

**DESAIN SISTEM DETEKSI PENCURIAN SEPEDA MOTOR DENGAN
MENGUNAKAN GPS DAN LOGIKA FUZZY METODE MAMDANI**

SKRIPSI

TEKNIK ELEKTRO KONSENTRASI TEKNIK KONTROL

Ditujukan untuk memenuhi persyaratan
memperoleh gelar Sarjana Teknik



GHOZY MAHENDRA

NIM. 175060307111027

UNIVERSITAS BRAWIJAYA

FAKULTAS TEKNIK

JURUSAN TEKNIK ELEKTRO

MALANG

2021



LEMBAR PENGESAHAN

**DESAIN SISTEM DETEKSI PENCURIAN SEPEDA MOTOR DENGAN
MENGUNAKAN GPS DAN LOGIKA FUZZY DENGAN METODE MAMDANI**

SKRIPSI

TEKNIK ELEKTRO KONSENTRASI TEKNIK KONTROL

Ditujukan untuk memenuhi persyaratan
memperoleh gelar Sarjana Teknik



Ghozy Mahendra

NIM. 175060307111027

Skripsi ini telah diperiksa dan disetujui oleh dosen
pembimbing pada tanggal 28 Juli 2021

Dosen Pembimbing 1

Dr. Ir. Sholeh Hadi Pramono, MS.
NIP. 195807281987011001

Dosen Pembimbing 2

Goegoes Dwi Nusantoro, ST., MT.
NIP. 197110132006041001

Mengetahui

Ketua Jurusan Teknik Elektro



Muhammad Aziz Muslim, S.T., M.T., Ph.D.
NIP. 19741203 200012 1 001



JUDUL SKRIPSI:

DESAIN SISTEM DETEKSI PENCURIAN SEPEDA MOTOR DENGAN
MENGUNAKAN GPS DAN LOGIKA FUZZY METODE MAMDANI

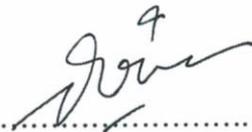
Nama Mahasiswa : Ghozy Mahendra
NIM : 175060307111027
Program Studi : Teknik Elektro
Konsentrasi : Teknik Kontrol

KOMISI PEMBIMBING

Ketua : Dr. Ir. Sholeh Hadi Pramono, MS.

.....

Anggota : Goegoes Dwi Nusantoro, ST., MT.

.....

TIM DOSEN PENGUJI:

Dosen Penguji I : Dr. Ir. Bambang Siswoyo, MT.

disetujui tanggal 28 Juli 2021 

Dosen Penguji II : M. Aziz Muslim, S.T., M.T., Ph.D.

disetujui tanggal 28 Juli 2021 

Tanggal Ujian : 19 Juli 2021

SK Penguji : No. 1073 Tahun 2021

LEMBAR ORISINALITAS SKRIPSI

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya dan berdasarkan hasil penelusuran berbagai karya ilmiah, gagasan dan masalah ilmiah yang diteliti dan diulas di dalam Naskah Skripsi ini adalah asli dari pemikiran saya, 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 dikutip dalam naskah ini dan disebutkan dalam sumber kutipan dan daftar pustaka.

Apabila ternyata di dalam naskah Skripsi ini dapat dibuktikan terdapat unsur-unsur jiplakan, saya bersedia Skripsi dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, pasal 25 ayat 2 dan pasal 70).

Malang, 24 Maret 2021

Mahasiswa,

Ghozy Mahendra
NIM. 175060307111027





RINGKASAN

Ghozy Mahendra, Jurusan Teknik Elektro, Fakultas Teknik, Universitas Brawijaya, Mei 2021, *Desain Sistem Deteksi Pencurian Sepeda Motor Menggunakan GPS Dan Logika Fuzzy Metode Mamdani*, Dosen Pembimbing 1: Dr. Ir. Sholeh Hadi Pramono, MS., Dosen Pembimbing 2: Goegoes Dwi Nusantoro, ST., MT.

GPS (*Global positioning system*) pada alat sistem pelacakan dan deteksi sepeda motor dari pencurian berfungsi untuk mendeteksi dan melacak keberadaan posisi sepeda motor saat terjadi perpindahan posisi awal atau posisi pada saat motor dalam keadaan parkir. Pendeteksian dan pelacakan sangat berguna dan bermanfaat untuk pengguna alat ini serta mempermudah pengguna untuk mengetahui letak atau posisi saat sepeda motor berpindah dari posisi awal. *Buzzer* digunakan sebagai *output* dari jarak yang dihasilkan oleh GPS

Digunakan logika *fuzzy* untuk mempermudah mengetahui besarnya *output* berupa bunyi *buzzer* pada saat motor berpindah letak dari posisi awal. Kontrol logika *fuzzy* yang digunakan adalah metode mamdani dengan beberapa tahap, yaitu fuzzifikasi, kaidah atur (*Rule-Base*) dan inferensi, dan defuzzifikasi. Metode fuzzifikasi yang digunakan adalah metode *min-max* sedangkan pada defuzzifikasi menggunakan metode *centroid*. Hasil dari respon secara keseluruhan sistem adalah semakin jauh letak motor dari posisi awal maka jeda bunyi *buzzer* semakin dekat

Kata Kunci: GPS, Logika *Fuzzy*, Sistem Deteksi Pencurian Sepeda Motor



SUMMARY

Ghozy Mahendra, *Department of Electrical Engineering, Faculty of Engineering, University of Brawijaya, May 2021 Motorcycle Theft Detection System Design Using GPS And Fuzzy Logic Mamdani Method, Academic Advisor 1: Dr. Ir. Sholeh Hadi Pramono, MS., Academic Advisor 2: Goegoes Dwi Nusantoro, ST., MT.*

GPS (Global positioning system) on a motorcycle tracking and detection system tool from theft serves to detect and track the whereabouts of the motorcycle's position when there is a change in the initial position or position when the motorcycle is parked. Detection and tracking are very useful and useful for users of this tool and make it easier for users to find out where or when the motorcycle moves from its initial position. The buzzer is used as the output of the distance generated by the GPS

Fuzzy logic is used to make it easier to find out the amount of output in the form of a buzzer sound when the motor moves from its initial position. The fuzzy logic control used is the Mamdani method with several stages, namely fuzzification, Rule-Base and inference, and defuzzification. The fuzzification method used is the min-max method while the defuzzification method uses the centroid method. The result of the overall response of the system is that the farther the motor is from the initial position, the closer the buzzer sound pauses

Keyword: *fuzzy logic, GPS, Motorcycle Detection System from Theft*



DAFTAR ISI

PENGANTAR.....	i
DAFTAR ISI.....	iii
DAFTAR GAMBAR.....	v
DAFTAR TABEL.....	vii
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	2
1.3 Batasan Penelitian.....	2
1.4 Tujuan.....	3
1.5 Manfaat.....	3
BAB II TINJAUAN PUSTAKA.....	5
2.1 NodeMCU ESP8266.....	5
2.2 GPS (<i>Global Positioning System</i>).....	6
2.3 <i>Buzzer</i>	9
2.4 <i>Driver Motor L298N</i>	10
2.5 <i>PWM (Pulse Width Modulation)</i>	11
2.6 <i>Logika Fuzzy</i>	11
2.7 <i>Penalaran Logika Fuzzy Metode Mamdani</i>	15
BAB III METODE PENELITIAN.....	19
3.1 <i>Penentuan Jenis dan Cara Pengambilan Data</i>	19
3.2 <i>Perancangan Blok Diagram Sistem</i>	20
3.3 <i>Spesifikasi Desain</i>	20
3.4 <i>Pembuatan Perangkat Keras</i>	21
3.5 <i>Pembuatan Perangkat Lunak</i>	23
BAB IV DESAIN DAN PERANCANGAN.....	25
4.1 <i>Perancangan Aplikasi Sistem</i>	25
4.2 <i>Perancangan Logika Fuzzy</i>	26
4.3 <i>Pengujian Sistem</i>	29
BAB V HASIL DAN PEMBAHASAN.....	31
5.1 <i>Pengujian GPS (Global Positioning System)</i>	31
5.2 <i>Pengujian Kontrol Logika Fuzzy</i>	31

5.3 Desain *Fuzzy* Pada Matlab 32

BAB VI PENUTUP..... 41

6.1 Kesimpulan 41

6.2 Saran 41

DAFTAR PUSTAKA 43



DAFTAR GAMBAR

Gambar 2.1	NodeMCU ESP8266.....	6
Gambar 2.2	<i>Global Positioning System</i> (GPS).....	8
Gambar 2.3	<i>Buzzer</i>	9
Gambar 2.4	<i>Driver Motor L298N</i>	10
Gambar 2.5	Representasi Linier Naik.....	13
Gambar 2.6	Representasi Linier Turun.....	13
Gambar 2.7	Representasi Kurva Segitiga.....	14
Gambar 2.8	Representasi Kurva Trapesium.....	14
Gambar 3.1	Langkah Pengambilan Data.....	19
Gambar 3.2	Blok Diagram Sistem.....	20
Gambar 3.3	Skema Pembuatan Perangkat Keras.....	21
Gambar 3.4	NodeMCU ESP8266.....	21
Gambar 3.5	<i>Global Positioning System</i> (GPS).....	22
Gambar 3.6	<i>Driver Motor L298N</i>	22
Gambar 3.7	<i>Buzzer</i>	23
Gambar 3.8	<i>Smartphone</i>	23
Gambar 3.9	Flowchart Sistem Kerja GPS.....	24
Gambar 4.1	Rancangan Aplikasi Antarmuka Pada <i>Smartphone</i>	25
Gambar 4.2	Rancangan Aplikasi Antarmuka Pada Laptop.....	26
Gambar 4.3	Himpunan Keanggotaan <i>Error</i>	27
Gambar 4.4	Himpunan Keanggotaan Bit <i>Buzzer</i>	29
Gambar 5.1	Desain <i>Fuzzy</i> Matlab.....	33
Gambar 5.2	Tampilan <i>Membership Function Error</i>	33
Gambar 5.3	Tampilan <i>Membership Function Output Bit Buzzer</i>	34
Gambar 5.4	Tampilan <i>Rule Editor</i>	34
Gambar 5.5	Tampilan <i>Rule Viewer</i> Dengan <i>Error 5</i>	35
Gambar 5.6	Tampilan <i>Rule Viewer</i> Dengan <i>Error 10</i>	35
Gambar 5.7	Tampilan <i>Rule Viewer</i> Dengan <i>Error 15</i>	36
Gambar 5.8	Tampilan <i>Rule Viewer</i> Dengan <i>Error 20</i>	36
Gambar 5.9	Tampilan <i>Rule Viewer</i> Dengan <i>Error 25</i>	37
Gambar 5.10	Tampilan <i>Rule Viewer</i> Dengan <i>Error 30</i>	37

Gambar 5.11 Tampilan *Rule Viewer* Dengan *Error 35*38

Gambar 5.12 Tampilan *Rule Viewer* Dengan *Error 40*38

Gambar 5.13 Tampilan *Rule Viewer* Dengan *Error 45*39

Gambar 5.14 Tampilan *Rule Viewer* Dengan *Error 50*39

Gambar 5.15 *Surface Output* Desain *Fuzzy Matlab*40

Gambar 5.16 Respon Sistem Jarak GPS Dengan *Output Bit Buzzer*.....40



DAFTAR TABEL

Tabel 4.1 Variabel dan Semesta Pembicaraan *Input* dan *Output*27

Tabel 4.2 Fuzzifikasi *Input Error*28

Tabel 4.3 Fuzzifikasi *Output Bit Buzzer*28

Tabel 5.1 Data Pengujian GPS31

Tabel 5.2 Data Hasil Pengujian Proses *Fuzzy*32



Halaman ini sengaja dikosongkan



BAB I PENDAHULUAN

1.1 Latar Belakang

Tahun 2018, tercatat total jumlah kendaraan bermotor sebesar 136.542.034 dan ditahun tersebut jumlah kasus pencurian sepeda motor mencapai 27.731. Jumlah tersebut sudah menurun apabila dibandingkan dengan tahun 2014, ketika jumlah mobil penumpang dan sepeda motor masih berjumlah 105.575.278, dan kasus curanmor sebanyak 42.165. Akan tetapi jumlah kasus pencurian sepeda motor cenderung masih tinggi jika dibandingkan dengan kasus pencurian sepeda motor di negara lain. Di kota Malang sendiri, di tahun 2017 terdapat 892 pelaporan kasus pencurian sepeda motor, dan hanya terselesaikan sebanyak 309 kasus, sedangkan di tahun 2018 terjadi penurunan kasus pencurian sepeda motor, terdapat 264 kasus pencurian sepeda motor, dan hanya dapat diselesaikan sebanyak 160 kasus. Walaupun terjadi penurunan dari tahun ke tahun, tetap saja kasus pencurian sepeda motor ini masih banyak terjadi di sekitar kita.

Di era teknologi saat ini, perkembangan alat-alat elektronik, sistem keamanan, kecerdasan buatan, dan program-program komputer yang canggih sudah merambah ke semua bidang. Hampir segala bidang pekerjaan membutuhkan teknologi yang terus diperbaharui demi mempermudah pekerjaan yang dilakukan. Selain untuk mempermudah pekerjaan yang dilakukan, perkembangan teknologi saat ini juga bisa kita rasakan di masalah keamanan. Banyak alat yang dapat membantu manusia untuk melakukan proteksi/ mengamankan suatu hal yang diinginkan.. Di lain pihak juga berkembang suatu teknologi kontrol yang tidak lagi memakai cara konvensional untuk mendapatkan suatu hasil yang diinginkan melalui persamaan matematika. Tetapi dengan menerapkan suatu sistem kemampuan manusia untuk mengendalikan sesuatu, yaitu dalam bentuk aturan-aturan jika-maka (*If-Then Rules*) sehingga proses pengendalian akan mengikuti pendekatan secara linguistik, sistem ini disebut dengan sistem kendali logika *fuzzy*, yang mana sistem kendali logika *fuzzy* merupakan sistem kendali yang tidak memiliki ketergantungan pada variabel-variabel proses kendali. Telah banyak aplikasi sistem kontrol dengan menggunakan sistem *fuzzy*, karena proses kendali ini relatif mudah dan fleksibel dirancang dengan tidak melibatkan model matematis yang rumit dari sistem

yang akan dikendalikan. Pengontrolan dengan sistem kendali logika *fuzzy* memiliki keuntungan dalam mengatasi permasalahan pada pengontrolan non linier dan adaptif.

Dalam implementasinya, tugas akhir ini akan menggunakan GPS untuk dapat mendeteksi dan melacak motor saat dicuri. Pengguna dapat mengirim perintah untuk melacak motor dengan menggunakan antarmuka yang terdapat pada *smartphone* yang terhubung dengan internet dan dengan mengklik ikon “Track” maka motor dapat terlacak keberadaannya.

1.2 Rumusan Masalah

GPS mampu bekerja untuk mendeteksi dan melacak keberadaan atau posisi motor saat motor berpindah tempat. GPS bekerja berdasarkan sinyal dari satelit yang diterima oleh alat penerima (*receiver*) di permukaan, dimana *GPS receiver* ini akan mengumpulkan informasi dari satelit GPS. Sebuah *GPS receiver* harus mengunci sinyal minimal tiga satelit untuk menghitung posisi 2D (*latitude* dan *longitude*) dan *track* pergerakan. Dalam skripsi ini komponen tersebut akan diintegrasikan untuk deteksi pengamanan sepeda motor dari pencurian. Proses integrasi dilakukan dengan menggunakan mikrokontroler yang dikendalikan dengan menggunakan algoritma Logika *Fuzzy* Metode Mamdani. Berdasarkan permasalahan ini dan penggunaan algoritma Logika *Fuzzy* Metode Mamdani sebagai solusi masalah, maka rumusan masalah dapat dibuat seperti berikut :

1. Bagaimana mengontrol jarak dan jeda bunyi *buzzer* menggunakan logika *fuzzy* dengan mikrokontroler NodeMCU ESP8266 ?
2. Bagaimana respon setelah menggunakan logika *fuzzy* metode mamdani pada GPS dengan *output* jeda bunyi *Buzzer* ?

