. *Compile* dan *upload* kode program yang sudah ditulis
4. Mengamati hasil *output* dan mencatat setiap data ketinggiannya
5. Menambah objek didepan sensor sebagai masukan sensor sesuai lebar jarak pada maket
6. Melakukan pengukuran manual menggunakan penggaris setiap perubahan nilai sensor
7. Kesimpulan

6.2.3 Hasil dan Analisis

Berikut adalah Tabel 6.3 yang menggunakan hasil dari pembacaan sensor Ultrasonik untuk mengukur jarak jalan pada maket, setelah melalui berbagai prosedur sehingga diperoleh data sebagai berikut.

Tabel 6.3 Hasil Pengujian Sensor Ultrasonik

No	Pengujian 1		Pengujian 2		Pengujian 3	
	Pengukuran Sensor (cm)	Pengukuran Manual	Pengukuran Sensor (cm)	Pengukuran Manual	Pengukuran Sensor (cm)	Pengukuran Manual

1	3	3	3	3	3	3
2	4	4	4	4	4	4
3	5	5	5	5	5	5
4	6	6	6	6	6	6
5	7	7	7	7	7	7
6	8	8	8	8	8	8
7	9	9	9	9	9	9
8	10	10	10	10	10	10
9	11	11	11	11	11	11
10	12	12	12	12	12	12
11	13	13	13	13	13	13
12	14	14	14	14	14	14
13	15	15	15	15	15	15
Selisih	0 cm		0 cm		0 cm	
Error	0 %		0 %		0 %	

Pada Tabel 6.3 di atas menunjukkan bahwa pengujian sensor Ultrasonik telah berhasil dilakukan. Sensor Ultrasonik yang digunakan dalam pengujian mengukur lebar jalan pada maket yang berjarak maksimal 15 cm. Dari pengujian didapatkan hasil pengukuran dengan sensor Ultrasonik memiliki selisih pengukuran sebesar 0 cm dan persentase *error* sebesar 0%.

6.3 Pengujian Motor Servo

6.3.1 Tujuan

Tujuan dilakukan pengujian servo adalah untuk mengetahui apakah servo dapat bekerja sesuai keinginan peneliti, yaitu sebagai aktuator pada penelitian juga layaknya gerbang tol bekerja.

6.3.2 Prosedur

1. Merancang servo dengan Arduino Mega menggunakan kabel penghubung agar servo dan mikrokontroler dapat terhubung sesuai dengan pin yang dibutuhkan.
2. Buka Arduino IDE dan tuliskan program untuk mengontrol servo
3. *Compile* dan *upload* kode program yang telah dituliskan
4. Amati hasil keluaran yang ditunjukkan servo dan mencatat posisi (derajat) motor servo
5. Melakukan pengukuran manual menggunakan busur derajat setiap perubahan nilai sensor
6. Kesimpulan

6.3.3 Hasil dan Analisis

Berikut adalah data hasil pengujian servo yang ditunjukkan oleh Tabel 6.4 di bawah ini.

Tabel 6.4 Hasil Pengujian Motor Servo

No	Pengujian 1		Pengujian 2		Pengujian 3	
	Posisi motor servo (derajat)	Pengukuran dengan busur derajat	Posisi motor servo (derajat)	Pengukuran dengan busur derajat	Posisi motor servo (derajat)	Pengukuran dengan busur derajat
1	0	0	0	0	0	0
2	18	20	18	20	18	20
3	36	40	36	40	36	40
4	54	56	54	56	54	56
5	72	73	72	73	72	73
6	90	90	90	90	90	90
7	108	110	108	110	108	110
8	126	128	126	128	126	128
9	144	140	144	140	144	140
10	162	155	162	155	162	155
11	180	180	180	180	180	180
Selisih	2.4 derajat		2.4 derajat		2.4 derajat	
Error	3.766 %		3.766 %		3.766 %	

Pada Tabel 6.4 di atas didapatkan hasil pengujian motor servo dimana pengujian dilakukan dengan mengamati perubahan posisi (derajat) pada motor servo yang telah diprogram dengan pengukuran posisi (derajat) yang ada pada busur derajat. Dari pengujian didapatkan selisih dari putaran motor servo dengan pengukuran manual busur derajat sebesar 2.4 derajat dan *error* masing – masing pengujian sebesar 3.766 %.

