

BAB 2

TINJAUAN PUSTAKA

2.1 Penelitian Terdahulu

Terdapat beberapa penelitian terdahulu yang menjadi referensi penulis dalam mengembangkan sistem pemantauan kondisi ban kendaraan bermotor pada penelitian tersebut, seperti yang dijabarkan pada Tabel 2.1.

Tabel 2.1 Penelitian terdahulu

No.	Penelitian	Ringkasan	Kelebihan dan Kekurangan
1.	<i>Wireless Monitoring System for Motorcycle Tire Air Pressure with Pressure Sensor and Voice Warning on Helmet using Fuzzy Logic</i> (Tawakal, M. I.) [6]	Penelitian ini membangun sistem pemantauan tekanan ban sepeda motor nirkabel berbasis logika <i>fuzzy</i> dengan menggunakan Arduino Uno, sensor MPX5700AP, dan sensor LM35, serta modul NRF24L01. Penelitian tersebut berfokus pada algoritma logika <i>fuzzy</i> Sugeno. Sistem yang dibangun dalam penelitian tersebut memiliki akurasi sebesar 86.6%.	Kelebihan Penelitian tersebut membandingkan akurasi logika <i>fuzzy</i> Sugeno dengan dan tanpa parameter suhu. Kekurangan Penelitian ini tidak menjabarkan secara detail mengenai <i>defuzzifikasi</i> dan evaluasi logika <i>fuzzy</i> .
2.	Rancang Bangun Alat <i>Monitoring</i> Tekanan Angin Ban secara <i>Real Time</i> Menggunakan Metode Tsukamoto pada Kendaraan Roda Empat (Setiawan, R. A.) [7]	Penelitian ini merancang dan mengimplementasikan alat pemantauan tekanan angin ban pada kendaraan roda empat secara <i>real-time</i> menggunakan logika <i>fuzzy</i> Tsukamoto. Sistem tersebut menggunakan Arduino Mega 2560, Arduino Uno, sensor MPX5700AP, sensor LM35, dan modul NRF24L01. Penelitian tersebut membandingkan data yang diperoleh dari sistem dengan hasil kuesioner mekanik,	Kelebihan Penelitian tersebut menjelaskan penerapan logika <i>fuzzy</i> pada sistem pemantauan ban dengan baik. Kekurangan Penelitian tersebut menggunakan cara yang konvensional dalam mengevaluasi metode logika <i>fuzzy</i> yang digunakan.

		dan akurasi yang diperoleh sebesar 85%.	
3.	Mon-Tire: <i>Portable Tire Pressure Monitoring System</i> (Kurniawan, R. M.) [8]	Penelitian ini membuat sistem pemantauan tekanan ban kendaraan bermotor roda dua secara <i>real-time</i> menggunakan sensor MD-PS002 dan mikrokontroler ESP32, yang dihubungkan dengan aplikasi pada <i>smartphone</i> menggunakan Bluetooth. Penelitian tersebut menggunakan metode regresi linear untuk mencari korelasi atau hubungan dari nilai tekanan dengan nilai sinyal keluaran sensor. Sistem yang dibuat dalam penelitian tersebut memiliki nilai kesalahan sebesar 2.7%.	<p>Kelebihan Penelitian tersebut mengkaji proses perancangan dan implementasi sistem pemantauan tekanan ban dengan sangat baik.</p> <p>Kekurangan Sistem yang dibangun memiliki nilai <i>error</i> yang cukup berarti.</p>
4.	<i>Development of a Frugal Onboard Tire Pressure Monitoring Control System</i> (Pai, A.) [9]	Penelitian ini membangun sistem pemantauan tekanan ban kendaraan dengan menggunakan Arduino Uno, sensor tekanan BMP180, dan RF <i>transmitter-receiver module</i> . Data yang diperoleh ditampilkan pada LCD 16x2 secara <i>real-time</i> . Penelitian tersebut menampilkan data rata-rata tekanan ban pada kondisi permukaan jalan yang berbeda, yaitu jalan beton, jalan beraspal, dan jalan tidak beraspal. Namun, penelitian tersebut tidak menjelaskan secara detail mengenai perhitungan yang dilakukan.	<p>Kelebihan Penelitian tersebut menghasilkan sistem pemantauan kondisi ban menggunakan sensor BMP180 yang dapat menunjukkan perbedaan rata-rata tekanan ban pada kondisi permukaan yang berbeda.</p> <p>Kekurangan Desain alat dalam penelitian tersebut tidak kompak dan menyebabkan pergerakan roda kendaraan menjadi terbatas.</p>
5.	Perancangan Alat Monitoring Tekanan Udara di Dalam Ban Kendaraan Bermotor Roda	Penelitian ini merancang dan membangun sistem pemantauan tekanan udara pada ban kendaraan roda empat dengan Arduino	<p>Kelebihan Alat yang dibangun memiliki desain yang kompak dan akurasi yang bagus dalam</p>

	Empat Menggunakan Sensor Tekanan Udara Berbasis Arduino (Novrizaldi, M. A.) [10]	Mega 2560, Arduino Nano, sensor tekanan MPX5700AP, dan modul NRF24L01. Sensor MPX5700 dalam penelitian tersebut dipasang pada ban dan data yang diperoleh dikirimkan ke <i>dashboard</i> kendaraan melalui modul NRF24L01 untuk diproses menggunakan Arduino Mega 2560 dan ditampilkan pada LCD. Sistem yang dibangun memiliki akurasi yang baik, yaitu 99.49%, dengan <i>error</i> sebesar 0.51%.	memantau tekanan ban. Kekurangan Penelitian tersebut tidak menjelaskan secara detail mengenai metode regresi linear yang digunakan.
--	--	--	--

