

**Hak Cipta :**

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :

a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.

b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta

2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

**BAB II  
TINJAUAN PUSTAKA**

**2.1. Penelitian Terdahulu**

Penelitian ini mengacu pada beberapa studi dalam 5 (lima) tahun terakhir yang membahas penggunaan metode *Adaptive Neuro-Fuzzy Inference System* (ANFIS) untuk analisis dan klasifikasi cuaca. Metode ini menggabungkan logika *fuzzy* dan jaringan saraf tiruan untuk memproses data lingkungan seperti suhu, tekanan udara, kelembapan, dan kecepatan angin. Sistem yang dikembangkan memanfaatkan empat parameter tersebut sebagai input utama, tanpa menggunakan data output secara eksplisit, untuk membentuk model klasifikasi cuaca yang adaptif dan akurat.

Tabel 2. 1 Penelitian Terdahulu

Nama Peneliti	Egi Pratama Muhammad Fatchan Ahmad Aguswin	Adam Sulthoni Akbar Candra Dewi Randy Cahya Wihandika	Ardytha Luthfiarta Aris Febriyanto Heru Lestiawan Wibowo Wicaksono
Judul dan Tahun	Klasifikasi Cuaca Menggunakan Data Historis dengan Algoritma Regresi Linear untuk Analisis Perubahan Suhu	Klasifikasi Cuaca Kota Denpasar menggunakan Algoritma ELM dengan Optimasi Quantum Delta Particle Swarm Optimization	Analisa Prakiraan Cuaca dengan Parameter Suhu, Kelembapan, Tekanan Udara, dan Kecepatan Angin Menggunakan Regresi Linear Berganda
Hasil Penelitian	Penelitian ini bertujuan membangun model klasifikasi suhu maksimum harian di Tokyo menggunakan algoritma regresi linear berbasis data historis cuaca selama 639 hari. Variabel yang digunakan meliputi suhu, minimum, kelembapan,	Penelitian ini menggunakan metode <i>Extreme Learning Machine</i> (ELM) yang dioptimasi dengan algoritma <i>Quantum Delta Particle Swarm Optimization</i> (QDPSO) untuk memklasifikasi cuaca di kota Denpasar. Parameter yang digunakan adalah suhu, kelembapan, tekanan udara, dan	Penelitian ini menggunakan metode regresi linear berganda untuk memklasifikasi curah hujan berdasarkan 4 variabel input: suhu, kelembapan, tekanan udara, dan kecepatan angin. Hasil analisis menunjukkan bahwa suhu, tekanan udara, dan kelembapan berpengaruh

**Hak Cipta :**

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :

a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.

b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta

2. Dilarang mengumumkannya dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

	<p>tekanan udara, kecepatan angin, dan curah hujan. Model yang dibangun menunjukkan hasil klasifikasi yang sangat akurat dengan nilai RMSE sebesar 0.80 dan R-squared sebesar 0.99, yang berarti model mampu menjelaskan hampir seluruh variabilitas data suhu maksimum harian. Penelitian ini menegaskan bahwa regresi linear efektif dalam memodelkan klasifikasi cuaca berbasis data historis.</p>	<p>kecepatan angin. Hasilnya menunjukkan bahwa penggunaan QDPSO secara signifikan meningkatkan akurasi klasifikasi. Dengan parameter optimal (jumlah partikel = 10, iterasi = 42, <math>g = 0.96</math>, dan 5 hidden neurons), model mencapai akurasi 100%, jauh lebih tinggi dibandingkan ELM tanpa QDPSO (yang hanya 39%). Ini menunjukkan bahwa kombinasi ELM dan QDPSO sangat efektif dalam klasifikasi cuaca berdasarkan data historis.</p>	<p>signifikan terhadap curah hujan, sementara kecepatan angin tidak signifikan. Persamaan regresi yang dihasilkan mampu menjelaskan 25,5% variabilitas curah hujan (<math>R^2 = 0,255</math>), sedangkan sisanya dipengaruhi oleh variabel lain di luar model. Nilai standar error (RMSE) sangat kecil (0,010), menunjukkan bahwa klasifikasi cukup akurat meskipun modelnya masih sederhana.</p>
Perbedaan	<ul style="list-style-type: none"> <li>• Penelitian sebelumnya menggunakan output terstruktur (supervised learning), sedangkan saya menyusun pendekatan dengan fokus pada input dan model <i>fuzzy adaptif</i> yang dapat bekerja tanpa label eksplisit output.</li> <li>• Penelitian sebelumnya menggunakan algoritma regresi linear, sedangkan saya menggunakan</li> </ul>	<ul style="list-style-type: none"> <li>• Penelitian sebelumnya menggunakan ELM dengan optimasi QDPSO, sedangkan saya menggunakan ANFIS (Adaptive Neuro-Fuzzy Inference System). HMI hanya dapat dilihat dari tampilan LCD Termokontrol saja. Belum ada tampilan HMI untuk melihat grafik dari pembacaan sensor</li> <li>• Model mereka bersifat black-box (ELM) yang sulit diinterpretasikan, sedangkan ANFIS</li> </ul>	<ul style="list-style-type: none"> <li>• Penelitian mereka menghasilkan model yang lebih sederhana tetapi kurang fleksibel terhadap pola data kompleks, sementara model saya lebih adaptif dan <i>fleksibel</i> terhadap variasi data karena kemampuan fuzzy dan learning dari ANFIS.</li> <li>• Interpretasi model regresi bersifat matematis langsung, sedangkan model ANFIS lebih transparan dan dapat dijelaskan</li> </ul>