6.4 Pengujian Fungsional Metode Fuzzy

6.4.1 Tujuan

Tujuan dilakukan pengujian fungsionalitas metode *Fuzzy* adalah untuk mengetahui keberhasilan sistem dalam menentukan durasi pembukaan pintu gerbang tol berdasarkan perhitungan *Fuzzy*.

6.4.2 Prosedur

Pengujian dilakukan dengan membandingkan nilai hasil perhitungan *Fuzzy* melalui sistem dan nilai hasil perhitungan manual *Fuzzy*. Sistem dapat dikatakan berhasil apabila nilai dari perhitungan melalui sistem hasilnya sesuai dengan

perhitungan manual. Pengujian dilakukan sebanyak 10 kali dengan data yang digunakan sebagai *input Fuzzy* dimasukan manual pada kode program. Berikut adalah prosedur yang harus dilakukan dalam pengujian fungsionalitas *Fuzzy*.

1. Menyambungkan Arduino ke PC.
2. Buka Arduino IDE, kemudian tuliskan program agar sistem berjalan menggunakan metode *Fuzzy*.
3. *Compile* dan *upload* kode program yang telah dituliskan.
4. Buka serial monitor pada Arduino IDE untuk mengamati hasil pengolahan *Fuzzy*
5. Memasukkan nilai *input* sebanyak 10 kali dan mengamati hasil yang mempresentasikan durasi pembukaan pintu gerbang tol.
6. Melakukan perhitungan *Fuzzy* secara manual yang akan digunakan untuk membandingkan untuk perhitungan dari sistem.
7. Kesimpulan.

6.4.3 Hasil dan Analisis

Berikut adalah data hasil pengolahan *Fuzzy* dari sistem dan secara manual. Data tersebut ditunjukkan pada Tabel 6.5 berikut

Tabel 6.5 Hasil Keluaran Logika *Fuzzy*

No.	Masukkan Nilai RFID	Masukkan Nilai Sensor Ultrasonik	<i>Output</i> Perhitungan Sistem	<i>Output</i> Perhitungan Manual
1	17 cm	55.5 cm	5.75 detik	5.75 detik
2	18 cm	51 cm	4 detik	4 detik
3	20 cm	46.5 cm	4 detik	4 detik
4	22 cm	42 cm	4.7 detik	4.7 detik
5	29 cm	60 cm	7.25 detik	7.25 detik
6	33 cm	42 cm	5.75 detik	5.75 detik
7	37 cm	55.5 cm	6.35 detik	6.35 detik
8	40 cm	33 cm	6.59 detik	6.59 detik
9	43 cm	60 cm	7.25 detik	7.25 detik
10	43 cm	37.5 cm	5.75 detik	5.75 detik

Berdasarkan Tabel 6.5 dapat dilihat keluaran dari kedua pengujian, yaitu keluaran dari sistem dan hasil dari perhitungan manual. Hasil dari keluaran sistem memiliki nilai yang sama dengan hasil perhitungan manual. Dengan demikian sistem sudah sesuai dalam melakukan perhitungan *Fuzzy*. dengan menggunakan 10 masukan yang memiliki kondisi yang berbeda, sistem dapat memberikan hasil keluaran yang sama dengan perhitungan manual. Berikut

adalah pengujian secara manual yang dilakukan berdasarkan rumus perhitungan *Fuzzy*. Diambil data dari sampel pada tabel 6.4 percobaan ke 1.

6.4.4 Perhitungan Manual

Contoh Kasus:

Misal *input* yang didapat sebagai berikut:

- Panjang kendaraan : 17 cm
 - Kondisi kepadatan : 55.5 cm
1. Fuzzifikasi, menghitung derajat keanggotaan masing – masing variabel.

