

BAB II

LANDASAN TEORI

2.1 Pertanian

Pertanian adalah kegiatan manusia dalam mengolah lahan dan memanfaatkan sumber daya alam untuk menghasilkan bahan pangan, bahan baku industri, atau energi terbarukan dengan menggunakan berbagai teknik budidaya yang ramah lingkungan dan berkelanjutan. Sektor ini memainkan peran penting dalam memenuhi kebutuhan dasar manusia dan mendukung perekonomian, terutama di negara agraris seperti Indonesia. Selain itu, perkembangan teknologi modern seperti *Internet of Things (IoT)* dan kecerdasan buatan semakin membantu meningkatkan efisiensi dan produktivitas di sektor ini (Julvin Saputri Mendrofa et al., 2024).

Kegiatan pertanian tidak hanya berfokus pada produksi, tetapi juga pada keberlanjutan lingkungan. Pengelolaan yang baik diperlukan untuk menjaga keseimbangan ekosistem dan mengurangi dampak negatif terhadap lingkungan, seperti emisi gas rumah kaca. Dalam konteks ini, pertanian berkelanjutan menjadi pendekatan penting untuk mendukung pelestarian lingkungan sekaligus memastikan kebutuhan pangan terpenuhi bagi generasi mendatang (Mortaza A Syafinuddin Hammada, 2024).

Pemantauan kondisi lingkungan seperti suhu tanah, kelembapan tanah, kelembapan udara, dan kondisi hujan merupakan aspek krusial dalam sistem pertanian yang efisien dan berkelanjutan. Suhu tanah, misalnya, dapat diklasifikasikan menjadi dingin, normal, dan panas, yang masing-masing memengaruhi aktivitas *mikroorganisme* serta kemampuan akar dalam menyerap nutrisi. Sementara itu, kelembapan tanah terbagi menjadi kering, lembap, dan basah, yang secara langsung berdampak pada proses *fisiologis* tanaman seperti *fotosintesis* dan penyerapan air. Keseimbangan antara suhu dan kelembapan tanah menjadi faktor penting dalam mendukung pertumbuhan tanaman secara *optimal* (Anis & Setia Budi, 2023).

kelembapan udara juga memiliki pengaruh signifikan terhadap proses *transpirasi* dan keseimbangan air dalam jaringan tanaman. Kelembapan udara umumnya dibagi menjadi tiga kategori, yaitu rendah, sedang, dan tinggi.

Kelembapan yang terlalu rendah dapat meningkatkan laju penguapan, sedangkan kelembapan yang terlalu tinggi berpotensi memicu serangan *patogen*, terutama yang disebabkan oleh jamur (Kartika Rahayu Tri Prasetyo Sari et al., 2020). Kondisi hujan di suatu wilayah, baik dari segi intensitas maupun keberadaannya, dapat memengaruhi strategi irigasi, pola tanam, dan pengelolaan sumber daya air dalam sistem pertanian.

2.2 Tanaman Padi

Padi (*Oryza sativa L.*) adalah tanaman pangan utama yang menjadi sumber *karbohidrat* bagi sebagian besar penduduk dunia, termasuk Indonesia. Tanaman ini termasuk dalam *famili Poaceae* dan memiliki siklus hidup semusim. Padi memiliki peran penting dalam ketahanan pangan nasional dan merupakan komoditas strategis dalam perekonomian Indonesia (Kementrian Pertanian, 2020).

2.2.1 Karakteristik Tanaman Padi

Tanaman padi (*Oryza sativa L.*) memiliki karakteristik *morfologi* yang unik dan *kompleks*, yang mendukung perannya sebagai tanaman utama dalam sistem pertanian *tropis*. Secara umum, tanaman padi terdiri dari bagian akar, batang, daun, bunga, dan biji. Sistem akar padi tergolong serabut yang dangkal, memungkinkan tanaman menyerap nutrisi dari lapisan tanah bagian atas. Batangnya berbentuk bulat, berongga, dan tersusun dalam ruas-ruas yang kokoh untuk menopang daun dan *malai*. Daun tanaman padi berbentuk panjang dan tipis, dengan permukaan yang mengandung *stomata* untuk proses *fotosintesis*. Bunga padi tersusun dalam *malai*, yang menjadi tempat terjadinya penyerbukan dan pembentukan biji. Setiap biji padi terbungkus oleh sekam yang melindunginya hingga waktu panen (Bintang Ahimsya et al., 2018).

Status kondisi tanaman padi merupakan indikator penting dalam menentukan tingkat kesehatan dan produktivitas tanaman. Secara umum, kondisi ini dibagi menjadi tiga, yaitu kritis, stres, dan *optimal*. Kritis menunjukkan gangguan berat seperti kekeringan atau serangan hama yang dapat menghambat pertumbuhan. Stres menandakan tekanan lingkungan yang masih dapat ditoleransi namun berdampak pada efisiensi *metabolisme* tanaman. Sementara itu, *optimal* menggambarkan

kondisi lingkungan yang mendukung pertumbuhan dan hasil panen secara maksimal (Sugiarto et al., 2018).

2.2.2 Siklus Hidup Tanaman Padi

Siklus hidup tanaman padi terbagi menjadi tiga fase utama, yaitu fase *vegetatif*, fase *reproduktif*, dan fase pematangan. Pada fase *vegetatif* yang berlangsung sekitar 0 hingga 60 hari, biji padi mulai berkecambah setelah ditanam di tanah yang lembap. Fase ini ditandai dengan pertumbuhan daun yang *intensif* untuk mendukung proses *fotosintesis*, pembentukan akar yang berfungsi menyerap air dan nutrisi, serta perkembangan batang sebagai penopang struktur tanaman. Selain itu, pada tahap ini tanaman juga mulai membentuk tunas samping atau anakan, yang memiliki peran penting dalam meningkatkan jumlah *malai* sebagai penentu hasil panen.