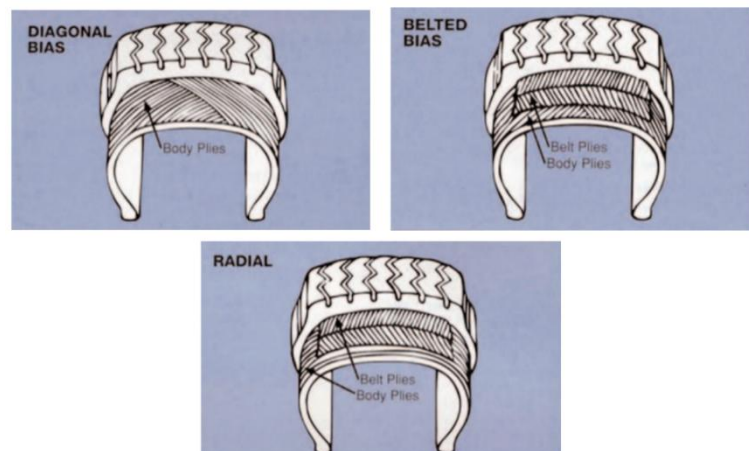
Berdasarkan analisis penelitian-penelitian terdahulu dan kajian teori yang dilakukan, penelitian tersebut mengembangkan solusi untuk mengatasi keterbatasan dan mengisi celah penelitian yang ada. Penelitian tersebut menerapkan logika *fuzzy* Sugeno dengan tiga parameter *input*, yaitu tekanan ban, suhu lingkungan, dan kecepatan rotasi ban, serta satu parameter *output*, yaitu kondisi ban. Logika *fuzzy* Sugeno dengan penambahan parameter *input* dilakukan dengan harapan dapat meningkatkan kemampuan model dalam mengidentifikasi kondisi ban dengan lebih baik.

Kemudian, berdasarkan kajian teori dan analisis penelitian terdahulu yang dilakukan, sensor tekanan MPX5700AP memiliki akurasi yang lebih baik dibandingkan sensor tekanan lainnya. Mikrokontroler ESP32 juga dipilih agar komunikasi antar-perangkat *transmitter* dan *receiver* menjadi lebih efektif, karena ESP32 sudah memiliki modul komunikasi *built-in* dan tidak memerlukan modul komunikasi tambahan, seperti NRF24L01, sehingga dapat mendukung desain alat yang lebih kompak. Dengan demikian, sistem pemantauan kondisi ban berbasis ESP32 berdasarkan parameter tekanan, suhu, dan kecepatan rotasi ban dengan menggunakan logika *fuzzy* Sugeno yang dirancang dan diimplementasikan pada kendaraan bermotor roda dua dalam penelitian tersebut diharapkan dapat menghasilkan sistem yang dapat memantau kondisi ban dengan optimal.

2.2 Ban Kendaraan

Ban adalah salah satu komponen penting dalam kendaraan yang berfungsi untuk menopang beban, meneruskan tenaga dari mesin ke jalan, serta memberikan kenyamanan dan kestabilan selama berkendara. Secara umum, ban merupakan struktur elastis berbentuk cincin yang terbuat dari karet dan bahan lainnya seperti kawat baja dan serat sintetis. Ban dirancang untuk memberikan traksi yang baik, meredam getaran, dan memastikan kontrol kendaraan yang optimal [11], [12].

Ada beberapa jenis ban kendaraan yang umum digunakan, di antaranya ban bias (*bias ply*), ban radial, dan ban *tubeless*. Ban bias memiliki lapisan kawat yang saling menyilang, memberikan daya tahan yang tinggi, namun kurang optimal pada kecepatan tinggi. Berbeda dengan ban bias, ban radial memiliki lapisan kawat yang tegak lurus terhadap lingkaran ban, sehingga memberikan kenyamanan lebih baik dan efisiensi bahan bakar yang lebih tinggi. Sementara itu, ban *tubeless* tidak menggunakan ban dalam sehingga lebih tahan bocor dan lebih aman digunakan [4].

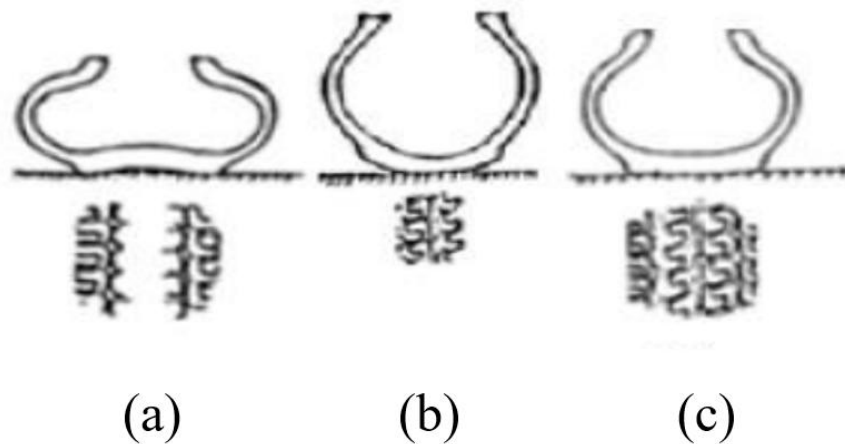


Gambar 2.1 Jenis-jenis ban

Sumber: NHTSA, 2005

Ban memiliki berbagai macam ukuran, yang disesuaikan dengan jenis kendaraan. Ukuran ban tidak hanya mengacu kepada dimensi ban, tetapi juga tekanan udara pada ban. Pada kendaraan bermotor, tekanan ban umumnya berada dalam rentang 22–36 psi, atau lebih spesifiknya; 22–25 psi untuk sepeda motor *trail*; 26–29 psi untuk sepeda motor transmisi otomatis (*matic*) dan manual (*underbone*/motor bebek); dan 32–36 psi untuk motor *sport* [13].