**Hak Cipta :**

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkannya dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

	metode <i>Adaptive Neuro-Fuzzy Inference System</i> (ANFIS).	milik saya lebih transparan karena menggunakan aturan <i>fuzzy</i> yang bisa dijelaskan secara logis.	dalam bentuk aturan logika <i>IF-THEN</i> .
Persamaan	<ul style="list-style-type: none"> <li>• Memanfaatkan parameter suhu, kelembaban, tekanan udara, dan kecepatan angin sebagai variabel input utama.</li> </ul>	<ul style="list-style-type: none"> <li>• Sama-sama memanfaatkan kecerdasan buatan untuk membangun sistem klasifikasi yang akurat.</li> </ul>	<ul style="list-style-type: none"> <li>• Sama-sama menggunakan pendekatan <i>supervised learning</i></li> </ul>

Berdasarkan beberapa referensi penelitian sebelumnya yang telah dijabarkan, maka dilakukan pengembangan dalam penelitian ini berupa perancangan sistem klasifikasi kondisi cuaca menggunakan metode *Adaptive Neuro-Fuzzy Inference System* (ANFIS). Penelitian ini berfokus pada pemanfaatan empat parameter utama, yaitu suhu, kelembapan, tekanan udara, dan kecepatan angin sebagai input untuk membangun model klasifikasi cuaca yang adaptif dan akurat. Tujuan dari pengembangan ini adalah untuk menghasilkan sistem klasifikasi yang mampu mengklasifikasikan kondisi cuaca secara tepat sehingga dapat digunakan sebagai dasar pengambilan keputusan atau peringatan dini terhadap perubahan cuaca.

## 2.2. Klasifikasi Cuaca

Klasifikasi cuaca adalah proses memperkirakan kondisi atmosfer pada masa mendatang berdasarkan observasi dan data historis dari parameter lingkungan seperti suhu, kelembapan, tekanan, dan angin. Informasi cuaca menjadi krusial dalam berbagai sektor, seperti pertanian, penerbangan, pelayaran, dan kehidupan sehari-hari (Wang et al., 2020). Sistem atmosfer yang dinamis menyebabkan klasifikasi cuaca menjadi tantangan karena variabel-variabelnya bersifat non-linier dan saling memengaruhi secara kompleks.

Ketidakpastian dalam data dan variabilitas pola cuaca membuat pendekatan konvensional sering kali tidak mencukupi. Oleh karena itu, pendekatan berbasis kecerdasan buatan (AI) seperti *Fuzzy Logic*, *Artificial Neural Network* (ANN), dan *Adaptive Neuro-Fuzzy Inference System* (ANFIS) digunakan karena mampu

mengolah data kompleks dan memberikan klasifikasi yang lebih adaptif (Sivakumar & Anbalagan, 2021).

### 2.3. Software (MATLAB)

*MATLAB* adalah singkatan dari *Matrix Laboratory*, sebuah perangkat lunak komputasi numerik yang dikembangkan oleh *MathWorks* dan banyak digunakan dalam dunia teknik, sains, dan akademik. *MATLAB* dirancang khusus untuk memudahkan pengguna dalam melakukan perhitungan matematis, analisis data, pemodelan sistem, simulasi, hingga visualisasi grafik dalam lingkungan yang terintegrasi. *MATLAB* mendukung berbagai jenis operasi numerik, seperti aljabar linier, statistik, pemrosesan sinyal, pemrosesan citra, kontrol sistem, dan masih banyak lagi. Salah satu keunggulan utama *MATLAB* adalah tersedianya berbagai toolbox khusus yang dapat digunakan untuk kebutuhan spesifik, seperti *Control System Toolbox*, *Signal Processing Toolbox*, dan *Fuzzy Logic Toolbox*.

*Fuzzy Logic Toolbox* dalam *MATLAB* memungkinkan pengguna untuk membangun sistem inferensi berbasis logika fuzzy, baik dengan pendekatan manual maupun yang bersifat adaptif. Di dalam toolbox ini terdapat fitur *ANFIS (Adaptive Neuro-Fuzzy Inference System)*, yaitu sebuah metode yang menggabungkan kelebihan logika fuzzy dan jaringan saraf tiruan. Dengan *ANFIS*, pengguna dapat membangun sistem prediktif yang mampu belajar langsung dari data. *ANFIS* sangat berguna dalam aplikasi-aplikasi di mana hubungan antara input dan output bersifat nonlinier atau sulit dipetakan secara eksplisit, seperti dalam klasifikasi cuaca, diagnosis medis, atau sistem kontrol industri. *MATLAB* menyediakan lingkungan yang sangat mendukung untuk pelatihan model *ANFIS*, mulai dari proses pre-processing data, pelatihan model, evaluasi performa, hingga visualisasi hasil klasifikasi secara menyeluruh dan interaktif.

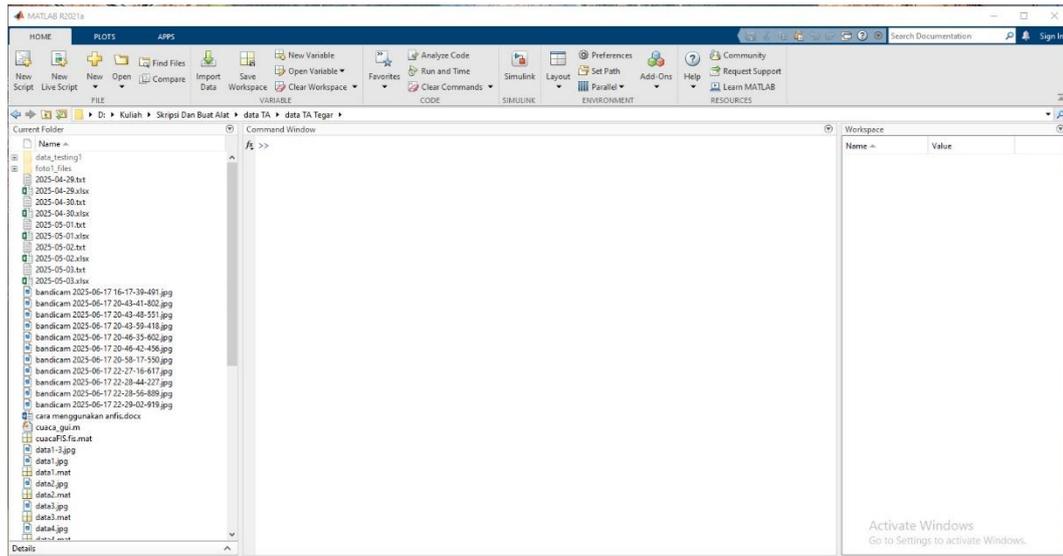
#### Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta



**Hak Cipta :**

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, -penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta



Gambar 2. 1 UI Matlab  
(Sumber: Dokume Pribadi)

## 2.4. Metode Parameter Input Klasifikasi Cuaca

Dalam sistem klasifikasi cuaca yang dikembangkan pada penelitian ini, digunakan empat parameter utama sebagai input, yaitu suhu udara, kelembapan, tekanan udara, dan kecepatan angin. Keempat parameter ini dipilih karena merupakan indikator utama kondisi atmosfer yang berperan penting dalam pembentukan pola cuaca. Setiap parameter memberikan informasi spesifik yang, jika dikombinasikan, dapat memberikan gambaran yang lebih menyeluruh mengenai kemungkinan perubahan cuaca yang akan terjadi.

Tabel 2. 2 Aturan Klasifikasi Cuaca

No.	Suhu	Tekanan	Kelembapan	Angin	Output (z)	Cuaca
1	Rendah	Tinggi	Rendah	Rendah	1.0	Cerah
2	Rendah	Tinggi	Rendah	Sedang	1.0	Cerah
3	Rendah	Sedang	Sedang	Rendah	2.0	Berawan
4	Sedang	Sedang	Sedang	Sedang	2.0	Berawan
5	Tinggi	Sedang	Tinggi	Tinggi	3.0	Hujan
6	Sedang	Rendah	Tinggi	Tinggi	3.0	Hujan
7	Tinggi	Rendah	Tinggi	Sedang	3.0	Hujan

**Hak Cipta :**

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkannya dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

#### 2.4.1 Suhu Udara

Suhu udara merupakan salah satu parameter paling dasar dalam sistem atmosfer. Nilai suhu menunjukkan tingkat energi panas yang ada di udara dan sangat memengaruhi proses penguapan dan kondensasi. Ketika suhu tinggi, penguapan air dari permukaan tanah atau laut akan meningkat, yang kemudian dapat menyebabkan pembentukan awan jika diikuti oleh pendinginan. Di sisi lain, suhu rendah seringkali dikaitkan dengan cuaca mendung atau hujan karena udara dingin memiliki kapasitas lebih rendah untuk menahan uap air. Dalam sistem klasifikasi cuaca, suhu dapat digunakan untuk memklasifikasi kecenderungan cuaca menjadi cerah atau hujan, tergantung pada hubungannya dengan parameter lainnya.

#### 2.4.2 Kelembapan

Kelembapan adalah ukuran kandungan uap air di atmosfer. Kelembapan relatif yang tinggi menunjukkan bahwa udara jenuh dengan uap air dan memiliki kemungkinan besar untuk terjadinya kondensasi, yang dapat menghasilkan awan atau bahkan hujan. Sebaliknya, kelembapan yang rendah menunjukkan udara kering dan biasanya diasosiasikan dengan cuaca cerah. Parameter ini sangat penting dalam klasifikasi cuaca karena perubahan kelembapan sangat memengaruhi terbentuknya awan, kabut, dan hujan. Kelembapan yang tinggi bersamaan dengan suhu rendah sering kali menjadi indikator kuat adanya potensi hujan.

#### 2.4.3 Tekanan Udara

Tekanan udara merupakan gaya yang diberikan udara terhadap permukaan bumi. Dalam meteorologi, tekanan udara sangat berperan dalam mengidentifikasi sistem cuaca. Tekanan tinggi biasanya menunjukkan kondisi cuaca yang stabil dan cerah, sedangkan tekanan rendah sering dikaitkan dengan kondisi atmosfer yang tidak stabil dan kemungkinan besar terjadi hujan atau badai. Perubahan tekanan udara dalam waktu singkat bisa menjadi tanda akan datangnya front cuaca. Oleh karena itu, memasukkan tekanan udara sebagai parameter input dalam sistem klasifikasi cuaca dapat membantu model mengenali kondisi transisi atmosfer yang kritis.

#### 2.4.4 Kecepatan Angin

Angin merupakan hasil dari perbedaan tekanan udara di atmosfer. Kecepatan dan arah angin dapat mengangkut massa udara yang membawa uap air, partikel,

atau panas dari satu wilayah ke wilayah lain. Kecepatan angin yang tinggi sering dikaitkan dengan perubahan cuaca yang cepat, misalnya datangnya awan hujan dari laut ke darat atau perubahan suhu mendadak. Selain itu, angin juga dapat berkontribusi terhadap distribusi kelembapan dan suhu di suatu area. Oleh karena itu, kecepatan angin menjadi variabel penting dalam memklasifikasi seberapa cepat perubahan cuaca akan terjadi dan arah pergerakan sistem cuaca.

### 2.5. Arduino IDE

*Arduino IDE* adalah perangkat lunak berbasis *open-source* yang digunakan untuk memrogram mikrokontroler seperti *Arduino Uno* dan *ESP8266*. Dalam penelitian ini, *Arduino IDE* digunakan untuk membaca data dari *sensor* dan mengirimkan ke komputer (Prasetyo & Nugraha, 2021). *Arduino IDE* menyediakan antarmuka pemrograman yang sederhana dan mudah dipahami, dengan bahasa pemrograman berbasis *C/C++*. Perangkat lunak ini juga mendukung berbagai pustaka (*library*) tambahan untuk *sensor* dan komunikasi *serial*, sehingga proses pengembangan sistem menjadi lebih efisien. Dalam konteks penelitian ini, *Arduino IDE* menjadi alat penting untuk mengontrol proses pengambilan data dari lingkungan secara *real-time* sebelum diproses lebih lanjut di *MATLAB* menggunakan model *ANFIS*.