Memasuki fase *reproduktif*, yang berlangsung antara 60 hingga 90 hari, tanaman padi mulai membentuk malai sebagai tempat berkembangnya bunga dan biji. Mekarnya bunga padi dalam fase ini menjadi penentu utama keberhasilan proses penyerbukan dan pembuahan. Fase ini sangat krusial dalam menentukan potensi hasil akhir dari tanaman.

pada fase pematangan yang berlangsung sekitar 90 hingga 120 hari, biji padi mulai mengalami pengisian dengan pati dan nutrisi, sehingga berat dan volumenya meningkat secara bertahap. Kematangan penuh ditandai dengan perubahan warna biji menjadi kuning atau keemasan, yang menunjukkan kesiapan tanaman untuk dipanen. Sepanjang siklus hidup ini, kondisi lingkungan seperti intensitas cahaya matahari, ketersediaan air, dan unsur hara tanah memainkan peran penting dalam mendukung proses *metabolisme* dan pertumbuhan tanaman padi secara *optimal* (Suspidayanti & Aries Rokhmana, 2021).

2.2.3 Kebutuhan Lingkungan

Tanaman padi membutuhkan kondisi lingkungan yang *spesifik* agar dapat tumbuh dan berkembang secara *optimal*. Kelembapan udara yang tinggi, berada pada kisaran 70–90%, sangat mendukung proses *fisiologis* tanaman seperti *fotosintesis* serta membantu mencegah terjadinya kekeringan *fisiologis*. Sementara

itu, suhu udara ideal untuk pertumbuhan padi berkisar antara 25–30°C. Apabila suhu turun di bawah 20°C atau meningkat di atas 35°C, berbagai proses *vital* seperti *fotosintesis* dan pembentukan biji dapat terganggu. Tidak hanya suhu udara, suhu tanah juga memegang peran penting dalam mendukung pertumbuhan akar serta aktivitas *mikroorganisme* tanah yang berkontribusi pada ketersediaan nutrisi; kisaran suhu *optimal* tanah berada pada 22–32°C.

kelembapan tanah yang ideal berkisar antara 60–80% kapasitas lapang, memastikan tanaman memperoleh pasokan air yang cukup tanpa menyebabkan kelebihan air yang berisiko menimbulkan stres akar. Ketersediaan air yang memadai sangat penting selama seluruh fase pertumbuhan, terutama pada fase *vegetatif* dan *reproduktif* yang membutuhkan pasokan air tinggi untuk pembentukan anakan dan *malai*. Oleh karena itu, sistem irigasi sawah yang dapat menjaga genangan air dalam batas ideal menjadi strategi utama dalam pemenuhan kebutuhan air tanaman padi secara berkelanjutan (Stuerz & Asch, 2019).

2.3 Internet Of Things (IoT)

Internet of Things (IoT) merupakan konsep teknologi yang menghubungkan berbagai perangkat melalui jaringan *internet* sehingga memungkinkan pengumpulan, pengolahan, dan pertukaran data secara *real-time*. *IoT* mengintegrasikan perangkat keras, perangkat lunak, serta konektivitas untuk menciptakan ekosistem yang dapat mendukung otomatisasi dan efisiensi sistem. Teknologi ini memanfaatkan sensor dan *aktuator* yang terhubung untuk memonitor lingkungan, mengontrol perangkat, dan memberikan data secara langsung kepada pengguna untuk pengambilan keputusan yang lebih baik (Judijanto et al., 2024).

IoT memungkinkan integrasi dunia fisik dengan dunia digital melalui pengumpulan data secara *real-time*, yang dapat digunakan untuk mengoptimalkan sistem dan mendukung pengambilan keputusan berbasis data. Dalam konteks pertanian, khususnya pada budidaya padi, penerapan *IoT* berperan penting dalam meningkatkan efisiensi dan produktivitas melalui pemantauan kondisi lingkungan pertumbuhan tanaman secara langsung. Teknologi ini memungkinkan pengambilan

keputusan yang lebih akurat dan tepat waktu. Pada penelitian ini, beberapa alat yang digunakan meliputi:

2.3.1 Mikrokontroler ESP32



Gambar 2.1 Microcontroller ESP32

(Sumber: <https://joy-it.net/en/products/SBC-NodeMCU-ESP32>)

ESP32 (Gambar 2.1) adalah mikrokontroler yang lebih canggih dibandingkan dengan ESP8266, yang dirancang khusus untuk aplikasi *Internet of Things (IoT)*. Dengan arsitektur *Dual-Core 32 bit*, ESP32 mampu menangani lebih banyak proses secara bersamaan, menawarkan performa yang lebih tinggi. Mikrokontroler ini dilengkapi dengan 48 pin dan 15 channel I/O, memberikan lebih banyak opsi konektivitas dan kontrol untuk berbagai aplikasi. Selain itu, ESP32 juga mendukung teknologi *Bluetooth*, menjadikannya solusi yang sangat baik untuk pengembangan sistem IoT dengan kebutuhan komunikasi nirkabel.

2.3.2 DS18B20 Waterproof Temperature Sensor

Sensor Suhu DS18B20 (Gambar 2.2) adalah sensor suhu digital yang dikembangkan oleh *Maxim Integrated*, dirancang khusus untuk mengukur suhu tanah dengan presisi tinggi. Sensor ini dapat mengukur suhu dalam rentang -55°C hingga 125°C dengan akurasi $\pm 0.5^{\circ}\text{C}$ pada rentang suhu -10°C hingga 85°C , yang menjadikannya ideal untuk aplikasi pertanian yang memerlukan pemantauan suhu tanah secara *real-time*.



Gambar 2.2 DS18B20 Waterproof Temperature Sensor

(Sumber: <https://makerselectronics.com/product/ds18b20-waterproof-temperature-sensor>)

Berikut adalah spesifikasi dari sensor suhu *DS18B20* sebagai berikut:

1. Menggunakan komunikasi satu kawat (*one-wire*) untuk pengiriman data.
2. Rentang pasokan daya: 3.0V hingga 5.5V.
3. Rentang suhu operasi: -55°C hingga +125°C.
4. Akurasi suhu: $\pm 0.5^\circ\text{C}$ (untuk rentang suhu antara -10°C hingga 85°C).