Tekanan ban merupakan salah satu faktor yang krusial dalam berkendara. Ban dengan tekanan normal akan memberikan kontak optimal antara ban dan jalan, sehingga meningkatkan efisiensi bahan bakar dan mengurangi keausan [14]. Sebaliknya, ban yang kempes (kurang tekanan) akan meningkatkan gesekan, menyebabkan ban cepat aus, dan berisiko menyebabkan kecelakaan. Sementara itu, ban yang terlalu keras (kelebihan tekanan) dapat mengurangi daya cengkeram, sehingga menyebabkan kendaraan menjadi kurang stabil dan meningkatkan risiko ban pecah.



Gambar 2.2 Ilustrasi ban dengan tekanan udara yang berbeda-beda

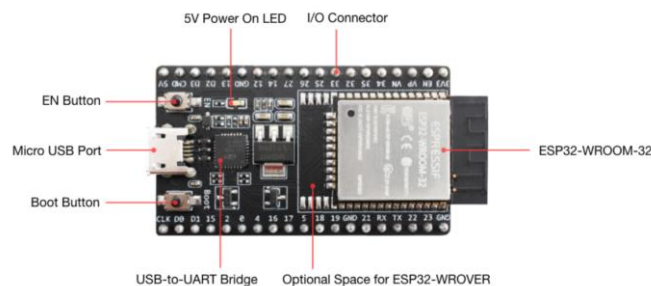
Sumber: Setyawan, 2019

Ilustrasi kondisi ban berdasarkan perbedaan tekanan udara dapat dilihat pada Gambar 2.2. Bagian (a) menunjukkan sebuah ban dengan kondisi tekanan udara yang kurang, sedangkan bagian (b) merupakan ban dalam kondisi tekanan udara yang terlalu tinggi, dan bagian (c) adalah ban dengan kondisi tekanan udara yang normal [11]. Ilustrasi pada Gambar 2.2 menunjukkan bahwa perbedaan kondisi tekanan udara pada ban memengaruhi bidang sentuh antara ban dan jalan. Ketika tekanan udara pada ban berada dalam kondisi normal, ban akan memiliki bidang sentuh yang cukup terhadap jalan. Tekanan udara yang terlalu tinggi akan mengurangi bidang sentuh, sementara tekanan angin yang terlalu rendah akan meningkatkan bidang sentuh. Perbedaan bidang sentuh ban terhadap jalan tersebut akan mengakibatkan perbedaan gaya gesek yang dialami ban, yang mana hal tersebut dapat memengaruhi kondisi ban.

Selain tekanan udara, suhu lingkungan dan kecepatan kendaraan juga merupakan dua faktor signifikan memengaruhi kondisi ban. Saat cuaca panas, temperatur aspal yang meningkat menyebabkan gesekan antara ban dan permukaan jalan menjadi lebih besar. Gesekan ini dapat memanaskan udara di dalam ban, sehingga tekanannya meningkat dan mendekati batas maksimal yang ditentukan pabrikan. Kondisi ini meningkatkan risiko ban pecah, terutama jika tekanan awal ban sudah melebihi standar. Sebaliknya, tekanan ban yang terlalu rendah juga menimbulkan masalah karena bagian tepi ban akan lebih banyak bersentuhan dengan jalan, mempercepat keausan dan meningkatkan risiko kerusakan struktural. Pada kecepatan tinggi, gesekan yang terjadi lebih intensif sehingga panas yang dihasilkan lebih besar, memperparah efek dari tekanan yang tidak sesuai [14], [15]. Oleh karena itu, menjaga tekanan ban dalam kisaran optimal dan memeriksa kondisi ban secara rutin sangat penting untuk menjaga keselamatan berkendara.

2.3 Mikrokontroler ESP32

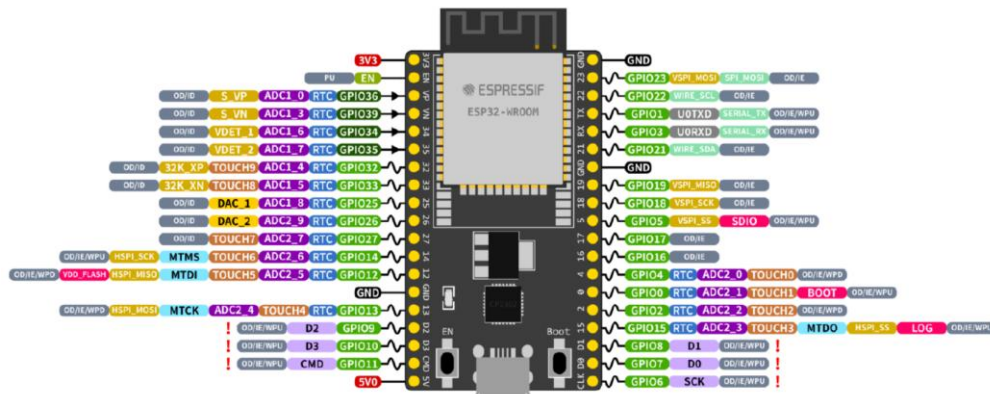
ESP32 adalah mikrokontroler *system on chip* (SoC) yang dikembangkan oleh Espressif Systems dengan desain teknologi prosesor 40 nm yang dibuat oleh TSMC [16]. ESP32 merupakan penerus mikrokontroler ESP8266 yang dilengkapi dengan fitur mode ganda, yaitu Wi-Fi dan Bluetooth, yang dapat digunakan untuk berbagai sistem aplikasi dan proyek berbasis *Internet of Things* (IoT). Selain itu, ESP32 juga dilengkapi dengan berbagai fitur tambahan dan keunggulan dibandingkan generasi sebelumnya, seperti inti CPU serta Wi-Fi yang lebih cepat, dukungan terhadap Bluetooth 4.2, *general purpose input output* (GPIO) yang lebih banyak, dan konsumsi daya yang rendah [17]. Komponen mikrokontroler ESP32 dapat dilihat pada Gambar 2.3.