Dengan memanfaatkan *Arduino IDE*, penulis dapat mengatur *interval* pengambilan data (misalnya setiap 2 menit), mengatur format data yang dikirim ke *MATLAB*, serta memastikan kestabilan komunikasi antara perangkat keras dan perangkat lunak. Kemampuan ini menjadikan *Arduino IDE* sebagai komponen penting dalam integrasi sistem klasifikasi cuaca berbasis *sensor* dan kecerdasan buatan. Dengan dukungan *library* yang luas, *Arduino IDE* memudahkan integrasi *sensor* suhu, kelembapan, tekanan, dan kecepatan angin. Program ini memungkinkan pengambilan data secara berkala dan pengiriman ke *MATLAB* untuk analisis lanjutan.

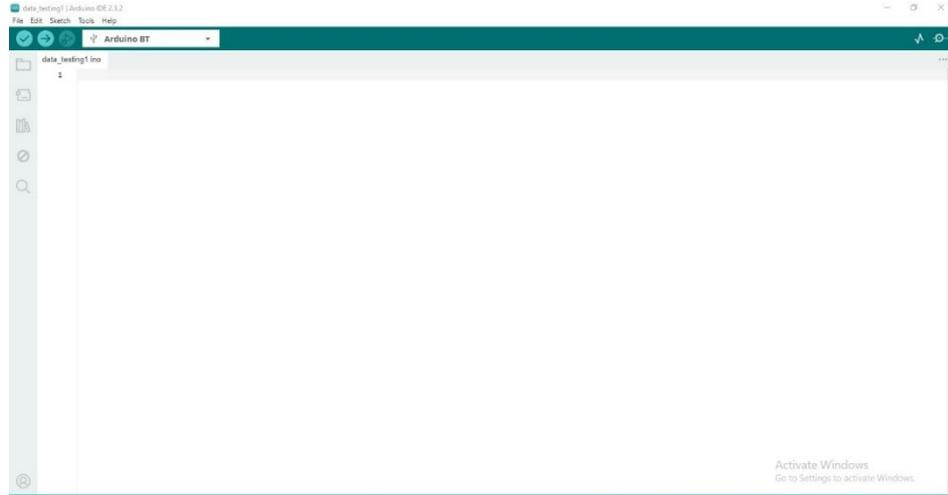
#### Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta



**Hak Cipta :**

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkannya dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta



Gambar 2. 2 UI Arduino IDE  
(Sumber: Dokume Pribadi)

## 2.6. ANFIS (*Adaptive Neuro-Fuzzy Inference System*)

ANFIS adalah metode kecerdasan buatan yang menggabungkan dua pendekatan populer, yaitu *logika fuzzy* dan jaringan saraf tiruan (*Artificial Neural Network*). *Logika fuzzy* memungkinkan sistem untuk menangani ketidakpastian dan data kabur menggunakan aturan berbasis linguistik seperti “jika suhu tinggi dan kelembapan rendah, maka cuaca cerah”. Di sisi lain, jaringan saraf memberikan kemampuan pembelajaran otomatis dari data yang tersedia.

Dalam konteks klasifikasi cuaca, ANFIS berfungsi untuk menganalisis hubungan *non-linier* antar parameter cuaca seperti suhu, kelembapan, tekanan, dan angin, kemudian menghasilkan output berupa kategori cuaca (cerah, berawan, hujan). ANFIS terdiri dari lima lapisan kerja yang dimulai dari proses *fuzzifikasi*, penghitungan kekuatan aturan, normalisasi, penentuan konsekuen, hingga *defuzzifikasi*. Metode ini unggul dalam fleksibilitas, transparansi, serta kemampuan belajar dari data historis, menjadikannya sangat cocok untuk digunakan dalam sistem klasifikasi cuaca yang kompleks dan dinamis.

### 2.6.1. Struktur Dasar ANFIS

ANFIS umumnya didasarkan pada *arsitektur Sugeno-type fuzzy inference system*, dan terdiri dari lima lapisan utama dalam bentuk jaringan saraf buatan. Misalkan kita memiliki dua input ( $x$  dan  $y$ ) dan satu output  $f$ , maka arsitektur ANFIS dengan dua aturan *fuzzy* adalah sebagai berikut:

### Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :

a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.

b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta

2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

### Contoh aturan fuzzy:

- Rule 1: If x is A1 and y is B1 then  $f1 = p1x + q1y + r1$
- Rule 2: If x is A2 and y is B2 then  $f2 = p2x + q2y + r2$

### 2.6.2. Variabel dan Himpunan Fuzzy

Dalam ANFIS, variabel input (misalnya suhu, kelembapan, tekanan, angin) dibagi menjadi himpunan *fuzzy* seperti:

- Suhu: Dingin, Sedang, Panas
- Kelembapan: Rendah, Sedang, Tinggi
- Tekanan: Rendah, Normal, Tinggi

Setiap himpunan *fuzzy* didefinisikan dengan fungsi keanggotaan (membership function), seperti

#### 2.6.2.1 Fungsi Gaussian (gausmf)

Fungsi *Gaussian* membentuk kurva lonceng yang halus dan simetris. Fungsi ini cocok ketika kita ingin model belajar dari data dengan transisi lembut antar kategori. Tidak ada sudut tajam, membuatnya ideal untuk sistem yang sensitif terhadap perubahan kecil.