1.3 Batasan Penelitian

Adapun batasan-batasan masalah pada penelitian ini adalah sebagai berikut:

1. Kendaraan bermotor yang digunakan dalam penelitian ini adalah sepeda motor merk Yamaha Aerox
2. Mikrokontroler yang digunakan adalah NodeMCU ESP8266 dan GPS digunakan sebagai data masukan
3. Indikasi gejala pencurian motor dibatasi pada saat motor bergerak dan hidup

4. Mikrokontroler diprogram dengan Arduino IDE.
5. Pengujian dilakukan dengan melihat keberhasilan alat mendeteksi gejala pencurian motor dan pengiriman notifikasi serta posisi motor ke ponsel pengguna.
6. Alat tidak bekerja jika tidak dinyalakan terlebih dahulu dari aplikasi pada ponsel pengguna.

1.4 Tujuan

Tujuan dari penelitian ini adalah merancang dan membuat sistem deteksi pencurian sepeda motor dan mendesain logika *fuzzy* pada sistem deteksi pencurian menggunakan GPS yang terhubung dengan mikrokontroler NodeMCU ESP8266

1.5 Manfaat

Manfaat dari perancangan sistem ini adalah sebagai berikut:

1. Mengurangi angka pencurian motor.
2. Mengetahui logika *fuzzy* dapat digunakan sebagai sistem kendali pada GPS yang terhubung dengan mikrokontroler NodeMCU ESP8266
3. Mengetahui logika *fuzzy* dapat mengendalikan bunyi buzzer sebagai output

Halaman ini sengaja dikosongkan



BAB II TINJAUAN PUSTAKA

Tinjauan pustaka berisi tinjauan bahan bacaan yang bersifat akademik untuk menyelesaikan permasalahan. Bahan bacaan yang digunakan didasarkan pada referensi-referensi dan hasil penelitian yang telah diuji kebenarannya. Teori yang dibahas adalah tentang karakteristik dan standardisasi dari komponen-komponen yang digunakan pada sistem ini, penjelasan tentang PWM (*Pulse Width Modulation*), penjelasan dan penalaran logika *fuzzy*.

2.1 NodeMCU ESP8266

Mikrokontroler adalah sebuah sistem komputer yang seluruh atau sebagian besar elemennya dikemas dalam satu *chip IC*, sehingga sering disebut *single chip microcomputer*. Mikrokontroler merupakan sistem komputer yang mempunyai salah satu atau beberapa tugas yang sangat spesifik. (Chamim, (2012). Dalam tugas akhir ini akan digunakan mikrokontroler NodeMCU ESP8266.

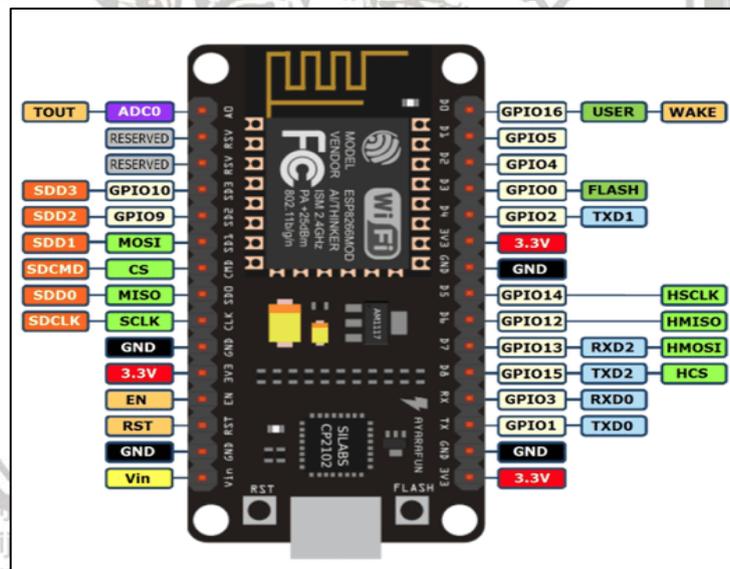
NodeMCU merupakan sebuah *open source platform IoT* dan pengembangan kit yang menggunakan bahasa pemrograman Lua untuk membantu dalam membuat *prototype* produk IoT atau bisa dengan memakai *sketch* dengan arduino IDE. Pengembangan kit ini didasarkan pada modul ESP8266, yang mengintegrasikan GPIO, PWM (*Pulse Width Modulation*), IIC, 1-Wire dan ADC (*Analog to Digital Converter*) semua dalam satu *board*. NodeMCU berukuran panjang 4.83cm, lebar 2.54cm, dan berat 7 gram. *Board* ini sudah dilengkapi dengan fitur WiFi dan *Firmware*nya yang bersifat *opensource*.

Spesifikasi yang dimiliki oleh NodeMCU sebagai berikut :

1. Board ini berbasis ESP8266 serial WiFi SoC (*Single on Chip*) dengan *onboard* USB to TTL. *Wireless* yang digunakan adalah IEEE 802.11b/g/n. 2.
2. Tantalum *capasitor* 100 *micro farad* dan 10 *micro farad*.
3. 3.3v LDO regulator.
4. *Blue led* sebagai indikator.
5. Cp2102 usb to UART *bridge*.
6. Tombol *reset*, port usb, dan tombol *flash*.

7. Terdapat 9 GPIO yang di dalamnya ada 3 pin PWM, 1 x ADC Channel, dan pin RX TX
8. 3 pin ground.
9. S3 dan S2 sebagai pin GPIO
10. S1 MOSI (*Master Output Slave Input*) yaitu jalur data dari master dan masuk ke dalam slave, sc cmd/sc.
11. S0 MISO (*Master Input Slave Input*) yaitu jalur data keluar dari slave dan masuk ke dalam master.
12. SK yang merupakan SCLK dari master ke slave yang berfungsi sebagai *clock*.
13. Pin Vin sebagai masukan tegangan.
14. Built in 32-bit MCU

Bentuk fisik dari NodeMCU ESP8266 dapat dilihat dalam gambar 2.1.



Gambar 2.1. NodeMCU ESP8266

Sumber: Google Image

2.2 GPS (*Global Positioning System*)

GPS adalah singkatan dari *Global Positioning System*, yang merupakan sistem navigasi dengan menggunakan teknologi satelit yang dapat menerima sinyal dari satelit.

Sinyal ini diterima oleh alat penerima (*receiver*) di permukaan, dimana GPS *receiver* ini akan mengumpulkan informasi dari satelit GPS. Sebuah GPS *receiver* harus mengunci

sinyal minimal tiga satelit untuk menghitung posisi 2D (*latitude* dan *longitude*) dan *track* pergerakan. Jika GPS *receiver* dapat menerima empat atau lebih satelit, maka dapat menghitung posisi 3D (*latitude*, *longitude* dan *altitude*). Jika sudah dapat menentukan posisi *user*, selanjutnya GPS dapat menghitung informasi lain, seperti kecepatan, arah yang dituju, jalur, tujuan perjalanan, jarak tujuan, matahari terbit dan matahari terbenam dan lain lain. Sinyal yang dikirimkan oleh satelit ke GPS akan digunakan untuk menghitung waktu perjalanan (*travel time*). Waktu perjalanan ini sering juga disebut sebagai *Time of Arrival* (TOA). Sesuai dengan prinsip fisika, bahwa untuk mengukur jarak dapat diperoleh dari waktu dikalikan dengan cepat rambat sinyal.

Dari beberapa pemakaian GPS di atas dikategorikan menjadi:

- a. Waktu. GPS receiver menerima informasi waktu dari jam atom yang mempunyai keakurasian sangat tinggi.
- b. Lokasi. GPS memberikan informasi lokasi:
 1. *Latitude*
 2. *Longitude*
 3. *Altitude*
- c. Kecepatan. Ketika berpindah tempat, GPS dapat menunjukkan informasi kecepatan berpindah tersebut.
- d. Arah perjalanan. GPS dapat menunjukkan arah tujuan. Simpan lokasi. Tempat-tempat yang sudah pernah atau ingin dikunjungi bisa disimpan oleh GPS *receiver*.
- e. Komulasi data. GPS *receiver* dapat menyimpan informasi *track*, seperti total perjalanan yang sudah pernah dilakukan, kecepatan rata-rata, kecepatan paling tinggi, kecepatan paling rendah, waktu/jam sampai tujuan, dan sebagainya.
- f. *Tracking*. Membantu untuk memonitoring pergerakan obyek. Membantu memetakan posisi tertentu, dan perhitungan jaringan terdekat.

Sistem yang serupa dengan GPS antara lain *GLONASS* Rusia, *Galileo* Uni Eropa, *IRNSS* India. Cara kerja GPS secara logik ada 5 langkah:

1. Memakai perhitungan "*triangulation*" dari satelit;
2. Untuk perhitungan "*triangulation*", GPS mengukur jarak menggunakan *travel time* sinyal radio;
3. Untuk mengukur *travel-time*, GPS memerlukan akurasi waktu yang tinggi;

4. Untuk perhitungan jarak, kita harus tahu dengan pasti posisi satelit dan ketinggian pada orbitnya;
5. Harus mengoreksi *delay* sinyal waktu perjalanan di atmosfer sampai diterima *receiver*.

Selain sebagai penunjuk jalan, GPS juga dapat digunakan sebagai alat untuk proteksi kendaraan kita, dengan dicampurkan program proteksi kendaraan, alat ini dapat membantu kita untuk mengamankan kendaraan kita, khususnya kendaraan roda 2. Bentuk fisik GPS dapat dilihat dalam Gambar 2.2 dan parameter GPS terdapat dalam Tabel 2.1.



Gambar 2.2. *Global Positioning System (GPS)*

Sumber: Google Image

Tabel 2.1 Parameter GPS

Parameter	Keterangan
Nama Produk	<i>GPS Ublox NEO-6M</i>
Produsen	<i>Ublox</i>
Website Produk	https://www.tokopedia.com/hwthinker/module-gps-u-blox-ublox-neo-6m-neo6m-gps-module-neo6mv2-arduino-drone
Harga	Rp 60.500
Tegangan VCC	3.3 – 5 VDC
Komunikasi	<i>UART SERIAL</i>
<i>Indoor GPS</i>	<i>-162 dBm tracking sensitivity</i>

2.3 Buzzer

Buzzer merupakan sebuah komponen elektronika yang berfungsi untuk mengubah getaran listrik menjadi getaran suara. Pada dasarnya prinsip kerja *buzzer* hampir sama dengan *loudspeaker*, jadi *buzzer* juga terdiri dari kumparan yang terpasang pada diafragma dan kemudian kumparan tersebut dialiri arus sehingga menjadi elektromagnet, kumparan tadi akan tertarik ke dalam atau keluar, tergantung dari arah arus dan polaritas magnetnya, karena kumparan dipasang pada diafragma maka setiap gerakan kumparan akan menggerakkan diafragma secara bolak-balik sehingga membuat udara bergetar yang akan menghasilkan suara. *Buzzer* biasa digunakan sebagai indikator bahwa proses telah selesai atau terjadi suatu kesalahan pada sebuah alat (*alarm*). Bentuk fisik *buzzer* dapat dilihat dalam Gambar 2.3 dan parameter *buzzer* terdapat dalam Tabel 2.2.



Gambar 2.3. Buzzer

Sumber : Google Image

Tabel 2.2 Parameter Buzzer

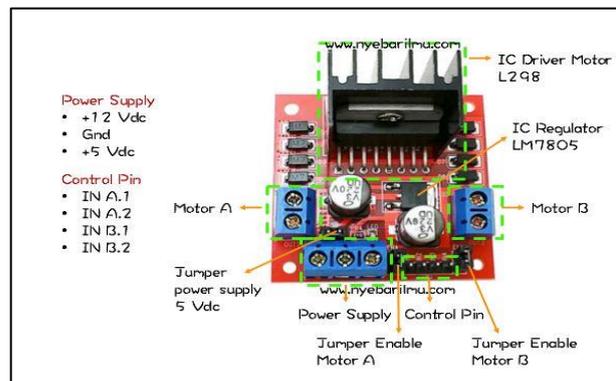
Parameter	Keterangan
Nama Produk	Active Buzzer
Produsen	Made in China
Website Produk	https://shopee.co.id/search?keyword=buzzer%20alarm
Harga	Rp 10.000
Input	3,3VDC \pm 300Hz
Output	12VDC \pm 4000Hz

2.4 Driver Motor L298N

Driver motor L298N merupakan *module driver* motor DC yang paling banyak digunakan atau dipakai di dunia elektronika yang difungsikan untuk mengontrol kecepatan serta arah perputaran motor DC. IC L298 merupakan sebuah IC tipe *H-bridge* yang mampu mengendalikan beban-beban induktif seperti *relay*, *solenoid*, motor DC dan motor *stepper*.

Pada IC L298 terdiri dari *transistor-transistor logic* (TTL) dengan gerbang *nand* yang berfungsi untuk memudahkan dalam menentukan arah putaran suatu motor dc maupun motor *stepper*. Untuk dipasaran sudah terdapat modul *driver* motor menggunakan ic l298 ini, sehingga lebih praktis dalam penggunaannya karena pin I/O nya sudah *ter-package* dengan rapi dan mudah digunakan.

Kelebihan akan modul *driver* motor L298N ini yaitu dalam hal kepresisian dalam mengontrol motor sehingga motor lebih mudah untuk dikontrol. Bentuk fisik dari *driver* motor L298N dapat dilihat dalam Gambar 2.4.



Gambar 2.4. Driver Motor L298N

Sumber: Google Image

Keterangan :

- *Enable A* : berfungsi untuk mengaktifkan bagian *output* motor A
- *Enable B* : berfungsi untuk mengaktifkan bagian *output* motor B
- *Jumper 5vdc* : sebagai mode pemilihan sumber tegangan 5Vdc, jika tidak di-*jumper* maka akan ke mode sumber tegangan 12 Vdc
- *Control Pin* : Sebagai kendali perputaran dan kecepatan motor yang dihubungkan ke Mikrokontroler

Spesifikasi dari *Driver Motor L298N*

- Menggunakan IC L298N (*Double H bridge Drive Chip*)
- Tegangan minimal untuk masukan *power* antara 5V-35V
- Tegangan operasional : 5V
- Arus untuk masukan antara 0-36mA
- Arus maksimal untuk keluaran per *Output A* maupun *B* yaitu 2A
- Daya maksimal yaitu 25W
- Dimensi modul yaitu 43 x 43 x 26mm
- Berat : 26g

2.5 PWM (*Pulse Width Modulation*)

Pulse Width Modulation (PWM) secara umum adalah sebuah cara memanipulasi lebar sinyal yang dinyatakan dengan pulsa dalam suatu perioda, untuk mendapatkan tegangan rata-rata yang berbeda. Beberapa Contoh aplikasi PWM adalah pemodulasian data untuk telekomunikasi, pengontrolan daya atau tegangan yang masuk ke beban, regulator tegangan, *audio effect* dan penguatan, serta aplikasi-aplikasi lainnya. Aplikasi PWM berbasis mikrokontroler biasanya berupa, pengendalian kecepatan motor DC, Pengendalian Motor Servo, Pengaturan nyala terang LED. Pada tugas akhir ini, PWM digunakan untuk mengontrol frekuensi dari *buzzer*.