Sensor ini sangat ideal digunakan untuk memantau suhu tanah, yang merupakan faktor kunci dalam menentukan kondisi pertumbuhan tanaman.

2.3.3 Soil Moisture Sensor YL-69

Soil Moisture Sensor YL-69 (Gambar 2.3) adalah sensor yang digunakan untuk mengukur kelembapan tanah, yang sangat penting dalam aplikasi pertanian dan sistem irigasi *otomatis*. Sensor ini bekerja dengan cara mengukur tingkat *resistansi* antara dua *elektroda* yang tertanam di dalam tanah, yang kemudian diubah menjadi nilai digital yang menunjukkan tingkat kelembapan tanah. *YL-69* dapat memberikan indikasi yang tepat tentang kebutuhan air tanah, memungkinkan pengguna untuk memantau kondisi tanah secara *real-time*.



Gambar 2.3 Soil Moisture Sensor YL-69

(Sumber : <https://www.canadarobotix.com/products/2638>)

Berikut adalah spesifikasi dari *Soil Moisture Sensor YL-69*:

1. Menggunakan sensor *resistif* untuk mendeteksi kelembapan tanah.
2. Rentang pasokan daya: 3.3V hingga 5V.
3. Rentang kelembapan tanah: 0-100% (dalam bentuk nilai *analog* atau digital).
4. *Output analog* dan digital, memudahkan integrasi dengan berbagai mikrokontroler.

Sensor *YL-69* cocok digunakan untuk aplikasi pemantauan kelembapan tanah dalam pertanian, yang membantu petani untuk mengatur jadwal irigasi berdasarkan kebutuhan kelembapan tanah yang lebih akurat.

2.3.4 *Raindrops Detection Module*

Raindrops Detection Module (Gambar 2.4) adalah modul sensor yang digunakan untuk mendeteksi keberadaan air hujan, yang sangat berguna dalam aplikasi otomatisasi rumah, sistem irigasi, dan perangkat *IoT* berbasis cuaca. Sensor ini bekerja dengan cara mendeteksi perubahan resistansi pada permukaan modul ketika terkena tetesan air hujan, yang kemudian diubah menjadi nilai *digital* atau *analog* yang menunjukkan nilai hujan. Modul ini memungkinkan pengguna untuk mengambil tindakan otomatis berdasarkan kondisi cuaca secara real-time.



Gambar 2.4 Raindrops Detection Module

(Sumber: <https://makerselectronics.com/product/raindrop-sensor-module>)

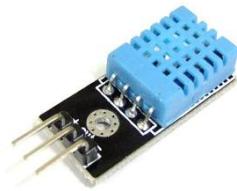
Berikut adalah spesifikasi dari *Raindrops Detection Module*:

1. Menggunakan sensor resistif untuk mendeteksi keberadaan air hujan.
2. Rentang pasokan daya: 3.3V hingga 5V.
3. Sensitivitas dapat disesuaikan menggunakan potensiometer.
4. Output analog dan digital, memudahkan integrasi dengan berbagai mikrokontroler.
5. Memiliki indikator *LED* untuk menunjukkan status deteksi hujan.

Raindrops Detection Module cocok digunakan dalam aplikasi pemantauan cuaca, membantu pengguna untuk membuat sistem otomatisasi berdasarkan nilai hujan

2.3.5 DHT11 Humidity Sensor

DHT11 Humidity Sensor (Gambar 2.5) adalah sensor digital yang digunakan untuk mengukur kelembapan dan suhu udara. Sensor ini sering digunakan dalam aplikasi otomatisasi rumah, sistem monitoring lingkungan, dan perangkat *IoT* karena kemampuannya untuk memberikan data lingkungan secara akurat dan efisien. Sensor *DHT11* bekerja dengan prinsip resistansi dan kapasitas untuk mendeteksi kelembapan, serta menggunakan elemen termistor untuk mengukur suhu.



Gambar 2.5 DHT11 Humidity Sensor

(Sumber: <https://www.jualarduinojogja.com/dht11-temperatur-and-humidity-sensor/>)

Berikut adalah spesifikasi dari *DHT11 Humidity Sensor*:

1. Rentang kelembapan: 20-90% RH (*Relative Humidity*).
2. Akurasi kelembapan: $\pm 5\%$ RH.
3. Rentang suhu: 0°C hingga 50°C .
4. Akurasi suhu: $\pm 2^{\circ}\text{C}$.
5. Tegangan operasi: 3.3V hingga 5V.
6. Output data digital menggunakan protokol komunikasi serial sederhana.

DHT11 Humidity Sensor dirancang untuk aplikasi yang memerlukan pengukuran lingkungan dengan biaya rendah namun tetap memberikan hasil yang memadai.

2.4 Fuzzy Mamdani

Logika *fuzzy* pertama kali diperkenalkan oleh Prof. Lotfi A. Zadeh pada tahun 1965. Landasan utama dari logika *fuzzy* adalah teori himpunan *fuzzy*, yang memungkinkan derajat keanggotaan suatu elemen tidak hanya bernilai 0 atau 1, tetapi dapat memiliki nilai di antara keduanya (Hidayati et al., 2024). Hal ini

membuat logika *fuzzy* mampu menangani ketidakpastian dan ketidakjelasan data, sehingga cocok digunakan dalam sistem cerdas yang meniru kemampuan manusia dalam pengambilan keputusan.

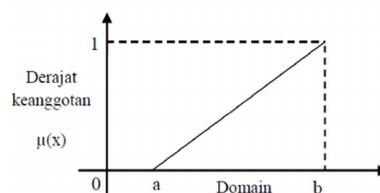
Dalam penerapan logika *fuzzy*, terdapat beberapa tahap perhitungan yang dilakukan, yaitu: penentuan nilai keanggotaan, *variabel linguistik*, proses *fuzzifikasi*, *inferensi*, pembuatan aturan (*rule*), dan *defuzzifikasi*. Logika *fuzzy* digunakan untuk mengkonversi data yang digambarkan dalam bentuk bahasa alami, seperti ukuran suhu yang bisa dinyatakan dengan kata-kata seperti dingin, hangat, atau panas. Logika *fuzzy* memungkinkan untuk menilai sejauh mana suatu nilai sesuai dengan kategori tertentu, sehingga mampu menangani ketidakpastian dalam data yang tidak bersifat pasti (Fatkhurrozi & Setiawan, 2024).