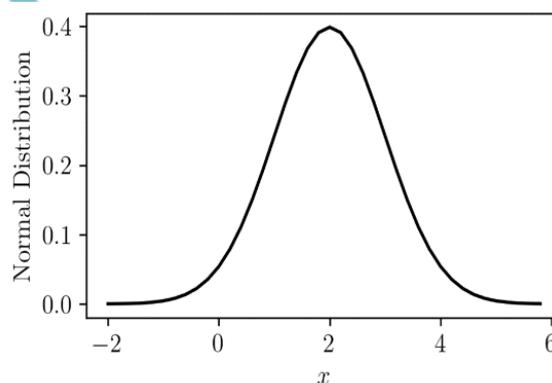
Persamaan:

$$\mu(x) = e^{-\frac{(x-c)^2}{2\sigma^2}}$$

Keterangan:

- $x$  = nilai *input*
- $c$  = pusat kurva (mean)
- $\sigma$  = lebar kurva (standar deviasi)

Visualisasi Konseptual:



**Hak Cipta :**

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

- Saat data memiliki transisi kontinu.
- Ideal untuk pengukuran fisik seperti suhu atau kelembapan.
- Cocok untuk klasifikasi cuaca real-time yang fluktuatif.

### 2.6.2.2 Fungsi Segitiga (trimf)

Merupakan fungsi keanggotaan yang paling sederhana dan paling umum. Berbentuk segitiga dengan tiga titik: awal, puncak, dan akhir. Transisinya bersifat linier lebih tajam dibanding Gaussian.

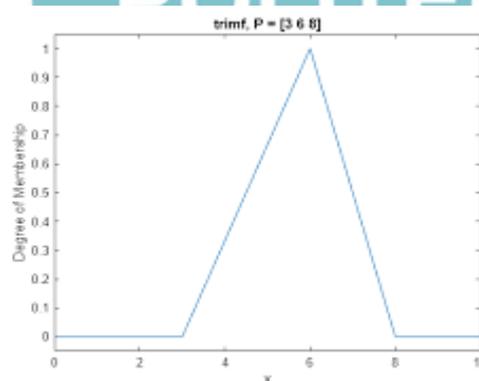
Persamaan:

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

Keterangan:

- $a$  = batas awal
- $b$  = puncak (nilai maksimal keanggotaan = 1)
- $c$  = batas akhir

**Visualisasi Konseptual:**



- Saat implementasi cepat diperlukan.
- Cocok untuk sistem dengan jumlah aturan *fuzzy* terbatas.
- Dapat digunakan untuk kategori seperti “rendah”, “sedang”, dan “tinggi” yang dibatasi tegas.

**Hak Cipta :**

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

**2.6.2.3 Fungsi Bell Umum (Generalized Bell / gbellmf)**

Fungsi ini adalah versi lebih fleksibel dari *Gaussian*. Bentuk kurvanya mirip lonceng, tetapi dapat dikendalikan dari tajam menjadi tumpul tergantung parameter. Sangat berguna saat kita ingin kontrol bentuk lebih halus.

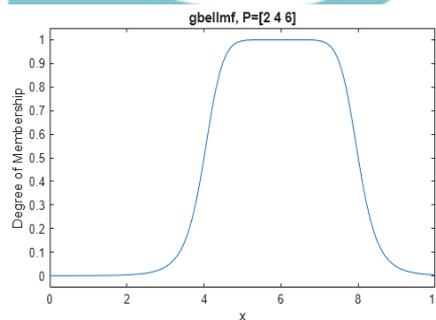
Persamaan:

$$\mu(x) = \frac{1}{1 + \left| \frac{x - c}{a} \right|^{2b}}$$

Keterangan:

- $a$  = mengatur lebar fungsi
- $b$  = mengatur ketajaman lengkungan
- $c$  = pusat fungsi

**Visualisasi Konseptual:**



Saat kita ingin transisi halus tapi masih bisa dikontrol ketajamannya.

- Cocok untuk sistem klasifikasi berbasis data real, seperti ANFIS dengan data lingkungan cuaca.

**2.6.3. Pembelajaran Parameter ANFIS**

Dalam sistem ANFIS, proses pembelajaran bertujuan untuk mengoptimalkan dua jenis parameter, yaitu:

1. **Parameter non-linier**, yaitu parameter dalam fungsi keanggotaan fuzzy (pada Layer 1).
2. **Parameter linier**, yaitu parameter konsekuen dari aturan fuzzy (pada Layer 4), biasanya berupa nilai konstanta atau persamaan linear dari input.

ANFIS menggunakan metode *hybrid learning algorithm*, yang merupakan gabungan dari dua pendekatan pembelajaran: *forward pass* dan *backward pass*.

**Hak Cipta :**

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

Proses ini dilakukan secara *iteratif* hingga sistem mencapai *error minimum* atau jumlah *epoch* yang ditentukan.

### 2.6.3.1 Forward Pass (Laju Maju)

Pada tahap ini:

- Parameter fungsi keanggotaan dianggap tetap.
- ANFIS menghitung output akhir sistem berdasarkan input data pelatihan.
- Kemudian, parameter linier ( $p, q, r$ ) pada bagian konsekuen dari aturan *fuzzy* dihitung menggunakan metode *Least Squares Estimation* (LSE).

Contoh fungsi konsekuen linier Sugeno:

$$f_i = p_i x + q_i y + r_i$$

Dengan:

- $f_i$ : output konsekuen dari aturan ke- $i$ ,
- $x, y$ : input,
- $p_i, q_i, r_i$ : parameter linier yang akan dihitung.

Metode *Least Squares Estimation* (LSE) menyelesaikan sistem persamaan untuk mendapatkan parameter linier yang meminimalkan *error* kuadrat antara *output* klasifikasi dan data aktual.

### 2.6.3.2 Backward Pass (Laju Mundur)

Pada tahap ini:

- Parameter linier hasil dari *forward pass* dianggap tetap.
- Parameter fungsi keanggotaan *fuzzy* diperbarui dengan metode *Gradient Descent*, berdasarkan *error* yang dihasilkan oleh sistem.