2.6 Logika Fuzzy

Logika *fuzzy* diperkenalkan pertama kali pada tahun 1965 oleh Prof Lutfi A. Zadeh seorang peneliti di Universitas California di Barkley dalam bidang ilmu komputer. Professor Zadeh beranggapan logika benar salah tidak dapat mewakili setiap pemikiran manusia, kemudian dikembangkanlah logika *fuzzy* yang dapat mempresentasikan setiap keadaan atau mewakili pemikiran manusia. Perbedaan antara logika tegas dan logika *fuzzy* terletak pada keanggotaan elemen dalam suatu himpunan. Jika dalam logika tegas suatu elemen mempunyai dua pilihan yaitu terdapat dalam himpunan atau bernilai 1 yang berarti benar dan tidak pada himpunan atau bernilai 0 yang berarti salah. Sedangkan dalam logika *fuzzy*, keanggotaan elemen berada di interval $[0,1]$. Logika *fuzzy* menjadi alternatif dari berbagai sistem yang ada dalam pengambilan keputusan karena logika *fuzzy* mempunyai kelebihan sebagai berikut:

- Logika *fuzzy* memiliki konsep yang sangat sederhana sehingga mudah untuk dipahami
- Logika *fuzzy* sangat fleksibel, artinya mampu beradaptasi dengan perubahan-perubahan dan ketidakpastian.
- Logika *fuzzy* memiliki toleransi terhadap data yang tidak tepat.
- Logika *fuzzy* mampu mensistemkan fungsi-fungsi non-linier yang sangat kompleks.
- Logika *fuzzy* dapat mengaplikasikan pengalaman atau pengetahuan dari para pakar secara langsung tanpa melalui proses pelatihan
- Logika *fuzzy* dapat bekerjasama dengan teknik-teknik kendali secara konvensional.
- Logika *fuzzy* didasarkan pada bahasa sehari-hari sehingga mudah dimengerti.

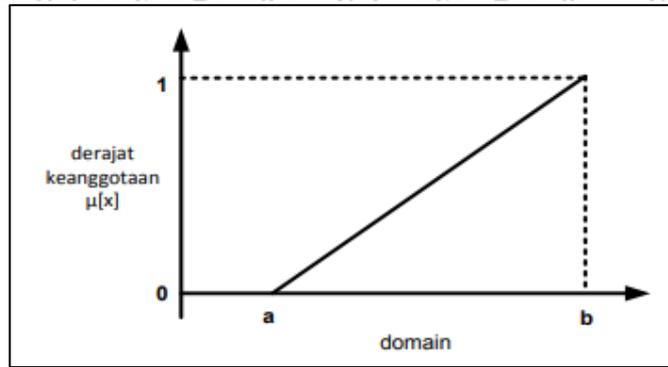
Logika *fuzzy* umumnya diterapkan pada masalah-masalah yang mengandung unsur ketidakpastian (*uncertainty*), ketidaktepatan (*imprecise*), *noisy*, dan sebagainya. Logika *fuzzy* memiliki beberapa komponen yang harus dipahami seperti himpunan *fuzzy*, fungsi keanggotaan, operator pada himpunan *fuzzy*, inferensi *fuzzy* dan defuzzifikasi.

A. Fungsi Keanggotaan

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

a. Representasi Linier

Pada representasi linear, pemetaan input ke derajat keanggotaannya digambarkan sebagai suatu garis lurus. Bentuk ini paling sederhana dan menjadi pilihan yang baik untuk mendekati suatu konsep yang kurang jelas. Ada 2 keadaan himpunan *fuzzy* yang linear. Pertama, kenaikan himpunan dimulai pada nilai domain yang memiliki derajat keanggotaan nol [0] bergerak ke kanan menuju ke nilai domain yang memiliki derajat keanggotaan lebih tinggi seperti pada Gambar 2.5.



Gambar 2.5. Representasi Linier Naik

Persamaan fungsi keanggotaan linier naik terdapat dalam Persamaan 2.1. (Jan Jantzen, 2007)

$$\mu[x] = \begin{cases} 0, & x \leq a \\ \frac{(x-a)}{(b-a)}, & a \leq x \leq b \\ 1, & x \geq b \end{cases} \dots\dots\dots(2.1)$$

Keterangan :

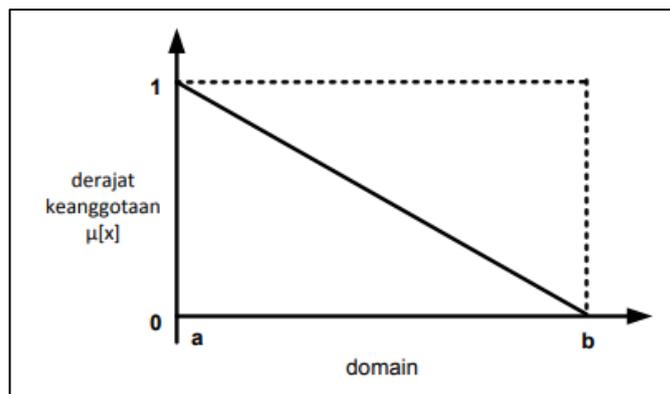
a = nilai domain terkecil saat derajat keanggotaan terkecil

b = derajat keanggotaan terbesar saat domain

c = nilai domain terbesar saat derajat keanggotaan terkecil

Kedua, merupakan kebalikan yang pertama. Garis lurus dimulai dari nilai domain dengan derajat keanggotaan tertinggi pada sisi kiri, kemudian bergerak menurun ke nilai domain yang memiliki derajat keanggotaan lebih rendah seperti pada Gambar

2.6.



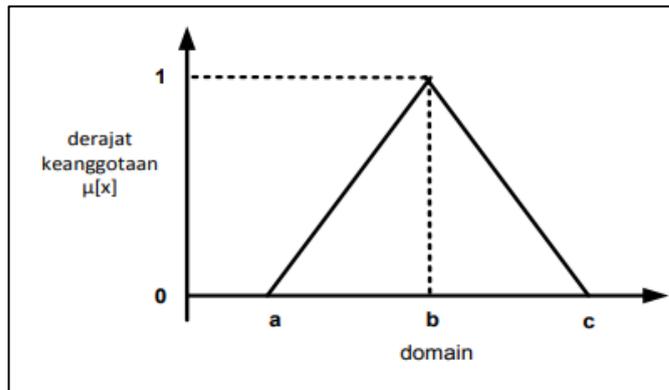
Gambar 2.6. Representasi Linier turun

Persamaan fungsi keanggotaan linier turun terdapat dalam Persamaan 2.2. (Jan Jantzen, 2007)

$$\mu[x] = \begin{cases} \frac{(b-x)}{(b-a)}, & a \leq x \leq b \\ 0, & x \geq b \end{cases} \dots\dots\dots(2.2)$$

b. Representasi Kurva Segitiga

Kurva Segitiga pada dasarnya merupakan gabungan antara 2 garis (linear) seperti terlihat pada Gambar 2.7.



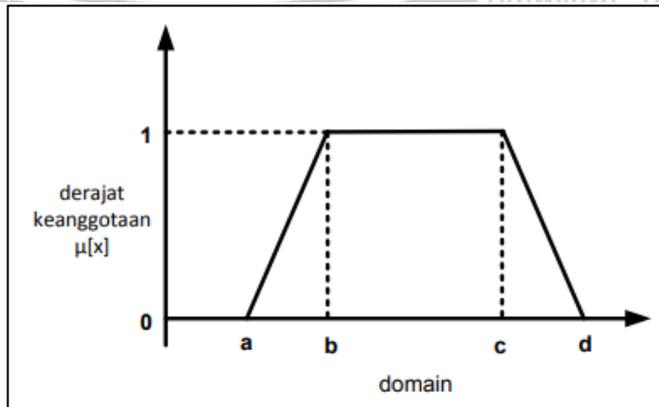
Gambar 2.7. Representasi Kurva Segitiga

Persamaan fungsi keanggotaan kurva segitiga terdapat dalam Persamaan 2.3 (Jan Jantzen, 2007)

$$\mu[x] = \begin{cases} 0, & x \leq a \text{ atau } x \geq c \\ \frac{(a-x)}{(b-a)}, & a \leq x \leq b \\ \frac{(c-x)}{(c-b)}, & b \leq x \leq c \end{cases} \dots\dots\dots(2.3)$$

c. Representasi Kurva Trapesium

Kurva Segitiga pada dasarnya seperti bentuk segitiga, hanya saja ada beberapa titik yang memiliki nilai keanggotaan 1 seperti pada Gambar 2.8.



Gambar 2.8. Representasi Kurva Trapesium

Persamaan fungsi keanggotaan kurva trapezium terdapat dalam Persamaan 2.4. (Jan Jantzen, 2007)

$$\mu[x] = \begin{cases} 0, & x \leq a \text{ atau } x \geq d \\ \left(\frac{x-a}{b-a}\right), & a \leq x \leq b \\ 1, & b \leq x \leq c \\ \left(\frac{d-x}{d-c}\right), & c \leq x \leq d \end{cases} \dots\dots\dots(2.4)$$

B. Operator Logika *Fuzzy*

Seperti halnya himpunan konvensional, ada beberapa operasi yang didefinisikan secara khusus untuk mengkombinasi dan memodifikasi himpunan *fuzzy*. Nilai keanggotaan sebagai hasil dari operasi 2 himpunan sering dikenal dengan nama fire strength atau α -predikat. Ada 3 operator dasar yang diciptakan oleh Zadeh, yaitu:

a. Operator AND

Operator ini berhubungan dengan operasi interseksi pada himpunan. α -predikat sebagai hasil operasi dengan operator AND diperoleh dengan mengambil nilai keanggotaan terkecil antar elemen pada himpunan- himpunan yang bersangkutan.

Persamaan operator AND terdapat dalam Persamaan 2.5. (Jan Jantzen, 2007)

$$\mu_{A \cap B} = \min(\mu_A[x], \mu_B[y]) \dots\dots\dots(2.5)$$

b. Operator OR

Operator ini berhubungan dengan operasi union pada himpunan. α -predikat sebagai hasil operasi dengan operator OR diperoleh dengan mengambil nilai keanggotaan terbesar antar elemen pada himpunan- himpunan yang bersangkutan.

Persamaan operator OR terdapat dalam Persamaan 2.6. (Jan Jantzen, 2007)

$$\mu_{A \cup B} = \max(\mu_A[x], \mu_B[x]) \dots\dots\dots(2.6)$$

c. Operator NOT

Operator ini berhubungan dengan operasi komplemen pada himpunan. α -predikat sebagai hasil operasi dengan operator NOT diperoleh dengan mengurangi nilai keanggotaan elemen pada himpunan yang bersangkutan dari 1. Persamaan operator

NOT terdapat dalam Persamaan 2.7. (Jan Jantzen, 2007)

$$\mu_A = 1 - \mu_A[x] \dots\dots\dots(2.7)$$

2.7 Penalaran Logika *Fuzzy* Metode Mamdani

Pada metode Mamdani, aplikasi fungsi implikasi menggunakan *MIN*, sedang komposisi aturan menggunakan metode *MAX*. Metode Mamdani dikenal juga dengan

metode *MAX-MIN*. Inferensi *output* yang dihasilkan berupa bilangan *fuzzy* maka harus ditentukan suatu nilai *crisp* tertentu sebagai *output*. Proses ini dikenal dengan defuzzifikasi. Ada beberapa tahapan untuk mendapatkan *output* yaitu:

A. Pembentukan himpunan *fuzzy*

Pada metode Mamdani baik variabel input maupun variabel *output* dibagi menjadi satu atau lebih himpunan *fuzzy*.

1. Aplikasi fungsi implikasi

Pada metode Mamdani, fungsi implikasi yang digunakan adalah *Min*.

2. Komposisi Aturan

Tidak seperti penalaran monoton, apabila sistem terdiri dari beberapa aturan, maka inferensi diperoleh dari kumpulan dan korelasi antar aturan.

Ada 3 metode yang digunakan dalam melakukan inferensi sistem *fuzzy* yaitu : *Max*, *Additive* dan Probabilistik OR

a) Metode *Max* (*Maximum*)

Pada metode ini solusi himpunan *fuzzy* diperoleh dengan cara mengambil nilai maksimum aturan, kemudian menggunakannya untuk memodifikasi daerah *fuzzy* dan mengaplikasikan 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 proposisi. Persamaan Metode *Max* terdapat dalam Persamaan 2.8. (Jan Jantzen, 2007)

$$\mu_{sf}[x_i] \leftarrow \max(\mu_{sf}[x_i], \mu_{kf}[x_i]) \dots \dots \dots (2.8)$$

dengan :

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

$\mu_{kf}[x_i]$ = nilai keanggotaan konsekuen fuzzy aturan ke-i

b) Metode *Additive* (*Sum*)

Pada metode ini, solusi himpunan *fuzzy* diperoleh dengan cara melakukan *bounded-sum* terhadap semua output daerah *fuzzy*. Persamaan Metode *Additive* terdapat dalam Persamaan 2.9. (Jan Jantzen, 2007)

$$\mu_{sf}[x_i] \leftarrow \min(1, \mu_{sf}[x_i] + \mu_{kf}[x_i]) \dots \dots \dots (2.9)$$

dengan :

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

$\mu_{kf}[x_i]$ = nilai keanggotaan konsekuen fuzzy aturan ke-i

c) Metode Probabilistik OR

Pada metode ini, solusi himpunan *fuzzy* diperoleh dengan cara melakukan produk terhadap semua output daerah *fuzzy*. Persamaan Probabilistik OR terdapat dalam persamaan 2.10. (Jan Jantzen, 2007)

$$\mu_{sf}[x_i] \leftarrow (\mu_{sf}[x_i] + \mu_{kf}[x_i]) - (\mu_{sf}[x_i] - \mu_{kf}[x_i]) \dots \dots \dots (2.10)$$

dengan :

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

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

B. Aturan *Fuzzy*

Aturan yang digunakan pada himpunan *fuzzy* adalah aturan *if-then*. Proposisi *fuzzy* dibedakan menjadi dua, proposisi *fuzzy atomic* dan proposisi *fuzzy compound*. Proposisi *fuzzy atomic* adalah pernyataan *single* dimana sebagai variabel linguistik dan adalah himpunan *fuzzy* dari . Proposisi *fuzzy compound* adalah gabungan dari proposisi *fuzzy atomic* yang dihubungkan dengan operator “or”, “and”, dan “not”.

C. Penegasan /Defuzzifikasi

Input dari proses Defuzzifikasi adalah suatu himpunan *fuzzy* 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, maka harus dapat diambil suatu nilai *crisp* tertentu sebagai *output*. Defuzzifikasi sebagai pemetaan dari himpunan *fuzzy* (C) ke himpunan tegas.