2.4.1 Fungsi Keanggotaan *Fuzzy*

Fungsi keanggotaan adalah representasi grafis yang menunjukkan tingkat keikutsertaan setiap nilai *input* dalam suatu himpunan *fuzzy*. Fungsi ini mengaitkan bobot dengan setiap *input* yang diproses, menggambarkan tumpang tindih antar *input*, dan menentukan respons akhir berupa *output* (Emirza Wira Saputra, 2020). Salah satu cara untuk memperoleh nilai keanggotaan adalah dengan menggunakan pendekatan fungsi, di mana terdapat beberapa fungsi yang dapat digunakan untuk menghitung nilai keanggotaan tersebut adalah:

1. *Representasi Linear*

Pada representasi *linear*, hubungan antara *input* dan derajat keanggotaan digambarkan dalam bentuk garis lurus. Bentuk ini merupakan yang paling sederhana dan sering dipilih untuk mendekati konsep yang kurang jelas. Setiap gambar disebut dalam penjelasan dibawah sebagai berikut:

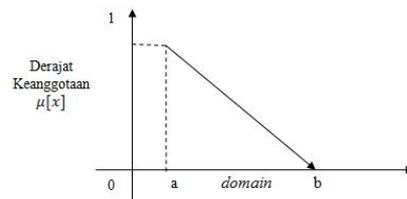


Gambar 2.6 Representasi *Linear Naik*

Fungsi Keanggotaan adalah sebagai berikut:

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

Selain *representasi linear* yang naik, terdapat juga *representasi linear* yang turun, di mana pemetaan input ke derajat keanggotaan digambarkan dengan garis lurus yang menurun.



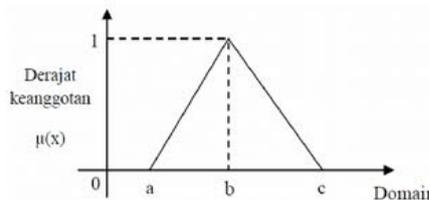
Gambar 2.7 Representasi Linear Turun

Fungsi Keanggotaan adalah sebagai berikut:

$$\mu(x) = \begin{cases} 0 & , x \leq a \\ \frac{b-x}{b-a} & , a \leq x \leq b \\ 1 & , x \geq b \end{cases} \quad (2.2)$$

2. Representasi Kurva Segitiga

Representasi kurva segitiga pada dasarnya merupakan gabungan dari dua garis linear yang saling bertemu, membentuk sebuah puncak segitiga. Kurva ini sering digunakan untuk menggambarkan nilai keanggotaan yang memiliki titik puncak dan dua sisi menurun atau meningkat.



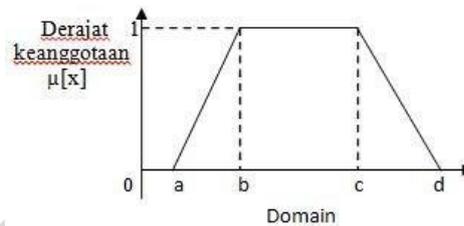
Gambar 2.8 Representasi Kurva Segitiga

Fungsi Keanggotaan adalah sebagai berikut:

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

3. Representasi Kurva Trapesium

Representasi kurva trapesium pada dasarnya mirip dengan bentuk segitiga, namun dengan perbedaan bahwa terdapat beberapa titik yang memiliki nilai keanggotaan 1, membentuk sisi datar di bagian atas kurva.



Gambar 2.9 Representasi Kurva trapesium

Fungsi Keanggotaan adalah sebagai berikut:

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

2.4.2 Variabel Linguistik

Variabel linguistik dalam logika *fuzzy* adalah variabel yang nilainya dinyatakan dengan kata-kata dalam bahasa alami, yang dicirikan oleh himpunan *fuzzy* yang didefinisikan dalam himpunan semesta tertentu (Adhi Gufron, 2010). Misalnya, variabel "suhu" dapat memiliki nilai *linguistik* seperti "dingin", "sedang", dan "panas". Setiap nilai *linguistik* ini disebut sebagai "terma" dan diwakili oleh himpunan *fuzzy* yang memiliki fungsi keanggotaan masing-masing. Fungsi keanggotaan ini menentukan derajat keanggotaan setiap elemen dalam himpunan semesta terhadap terma tertentu.

Sebagai contoh, *variabel linguistik* "kelembapan udara" dapat memiliki nilai dalam *interval* [0, 100], dengan rentang 0 hingga 100 menunjukkan persentase kelembapan. Didefinisikan tiga himpunan *fuzzy*: "rendah", "sedang", dan "tinggi" dalam *interval* tersebut. Maka, pernyataan seperti "kelembapan udara adalah rendah", "kelembapan udara adalah sedang", atau "kelembapan udara adalah tinggi" dapat digunakan untuk menggambarkan kondisi kelembapan udara secara *kualitatif*

2.4.3 Fuzzifikasi

Fuzzifikasi adalah proses dalam logika *fuzzy* yang bertujuan untuk mengubah data input berupa nilai numerik (*crisp*) menjadi nilai *fuzzy* berdasarkan fungsi keanggotaan yang telah didefinisikan. Dalam sistem *fuzzy*, data *crisp* biasanya diperoleh dari sensor atau hasil pengukuran, yang kemudian dikonversi menjadi derajat keanggotaan terhadap himpunan *fuzzy* tertentu. Proses ini penting karena sistem *fuzzy* bekerja dengan data yang bersifat *kualitatif* atau *linguistik*, seperti "rendah", "sedang", atau "tinggi".