Gambar 2.3 Bagian ESP32

Sumber: Espressif, 2024

ESP32 menggunakan mikroprosesor Tensilica Xtensa LX6 *dual-core* atau *single-core* dengan *clock rate* hingga 240 MHz. Dengan dimensi yang kecil ($25.5 \times 18.0 \times 2.8$ mm), ESP32 menggunakan modul yang komponen-komponennya sudah terintegrasi dengan *built-in antenna switches*, *power amplifier*, *low-noise receive amplifier*, *filters*, dan *power management modules*. Selain itu, ESP32 memiliki hingga 34 GPIO (*general-purpose input-output*) yang dapat dikonfigurasi sebagai *input*, *output*, ADC (*analog to digital converter*), DAC (*digital to analog converter*), PWM (*pulse width modulation*), I2C (*inter-integrated circuit*), SPI (*serial peripheral interface*), UART (*universal asynchronous receiver transmitter*), dan fungsi lainnya. Jumlah dan fungsi *pin* tersebut dapat bervariasi, tergantung pada jenis modul ESP32 yang digunakan, seperti ESP32-WROOM-32 atau ESP32-WROVER. *Pinout* ESP32 dapat dilihat pada Gambar 2.4.



Gambar 2.4 *Pinout* ESP32

Sumber: Espressif, 2024

2.4 Sensor Tekanan MPX5700AP

Sensor MPX5700AP adalah sensor tekanan absolut berbasis silikon yang dirancang untuk mengukur tekanan udara atau gas dalam berbagai aplikasi. Sensor tersebut menggunakan satu *port* untuk mendeteksi tekanan relatif terhadap ruang vakum internalnya. Dengan dilengkapi teknologi pengondisian sinyal dan kalibrasi suhu internal, sensor tersebut dapat memberikan pengukuran yang akurat dan stabil, bahkan dalam kondisi suhu yang beragam [18], [19]. Bentuk sensor MPX5700AP dapat dilihat pada Gambar 2.5.



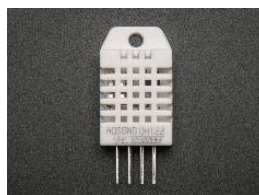
Gambar 2.5 Sensor MPX5700AP

Sumber: Waworundeng, 2019

Sensor MPX5700AP memiliki tingkat kesalahan maksimum sebesar 2.5% pada suhu 0 °C hingga 85 °C dan dapat mengukur tekanan dalam rentang 15 kPa hingga 700 kPa (2.18 hingga 101.5 Psi). Sensor ini beroperasi pada tegangan suplai 4.75 VDC hingga 5.25 VDC, serta mampu bekerja dalam suhu lingkungan dari -40 °C hingga 125 °C [18]. Dengan akurasi yang tinggi dan kemampuannya untuk bekerja pada berbagai kondisi, sensor tersebut sering digunakan untuk memonitor tekanan udara pada berbagai objek, termasuk tekanan udara pada ban kendaraan.

2.5 Sensor Suhu DHT22

Sensor DHT22 merupakan sensor digital yang dapat mengukur suhu dan kelembapan relatif udara. Sensor tersebut bekerja dengan mengeluarkan sinyal digital hasil pengukuran yang telah terkalibrasi secara digital, sehingga dapat digunakan tanpa memerlukan pengolahan data tambahan. Selain itu, beberapa hasil penelitian mengklaim bahwa sensor DHT22 memiliki nilai akurasi yang lebih baik dibandingkan dengan sensor lainnya yang dapat mengukur suhu, seperti sensor DHT11 dan LM35 [20].



Gambar 2.6 Sensor DHT22

Sumber: Puspasari, 2020

Sensor DHT22 terdiri dari empat *pin*, yaitu VCC, SDA, NC, dan GND. Secara teknis, sensor tersebut mengandalkan elemen kapasitif untuk mengukur kelembapan dan termistor untuk suhu. Selain dikenal memiliki akurasi yang baik, sensor tersebut juga memiliki respons akuisisi data yang cepat, sehingga sangat mudah digunakan dengan mikrokontroler [21].

2.6 Sensor *Hall Effect* SS41F

Sensor SS41F merupakan sensor *hall effect* yang dapat mendeteksi medan magnet dengan tingkat sensitivitas tinggi. Sensor tersebut bekerja dengan mendeteksi keberadaan medan magnet dari magnet permanen atau elektromagnet. Ketika medan magnet yang memenuhi ambang batas terdeteksi oleh sensor SS41F, status dari *output* sensor akan berubah secara digital (*HIGH/LOW*) [22].



Gambar 2.7 Sensor SS41F

Sumber: Larin, 2022

Sensor *hall effect* SS41F menghasilkan sinyal digital yang dapat langsung diproses oleh mikrokontroler tanpa memerlukan rangkaian tambahan, seperti ADC. Dengan demikian, sensor tersebut dapat digunakan untuk aplikasi yang memerlukan deteksi perubahan status berbasis *hall effect*, misalnya seperti pada pengukuran kecepatan putaran motor [23].

2.7 LCD 20x4 I2C

Liquid crystal display (LCD) 20x4 adalah modul layar yang dapat menampilkan karakter dalam format matriks 20 kolom dan 4 baris. Modul tersebut mampu menampilkan hingga 80 karakter secara bersamaan, sehingga cocok untuk aplikasi yang membutuhkan tampilan informasi yang cukup banyak sekaligus [24]. Bentuk LCD 20x4 dan modul I2C dapat dilihat pada Gambar 2.8.



Gambar 2.8 LCD 20x4

Sumber: Sulistyawan, 2021

LCD 20x4 biasanya menggunakan teknologi *backlight* untuk meningkatkan keterbacaan dalam kondisi pencahayaan yang rendah. Modul tersebut juga kompatibel dengan berbagai mikrokontroler dan dilengkapi dengan modul *interface inter-integrated circuit* (I2C) yang memungkinkan komunikasi lebih sederhana dengan hanya dua jalur data, yaitu *serial data* (SDA) dan *serial clock* (SCL) [25].