Gradient descent bekerja dengan menghitung turunan parsial *error* terhadap masing-masing parameter non-linier dari fungsi keanggotaan dan memperbarui nilai parameternya menggunakan rumus:

$$\theta_{new} = \theta_{old} - \eta \frac{\partial E}{\partial \theta}$$

Dengan:

- $\theta$ : parameter fungsi keanggotaan (misal pusat atau lebar dari fungsi *Gaussian*),
- $\eta$ : *learning rate*,
- $E$ : *error* (selisih antara *output* sistem dan target),

**Hak Cipta :**

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

- $\frac{\partial E}{\partial \theta}$ : turunan parsial error terhadap parameter.

### 2.6.3.3 Rangkaian Proses Hybrid Learning dalam ANFIS

1. Data input diberikan ke sistem.
2. Di layer pertama, fungsi keanggotaan *fuzzy* diterapkan ke input.
3. Aturan *fuzzy* diaktifkan dan bobot dihitung.
4. Di layer 4, konsekuen linear digunakan untuk menghitung *output* setiap aturan.
5. *Output* akhir diperoleh dari rata-rata berbobot semua *output* aturan.
6. *Error* (selisih *output* aktual dan target) dihitung.
7. Dengan *error* tersebut:
  - *Forward pass* menghitung parameter linier (dengan LSE).
  - *Backward pass* memperbarui parameter non-linier (dengan *Gradient Descent*).
8. Proses diulang hingga konvergen (*error* kecil atau jumlah *epoch* tercapai).

### 2.6.4. Implementasi ANFIS

Di *MATLAB*, *training* dilakukan dengan fungsi:

```
[trainedFis, trainError] = anfis(trainingData, initFis, numEpoch);
```

- *trainingData*: berisi data input dan output target
- *initFis*: struktur awal FIS
- *trainedFis*: FIS hasil pelatihan
- *trainError*: nilai error tiap epoch

*Adaptive Neuro-Fuzzy Inference System* (ANFIS) adalah metode hibrida yang menggabungkan kelebihan dari logika fuzzy dan jaringan saraf tiruan (neural network). ANFIS dirancang untuk memetakan input ke output berdasarkan aturan-aturan fuzzy dengan parameter-parameter yang disesuaikan menggunakan teknik pembelajaran dari jaringan saraf. Metode ini digunakan secara luas dalam sistem klasifikasi, pengendalian, dan pengambilan keputusan karena kemampuannya dalam menangani ketidakpastian serta pembelajaran dari data.

### 2.6.5. Struktur ANFIS terdiri dari lima lapisan utama:

- Layer 1: *Fuzzifikasi (Membership Functions)* Setiap node pada layer ini berfungsi sebagai fungsi keanggotaan (*membership function*). Fungsi ini digunakan untuk

**Hak Cipta :**

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

menentukan derajat keanggotaan setiap input terhadap himpunan *fuzzy* tertentu. Fungsi keanggotaan yang umum digunakan adalah:

$$\bullet \text{ Fungsi segitiga (triangular): } \mu(x) = \begin{cases} 0, & x \leq a \\ \frac{x-a}{b-a}, & a < x \leq b \\ \frac{c-x}{c-b}, & b < x < c \\ 0, & x \geq c \end{cases}$$

- Fungsi bell-shaped (gaussian, generalized bell), sigmoid, dan lainnya.
- Layer 2: Aturan *Fuzzy (Rule Layer)* Node pada layer ini berperan sebagai *rule nodes*. Setiap node mewakili satu aturan *fuzzy*. Fungsi dari node ini adalah mengalikan semua derajat keanggotaan dari layer sebelumnya:  $w_i = \mu_{A_i}(x) \cdot \mu_{B_i}(y)$  Dimana  $w_i$  adalah kekuatan aturan ke-i.
- Layer 3: Normalisasi Setiap node pada layer ini menghitung rasio bobot:  $\bar{w}_i = \frac{w_i}{\sum_j w_j}$  Ini digunakan untuk menormalkan kekuatan semua aturan fuzzy.
- Layer 4: Konsekuen (Output Linear Functions) Node pada layer ini berfungsi sebagai konsekuen dari aturan *fuzzy*, biasanya dalam bentuk linear dari input:  $\bar{w}_i f_i = \bar{w}_i(p_i x + q_i y + r_i)$  Di mana  $p_i, q_i, r_i$  adalah parameter yang akan dilatih.
- Layer 5: Output Merupakan hasil penjumlahan seluruh output dari layer 4:  $\text{Output} = \sum_i \bar{w}_i f_i$

### 2.6.6. Metode Inferensi Fuzzy dalam ANFIS

ANFIS secara spesifik menggunakan metode inferensi Sugeno (bukan Mamdani) karena struktur output dari setiap aturan *fuzzy* dalam ANFIS harus berbentuk fungsi linear dari *input* atau konstanta. Berbeda dengan metode Mamdani yang menggunakan output *fuzzy* (dengan fungsi keanggotaan), metode Sugeno mempermudah proses *defuzzifikasi* karena *outputnya* langsung berupa nilai numerik.

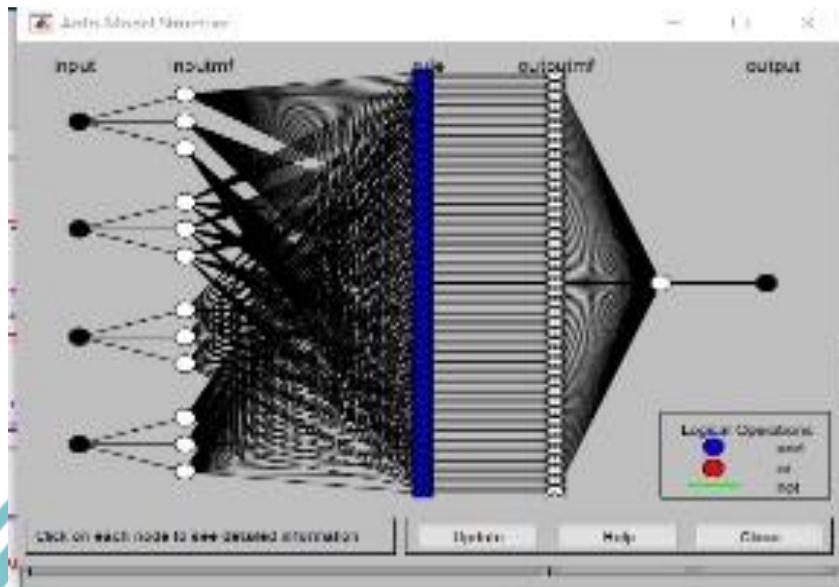
Contoh aturan Sugeno: "Jika suhu panas dan kelembapan rendah, maka cuaca = 0.8" Atau lebih umum: "Jika x adalah A dan y adalah B, maka output =  $px + qy + r$ "

Metode Sugeno cocok dalam implementasi ANFIS karena menghasilkan fungsi yang dapat dilatih melalui algoritma *least squares* dan *backpropagation*.

### Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkannya dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

## 2.6.7 Gambar Struktur ANFIS



Gambar 2. 3 Struktur Anfis  
(Sumber: Dokume Pribadi)

Dengan struktur dan fungsi-fungsi di atas, ANFIS dapat melakukan klasifikasi berdasarkan pola-pola data historis yang telah diberikan selama proses pelatihan. Dalam konteks penelitian ini, ANFIS digunakan untuk memklasifikasi kondisi cuaca berdasarkan suhu, kelembapan, tekanan udara, dan kecepatan angin.

- ANFIS menggunakan metode Sugeno secara default.
- Jika menggunakan metode Mamdani, maka *output fuzzy* akan memerlukan proses *defuzzifikasi* seperti *centroid*, *bisektor*, dsb.
- Namun karena ANFIS dirancang dengan konsekuen fungsi linier, maka metode Mamdani tidak digunakan.

Pengetahuan mendalam tentang struktur ini sangat penting agar dapat menyesuaikan parameter dan memahami hasil klasifikasi dalam pengembangan sistem klasifikasi kondisi cuaca berbasis ANFIS.

## 2.7. Komponen Sistem

Berikut ini adalah penjelasan komponen utama yang digunakan dalam sistem klasifikasi cuaca berbasis Arduino dan ANFIS:

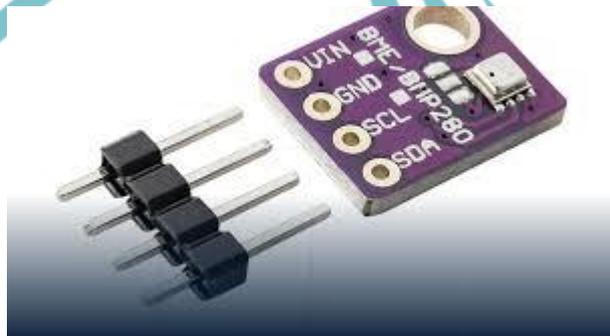
### 2.7.1 Sensor BME280

Sensor BME280 adalah sensor digital buatan Bosch Sensortec yang dapat mengukur suhu, tekanan udara, dan kelembapan relatif secara bersamaan. Sensor

**Hak Cipta :**

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

ini banyak digunakan dalam aplikasi monitoring cuaca dan perangkat berbasis IoT karena ukurannya yang kecil, konsumsi daya yang rendah, serta akurasi yang tinggi. BME280 mendukung antarmuka komunikasi I2C dan SPI, sehingga mudah diintegrasikan dengan mikrokontroler seperti Arduino atau ESP32. Sensor ini beroperasi pada tegangan 1.8V hingga 3.6V dan mampu mengukur suhu dari -40 hingga 85°C, tekanan dari 300 hingga 1100 hPa, serta kelembapan dari 0% hingga 100% RH, menjadikannya pilihan ideal untuk sistem klasifikasi cuaca dan pemantauan lingkungan.



*Gambar 2. 4 Sensor BME280*

(Sumber: <https://blog.indobot.co.id/menerapkan-sensor-bme280-dengan-esp32-membaca-tekanan-udara-suhu-dan-kelembaban/>)

### 2.7.2. Sensor Anemometer

Sensor anemometer yang digunakan dalam sistem ini adalah tipe ZTS-3000-FSJT-N01, yang berfungsi untuk mengukur kecepatan angin secara real-time. Mekanisme kerja sensor ini dimulai dari putaran baling-baling akibat tiupan angin, yang kemudian dikonversi menjadi pulsa elektrik (satu pulsa per putaran). Pulsa tersebut selanjutnya diolah oleh rangkaian elektronik internal sensor menjadi data kecepatan angin dalam satuan meter per detik (m/s). Data kecepatan yang dihasilkan dikomunikasikan ke mikrokontroler ESP32 menggunakan protokol Modbus RTU untuk selanjutnya diproses dan ditampilkan dalam sistem.

**Hak Cipta :**

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkannya dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta



*Gambar 2. 5 Sensor Anemometer*

(Sumber: <https://www.blibli.com/p/wind-speed-anemometer-sensor-kecepatan-angin-arduino-compatible/ps--ITE-70030-00239>)

### 2.7.3. Kabel Jumper

Kabel jumper adalah kabel penghubung pendek yang digunakan untuk menyambungkan pin antar komponen elektronik dalam rangkaian, terutama pada papan percobaan (breadboard) atau langsung ke mikrokontroler seperti Arduino atau ESP32. Kabel ini sangat umum digunakan dalam proses prototipe karena sifatnya yang fleksibel, mudah dipasang dan dilepas tanpa perlu disolder. Kabel jumper tersedia dalam tiga jenis utama, yaitu kabel jumper male-to-male (ujung jarum ke jarum), male-to-female (jarum ke lubang), dan female-to-female (lubang ke lubang), yang masing-masing digunakan sesuai jenis pin yang ingin dihubungkan.