Ada beberapa metoda yang dipakai dalam defuzzifikasi:

- a. Metode *Centroid* : Pada metode ini penetapan nilai *crisp* dengan cara mengambil titik pusat daerah *fuzzy*. Persamaan Metode *Centroid* terdapat dalam Persamaan 2.11 untuk bagian kontinu sedangkan Persamaan 2.12 untuk bagian diskrit. (Jan Jantzen, 2007)

$$z^* = \frac{\int_a^b z \mu_C(z) dz}{\int_a^b \mu_C(z) dz} \dots \dots \dots (2.11)$$

$$z^* = \frac{\sum_{j=i}^n \mu_C(z_j) z_j}{\sum_{j=i}^n \mu_C(z_j)} \dots \dots \dots (2.12)$$

Keterangan :

Pembilang = Momen

Penyebut = Luas Daerah

n = Jumlah tingkat kuantisasi dari output

C = himpunan *fuzzy* yang didefinisikan pada *output* (z)

- b. Metode Bisektor : Pada metode ini , solusi *crisp* diperoleh dengan cara mengambil nilai pada domain *fuzzy* yang memiliki nilai keanggotaan seperti dari jumlah total nilai keanggotaan pada daerah *fuzzy*. Persamaan Metode Bisektor terdapat dalam Persamaan 2.13. (Jan Jantzen, 2007)

$$z_p = \int_a^p \mu(z) dz = \int_{ap}^b u(z) dz \dots \dots \dots (2.13)$$

Keterangan :

$p = x$ yang membagi daerah inferensi menjadi dua bagian yang sama besar

- c. Metode *Means of Maximum* (MoM) : Pada metode ini, solusi *crisp* diperoleh dengan cara mengambil nilai rata-rata domain yang memiliki nilai keanggotaan maksimum.
- d. Metode *Largest of Maximum* (LoM) : Pada metode ini, solusi *crisp* diperoleh dengan cara mengambil nilai terbesar dari domain yang memiliki nilai keanggotaan maksimum.
- e. Metode *Smallest of Maksimum* (SoM) : Solusi *crisp* diperoleh dengan cara mengambil nilai terkecil dari domain yang memiliki nilai keanggotaan maksimum.

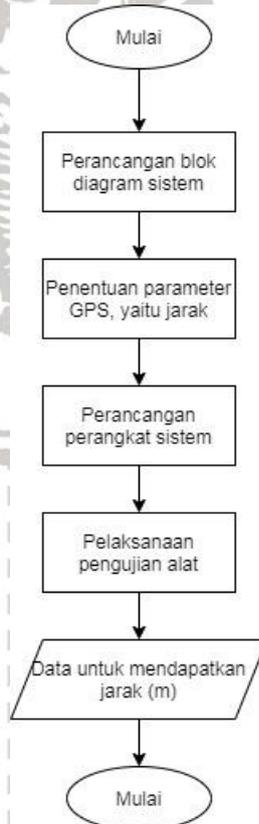
BAB III METODE PENELITIAN

Untuk menyelesaikan rumusan masalah dan merealisasikan tujuan penelitian yang terdapat di bab pendahuluan maka diperlukan langkah-langkah untuk menyelesaikan masalah tersebut. Metode penelitian pada skripsi ini meliputi:

1. Penentuan jenis dan cara pengambilan data
2. Perancangan blok diagram sistem
3. Spesifikasi desain
4. Pembuatan perangkat keras
5. Pembuatan perangkat lunak

3.1 Penentuan Jenis dan Cara Pengambilan Data

Data-data yang diperlukan dalam kajian penelitian terdiri atas data primer dan data sekunder. Data primer merupakan data hasil pengukuran jarak pada GPS dari posisi awal hingga posisi yang telah ditentukan. Ditampilkan diagram alir metode pengambilan data primer melalui eksperimen dalam Gambar 3.1.

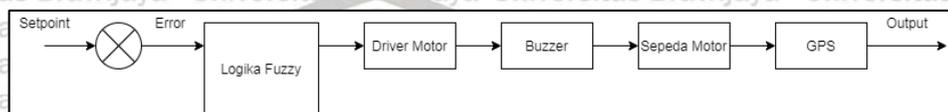


Gambar 3.1. Langkah Pengambilan Data

Data sekunder bersumber dari buku referensi, jurnal, skripsi, internet, dan forum-forum resmi. Data sekunder yang digunakan dalam pembahasan skripsi ini diperlukan sebagai bahan yang mendasari konsep-konsep yang terkait dengan jarak yang dihasilkan oleh GPS.

3.2 Perancangan Blok Diagram Sistem

Perancangan diagram blok sistem menjelaskan sistem secara garis besar dan diharapkan alat dapat bekerja sesuai dengan spesifikasi desain. Blok diagram sistem deteksi dapat dilihat dalam Gambar 3.2.



Gambar 3.2. Blok Diagram Sistem

Keterangan :

1. Logika Fuzzy : Kontroler
2. *Driver Motor* : *Buffer*
3. *Buzzer* : Aktuator
4. *Sepeda Motor* : *Plant*
5. *GPS* : *Plant*

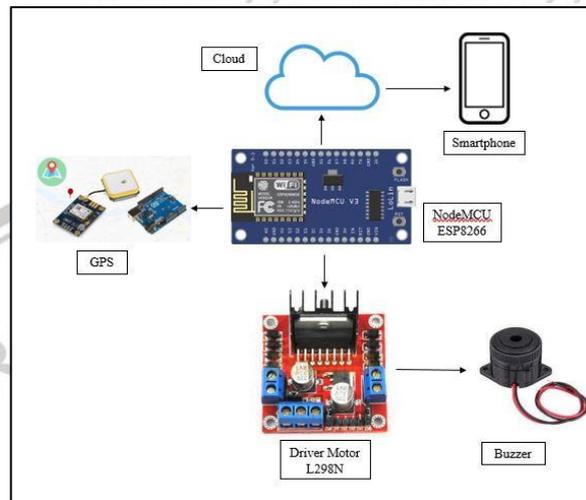
3.3 Spesifikasi Desain

Spesifikasi desain ditentukan dengan tujuan agar dapat membuat sistem sesuai dengan yang diharapkan. Desain yang dirancang akan memiliki spesifikasi sebagai berikut.

1. GPS digunakan dengan tujuan mampu mendeteksi dan melacak saat diperintah
2. *Board NodeMCU ESP8266* mampu memproses data yang diambil dari GPS dan *buzzer* serta melakukan proses klarifikasi.
3. Modul Wifi digunakan dengan tujuan mampu menyambungkan mikrokontroler *NodeMCU ESP8266*.
4. *Smartphone* mampu menampilkan data dari *input* GPS
5. *Arduino IDE* digunakan untuk memprogram board *NodeMCU ESP8266*
6. *Buzzer* digunakan untuk *output* dari jarak yang dihasilkan GPS

3.4 Pembuatan Perangkat Keras

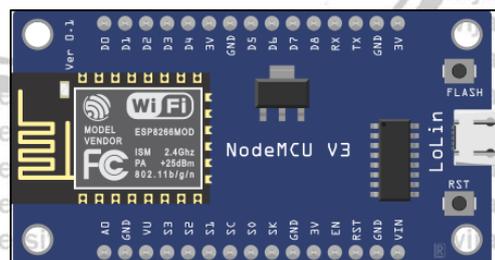
Pembuatan perangkat keras dilakukan sebagai langkah awal sebelum terbentuknya suatu sistem beserta pemrogramannya. Hal ini dimaksudkan agar sistem pengontrolan GPS dapat bekerja sesuai dengan deskripsi awal yang telah direncanakan. Skema pembuatan perangkat keras dapat dilihat dalam Gambar 3.3.



Gambar 3.3. Skema Pembuatan Perangkat Keras

1. NodeMCU ESP8266

Mikrokontroler NodeMCU ESP8266 digunakan sebagai sistem utama atau otak dari alat untuk menggerakkan atau mengaktifkan relay ketika diperintahkan. Bentuk fisik dari NodeMCU ESP8266 dapat dilihat dalam Gambar 3.4.



Gambar 3.4. NodeMCU ESP8266

Sumber : Google Image

2. GPS

GPS digunakan sebagai pendeteksi dan pelacakan pada sepeda motor yang dapat dilihat dalam Gambar 3.5

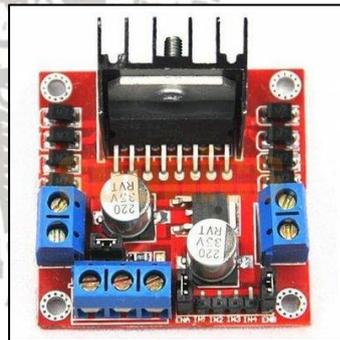


Gambar 3.5. Global Positioning System (GPS)

Sumber : Google Image

3. Driver Motor L298N

Driver motor digunakan sebagai pengontrol PWM pada buzzer dapat dilihat dalam Gambar 3.6.



Gambar 3.6. Driver Motor L298N

Sumber : Google Image

4. Buzzer

Buzzer digunakan sebagai pengingat atau alarm pada sepeda motor dapat dilihat dalam Gambar 3.7.



Gambar 3.7. Buzzer

Sumber : Google Image

5. Smartphone

Smartphone digunakan sebagai penerima data dari mikrokontroler berupa notifikasi dan sebagai antarmuka dari aplikasi alat dapat dilihat dalam Gambar 3.8.



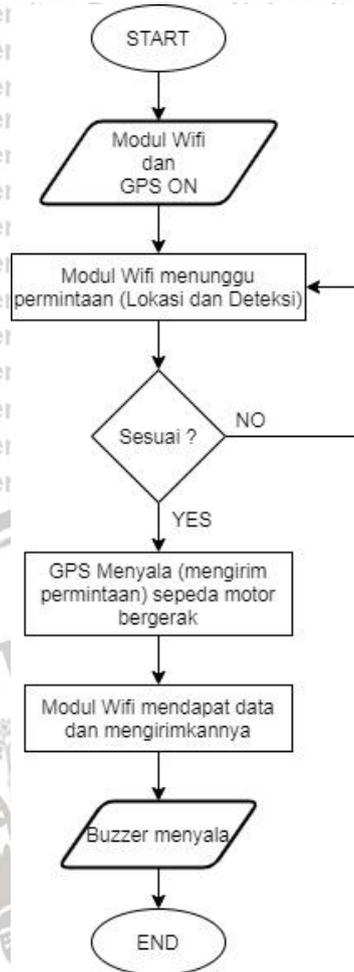
Gambar 3.8. Smartphone

Sumber : Google Image

3.5 Pembuatan Perangkat Lunak

Perancangan perangkat lunak berkaitan dengan *flowchart* program yang akan diimplementasikan ke mikrokontroler NodeMCU ESP8266 dan terhubung dengan GPS.

Flowchart sistem dapat dilihat dalam Gambar 3.9.



Gambar 3.9. Flowchart Sistem Kerja GPS

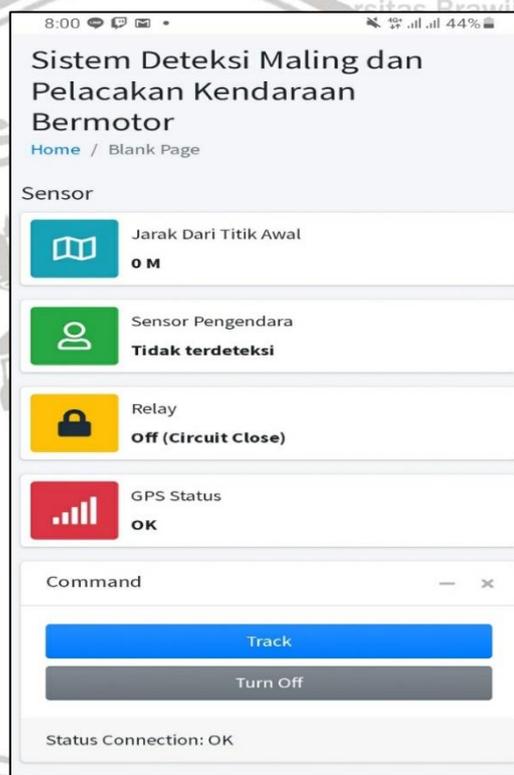
Komponen ini aktif ketika sistem berjalan secara otomatis semua komponen akan ikut aktif seperti Modul Wifi dan GPS, kemudian Modul Wifi meminta data ke GPS dan NodeMCU ESP8266 yang bekerja sebagai otak utama sistem. Modul Wifi menyala dan mengirim sinyal ke pengguna melalui aplikasi antarmuka yang ada pada *smartphone* dan sudah terkoneksi dengan internet serta *buzzer* menyala.

BAB IV DESAIN DAN PERANCANGAN

Bab IV menjelaskan desain dan perancangan. Desain dan perancangan yang disajikan dari penelitian meliputi perancangan aplikasi sistem, perancangan logika *fuzzy* dan pengujian sistem.

4.1 Perancangan Aplikasi Sistem

Aplikasi antarmuka pada *smartphone* akan diberi nama “Sistem Deteksi Maling dan Pelacakan Kendaraan Bermotor”. Rancangan antarmuka aplikasi pada *smartphone* dapat dilihat dalam Gambar 4.1 dan pada laptop dapat dilihat dalam Gambar 4.2.

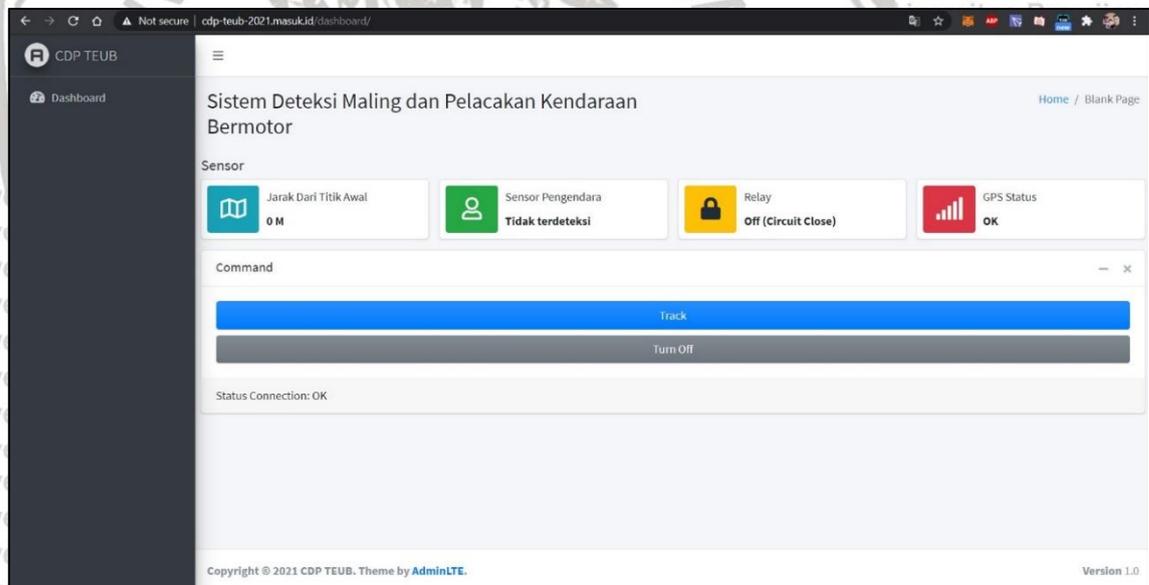


Gambar 4.1. Rancangan Aplikasi Antarmuka Pada Smartphone

Berikut penjelasan untuk setiap bagian dari antarmuka aplikasi:

- Judul halaman: menunjukkan judul halaman. Dalam aplikasi ini hanya akan ada 3 halaman, yaitu halaman “*HOME*” (halaman utama), halaman “Lokasi Motor”, dan halaman “*Alert*” yang hanya akan muncul ketika pengguna mengetuk notifikasi dari aplikasi *android* ini.