Panjang Kendaraan

$$\text{Gol. 1} \quad [17] = 1$$

$$\text{Gol. 2} \quad [17] = 0$$

$$\text{Gol. 3} \quad [17] = 0$$

Kondisi Kepadatan

$$\text{Sepi} \quad [55.5] = 0$$

$$\text{Sedang} \quad [55.5] = 0$$

$$\text{Padat} \quad [55.5] = 1$$

2. Fungsi Implikasi (MIN)

$$\begin{aligned} R1 \text{ MIN} &= \text{MIN} (\mu_{\text{RFIDGol.1}} \cap \mu_{\text{ultrasonikSepi}}) \\ &= \text{MIN} (1 \cap 0) \\ &= \text{MIN} 0 \end{aligned}$$

$$\begin{aligned} R2 \text{ MIN} &= \text{MIN} (\mu_{\text{RFIDGol.1}} \cap \mu_{\text{ultrasonikSedang}}) \\ &= \text{MIN} (1 \cap 0) \\ &= \text{MIN} 0 \end{aligned}$$

$$\begin{aligned} R3 \text{ MIN} &= \text{MIN} (\mu_{\text{RFIDGol.1}} \cap \mu_{\text{ultrasonikPadat}}) \\ &= \text{MIN} (1 \cap 1) \\ &= \text{MIN} 1 \end{aligned}$$

$$\begin{aligned} R4 \text{ MIN} &= \text{MIN} (\mu_{\text{RFIDGol.2}} \cap \mu_{\text{ultrasonikSepi}}) \\ &= \text{MIN} (0 \cap 0) \\ &= \text{MIN} 0 \end{aligned}$$

$$\begin{aligned} R5 \text{ MIN} &= \text{MIN} (\mu_{\text{RFIDGol.2}} \cap \mu_{\text{ultrasonikSedang}}) \\ &= \text{MIN} (0 \cap 0) \\ &= \text{MIN} 0 \end{aligned}$$

$$\begin{aligned} R6 \text{ MIN} &= \text{MIN} (\mu_{\text{RFIDGol.2}} \cap \mu_{\text{ultrasonikPadat}}) \\ &= \text{MIN} (0 \cap 1) \\ &= \text{MIN} 0 \end{aligned}$$

$$\begin{aligned} R7 \text{ MIN} &= \text{MIN} (\mu_{\text{RFIDGol.3}} \cap \mu_{\text{ultrasonikSepi}}) \\ &= \text{MIN} (0 \cap 0) \\ &= \text{MIN} 0 \end{aligned}$$

$$\begin{aligned} R8 \text{ MIN} &= \text{MIN} (\mu_{\text{RFIDGol.3}} \cap \mu_{\text{ultrasonikSedang}}) \\ &= \text{MIN} (0 \cap 0) \\ &= \text{MIN} 0 \end{aligned}$$

$$\begin{aligned} R9 \text{ MIN} &= \text{MIN} (\mu_{\text{RFIDGol.3}} \cap \mu_{\text{ultrasonikPadat}}) \\ &= \text{MIN} (0 \cap 1) \\ &= \text{MIN} 0 \end{aligned}$$

3. Komposisi Aturan (MAX)

$$\begin{aligned} \text{Pembukaan pintu cepat} &= \text{MAX} (R1, R2, R4) \\ &= \text{MAX} (0 ; 0 ; 0) \\ &= \text{MAX} 0 \end{aligned}$$

$$\begin{aligned} \text{Pembukaan pintu sedang} &= \text{MAX} (R3, R5, R7) \\ &= \text{MAX} (1 ; 0 ; 0) \\ &= \text{MAX} 1 \end{aligned}$$

$$\begin{aligned} \text{Pembukaan pintu lambat} &= \text{MAX} (R6, R8, R9) \\ &= \text{MAX} (0 ; 0 ; 0) \\ &= \text{MAX} 0 \end{aligned}$$

4. Defuzzifikasi

$$Z = \frac{((3,3+3,7+4,3+4,7) 0 + (5+5,5+6+6,5) 1 + (6,7+7,2+7,3+7,8) 0)}{((4*0) + (4*1) + (4*0))} = \frac{26}{4} = 5,75$$

Dari pengujian perhitungan manual di atas, didapatkan nilai defuzzifikasi sebesar 5,75 yang artinya durasi pembukaan pintu gerbang tol sebesar 5,75 detik berdasarkan panjang kendaraan sebesar 17 cm dan kondisi kepadatan sebesar 55,5 cm.