2.8 Buzzer

Buzzer adalah komponen elektronika yang berfungsi mengubah energi listrik menjadi gelombang bunyi. Prinsip kerjanya mirip dengan *loudspeaker*, di mana arus listrik dialirkan melalui elektromagnet yang memengaruhi gelombang pada diafragma. Gelombang tersebut akan tertarik ke luar atau ke dalam bergantung pada polaritas magnet dan arah arus, sehingga menghasilkan gerakan yang memicu getaran udara untuk menghasilkan suara [26].



Gambar 2.9 Buzzer

Sumber: Natsir, 2019

Buzzer biasanya memiliki dua kaki, yaitu positif dan negatif, yang dihubungkan ke tegangan listrik sesuai spesifikasinya, biasanya dalam rentang 3V hingga 12V. Dengan frekuensi suara berkisar antara 1–5 kHz, *buzzer* sering dimanfaatkan sebagai indikator proses selesai atau peringatan adanya kesalahan dalam sistem [27].

2.9 ESP-NOW

ESP-NOW adalah protokol komunikasi nirkabel yang memungkinkan pertukaran data secara langsung antar perangkat berbasis ESP32 atau ESP8266 tanpa memerlukan koneksi Wi-Fi. Protokol komunikasi tersebut dikembangkan oleh Espressif dan bersifat *connectionless*, yang berarti bahwa perangkat tidak perlu membentuk koneksi secara formal sebelum mengirim data, sehingga proses komunikasi menjadi lebih cepat dan efisien. Identifikasi antar-perangkat dilakukan melalui *MAC address*, yang berfungsi sebagai alamat unik untuk setiap perangkat ESP yang terlibat dalam komunikasi. Setiap paket data dalam

ESP-NOW dikemas dalam bentuk *action frame* khusus dan dapat memuat *payload* hingga 250 byte. ESP-NOW mendukung komunikasi *unicast* (satu pengirim ke satu penerima) dan juga dapat diatur untuk skenario *multicast* dalam aplikasi tertentu. Dalam mode *station*, ESP-NOW dapat menangani hingga 10 *peer* terenkripsi, yang artinya komunikasi dapat diamankan dengan metode *pairing* sebelumnya. Selain itu, satu perangkat ESP juga dapat menangani hingga 20 *peer* secara keseluruhan dalam mode campuran (*mixed mode*) [28], [29].

Keunggulan utama ESP-NOW adalah kemampuannya untuk mempertahankan koneksi antar perangkat, bahkan setelah terjadi *restart*, selama proses *pairing* telah dilakukan sebelumnya dan tidak dihapus dari memori perangkat. Dengan *bit rate default* 1 Mbps, protokol ini memiliki kecepatan yang memadai untuk pertukaran data ringan hingga menengah, serta sangat cocok untuk digunakan dalam sistem pengambilan data dari *node* sensor di lingkungan yang tidak memiliki jaringan lokal (Wi-Fi *router*). Penelitian [30] menunjukkan bahwa ESP-NOW mampu melakukan komunikasi dua arah di antara dua ESP32 dengan stabil. Perangkat-perangkat tersebut juga dapat saling berganti peran, baik sebagai *master* (pengirim data), maupun *slave* (penerima data). Selain itu, ESP-NOW juga memiliki performa komunikasi yang efisien dengan jangkauan optimal hingga 25 meter, dan bahkan mampu mencapai hingga 50 meter, meskipun dengan peningkatan risiko gangguan transmisi karena faktor lingkungan, seperti hambatan fisik atau interferensi sinyal.

2.10 Logika *Fuzzy*

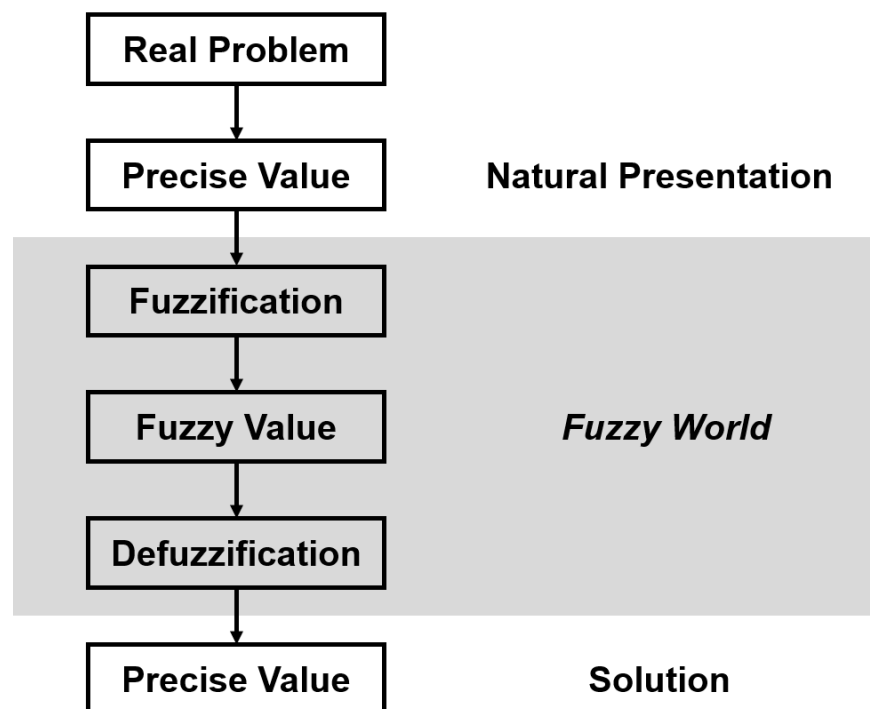
2.10.1 Pengertian Logika *Fuzzy*

Logika *fuzzy* adalah sebuah metode yang memungkinkan komputer memahami deskripsi parameter dengan lebih fleksibel, di mana parameter tidak dianggap sebagai nilai pasti yang presisi, melainkan sebagai nilai yang bersifat bias (*fuzzy*). Secara definitif, logika *fuzzy* dapat diartikan sebagai metode yang memungkinkan penggunaan bahasa alamiah dalam logika [31]. Logika *fuzzy* dikembangkan oleh Lotfi Azker Zadeh, seorang ilmuwan Amerika Serikat berkebangsaan Iran, melalui tulisannya pada tahun 1965 tentang teori himpunan *fuzzy* [32].