- b. Nama aplikasi: menampilkan nama aplikasi. Dalam hal ini, nama aplikasi adalah “Sistem Deteksi Maling dan Pelacakan Kendaraan Bermotor
- c. Jarak Dari Titik Awal: Memberitahukan posisi letak motor sejauh mana dari titik awal.
- d. Sensor Pengendara: Memberitahukan ada atau tidaknya seseorang diatas jok motor, yang dibaca oleh sensor ultrasonik
- e. Relay: Memberitahukan kondisi relay yang *on* atau *off*
- f. GPS : menunjukkan status lokasi dari kendaraan pengguna secara *real time*
- g. *Track*: Berfungsi untuk melacak kendaraan pengguna secara *real time*, disaat tombol ini ditekan, lokasi motor pengguna akan langsung diberitahu dilokasi mana secara *real time*
- h. *Turn off* : Berfungsi untuk mematikan mesin dari kendaraan bermotor pengguna, disaat tombol ini ditekan motor pengguna akan langsung secara otomatis akan mati



Gambar 4.2 Rancangan Aplikasi Antarmuka Pada Laptop

4.2. Perancangan Logika Fuzzy

Untuk merancang pengendali dalam penelitian ini, maka diterapkan logika *fuzzy* pada sistem deteksi menggunakan GPS dengan metode Mamdani. Sistem kendali dirancang dengan dua masukan yang berupa *Error* yang merupakan selisih antara setpoint dengan pembacaan *buzzer* dan *Delta Error* yang merupakan turunan pertama dari nilai *Error*.

Kedua input *Error* dan *Delta Error* akan diproses oleh *fuzzy logic controller* (FLC) untuk mendapatkan nilai *output* berupa sinyal PWM untuk mengontrol suara *buzzer*.

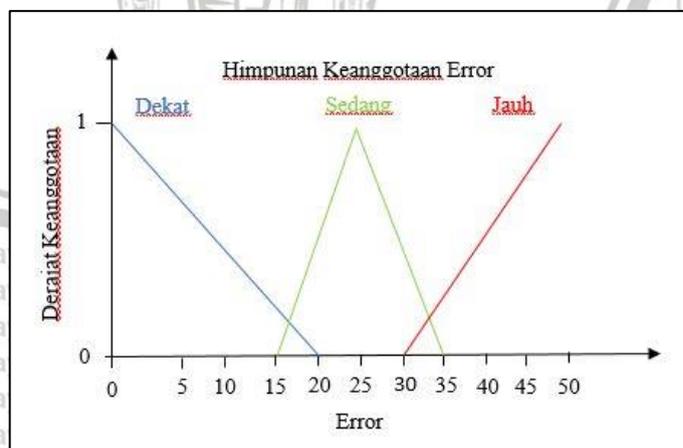
a. Pembentukan himpunan *fuzzy*

Pengolahan data dilakukan dengan menentukan variabel dan semesta pembicaraan, dilanjutkan dengan pembentukan himpunan *fuzzy*. Penentuan variabel dan semesta pembicaraan dari hasil pengambilan data dapat diperoleh pada Tabel 4.1.

Tabel 4.1. Variabel dan Semesta Pembicaraan *Input* dan *Output*

Fungsi	Nama Variabel	Semesta Pembicaraan	Keterangan
<i>Input</i>	<i>Error</i>	[0-50]	<i>Error</i> jarak dari titik awal
<i>Output</i>	<i>Buzzer</i>	[0-100]	Bit

Dari data-data yang sudah didapatkan sebagai input untuk mengetahui hasil *output*, maka data tersebut dibuat sesuai dengan rentang tingkatan *fuzzy*. Untuk himpunan keanggotaan *error* terdapat dalam Gambar 4.3. Fuzzifikasi input *error* terdapat dalam Tabel 4.2, dan fuzzifikasi *output* Bit *buzzer* terdapat dalam Tabel 4.3.



Gambar 4.3. Himpunan Keanggotaan *Error*

Tabel 4.2. Fuzzifikasi *Input Error*

No	Nilai Pembacaan Jarak (m)	Himpunan <i>Fuzzy</i>
1	0-20	Dekat
2	15-35	Sedang
3	30-50	Jauh

Fungsi dari Himpunan keanggotaan *Error* :

$$\mu_{Dekat}(x) = \begin{cases} 0, & x \leq 0 \text{ atau } x \geq 20 \\ \frac{20-x}{20}, & 0 < x < 20 \end{cases}$$

$$\mu_{Sedang}(x) = \begin{cases} 0, & x \leq 15 \text{ atau } x \geq 35 \\ \frac{x-15}{20}, & 15 < x < 25 \\ \frac{35-x}{20}, & 25 \leq x < 35 \end{cases}$$

$$\mu_{Jauh}(x) = \begin{cases} 0, & x \leq 30 \\ \frac{x-30}{20}, & 30 < x < 50 \\ 1, & x \geq 50 \end{cases}$$

Tabel 4.3. Fuzzifikasi *Output Bit Buzzer*

No	Nilai <i>Output Bit Buzzer</i> per detik	Himpunan <i>Fuzzy</i>
1	0-40	Pelan
2	30-70	Sedang
3	60-100	Cepat

b. Aplikasi fungsi implikasi

Menyusun basis aturan, yaitu aturan-aturan berupa implikasi-implikasi *fuzzy* yang menyatakan relasi antara variabel *input* dengan variabel *output*. Pada penelitian ini menggunakan metode Mamdani, fungsi implikasi yang digunakan adalah *min-max* dengan bentuk sebagai berikut:

[R1] IF *Error* DEKAT THEN *Output Bit Buzzer* PELAN

[R2] IF *Error* SEDANG THEN *Output Bit Buzzer* SEDANG

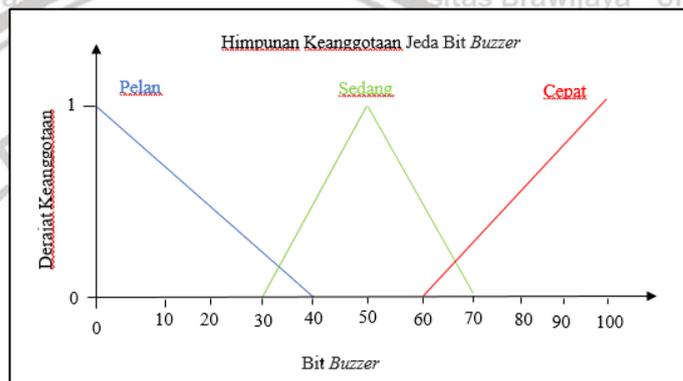
[R3] IF *Error* JAUH THEN *Output Bit Buzzer* CEPAT

c. Penegasan (defuzifikasi)

Masukan dari proses penegasan adalah suatu himpunan *fuzzy* yang diperoleh dari komposisi aturan-aturan *fuzzy*, sedangkan *output* yang dihasilkan merupakan suatu bilangan *real* yang tegas. Sehingga jika diberikan suatu himpunan *fuzzy* dalam *range* tertentu, maka dapat diambil suatu nilai tegas tertentu sebagai *output*.

Defuzifikasi menggunakan metode *centeroid*. Pada metode ini, solusi tegas diperoleh dengan cara mengambil titik pusat daerah *fuzzy*. Untuk himpunan keanggotaan Bit *buzzer* terdapat dalam Gambar 4.4.

$$z^* = \frac{\int_a^b z \mu_C(z) dz}{\int_a^b \mu_C(z) dz}$$



Gambar 4.4. Himpunan Keanggotaan Bit *Buzzer*

4.3 Pengujian Sistem

Setelah sistem dari “sistem pelacakan dan deteksi sepeda motor dari pencurian” dibuat secara keseluruhan, maka langkah selanjutnya adalah melakukan pengujian dan analisis terhadap keseluruhan sistem. Pengujian yang dilakukan meliputi:

1. Uji performa GPS.

Pengujian ini dilakukan untuk mengetahui apakah GPS berkerja sesuai harapan.

2. Uji performa sistem.

Pengujian ini dilakukan untuk mengetahui performa pengiriman data. Ada atau tidaknya *error* saat proses pengiriman data.

3. Uji performa aplikasi *Android*.

Pengujian ini dilakukan untuk mengetahui apakah aplikasi *Android* dapat berjalan sesuai dengan harapan.

4. Uji sistem secara keseluruhan.

Pengujian ini dilakukan untuk mengetahui apakah alat secara keseluruhan dapat berjalan sesuai dengan harapan.



BAB V HASIL DAN PEMBAHASAN

Pada bagian ini akan dilakukan pengujian alat agar diketahui bekerja atau tidaknya fungsi-fungsi yang telah dibuat serta kesesuaian hasil dengan yang sudah direncanakan. Pengujian dilakukan per blok untuk dapat mengetahui permasalahan di tiap blok.

5.1 Pengujian GPS (*Global Positioning System*)

Pengujian GPS dilakukan untuk mendapatkan perbandingan data GPS dan data praktik. Selain itu, pengujian ini juga mendapatkan %*error* dari komponen GPS. Data pengujian GPS terdapat dalam Tabel 5.1.

$$\text{Perhitungan \%error} : \frac{\text{Data praktik} - \text{Data GPS}}{\text{Data Praktik}} \times 100\%$$

Tabel 5.1. Data Pengujian GPS

No	Data GPS (m)	Data Pengujian (m)	% <i>error</i>
1	10	10,5	4,7%
2	20	20,7	3,3%
3	30	31,4	4,4%
4	40	40,8	1,9%
5	50	50,5	0,9%

Pada Tabel 5.1, dapat dilihat bahwa hasil pengujian GPS memiliki *error* dibawah 5% dan rata-rata *error* pada GPS yaitu 3,05%. Dapat disimpulkan bahwa pengujian GPS sesuai dengan data dan bekerja dengan baik.

5.2 Pengujian Kontrol Logika Fuzzy

Tujuan pengujian ini adalah untuk mengetahui apakah nilai keluaran *fuzzy* hasil dari proses *rule base evaluation* dan proses defuzzyfikasi sudah sesuai dengan yang diharapkan pada sistem. Metode fuzifikasi yang digunakan adalah *centroid*. Data hasil pengujian proses *fuzzy* terdapat dalam Tabel 5.2.

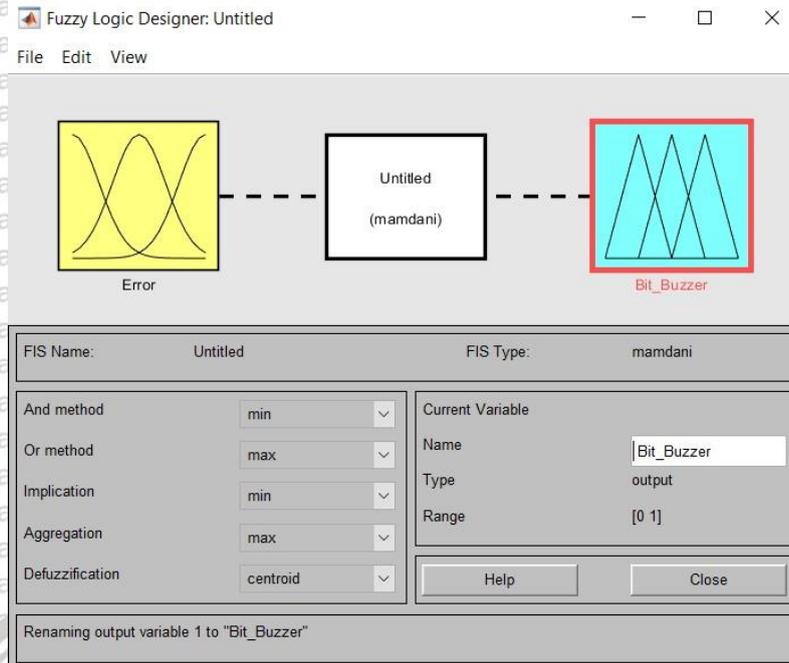
Tabel 5.2. Data Hasil Pengujian Proses *Fuzzy*

No	Error (m)	Output
1	0	0
2	5	13.7
3	10	15.3
4	15	17.4
5	20	50
6	25	50
7	30	50
8	35	82.6
9	40	84.7
10	45	86.3
11	50	87

Pada Tabel 5.2 dapat dilihat bahwa hasil dari pengujian proses *fuzzy* memiliki tingkat jeda bit *buzzer* semakin cepat. *Error* menunjukkan letak sepeda motor ketika berpindah dari posisi awal atau dalam keadaan parkir. *Output* menunjukkan jeda bit pada *buzzer* per detik.

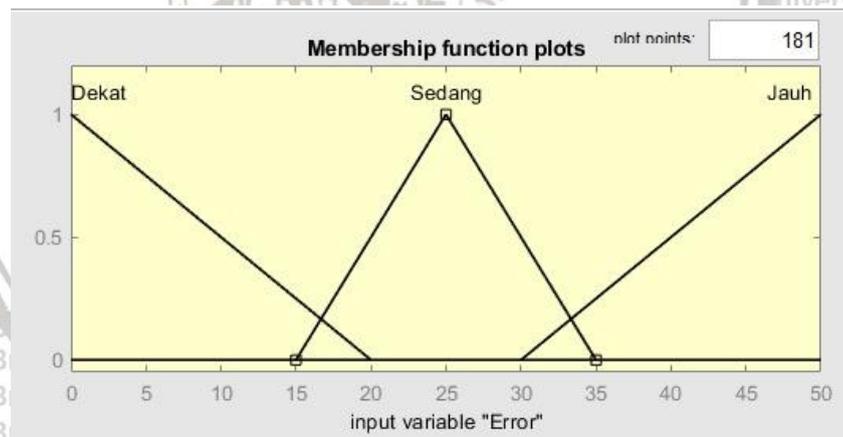
5.3 Desain *Fuzzy* Pada Matlab

Desain *fuzzy* dilakukan pada aplikasi matlab, *fuzzy* yang digunakan adalah *fuzzy* metode Mamdani. Sebagai representasi sistem kontrol logika *fuzzy* maka dilakukan desain *fuzzy rule* pada matlab dengan memasukan fungsi keanggotaan jarak dan *output* yang dapat dilihat dalam Gambar 5.1.

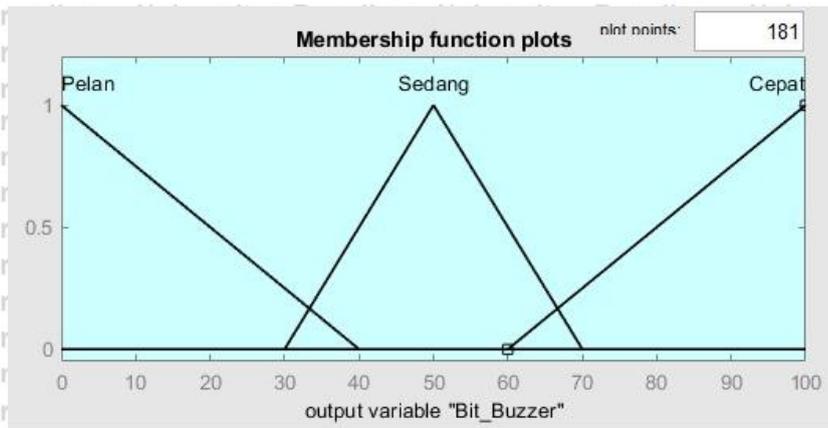


Gambar 5.1. Desain *Fuzzy Matlab*

Berikutnya adalah tampilan *membership function*, disinilah tempat untuk mengkomposisikan himpunan *fuzzy*. Himpunan *fuzzy error* dapat dilihat dalam Gambar 5.2 dan himpunan *fuzzy output bit buzzer* dapat dilihat dalam Gambar 5.3.