Sebagai contoh, pada *variabel linguistik* "kelembapan udara", jika nilai *crisp* kelembapan adalah 70%, maka *fuzzifikasi* menentukan sejauh mana nilai ini termasuk ke dalam kategori linguistik seperti "sedang" atau "tinggi".

2.4.4 Inferensi Rule

Penalaran dalam logika *fuzzy* dilakukan dengan menggunakan input *fuzzy* dan aturan *fuzzy* (*fuzzy rules*) yang telah ditentukan sebelumnya untuk menghasilkan output dalam bentuk *fuzzy*. Terdapat berbagai metode yang dapat diterapkan untuk aturan *fuzzy* ini. Di bawah ini, dijelaskan tiga metode yang paling umum digunakan, yaitu metode *Tsukamoto*, metode *Mamdani*, dan metode *Sugeno*.

1. Metode *Tsukamoto*

Pada metode *Tsukamoto*, setiap konsekuen dalam aturan berbentuk *IF-THEN* direpresentasikan sebagai himpunan *fuzzy* dengan fungsi keanggotaan yang bersifat monoton. Akibatnya, *output* inferensi dari setiap aturan dihasilkan dalam bentuk nilai tegas (*crisp*) yang dihitung berdasarkan (α)-predikat (*fire strength*). Nilai akhir diperoleh dengan menghitung rata-rata berbobot dari hasil-hasil tersebut.

2. Metode *Mamdani*

Metode *Mamdani*, yang juga dikenal sebagai Metode *Max-Min*, pertama kali diperkenalkan oleh Ibrahim Mamdani pada tahun 1975. Proses untuk menghasilkan *output* dalam metode ini terdiri dari empat langkah utama, yaitu:

- a) Membentuk himpunan *fuzzy*.
- b) Mengaplikasikan fungsi implikasi berdasarkan aturan yang telah ditetapkan.
- c) Melakukan komposisi aturan untuk menggabungkan hasil dari semua aturan.
- d) Melakukan penegasan (*defuzzifikasi*) untuk menghasilkan output tegas (*crisp*).

3. Metode *Sugeno*

Penalaran dengan metode *Sugeno* memiliki kemiripan dengan metode *Mamdani*, tetapi perbedaannya terletak pada *output* sistem yang berupa nilai tegas (*crisp*), bukan himpunan *fuzzy*. Metode ini dikembangkan oleh Takagi, Sugeno, dan Kang (TSK) pada tahun 1985.

Terdapat dua model utama dalam sistem inferensi *fuzzy* menggunakan metode TSK, yaitu model TSK orde-0 dan model TSK orde-1.

a) Model *Fuzzy Sugeno Orde-0*

Secara umum, bentuk model *fuzzy Sugeno orde-0* dapat ditulis sebagai:

$$\begin{aligned} & \text{IF } (x_1 \text{ adalah } A_1) \text{ o } (x_2 \text{ adalah } A_2) \text{ o } \dots \text{ o } (x_n \text{ adalah } A_n) \\ & \text{THEN } z = k \end{aligned}$$

Di mana X_n merupakan variabel *linguistik*, A_n adalah nilai *linguistik*, o adalah operator *fuzzy* (misalnya *AND* atau *OR*), dan k adalah konstanta tegas.

b) Model *Fuzzy Sugeno Orde-1*

Secara umum, bentuk model *fuzzy Sugeno orde-1* adalah:

$$\begin{aligned} & \text{IF } (x_1 \text{ adalah } A_1) \text{ o } (x_2 \text{ adalah } A_2) \text{ o } \dots \text{ o } (x_n \text{ adalah } A_n) \text{ THEN} \\ & z = (p_1 * x_1) + \dots + (p_n * x_n) + q \end{aligned}$$

Dalam model ini, X_n adalah variabel *linguistik*, A_n adalah nilai *linguistik*, o adalah operator *fuzzy* seperti *AND* atau *OR*, sedangkan P_n dan q adalah konstanta tegas.

2.4.5 Defuzzifikasi

Defuzzifikasi atau penegasan merupakan proses untuk mengubah *output* sistem *fuzzy* yang berbentuk himpunan derajat keanggotaan menjadi nilai *crisp* atau nilai tegas. Proses ini sangat penting karena hasil akhir dari sistem logika *fuzzy* harus dapat ditafsirkan dalam bentuk nilai *numerik* yang nyata, sehingga dapat digunakan dalam pengambilan keputusan atau pengendalian sistem. Berbagai metode *defuzzifikasi* telah dikembangkan dan diterapkan, tergantung pada karakteristik masalah dan kebutuhan sistem yang dirancang.

Salah satu metode *defuzzifikasi* yang paling umum digunakan adalah metode *Centroid*, yang juga dikenal dengan istilah *Center of Area (CoA)* atau *Center of Gravity (CoG)*. Metode ini bekerja dengan mencari titik pusat massa dari area di bawah kurva fungsi keanggotaan. Dengan kata lain, metode ini menghitung rata-rata berbobot dari seluruh nilai dalam domain output *fuzzy* berdasarkan derajat keanggotaannya. Secara matematis, nilai *crisp* (γ) dapat dihitung menggunakan rumus:

$$\gamma = \frac{\sum \gamma \mu_R(\gamma)}{\sum \mu_R(\gamma)} \quad (2.5)$$

Dimana γ adalah nilai *crisp* yang dihasilkan, γ adalah *input crisp*, dan μ_R adalah derajat keanggotaan dari γ .

2.5 Message Queuing Telemetry Transport (MQTT)

Message Queuing Telemetry Transport (MQTT) adalah protokol komunikasi berbasis *publish-subscribe* yang dirancang untuk mengirimkan pesan antara perangkat *Internet of Things (IoT)* dengan *overhead* rendah dan konsumsi energi yang efisien. MQTT bekerja di atas protokol *TCP/IP* dan bersifat *open source*, memungkinkan komunikasi yang andal antara perangkat dengan sumber daya terbatas dan jaringan dengan *bandwidth* rendah. Beberapa penelitian di Indonesia telah mengimplementasikan protokol *MQTT* dalam berbagai aplikasi *IoT*. Misalnya, sebuah studi mengembangkan sistem *smart home* berbasis *ESP-32* dengan *multi-sensor* menggunakan protokol *MQTT* untuk meningkatkan efisiensi energi di lingkungan rumah tangga (Makatita & Hakim, 2024).