6.5 Pengujian Keseluruhan Sistem

6.5.1 Tujuan

Tujuan dilakukan pengujian keseluruhan sistem adalah untuk mengetahui akurasi sistem dalam menentukan durasi pembukaan pintu gerbang tol secara keseluruhan.

6.5.2 Prosedur

Untuk melakukan pengujian keseluruhan sistem, prosedur yang harus dilakukan adalah sebagai berikut.

1. Merangkai Sensor Ultrasonik, RFID, Arduino Mega 2560, motor servo, dan menggunakan kabel penghubung agar setiap komponen terhubung satu sama lain dan sesuai dengan pin yang digunakan.
2. Buka Arduino IDE, kemudian tuliskan program agar sistem berjalan sesuai dengan perancangan yang telah dibuat.
3. *Compile* dan *upload* kode program yang telah dituliskan.
4. Amati saat keseluruhan sistem berjalan.
5. Membandingkan *output* sistem dengan 10 data sampel
6. Kesimpulan.

6.5.3 Hasil dan Analisis

Berikut adalah data hasil dari pengujian keseluruhan sistem yang ditunjukkan pada Tabel 6.6 berikut

Tabel 6.6 Pengujian Keseluruhan Sistem

No.	Nilai Panjang Kendaraan (EEPROM)	Nilai Panjang Kendaraan (Penggaris)	Nilai Kondisi Kepadatan (Maket)	Nilai Kondisi Kepadatan (Penggaris)	Output Sistem	Output Data Sampel
1	17 cm	17.9 cm	55.5 cm	55.5 cm	5.75 detik	5.25 detik
2	18 cm	18.5 cm	51 cm	51 cm	4 detik	4.13 detik
3	20 cm	20.575 cm	46.5 cm	46.5 cm	4 detik	4.27 detik
4	22 cm	22.53 cm	42 cm	42 cm	4.7 detik	4.55 detik
5	29 cm	29.8 cm	60 cm	60 cm	7.25 detik	7.01 detik
6	33 cm	33.85 cm	42 cm	42 cm	5.75 detik	5.67 detik
7	35 cm	35.075 cm	55.5 cm	55.5 cm	6.35 detik	6.01 detik
8	37 cm	37.675 cm	33 cm	33 cm	6.59 detik	6.77 detik
9	40 cm	40.775 cm	60 cm	60 cm	7.25 detik	6.89 detik
10	42 cm	43.525 cm	37.5 cm	37.5 cm	5.75 detik	5.81 detik

Berdasarkan data hasil pengujian keseluruhan sistem yang ditunjukkan pada Tabel 6.6 didapatkan data untuk menghitung persentase *error* yang ditunjukkan pada Tabel 6.7 berikut.

Tabel 6.7 Persentase *Error*

<i>Output Data Sampel</i>	<i>Output Sistem</i>	Selisih	Persentase <i>Error</i> (%)
5.25	5.75	0.50	9.52
4.13	4	0.13	3.38
4.27	4	0.27	6.32
4.55	4.7	0.15	3.29
7.01	7.25	0.24	3.42
5.67	5.75	0.08	1.41
6.01	6.35	0.34	5.65
6.77	6.59	0.18	2.65
6.89	7.25	0.36	5.22
5.81	5.75	0.06	1.03
Rata - rata		0.232	4.189 %

Tabel 6.7 merupakan tabel rata – rata persentase *error* dari perhitungan logika *Fuzzy* yang menghasilkan data keluaran dari sistem untuk dibandingkan dengan data sampel yang telah diuji dengan responden sistem. Berdasarkan perhitungan persentase *error* sistem, secara keseluruhan perhitungan logika *Fuzzy* untuk durasi pembukaan pintu gerbang tol memiliki persentase *error* sebesar 4.189 %.