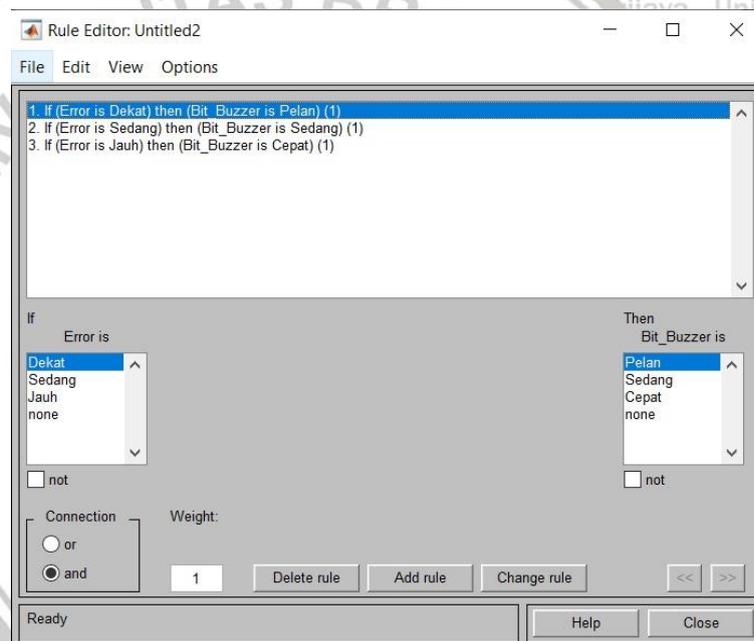


Gambar 5.2. Tampilan *Membership Function Error*



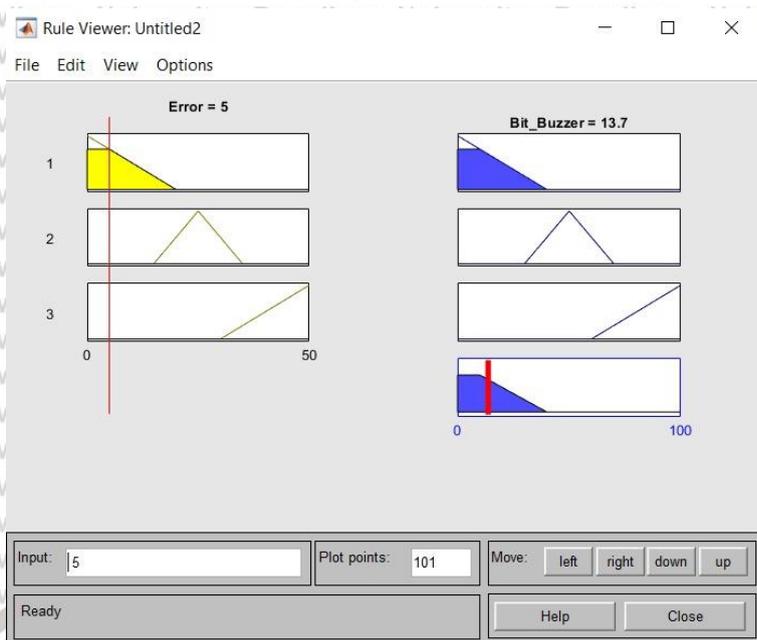
Gambar 5.3. Tampilan *Membership Function Output Bit Buzzer*

Setelah menentukan semua *input* dan *output*, maka selanjutnya yaitu memasukkan *rule-base* yang telah dirancang pada *rule editor*. Hasilnya dapat dilihat dalam Gambar 5.4

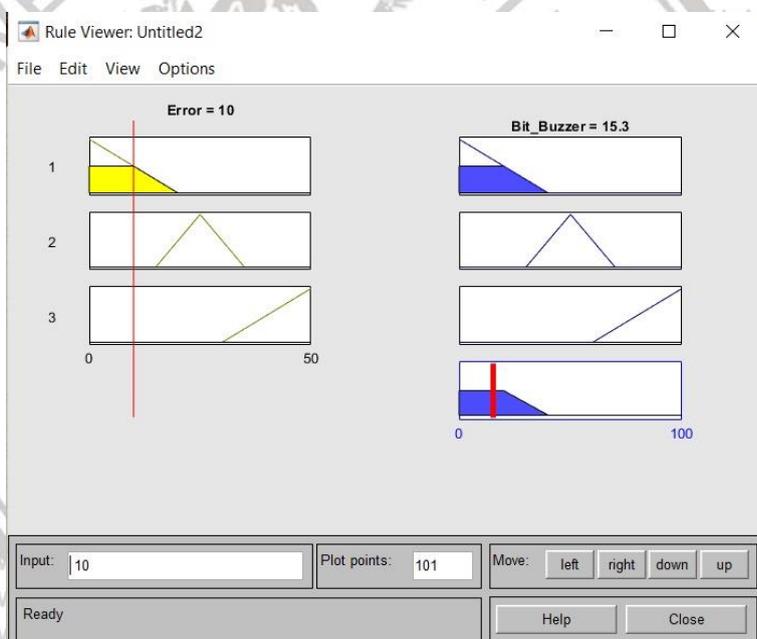


Gambar 5.4. Tampilan *Rule Editor*

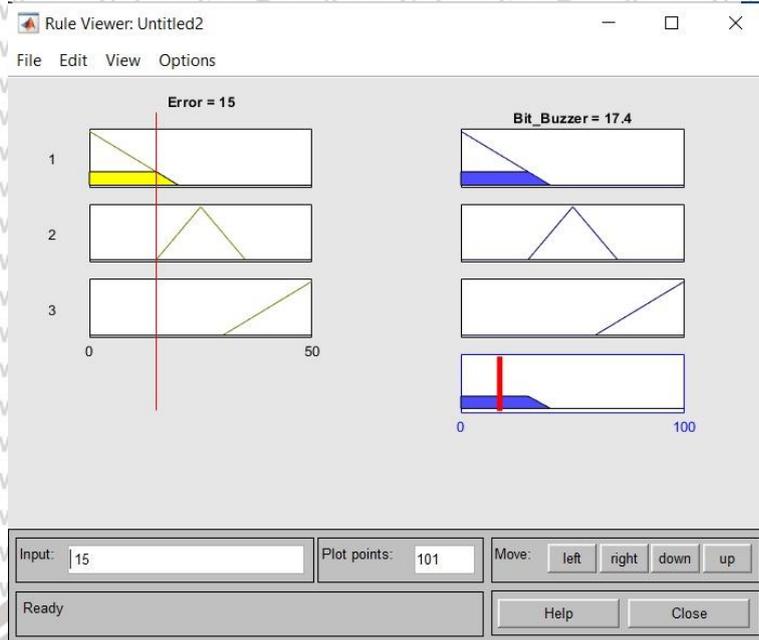
Rule viewer yang dimana dapat memasukkan *input* dan melihat hasil *outputnya*. Berikut disajikan *Rule viewer* pada proses pengujian *fuzzy*.



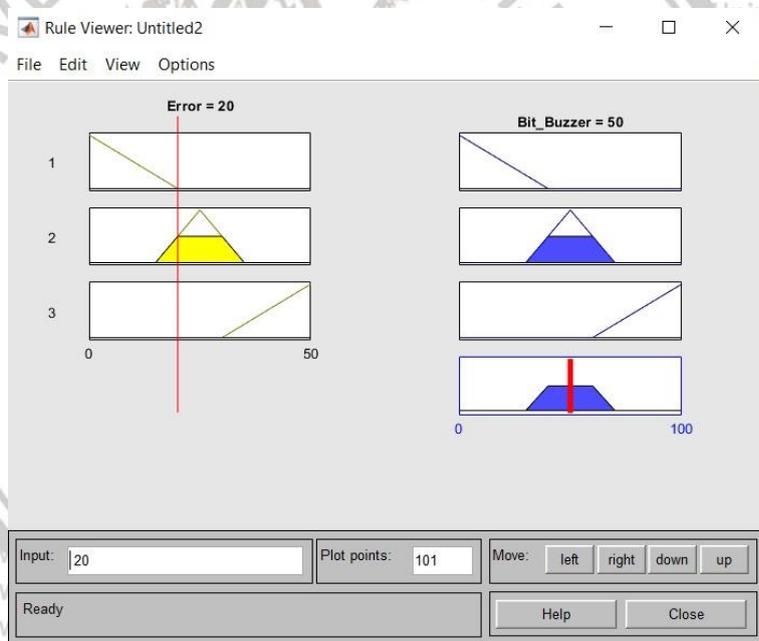
Gambar 5.5. Tampilan *Rule Viewer* Dengan *Error* 5



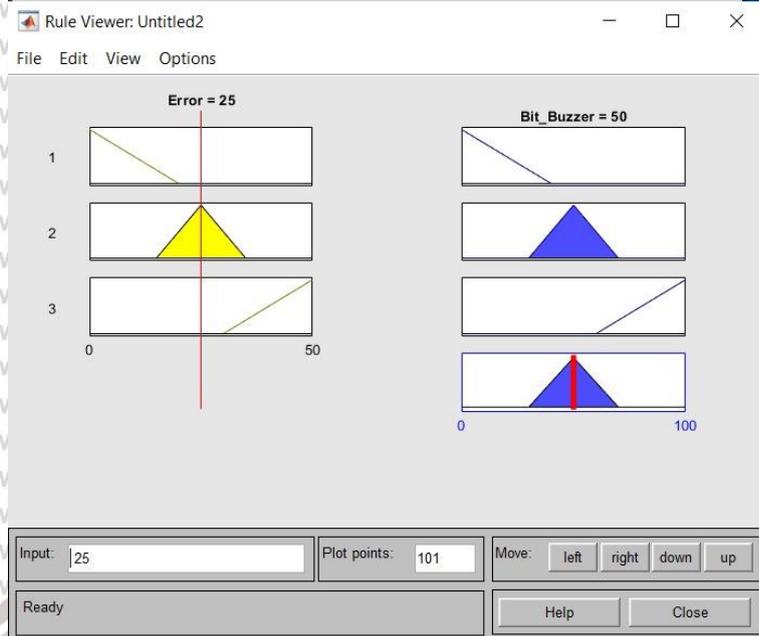
Gambar 5.6. Tampilan *Rule Viewer* Dengan *Error* 10



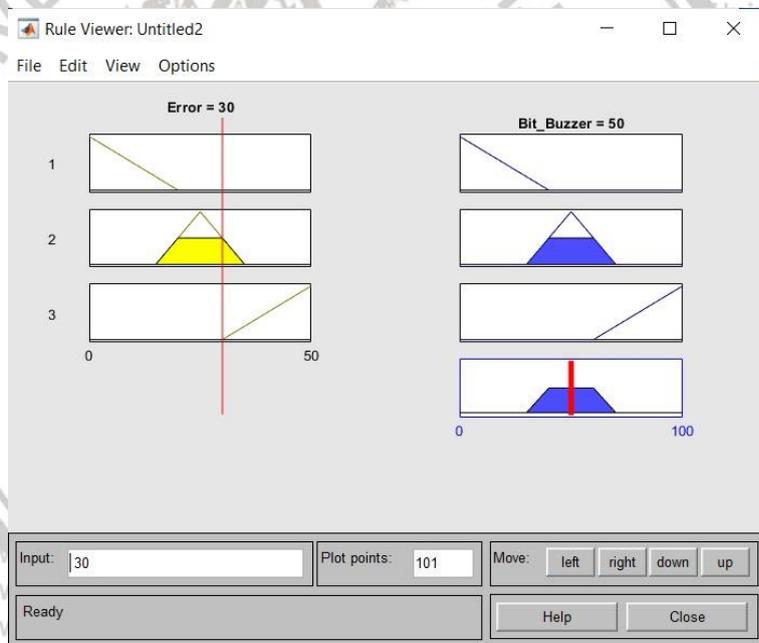
Gambar 5.7. Tampilan *Rule Viewer* Dengan *Error 15*



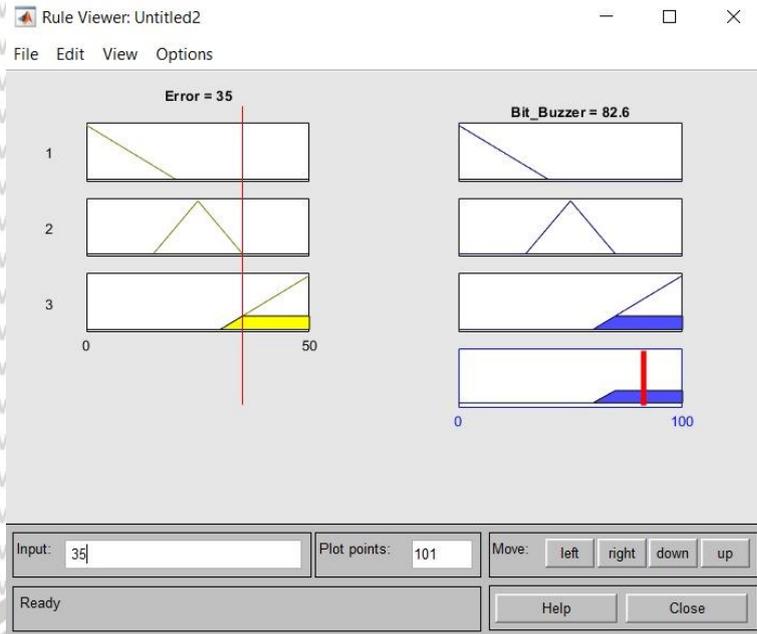
Gambar 5.8. Tampilan *Rule Viewer* Dengan *Error 20*



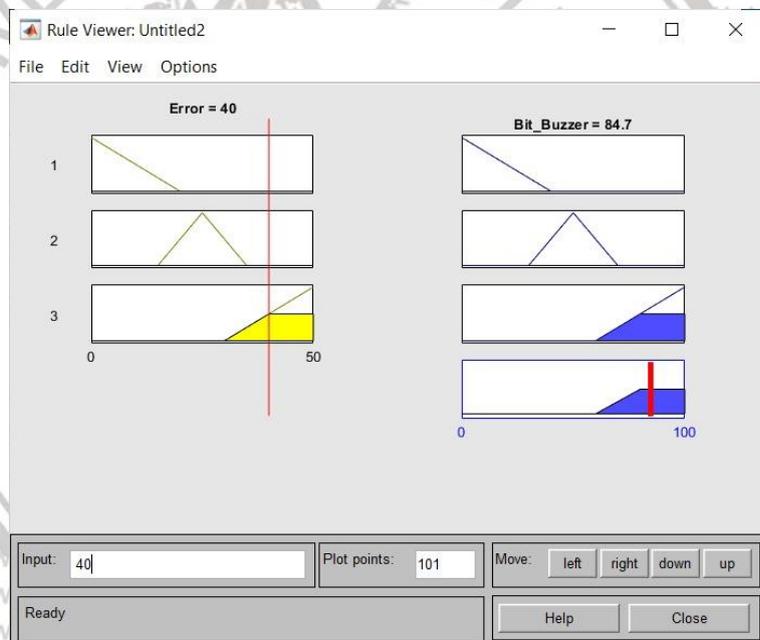
Gambar 5.9. Tampilan *Rule Viewer* Dengan *Error 25*