2.5.1 Konsep dasar MQTT

2.5.1.1 Publish/Subscribe

Dalam protokol *MQTT*, *publisher* berperan sebagai pengirim pesan, sedangkan *subscriber* mendaftar ke topik tertentu untuk menerima informasi yang relevan bagi mereka. Model komunikasi ini dikenal sebagai *Publish/Subscribe*, di mana *subscriber* akan mendapatkan setiap pesan yang diterbitkan pada topik yang mereka ikuti. Di sisi lain, klien juga dapat mengirim pesan ke suatu topik, sehingga semua *subscriber* yang terhubung ke topik tersebut dapat menerima pesan yang telah dipublikasikan.

2.5.1.2 Topics and Subscriptions

Dalam protokol *MQTT*, *publisher* mengirimkan pesan ke topik yang berfungsi sebagai identifikasi atau kategori pesan. *Subscriber* kemudian mendaftar ke topik tertentu agar dapat menerima pesan yang sesuai dengan kebutuhan mereka. Proses berlangganan ini dapat disesuaikan untuk membatasi data yang diterima hanya dari topik yang diinginkan. Selain itu, *MQTT* mendukung dua jenis *wildcard* memungkinkan *subscriber* untuk mengakses pesan dari berbagai topik yang masih saling berhubungan.

2.5.1.3 Retained messages

Dalam *MQTT*, setelah pesan dikirimkan ke semua klien yang berlangganan, *broker* dapat menyimpannya sebagai *retained message*. Jika ada klien baru yang berlangganan ke topik yang sama, *broker* akan langsung mengirimkan pesan yang tersimpan tersebut tanpa harus menunggu pesan baru dikirimkan oleh *publisher*. Fitur ini sangat berguna untuk memberikan informasi terbaru kepada klien yang baru terhubung, seperti status perangkat atau data sensor terakhir.

2.5.1.4 Clean sessions and reliable connections

Ketika seorang *subscriber* terhubung ke *broker*, pengaturan *clean session* menentukan apakah sesi akan bersifat sementara atau permanen. Jika *clean session* diset ke *false*, maka *broker* akan menyimpan informasi langganan serta pesan yang dikirim dengan tingkat *QoS* tertinggi untuk dikirim kembali saat koneksi klien dipulihkan. Fitur ini berguna untuk memastikan bahwa klien tidak kehilangan pesan

penting saat mengalami gangguan koneksi. Namun, penggunaan *flag* ini bersifat opsional dan dapat disesuaikan dengan kebutuhan komunikasi.

2.5.1.5 *Last Will and Testament (LWT)*

Seorang klien dapat mengonfigurasi *broker* untuk mengirimkan *will message* ke topik tertentu jika terjadi pemutusan koneksi yang tidak terduga. Pesan ini berguna dalam sistem yang memerlukan respons cepat, seperti keamanan atau *alarm*, di mana *administrator* atau sistem pemantauan akan segera diberi tahu jika sebuah sensor atau perangkat kehilangan koneksi. Fitur ini memastikan bahwa gangguan komunikasi dapat terdeteksi dan ditindaklanjuti dengan cepat.

2.5.1.6 *Quality of service levels*

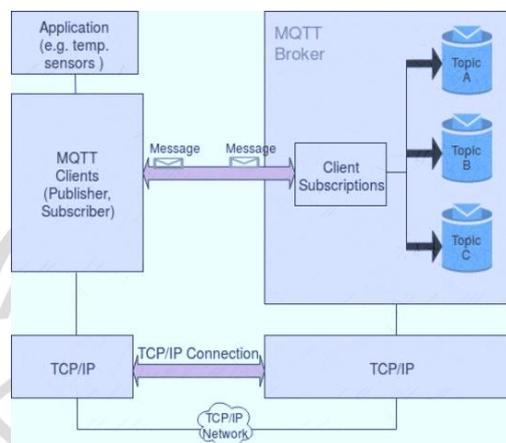
Protokol ini mendefinisikan *Quality of Service (QoS)* sebagai perjanjian antara pengirim dan penerima mengenai tingkat keandalan dalam distribusi data. *MQTT* mendukung tiga tingkat *QoS*, masing-masing dengan tingkat jaminan yang berbeda:

1. *QoS 0 (At most once)* Pesan dikirimkan paling banyak satu kali, tanpa jaminan bahwa pesan akan benar-benar diterima oleh penerima. Jika terjadi gangguan selama transmisi, pesan bisa saja hilang.
2. *QoS 1 (At least once)* Pesan dikirim minimal satu kali, yang berarti pesan dapat diterima lebih dari satu kali oleh *subscriber* jika terjadi retransmisi. *Broker* menandai pesan duplikat dengan *duplicate flag* yang diatur ke nilai 1.
3. *QoS 2 (Exactly once)* Pesan dikirim tepat satu kali, menggunakan mekanisme *4-way handshake* untuk memastikan tidak ada duplikasi maupun kehilangan pesan, menjadikannya tingkat *QoS* yang paling andal.

Pemilihan *QoS* bergantung pada kebutuhan sistem. Untuk *monitoring* kondisi lahan, *QoS 1 (At Least Once)* lebih sesuai karena memastikan data *real-time* diterima dengan andal tanpa mengorbankan kecepatan. Meskipun ada kemungkinan duplikasi, sistem dapat menyaringnya untuk menjaga efisiensi dan responsivitas.

2.5.2 Arsitektur MQTT

Arsitektur MQTT umumnya terbagi menjadi dua komponen utama, sebagaimana ditampilkan pada (Gambar 2.11). Berikut adalah penjelasan singkat mengenai masing-masing komponen.