Secara umum berdasarkan 10 skenario yang telah diuji masing – masing, terjadi selisih pada data sampel dan perhitungan oleh sistem pada perhitungan logika *Fuzzy* yang disebabkan oleh perbedaan pembulatan oleh sistem juga perbedaan data panjang kendaraan yang digunakan sistem dengan yang digunakan data sampel, tetapi pada perbandingan *output* sistem dengan manual, seluruh perhitungan sistem sesuai dengan perhitungan manual.

BAB 7 PENUTUP

Pada bab ini membahas tentang kesimpulan yang diperoleh dari penelitian yang telah dilakukan yaitu “Rancang Bangun Pengendali Pintu Gerbang Tol dengan RFID Menggunakan Logika *Fuzzy*” dan saran-saran untuk pengembangan skripsi atau penelitian lebih lanjut.

7.1 Kesimpulan

Kesimpulan yang dapat diperoleh dari pengujian dan analisa yang telah dilakukan dalam penelitian ini adalah sebagai berikut.

1. Proses perancangan rancang bangun sistem pintu gerbang tol dengan RFID dengan mikrokontroller adalah dengan menyambungkan sensor Ultrasonik, RFID *reader*, dan motor servo menjadi sebuah sistem yang saling terhubung. Proses akuisisi data pada sensor Ultrasonik berupa pengukuran lebar jalan yang ada pada maket untuk mendeteksi adanya objek jika nilai pembacaan lebar jalan berubah dengan satuan cm (centimeter). Proses akuisisi data pada RFID meliputi pembacaan *Tag* RFID yang berisi nilai panjang kendaraan oleh RFID *reader* dimana data panjang kendaraan tersebut disimpan pada EEPROM Arduino. Sedangkan motor servo difungsikan sebagai aktuator dalam sistem. Keseluruhan sensor dan aktuator yang digunakan pada penelitian ini dihubungkan ke Arduino Mega sebagai pengendali sistem.
2. Implementasi logika *Fuzzy* Mamdani pada rancang bangun sistem pengendali gerbang tol dengan RFID meliputi 4 proses yaitu fuzzifikasi, pembentukan basis pengetahuan *Fuzzy (rule)*, fungsi implikasi MIN dan komposisi antar rule menggunakan MAX, dan terakhir pengambilan keputusan defuzzifikasi menggunakan centroid. Sistem gerbang tol berdasarkan kondisi kepadatan dan panjang kendaraan berhasil dirancang dan diimplementasikan. Terbukti sistem dapat membaca lebar jalan pada maket serta mengambil data panjang kendaraan pada EEPROM Arduino dan mengolah data tersebut menggunakan metode *Fuzzy*.
3. Setelah sistem diimplementasikan, dilakukan pengujian dengan skenario *input* yang berbeda - beda, hasil perhitungan sistem kemudian dibandingkan dengan data sampel yang didapat dari beberapa percobaan, secara keseluruhan perhitungan logika *Fuzzy* pada durasi pembukaan pintu gerbang tol memiliki rata – rata selisih 0,232 dan memiliki rata – rata persentase *error* sebesar 4,189%.

7.2 Saran

Berdasarkan hasil penelitian perancangan dan pengujian rancang bangun pengendali gerbang tol dengan RFID menggunakan logika *Fuzzy*, saran yang bisa diberikan oleh peneliti untuk pengembangan sistem selanjutnya adalah:



1. Sebagai pengembangan lebih lanjut diharapkan ada penambahan parameter yang lebih luas serta penggunaan RFID yang lebih canggih.
2. Sistem menggunakan media penyimpanan selain EEPROM Arduino agar kapasitas penyimpanan lebih besar.
3. Sistem menggunakan metode lain selain *Fuzzy* untuk mengetahui metode mana yang lebih akurat saat diterapkan pada sistem.
4. Sistem diimplementasikan pada kondisi *real*