Saat ini, logika *fuzzy* telah diterapkan dalam berbagai bidang, mulai dari teori sistem kendali hingga kecerdasan buatan (*artificial intelligence*). Secara umum, logika *fuzzy* digunakan untuk menangani berbagai permasalahan yang melibatkan ketidakpastian, gangguan (*noise*), dan lain sebagainya. Selain itu, logika *fuzzy* juga berperan sebagai penghubung antara bahasa mesin yang sangat presisi dan bahasa manusia yang lebih berfokus pada makna dan interpretasi [33].

2.10.2 Tahapan Logika *Fuzzy*

Data yang diperoleh dari alam sering kali berbentuk nilai pasti, atau disebut juga sebagai *crisp input* (*input* mutlak). Nilai tersebut kemudian perlu diubah menjadi nilai *fuzzy* (bias) melalui proses *fuzzifikasi*. Selanjutnya, nilai *fuzzy* tersebut diolah melalui proses *defuzzifikasi* untuk memperoleh kembali sebuah nilai pasti, atau disebut juga sebagai *crisp output* (*output* mutlak) [31]. Tahapan algoritma logika *fuzzy* dapat dilihat pada Gambar 2.10.



Gambar 2.10 Tahapan algoritma logika *fuzzy*

Sumber: Utama, 2021

2.10.3 Himpunan *Fuzzy*

Himpunan *fuzzy* merupakan sekumpulan elemen yang merepresentasikan kondisi atau keadaan tertentu dalam sebuah variabel *fuzzy* [34]. Himpunan *fuzzy* memiliki dua atribut utama, yaitu:

a) Linguistik

Linguistik merujuk pada penamaan kelompok yang menggambarkan suatu kondisi tertentu menggunakan bahasa alami. Contoh: rendah, sedang, dan tinggi.

b) Numerik

Numerik adalah nilai berupa angka yang merepresentasikan ukuran dari suatu variabel. Contoh: 10, 20, 30, dan sebagainya.

Selain itu, terdapat beberapa konsep dasar dalam logika *fuzzy* yang perlu dipahami, yaitu:

a) Variabel *Fuzzy*

Variabel *fuzzy* adalah variabel yang dianalisis dalam sebuah sistem *fuzzy*. Contoh: penghasilan (rendah, sedang, tinggi), umur (muda, dewasa, tua), temperatur (dingin, hangat, panas), dan sebagainya.

b) Semesta Pembicaraan

Semesta pembicaraan mencakup seluruh nilai yang dapat digunakan dalam sebuah variabel *fuzzy*. Contoh: semesta pembicaraan untuk variabel temperatur adalah [10, 100].

c) Domain Himpunan *Fuzzy*

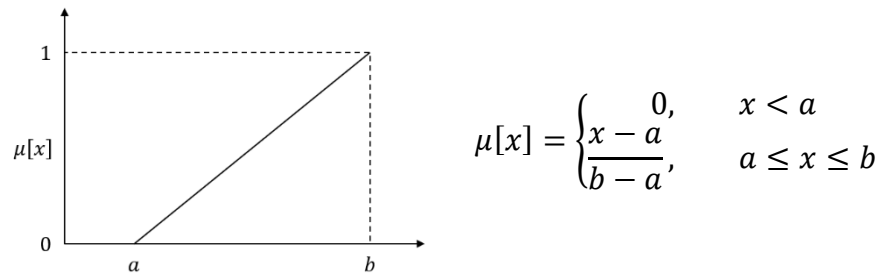
Domain himpunan *fuzzy* merupakan nilai-nilai dalam semesta pembicaraan yang dapat digunakan untuk membentuk sebuah himpunan *fuzzy* tertentu.

2.10.4 Fungsi Keanggotaan *Fuzzy*

Fungsi keanggotaan, atau *membership function*, adalah representasi grafis yang menunjukkan derajat keanggotaan (*membership degree*) dari setiap variabel *input* pada rentang nilai dari 0 hingga 1. Secara praktis, fungsi tersebut digunakan sebagai pendekatan untuk menentukan nilai dari derajat keanggotaan domain himpunan *fuzzy* [35]. Terdapat beberapa fungsi keanggotaan yang umum digunakan, yaitu:

a) Representasi Linear Naik

Pada representasi linear naik, derajat keanggotaan meningkat secara linier seiring bertambahnya nilai *input*. Grafiknya berbentuk garis lurus yang dimulai dari domain dengan derajat keanggotaan nol dan naik menuju domain dengan derajat keanggotaan yang lebih tinggi.

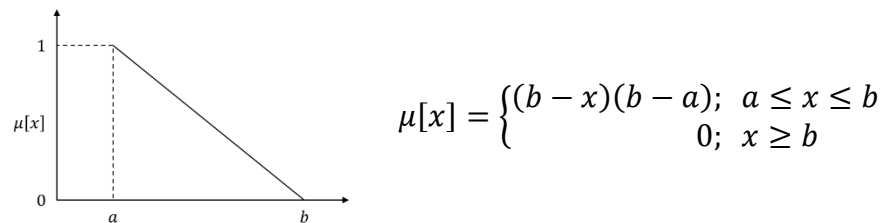


Gambar 2.11 Grafik representasi linear naik dan persamaannya

Sumber: Panjaitan, 2021

b) Representasi Linear Turun

Representasi ini menggambarkan penurunan derajat keanggotaan secara linier. Grafik dimulai dari domain dengan derajat keanggotaan tertinggi, kemudian turun menuju domain yang derajat keanggotaannya rendah.