Gambar 2.10 Arsitektur MQTT

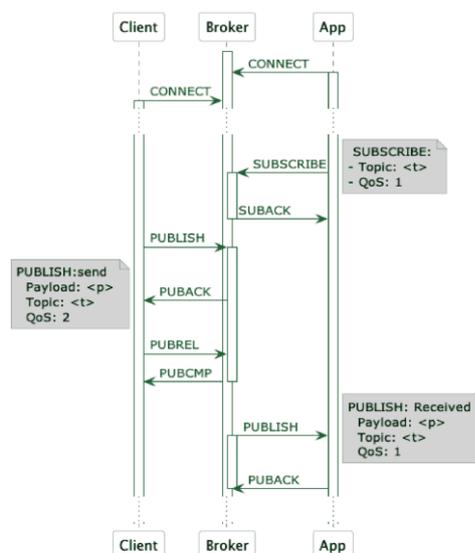
2.5.2.1 Client

Client dalam MQTT adalah perangkat atau aplikasi yang berkomunikasi melalui broker menggunakan model *Publish/Subscribe*. *Client* dapat berperan sebagai *publisher*, *subscriber*, atau keduanya. Sebagai *publisher*, *client* mengirimkan pesan ke *broker* berdasarkan topik tertentu, sementara *subscriber* berlangganan ke topik yang relevan untuk menerima pesan yang dikirimkan oleh *broker*.

2.5.2.2 Broker

Broker adalah komponen utama dalam arsitektur MQTT yang berfungsi sebagai perantara komunikasi antara *client*, baik sebagai *publisher* maupun *subscriber*. *Broker* menerima pesan dari *publisher*, menyimpannya jika diperlukan, dan mendistribusikannya kepada *subscriber* berdasarkan topik yang relevan. Untuk memastikan keandalan pengiriman, *broker* menerapkan *Quality of Service (QoS)* dengan tiga tingkat jaminan, serta menangani koneksi *client*, termasuk sesi yang terputus dan *retained messages* bagi *subscriber* baru. Selain itu, *broker* mendukung autentikasi dan enkripsi untuk keamanan data. Beberapa *broker* populer seperti *Eclipse Mosquitto*, *HiveMQ*, dan *EMQX* menyediakan fitur skalabilitas serta

integrasi dengan *cloud*, menjadikan *MQTT* efisien, andal, dan ideal untuk sistem berbasis IoT (Harnanta et al., 2020).



Gambar 2.11 Alur Kerja MQTT

(Sumber: <https://rb.gy/216o1w/>)

Dalam penelitian ini, peneliti menerapkan *MQTT* dalam sistem *monitoring* lahan tanaman padi guna memastikan pengiriman data yang efisien dan *real-time*. Dengan mengadopsi *model publish/subscribe*, *MQTT* memungkinkan data dari sensor lingkungan, seperti kelembaban tanah dan suhu udara, untuk dikirimkan secara langsung dan cepat ke *klien* yang terhubung, bahkan dalam kondisi jaringan dengan *bandwidth* terbatas. Selain itu, penggunaan *MQTT* mendukung konsumsi daya yang rendah serta stabilitas dalam pengiriman data, yang sangat penting untuk sistem *monitoring* pertanian yang membutuhkan pembaruan informasi secara berkala guna memberikan rekomendasi irigasi dan pemupukan yang lebih akurat.

2.6 Matriks Pengujian Performa

Implementasi dan pengujian sistem berbasis *Fuzzy Logic Mamdani* dilakukan untuk memonitor lahan pertanian padi. Sistem ini dirancang untuk membantu pengambilan keputusan dalam pengelolaan lahan, seperti pengaturan irigasi, pemupukan, dan pemantauan kondisi lingkungan. Evaluasi sistem dilakukan dengan menggunakan metode *Confusion Matrix* untuk mengukur akurasi sistem

dalam mengklasifikasikan data. *Confusion Matrix* merupakan alat analisis yang digunakan untuk mengevaluasi performa sistem dalam mengenali data dari kelas yang berbeda. Nilai *True Positive (TP)* dan *True Negative (TN)* menunjukkan bahwa sistem berhasil melakukan klasifikasi dengan benar, sedangkan *False Positive (FP)* dan *False Negative (FN)* menunjukkan kesalahan dalam klasifikasi data. Rumus perhitungan akurasi adalah sebagai berikut:

$$\text{Accuracy} = \frac{TP + TN}{TP + FP + FN + TN} \quad (2.6)$$

$$\text{Sensivitas (TPR)(Recall)} = \frac{TP}{TP + FN} \quad (2.7)$$

$$\text{Spesifisitas (TNR)} = \frac{TN}{TN + Fp} \quad (2.8)$$

$$\text{FPR} = \frac{FP}{TN + FP} \quad (2.9)$$

$$\text{FNR} = \frac{FN}{TP + FN} \quad (2.10)$$

$$\text{Precision} = \frac{TP}{TP + FP} \quad (2.11)$$

$$\text{F1 - Score} = 2x \frac{\text{Recall} \times \text{Precision}}{\text{Recall} + \text{Precision}} \quad (2.12)$$

$$\text{Negative Prediktive Value} = \frac{FPTN}{TN + FN} \quad (2.13)$$

Keterangan:

1. *True Positive (TP)* dan *True Negative (TN)* terjadi saat prediksi model sesuai dengan hasil sebenarnya, baik untuk kelas positif maupun negatif.
2. *False Positive (FP)* dan *False Negative (FN)* adalah kesalahan prediksi di mana model salah mengklasifikasikan data sebagai positif atau negatif.
3. Akurasi (*Accuracy*) mengukur seberapa sering model membuat prediksi yang benar dari seluruh data yang diuji.
4. Sensitivitas (*Recall/TPR*) menunjukkan kemampuan model dalam mendeteksi kelas positif dengan benar.
5. Spesifisitas (*Specificity/TNR*) mengukur seberapa baik model mengenali kelas negatif.