DAFTAR PUSTAKA

- Ajie, S., 2015. *membaca dan menulis data ke EEPROM arduino*. [Online] Available at: <http://saptaji.com/2015/02/02/membaca-dan-menulis-data-ke-eprom-arduino/> [Accessed Monday Januari 2018].
- Anne Marie Helmenstine, P., 2018. *How to Calculate Percent Error*. [Online] Available at: <https://www.thoughtco.com/how-to-calculate-percent-error-609584> [Accessed Wednesday August 2018].
- Arduino, 2017. *arduino-mega-2560*. [Online] Available at: <http://www.arduino.org/products/boards/arduino-mega-2560> [Accessed 19 July 2017].
- Ayuningtias, L. P., 2017. Jurnal Teknik Informatika. *Analisa Perbandingan Logic Fuzzy Metode Tsukamoto, Sugeno, dan Mamdani (Studi Kasus : Prediksi Jumlah Pendaftar Mahasiswa Baru Fakultas Sains dan Teknologi Universitas Islam Negeri Sunan Gunung Djati Bandung)*.
- Components, 2017. *Servo Motor SG-90*. [Online] Available at: <https://components101.com/servo-motor-basics-pinout-datasheet>
- Dardak, A. H., 2009. *Geometri Jalan Bebas Hambatan Untuk Jalan Tol*. s.l.:Departemen Pekerjaan Umum Direktorat Jendral Bina Marga.
- e-Gizmo, 2017. *Datasheet RC522 Card Reader and tag*. s.l.:e-Gizmo Mechanic Control.
- Elecfreaks, 2005. *Datasheet Ultrasonic Distance Sensor*. s.l.:Parallax.
- Elektro, Z., 2014. *Motor Servo*. [Online] Available at: <http://zoniaelektro.net/motor-servo/> [Accessed 5 February 2017].
- Eridani, D., 2011. Simulasi Gerbang Tol Menggunakan RFID. *Makalah Seminar Tugas Akhir*, p. 2.
- Geddes, M., 2016. *Arduino Project Handbook : 25 Practical Project To Get You Started*. San Francisco: No Starch Press.
- Kece, A., 2017. *Nyebarilmu.com*. [Online] Available at: <https://www.nyebarilmu.com/tutorial-arduino-mengakses-module-rfid-rc522/> [Accessed Wednesday January 2018].
- Kusumadewi, S., 2010. *Aplikasi Logika Fuzzy*. 2nd ed. Yogyakarta: Graha Ilmu.
- Monk, S., 2016. *adafruit-raspberry-pi-lesson-8-using-a-servo-motor*. [Online] Available at: <https://learn.adafruit.com/adafruit-raspberry-pi-lesson-8-using-a->

[servo-motor?view=all](#)

[Accessed 19 July 2017].

Prabowo, D., 2018. *kompascom*. [Online] Available at: <https://properti.kompas.com/read/2018/05/03/144244121/begini-kemudahan-gpn-untuk-transaksi-tol>

[Accessed Friday June 2018].

Prasetya, A. E., 2017. Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer. *Rancang Bangun Pengendali Pintu Air Sungai dengan Menggunakan Logika Fuzzy dan Simple Additive Weigjting*.

R, M. A., 2017. *detikcom*. [Online] Available at: <https://news.detik.com/berita/3505640/seluruh-gerbang-tol-tak-layani-transaksi-tunai-mulai-oktober-2017>

[Accessed Tuesday September 2017].

Santoso, H., 2015. *Cara Kerja Sensor Ultrasonik, Rangkaian, & Aplikasinya*. [Online]

Available at: <http://www.elangsakti.com/2015/05/sensor-ultrasonik.html>

[Accessed Monday September 2017].

Sighlia, P., 2016. International Journal of Engineering Trends and Technology (IJETT). *IoT Based RFID Gate Automation System*.

Sutikno, I. W., 2014. Perbandingan Metoda Deffuzifikasi Sistem Kendali Logika Fuzzy. *Jurnal Masyarakat Informatika*, Volume 2.

T.Sutojo, S., 2011. *Kecerdasan Buatan*. Yogyakarta: C.V ANDI OFFSET.

Tjahyadi, C., 2017. *Sensor Ultrasonik HC-SR04*. [Online] Available at: <http://christianto.tjahyadi.com/belajar-mikrokontroler/sensor-ultrasonik-hc-sr04.html>

[Accessed Wednesday September 2017].

Vicky Primandani, T. W. W., 2012. Purwarupa Sistem Pembayaran Retribusi Jalan Tol Berbasis RFID. *IJEIS*, Volume 2, pp. 11 - 20.