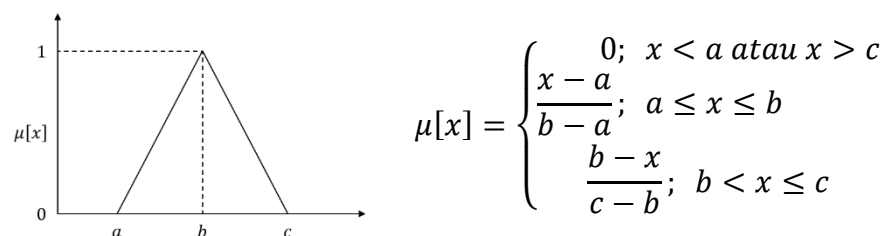


Gambar 2.12 Grafik representasi linear turun dan persamannya

Sumber: Panjaitan, 2021

c) Representasi Kurva Segitiga

Kurva segitiga merupakan kombinasi dari representasi linear naik dan linear turun, yang membentuk grafik segitiga.

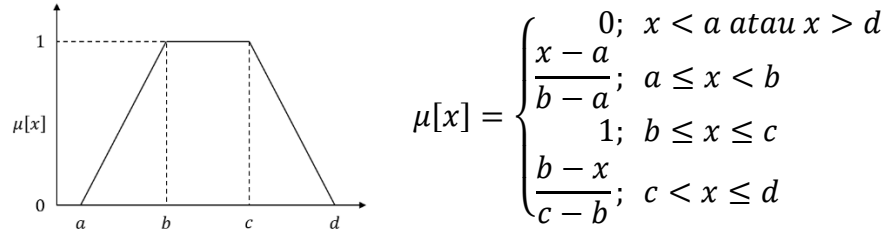


Gambar 2.13 Grafik representasi kurva segitiga dan persamannya

Sumber: Panjaitan, 2021

d) Representasi Kurva Trapesium

Kurva trapesium mirip dengan kurva segitiga, namun memiliki bagian datar pada titik-titik tertentu, di mana nilai keanggotaan bernilai satu.

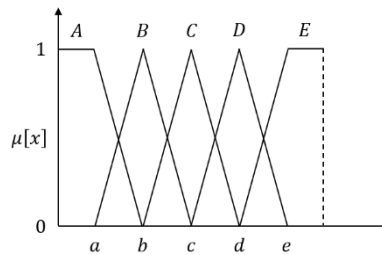


Gambar 2.14 Grafik representasi kurva trapesium dan persamaannya

Sumber: Panjaitan, 2021

e) Representasi Kurva Bentuk Bahu

Kurva bahu digunakan untuk menggambarkan daerah *fuzzy* dengan nilai derajat keanggotaannya yang tetap konstan di ujung grafik.

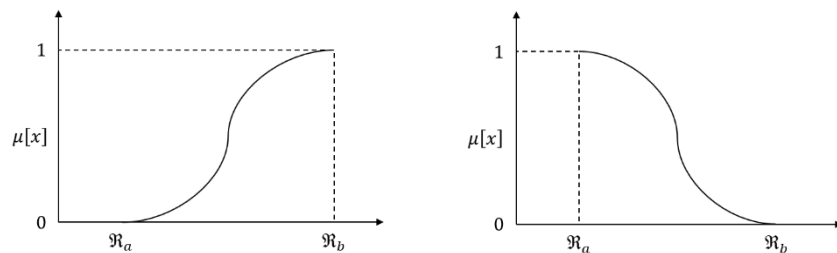


Gambar 2.15 Grafik representasi kurva bentuk bahu

Sumber: Panjaitan, 2021

f) Representasi Kurva Sigmoid (Kurva-S)

Kurva-S menunjukkan perubahan derajat keanggotaan secara non-linear. Pada kurva-S naik, grafik bergerak dari sisi kiri (derajat keanggotaan nol) menuju sisi kanan (derajat keanggotaan satu). Sebaliknya, pada kurva-S turun, grafik bergerak dari sisi kanan (derajat keanggotaan satu) menuju sisi kiri (derajat keanggotaan nol).



Gambar 2.16 Grafik representasi kurva sigmoid

Sumber: Panjaitan, 2021

2.10.5 Basis Aturan *Fuzzy* dan Fungsi Implikasi *Fuzzy*

Basis aturan *fuzzy*, atau disebut juga sebagai *fuzzy rule base*, merupakan komponen penting dalam penerapan logika *fuzzy* secara menyeluruh untuk menyelesaikan permasalahan klasik. Selain proses *fuzzifikasi* dan *defuzzifikasi* yang bertujuan mengubah nilai *crisp-fuzzy-crisp*, basis aturan *fuzzy* diperlukan sebagai kerangka analisis yang menggambarkan hubungan antar-parameter dalam bentuk linguistik, yang menyerupai pola pemikiran manusia. Basis aturan tersebut mengombinasikan komponen-komponen penentuan *fuzzy* dengan aturan tertentu yang telah ditetapkan [31].

Dasar dari basis aturan *fuzzy* adalah logika matematika sederhana yang dikenal sebagai koneksi logika *fuzzy*. Koneksi ini terdiri dari dua jenis utama, yaitu *fuzzy conjunction connection* (AND) dan *fuzzy disjunction connection* (OR). Koneksi AND akan bernilai benar apabila semua kondisi yang terhubung bernilai benar; jika salah satu atau semua kondisi salah, maka hasilnya bernilai salah. Sebaliknya, koneksi OR akan bernilai benar jika minimal satu kondisi bernilai benar; apabila semua kondisi salah, hasilnya juga bernilai salah. Dalam koneksi AND, nilai terkecil (minimum) dari kondisi yang terhubung dipilih sebagai hasil, sedangkan pada koneksi OR, nilai terbesar (maksimum) yang menjadi hasil.