6. *False Positive Rate (FPR)* dan *False Negative Rate (FNR)* mengindikasikan seberapa sering model salah dalam mengklasifikasikan data negatif atau positif.
7. *Precision* menilai seberapa akurat prediksi positif yang dilakukan model.
8. *F1-Score* adalah keseimbangan antara presisi dan *recall*, berguna jika keduanya sama-sama penting.
9. *Negative Predictive Value (NPV)* menunjukkan keakuratan prediksi negatif yang dibuat model.

2.7 Review Artikel

Sebagai langkah untuk memperkuat topik penelitian, penulis melakukan analisis terhadap hasil penelitian sebelumnya yang relevan. Beberapa temuan dari studi terdahulu yang terkait dengan topik ini disajikan dalam Tabel 2.1.

Tabel 2.1 Hasil Review Artikel

Masalah	Hasil Penelitian	Dataset	Penulis
Sistem irigasi berbasis waktu menyebabkan akar busuk, penyakit <i>fungisarium</i> , dan penurunan panen hingga 50%, terutama pada musim pancaroba.	Logika <i>fuzzy</i> pada <i>NodeMCU 8266</i> berhasil mengatur durasi penyemprotan dengan rata-rata waktu respon 3,6 detik, meningkatkan efisiensi irigasi sesuai kondisi suhu dan kelembapan.	Dataset terdiri dari 10 data simulasi suhu ($^{\circ}\text{C}$), kelembapan (%), dan durasi penyemprotan (detik)	(Prasetyo et al., 2022)
<i>Greenhouse</i> sulit menjaga stabilitas suhu, kelembapan, dan cahaya secara optimal dengan sistem konvensional.	Sistem berbasis <i>fuzzy</i> dengan <i>Arduino Mega 2560</i> sukses mengontrol penyiraman dan pencahayaan dengan akurasi hingga 99,6%.	Dataset terdiri dari 10 set data suhu ($^{\circ}\text{C}$), kelembapan (%), ketinggian air (%), dan intensitas cahaya (lux)	(Kurniawan & Witanti, 2021)
Pengelolaan kelembapan tanah untuk tanaman cabai sering kali sulit dilakukan secara real-time,	Sistem <i>IoT</i> dengan <i>NodeMCU ESP8266</i> dan <i>fuzzy logic</i> berhasil menjaga kelembapan tanah optimal dan efisien.	4 set data hasil pengujian mencakup kelembapan tanah (%RH), suhu udara ($^{\circ}\text{C}$), dan	(Diva Putra Romadan et al., 2025)

Masalah	Hasil Penelitian	Dataset	Penulis
sehingga menyebabkan penyiraman berlebih atau kurang		kelembapan udara (%)	
masalah utama dalam aktivitas penyiraman tanaman yang tidak tepat waktu karena kesibukan pemilik, yang dapat menyebabkan tanaman mati atau tumbuh dengan buruk.	Sistem ini dapat secara otomatis menyiram tanaman berdasarkan pembacaan sensor suhu, kelembapan tanah, dan intensitas cahaya.	Dataset meliputi data suhu ($^{\circ}\text{C}$), kelembapan udara (%), kelembapan tanah (%), ketinggian air (%), dan intensitas cahaya (<i>lux</i>). Penelitian menggunakan 10 set data hasil pengujian	(Bintan Kurnia Putra, 2022)
pencampuran larutan nutrisi tanaman <i>hidroponik</i> yang masih manual, membutuhkan tenaga manusia dan waktu lama, serta sulit memantau kondisi larutan secara jarak jauh.	Berdasarkan pengujian, sistem bekerja dengan rata-rata error kurang dari 7%, menghasilkan waktu kontrol lebih efisien dibanding metode manual, serta memberikan kontrol nutrisi dan pemantauan jarak jauh secara <i>real-time</i> .	Dataset mencakup kandungan nutrisi (<i>TDS</i> dalam PPM) dan ketinggian air (cm) pada <i>hidroponik</i>	(Fatori, 2022)
penyiraman dan pemupukan tanaman secara konvensional yang memakan waktu, tidak efisien, dan rentan terhadap ketidaktepatan dalam memenuhi kebutuhan tanaman.	Sistem memanfaatkan sensor untuk memantau kelembapan tanah, suhu udara, dan ketinggian air, serta dikendalikan melalui aplikasi <i>Node-RED</i> . Pengujian menunjukkan bahwa sistem bekerja dengan baik, menghasilkan rata-rata error 0,3% pada sensor <i>DHT-22</i> selama 3 jam.	Dataset meliputi kelembapan tanah (10-100%), suhu udara (39,4–46,2 $^{\circ}\text{C}$), kelembapan udara (34,3–40,4%), dengan data diambil selama 3 jam melalui sensor <i>DHT-22</i> dan <i>sensor soil moisture</i> .	(Rinaldi et al., 2022)

Masalah	Hasil Penelitian	Dataset	Penulis
kesuburan tanah yang menjadi faktor utama dalam keberhasilan bertani, khususnya tingkat <i>pH</i> tanah yang terlalu asam dan kelembapan tanah yang rendah	Pengujian menunjukkan sistem dapat memberikan output mode penyiraman yang sesuai (Mode 1, Mode 2, Mode 3) dengan rata-rata <i>error</i> rendah.	Dataset terdiri dari pengukuran: 4 sampel <i>pH</i> tanah (dengan error rata-rata 0,375%), 5 kategori kelembapan tanah, serta hasil fuzzy untuk 9 kombinasi parameter.	(Setiatno Putera et al., 2023)
kurang optimalnya pemantauan suhu dan kelembapan pada budidaya bunga krisan secara manual, yang memengaruhi kualitas bunga.	menghasilkan alat berbasis <i>IoT</i> dengan logika fuzzy Sugeno yang dapat mengontrol kipas dan pompa secara otomatis sesuai suhu dan kelembapan, dengan hasil presisi tinggi.	menggunakan 13 kali pengujian PWM kipas dan 8 kali pengujian PWM pompa DC, dengan rata-rata selisih hasil fuzzy sebesar 0,04 untuk kipas dan 0,08 untuk pompa.	(Fono et al., 2023)