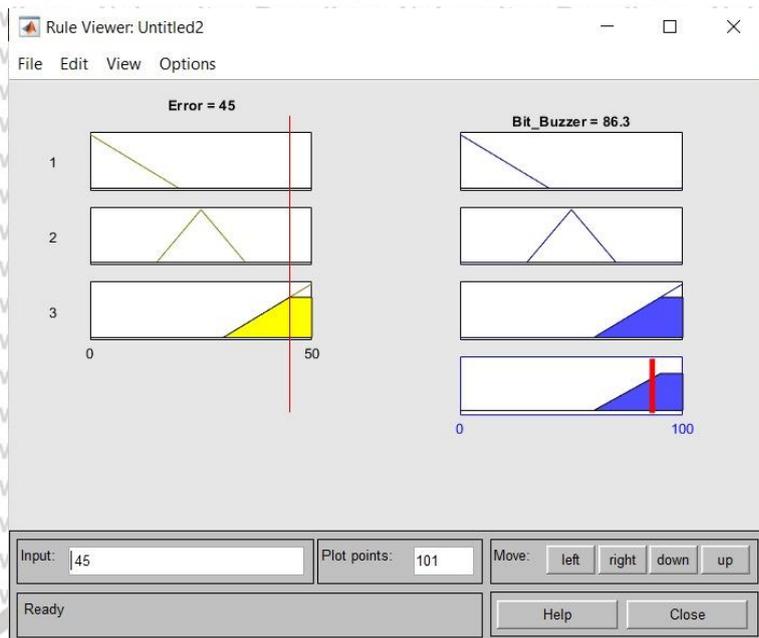
Gambar 5.10. Tampilan *Rule Viewer* Dengan *Error 30*



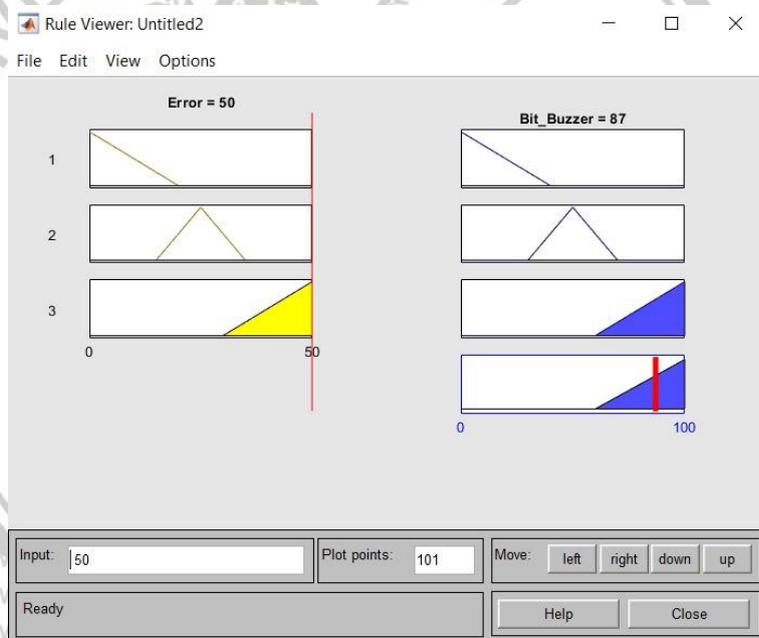
Gambar 5.11. Tampilan *Rule Viewer* Dengan *Error 35*



Gambar 5.12. Tampilan *Rule Viewer* Dengan *Error 40*

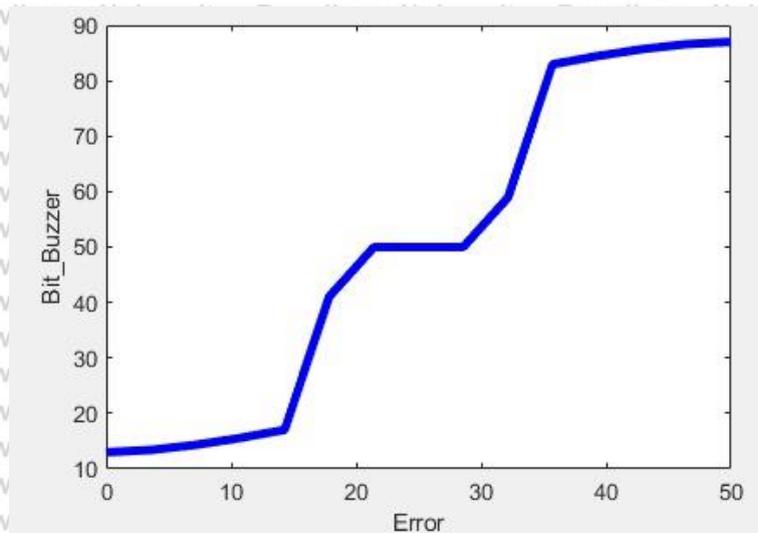


Gambar 5.13. Tampilan *Rule Viewer* Dengan *Error 45*



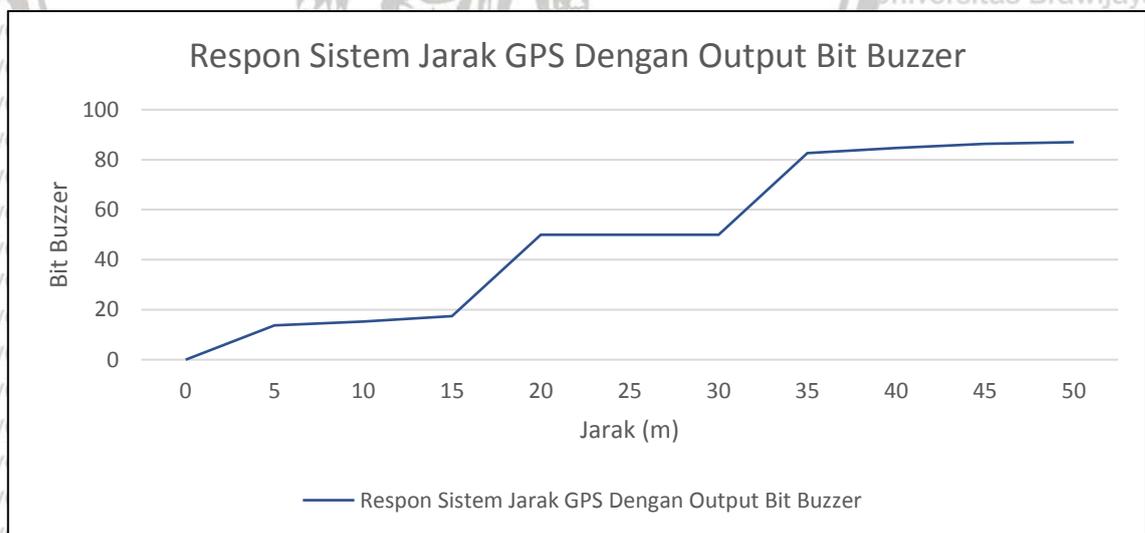
Gambar 5.14. Tampilan *Rule Viewer* Dengan *Error 50*

Hasil secara keseluruhan keluaran desain *fuzzy rule* maka dapat dilihat melalui fasilitas *surface* yang tersedia sehingga menghasilkan ruang solusi. Ruang solusi tersebut terdapat dalam Gambar 5.15.



Gambar 5.15. Surface Output Desain Fuzzy Matlab

Pada saat pengujian sistem, sistem diuji menggunakan 11 *inputan* jarak yaitu : 0m, 5m, 10m, 15m, 20m, 25m, 30m, 35m, 40m, 45m, 50m. Respon keluaran sistem terdiri dari grafik *output* yaitu jarak pada GPS sebagai *input* dan jeda bit pada Buzzer sebagai *output*. Grafik respon sistem jarak pada GPS hasil kalkulasi oleh logika *fuzzy* yang akan ditunjukkan dalam Gambar 5.16.



Gambar 5.16. Respon Sistem Jarak GPS Dengan *Output* Bit Buzzer

BAB VI PENUTUP

6.1 Kesimpulan

Setelah dilakukan berbagai macam pengujian dan analisis data, maka dapat disimpulkan sebagai berikut:

1. Logika *fuzzy* menggunakan metode Mamdani dapat digunakan untuk jarak pada GPS dengan akurasi yang cukup baik.
2. Keakuratan pembacaan dari data GPS jauh lebih baik yaitu dengan *error* dibawah 5% dan rata-rata *error*nya adalah 3,05%.
3. Hasil dari pengujian sistem jarak menunjukkan bahwa semakin jauh jarak GPS dari titik awal maka jeda bit *Buzzer* semakin cepat.

6.2 Saran

Saran yang dapat diberikan untuk penelitian selanjutnya adalah dengan memodifikasi himpunan masukan dan keluaran *fuzzy* serta *fuzzy rule* agar respon sistem dapat mencapai *output* yang diinginkan.

Halaman ini sengaja dikosongkan



DAFTAR PUSTAKA

Mochammad Rusli, 2017, Dasar-Dasar Perancangan Kendali Logika Fuzzy. Universitas Brawijaya Press, Malang.

Kusumadewi, Sri dan Hari Purnomo. 2004. Aplikasi Logika Fuzzy untuk Pendukung Keputusan. Yogyakarta: Graha Ilmu.

Suwintana, I Ketut. 2013. Sistem Inferensi Fuzzy Mamdani Berbasis Web, Jurnal Matrix Vol.3 No.1. Bali: Politeknik Negeri Bali.

Widiyantoro, Andreas, T. Sutojo Dan Sudaryanto. 2014. Menerapkan Logika Fuzzy Mamdani untuk Menentukan Harga Jual Batik Vol.13 No.2: 69-74. Semarang: Universitas Dian Nuswantoro.

Winarno, Totok. 2009. Aplikasi Fuzzy Logic Sebagai Kontrol Posisi Motor DC. Malang: Politeknik Negeri Malang.

Sutikno Dan Indra Waspada. Perbandingan Metode Defuzzifikasi Sistem Kendali Logika Fuzzy Model Mamdani pada Motor DC. Semarang: Universitas Diponegoro.

"Aplikasi Teknologi Global Positioning System (GPS) Dan Telepon Selular(GSM) Untuk Monitoring Titik Akses Kendaraan Dinas Unnes", journal.unnes.co.id, 1 Juli 2012, 14 November

2020, <https://journal.unnes.ac.id/nju/index.php/saintekno/article/view/5539/4407>

"Angka Curanmor Di Era Pandemi", SuaraMerdeka.com, 29 Juli 2020, 14 November 2020,

<https://www.suaramerdeka.com/news/nasional/236419-angka-curanmor-di-erapandemi-cenderungnaik#:~:text=Meski%20bertambah%20di%20saat%20pandemi,jumlah%20kendaraan%20terlihat%20makin%20menurun>

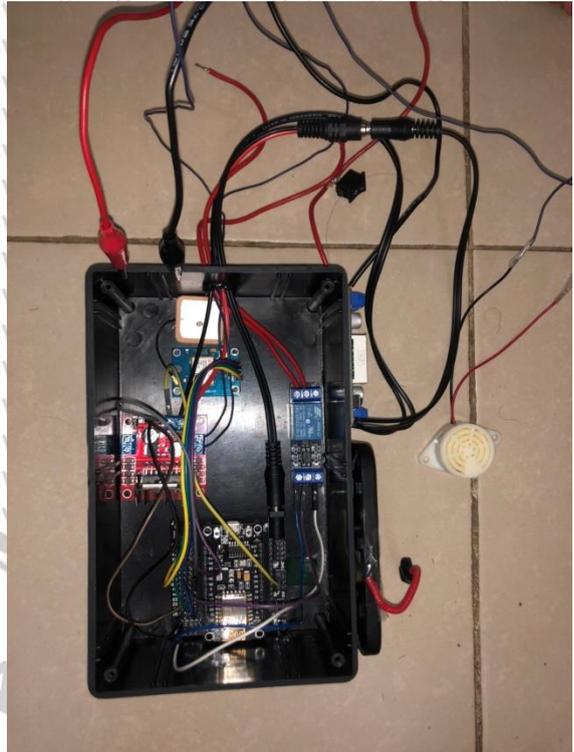


LAMPIRAN 1

Foto Alat



Dokumentasi alat tampak atas



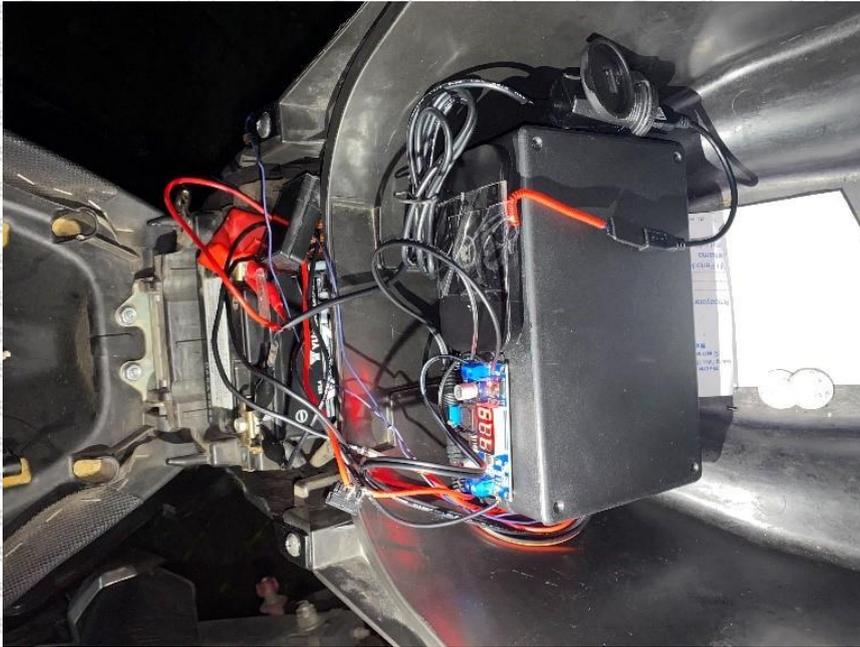
Dokumentasi rangkaian elektrik alat



Dokumentasi alat bagian GPS



Dokumentasi alat bagian buzzer



Dokumentasi alat tampak depan didalam bagasi



Dokumentasi alat tampak atas didalam bagasi



LAMPIRAN 2
Listing Program

```
#include <TinyGPS++.h>
#include <SoftwareSerial.h>
float lat, lon;
int year, month, date, hour, minute, second;
String date_str, time_str, lat_str, lon_str;
```

```
float park_lat, park_lon;
```

```
TinyGPSPlus gps;
```

```
int park = 0;
```

```
unsigned long get_distance;
```

```
SoftwareSerial GpsSerial(4, 5);
```

```
void setup() {
```

```
  // put your setup code here, to run once:
```

```
  Serial.begin(9600);
```

```
  if(GpsSerial.available() > 0)
```

```
  {
```

```
    Serial.println("GPS Serial OK");
```

```
  }
```