Seperti yang telah dijelaskan pada paragraf sebelumnya, setiap aturan atau proposisi pada basis pengetahuan *fuzzy* akan berhubungan dengan suatu relasi *fuzzy*. Bentuk umum dari aturan yang digunakan dalam fungsi implikasi adalah:

$$IF\ x\ is\ A,\ THEN\ y\ is\ B,$$

dengan x dan y adalah skalar, serta A dan B adalah himpunan *fuzzy*. Proposisi yang mengikuti *IF* disebut sebagai anteseden (*antecedent*), sedangkan proposisi yang mengikuti *THEN* disebut sebagai konsekuen (*consequent*). Proposisi tersebut dapat diperluas dengan menggunakan operator *fuzzy*, seperti:

$$IF(x_1\ is\ A_1) \cdot (x_2\ is\ A_2) \cdot (x_3\ is\ A_3) \cdot \dots \cdot (x_n\ is\ A_n),\ THEN\ y\ is\ B,$$

dengan simbol \cdot adalah simbol untuk operator *OR* atau *AND*.

2.10.6 Sistem Inferensi *Fuzzy* Sugeno

Sistem inferensi *fuzzy*, atau disebut juga *fuzzy inference system*, merupakan sebuah kerangka kerja komputasi yang sangat populer dalam penerapan logika *fuzzy*. Sistem ini didasarkan pada tiga komponen utama dari teori *fuzzy*, yaitu teori himpunan *fuzzy*, aturan-aturan *fuzzy* dalam bentuk *if-then*, dan proses penalaran *fuzzy* yang menyerupai cara manusia mengambil keputusan berdasarkan informasi yang tidak pasti atau ambigu. Sistem tersebut memungkinkan pengambilan keputusan yang bersifat toleran terhadap ketidakpastian dan tidak memerlukan data yang sepenuhnya presisi [32].

Pada dasarnya, sistem inferensi *fuzzy* mendefinisikan pemetaan non-linear dari vektor data *input* menjadi *output* skalar melalui proses yang mencakup beberapa tahapan penting. Proses tersebut dimulai dari penggunaan fungsi keanggotaan *fuzzy* untuk *input* dan *output*, lalu dilanjutkan dengan penerapan operator-operator *fuzzy* yang menggabungkan nilai keanggotaan dari setiap variabel, yang kemudian diolah menggunakan aturan *if-then fuzzy* yang telah ditentukan sebelumnya dalam basis aturan (*rule base*). Setelah itu, dilakukan proses agregasi untuk menggabungkan hasil dari beberapa aturan yang aktif, lalu diakhiri dengan proses *defuzzification* (*defuzifikasi*) untuk mengubah hasil *fuzzy* menjadi nilai *output crisp* yang dapat digunakan oleh sistem, misalnya sebagai indikator kondisi, nilai keputusan, dan lain sebagainya [31].

Sistem inferensi *fuzzy* terdiri dari empat komponen utama, yaitu *fuzzifier*, *inference engine* (mesin inferensi), *rule base* (basis aturan), dan *defuzzifier*. *Fuzzifier* berfungsi untuk mengubah *input* numerik atau *crisp* menjadi derajat keanggotaan dalam himpunan *fuzzy*. Kemudian, *inference engine* merupakan mesin inferensi yang melakukan proses evaluasi terhadap aturan *fuzzy* dan menghasilkan *output fuzzy* berdasarkan *input* yang diberikan. Sementara itu, terdapat *rule*, yaitu basis aturan yang berisi sekumpulan aturan *fuzzy if-then* yang digunakan untuk menyusun logika pengambilan keputusan, serta *defuzzifier* yang bertugas mengubah *output fuzzy* menjadi nilai *crisp* sebagai hasil akhir sistem [32].

Terdapat tiga metode dalam sistem inferensi *fuzzy*, yaitu Tsukamoto, Mamdani, dan Sugeno. Penelitian tersebut menggunakan sistem inferensi *fuzzy* Sugeno, yang diperkenalkan oleh Michio Sugeno pada tahun 1985. Metode tersebut memiliki ciri khas, di mana *output* dari aturan *fuzzy* bukan berupa himpunan *fuzzy*, melainkan persamaan linear atau konstanta. Terdapat dua model inferensi Sugeno, yaitu model *fuzzy* Sugeno orde-nol dan model *fuzzy* Sugeno orde-satu [32].

Pada model *fuzzy* Sugeno orde-nol, bentuk umum proposisinya dijabarkan sebagai berikut:

$$IF(x_1 \text{ is } A_1) \cdot (x_2 \text{ is } A_2) \cdot (x_3 \text{ is } A_3) \cdot \dots \cdot (x_n \text{ is } A_n), THEN z = k,$$

dengan A_n adalah himpunan *fuzzy* ke- n sebagai anteseden dan k adalah suatu konstanta *crisp* sebagai konsekuen. Pada model *fuzzy* Sugeno orde-satu, bentuk umum proposisinya dijabarkan sebagai berikut:

$$IF(x_1 \text{ is } A_1) \cdot (x_2 \text{ is } A_2) \cdot (x_3 \text{ is } A_3) \cdot \dots \cdot (x_n \text{ is } A_n),$$

$$THEN z = p_1x_1 + p_2x_2 + \dots + p_nx_n + k,$$

dengan A_n adalah himpunan *fuzzy* ke- n sebagai anteseden dan p_n adalah suatu konstanta *crisp* ke- n dan k juga merupakan konstanta dalam konsekuen [32].

Dalam sistem inferensi *fuzzy* Sugeno, metode defuzzifikasi yang umum digunakan adalah *weighted average* (rata-rata tertimbang) untuk memperoleh nilai *output crisp* [36]. Persamaan untuk menghitung nilai *output* final (*defuzzifikasi*) dari sistem inferensi *fuzzy* Sugeno dapat dilihat pada Persamaan 2.1 [31].

$$Z = \frac{\sum(w_i \times z_i)}{\sum w_i} \quad (2.1)$$

di mana:

Z : nilai *output* final

w_i : nilai keanggotaan/ derajat aktivasi setiap aturan *fuzzy*

z_i : nilai *output* dari setiap fungsi aturan *fuzzy*