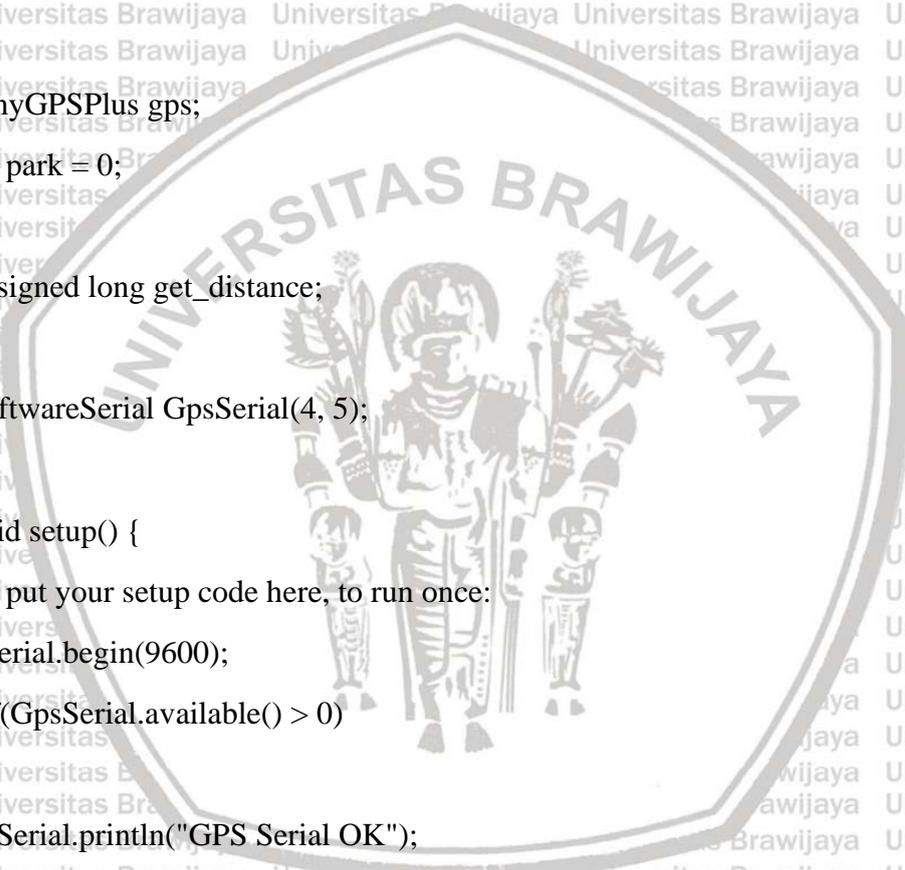
```
  GpsSerial.begin(9600);
```

```
  pinMode(D1, INPUT); // PARK ON
```

```
  pinMode(D2, INPUT); // PARK OFF
```

```
  pinMode(D8, OUTPUT);
```

```
}
```

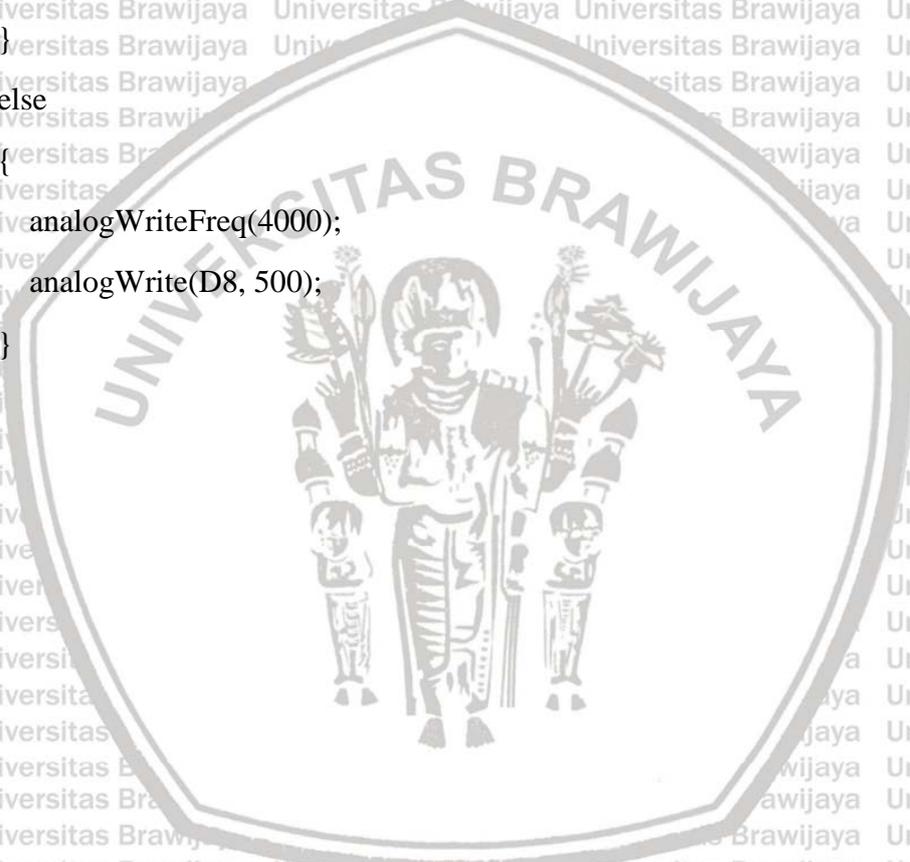


```
void loop() {  
  // Ambil Data GPS  
  while(GpsSerial.available() > 0)  
  {  
    if(gps.encode(GpsSerial.read()))  
    {  
      if(gps.location.isValid())  
      {  
        lat = gps.location.lat();  
        lat_str = String(lat, 6);  
        lon = gps.location.lng();  
        lon_str = String(lon, 6);  
      }  
    }  
    if(digitalRead(D1) == HIGH) // Nyalakan mode parkir  
    {  
      park_lat = lat;  
      park_lon = lon;  
      park = 1;  
    }  
    if(digitalRead(D1) == HIGH) // Matikan Mode Parkir  
    {  
      park = 0;  
    }  
    if(park == 1){ // Jika mode parkir nyala dan jarak lebih dari 30 maka buzzer menyala.  
    pada jarak 50 meter lebih maka suara maksimal / keras  
      get_distance = (unsigned long)gps.distanceBetween(lat, lon, park_lat, park_lon);  
      if(get_distance <= 30 )  
      {
```

UNIVERSITAS BRAWIJAYA



```
analogWrite(D8, 0);  
}  
else if(get_distance > 30 && get_distance <= 50)  
{  
  int freq = map(get_distance, 30, 50, 3800, 4000);  
  int pwm = map(get_distance, 30, 50, 200, 500);  
  analogWriteFreq(4000);  
  analogWrite(D8, 500);  
}  
else  
{  
  analogWriteFreq(4000);  
  analogWrite(D8, 500);  
}
```





LAMPIRAN 3

Data Sheet

1. Data Sheet NodeMCU ESP8266

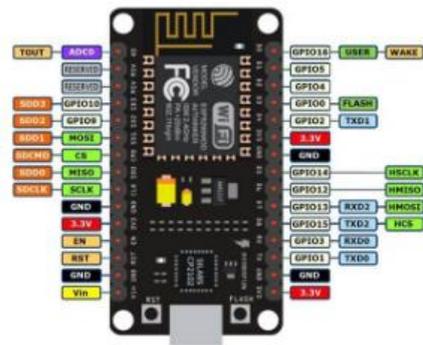
NodeMCU ESP8266

📅 22 April, 2020

🔍 0



NodeMCU ESP8266



NodeMCU ESP8266 Pinout

NodeMCU Development Board Pinout Configuration

Pin Category	Name	Description
Power	Micro-USB, 3.3V, GND, Vin	<p>Micro-USB: NodeMCU can be powered through the USB port</p> <p>3.3V: Regulated 3.3V can be supplied to this pin to power the board</p> <p>GND: Ground pins</p> <p>Vin: External Power Supply</p>
Control Pins	EN, RST	The pin and the button resets the microcontroller
Analog Pin	A0	Used to measure analog voltage in the range of 0-3.3V
GPIO Pins	GPIO1 to GPIO16	NodeMCU has 16 general purpose input-output pins on its board

SPI Pins	SD1, CMD, SD0, CLK	NodeMCU has four pins available for SPI communication.
UART Pins	TXD0, RXD0, TXD2, RXD2	NodeMCU has two UART interfaces, UART0 (RXD0 & TXD0) and UART1 (RXD1 & TXD1). UART1 is used to upload the firmware/program.
I2C Pins		NodeMCU has I2C functionality support but due to the internal functionality of these pins, you have to find which pin is I2C.

NodeMCU ESP8266 Specifications & Features

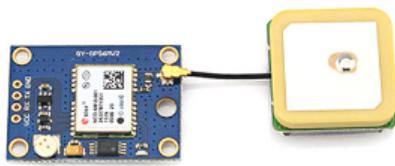
- Microcontroller: Tensilica 32-bit RISC CPU Xtensa LX106
- Operating Voltage: 3.3V
- Input Voltage: 7-12V
- Digital I/O Pins (DIO): 16
- Analog Input Pins (ADC): 1
- UARTs: 1
- SPIs: 1
- I2Cs: 1
- Flash Memory: 4 MB
- SRAM: 64 KB
- Clock Speed: 80 MHz
- USB-TTL based on CP2102 is included onboard, Enabling Plug n Play
- PCB Antenna
- Small Sized module to fit smartly inside your IoT projects

2. Data Sheet GPS

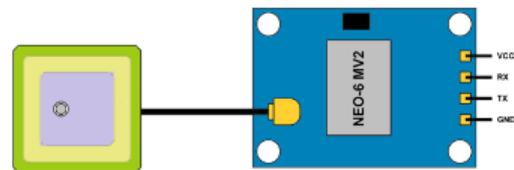
NEO-6MV2 GPS Module

📅 16 October, 2018

🗨️ 0



NEO-6MV2 GPS Module



NEO-6MV2 GPS Module Pinout

NEO-6MV2 GPS Module Pin Configuration

The module has four output pins and we will describe the function each pin of them below. The powering of module and communication interface is done through these four pins.

Pin Name	Description
VCC	Positive power pin
RX	UART receive pin
TX	UART transmit pin
GND	Ground

Features and Electrical Characteristics

- Standalone GPS receiver
- Anti-jamming technology
- UART Interface at the output pins (Can use SPI ,I2C and USB by soldering pins to the chip core)
- Under 1 second time-to-first-fix for hot and aided starts
- Receiver type: 50 Channels - GPS L1 frequency - SBAS (WAAS, EGNOS, MSAS, GAGAN)
- Time-To-First-fix: For Cold Start 32s, For Warm Start 23s, For Hot Start <1s
- Maximum navigation update rate: 5Hz
- Default baud rate: 9600bps
- EEPROM with battery backup
- Sensitivity: -160dBm
- Supply voltage: 3.6V
- Maximum DC current at any output: 10mA
- Operation limits: Gravity-4g, Altitude-50000m, Velocity-500m/s
- Operating temperature range: -40°C TO 85°C

3. Data Sheet Driver Motor L298N



Handson Technology

User Guide

L298N Dual H-Bridge Motor Driver

This dual bidirectional motor driver, is based on the very popular L298 Dual H-Bridge Motor Driver Integrated Circuit. The circuit will allow you to easily and independently control two motors of up to 2A each in both directions. It is ideal for robotic applications and well suited for connection to a microcontroller requiring just a couple of control lines per motor. It can also be interfaced with simple manual switches, TTL logic gates, relays, etc. This board equipped with power LED indicators, on-board +5V regulator and protection diodes.

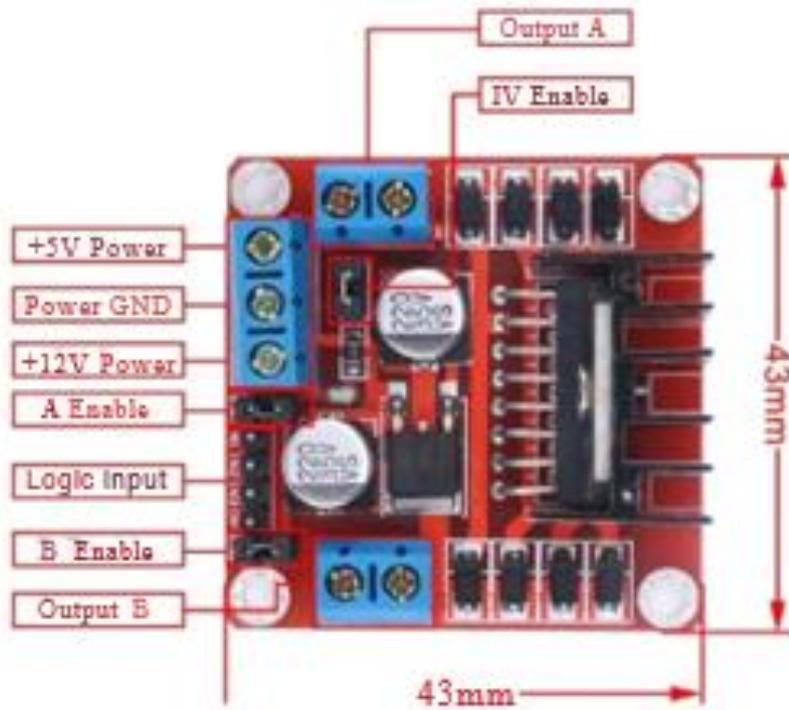


SKU: MDU-1049

Brief Data:

- Input Voltage: 3.2V~40Vdc.
- Driver: L298N Dual H Bridge DC Motor Driver
- Power Supply: DC 5 V - 35 V
- Peak current: 2 Amp
- Operating current range: 0 ~ 36mA
- Control signal input voltage range :
 - Low: $-0.3V \leq V_{in} \leq 1.5V$.
 - High: $2.3V \leq V_{in} \leq V_{ss}$.
- Enable signal input voltage range :
 - Low: $-0.3 \leq V_{in} \leq 1.5V$ (control signal is invalid).
 - High: $2.3V \leq V_{in} \leq V_{ss}$ (control signal active).
- Maximum power consumption: 20W (when the temperature $T = 75^\circ C$).
- Storage temperature: $-25^\circ C \sim +130^\circ C$.
- On-board +5V regulated Output supply (supply to controller board i.e. Arduino).
- Size: 3.4cm x 4.3cm x 2.7cm

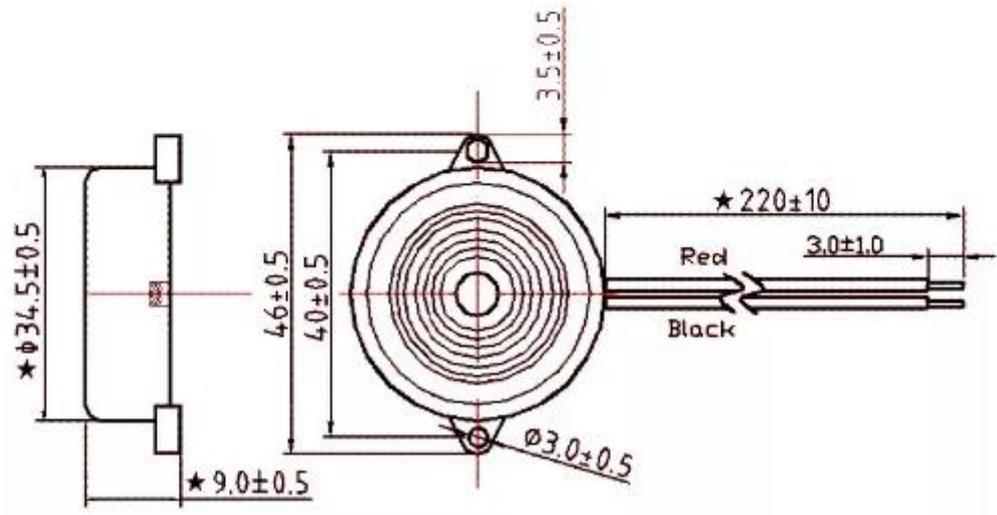
Board Dimension & Pin Function:



4. Data Sheet Buzzer

	Piezo Buzzer	
	www.swt99.com	Email: swt9@swt99.com

Part No:	PSE3590+1012WC
Rated Voltage (Vp-p)	12
Operating Voltage (Vp-p/max)	1--25
Max. Current Consumption (mA)	20
Min. Sound Pressure level at30cm (dB)	87 (12Vp-p square wave1.0kHz)
Resonant Frequency (KHZ)	1.0±0.5
Capacitance at 120Hz (nF)	50±30%
Operating Temperature (°C)	-20~+70
Storage Temperature (°C)	-30~+80
Housing Material	ABS



线材:1007#30