Dalam sistem berbasis IoT dan mikrokontroler, kabel jumper berperan penting dalam menyambungkan sensor, modul komunikasi, maupun aktuator ke pin digital maupun analog pada mikrokontroler. Kabel ini biasanya memiliki panjang dan warna yang bervariasi, sehingga memudahkan pengguna dalam mengatur koneksi serta mengidentifikasi jalur sinyal. Penggunaan kabel jumper sangat membantu dalam eksperimen elektronik karena memudahkan perbaikan atau perubahan rangkaian tanpa merusak komponen.

**Hak Cipta :**

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkannya dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta



*Gambar 2. 6 Kabel Jumper*

(Sumber: <https://www.brontoseno.com/product/kabel-jumper-male-to-female-20cm-10pcs/>)

#### 2.7.4. Power Bank 12V 2A

Power bank 12V 2A adalah perangkat penyimpanan daya portabel yang dirancang untuk menyediakan tegangan output sebesar 12 volt dengan arus maksimum 2 ampere. Berbeda dengan power bank biasa yang umumnya hanya menyediakan output 5V untuk mengisi daya ponsel, power bank ini digunakan untuk perangkat elektronik yang membutuhkan tegangan dan arus lebih tinggi, seperti mikrokontroler, router, kamera CCTV, maupun sistem sensor berbasis IoT.

Dalam sistem klasifikasi cuaca berbasis Arduino atau ESP32, power bank 12V 2A berfungsi sebagai sumber daya utama yang memberikan suplai tegangan stabil ke seluruh rangkaian, termasuk mikrokontroler dan sensor seperti BME280 dan anemometer. Tegangan 12V sering kali diperlukan karena beberapa modul atau converter dalam sistem hanya dapat bekerja dengan input minimal 9–12V. Arus 2A memungkinkan sistem berjalan stabil meskipun beberapa komponen aktif secara bersamaan. Kelebihan lain dari power bank jenis ini adalah portabilitasnya, sehingga cocok digunakan untuk sistem monitoring di luar ruangan atau tempat yang tidak memiliki sumber listrik tetap.

**Hak Cipta :**

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkannya dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta



*Gambar 2. 7 Power Bank*

(Sumber: <https://www.blibli.com/p/powerbank-xiaomi-mi-2c-2000mah-power-bank-9v-2a-12v-1-5a-18w/ps--MAH-60042-02695>)

### 2.7.5 MODBUS

Modbus MAX485 adalah modul komunikasi berbasis protokol Modbus RTU (Remote Terminal Unit) yang menggunakan chip MAX485 sebagai konverter sinyal TTL ke RS-485. Modul ini sangat berguna dalam sistem mikrokontroler seperti Arduino atau ESP32 untuk berkomunikasi dengan perangkat industri yang menggunakan standar RS-485, seperti sensor kecepatan angin, PLC, dan perangkat monitoring lainnya.

Modul MAX485 bekerja dengan cara mengubah sinyal UART (Universal Asynchronous Receiver Transmitter) dari mikrokontroler yang bersifat TTL (Transistor-Transistor Logic) menjadi sinyal diferensial RS-485 yang mampu mentransmisikan data pada jarak jauh hingga ratusan meter dengan gangguan yang rendah. RS-485 menggunakan dua jalur komunikasi (A dan B) secara diferensial, sehingga tahan terhadap noise dan cocok untuk lingkungan industri.

Dalam konteks penggunaan dengan sensor anemometer, modul MAX485 berperan sebagai jembatan antara ESP32 (melalui port serial) dengan sensor yang mendukung protokol Modbus. Protokol Modbus RTU sendiri menggunakan format master-slave, di mana ESP32 berperan sebagai master yang mengirim permintaan (request) ke sensor (slave), dan sensor akan mengirimkan respons (response) berupa data pengukuran.

Penggunaan MAX485 membutuhkan pengaturan pin kontrol RE (Receive Enable) dan DE (Driver Enable) yang menentukan mode baca atau tulis. Biasanya,

**Hak Cipta :**

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkannya dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

pin ini dihubungkan bersama dan dikendalikan oleh mikrokontroler untuk beralih antara mode pengiriman dan penerimaan data.

Secara keseluruhan, modul Modbus MAX485 memungkinkan komunikasi data yang andal dan efisien dalam sistem IoT yang membutuhkan konektivitas dengan perangkat industri jarak jauh menggunakan protokol RS-485.



Gambar 2. 8 Modbus

(Sumber: <https://www.arduinoindonesia.id/2023/05/komunikasi-data-dengan-modbus-arduino-rs485.html>)

### 2.7.6 ESP32

ESP32 adalah mikrokontroler yang dikembangkan oleh Espressif Systems dan merupakan penerus dari ESP8266. Mikrokontroler ini dirancang khusus untuk aplikasi Internet of Things (IoT) karena memiliki kemampuan konektivitas nirkabel yang kuat dan performa pemrosesan yang tinggi. ESP32 dilengkapi dengan prosesor dual-core Tensilica LX6 dengan kecepatan hingga 240 MHz, serta memori RAM dan flash yang cukup besar untuk mendukung berbagai aplikasi berbasis sensor, kendali, dan komunikasi data.

Salah satu keunggulan utama ESP32 adalah integrasi koneksi Wi-Fi dan Bluetooth (baik Classic maupun BLE – Bluetooth Low Energy), yang memungkinkan perangkat ini terhubung ke internet, server cloud, atau perangkat lain secara nirkabel. Selain itu, ESP32 memiliki banyak antarmuka input/output seperti GPIO, ADC, DAC, SPI, I2C, UART, PWM, dan touch sensor, yang menjadikannya fleksibel untuk digunakan dalam berbagai sistem kendali dan pengumpulan data.

Dalam proyek sistem klasifikasi cuaca, ESP32 berperan sebagai pusat kendali yang menghubungkan sensor-sensor seperti BME280 (untuk membaca suhu, kelembapan, dan tekanan udara) serta anemometer (untuk membaca

**Hak Cipta :**

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkannya dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

kecepatan angin), kemudian memproses dan mengirimkan data ke platform monitoring seperti Arduino IoT Cloud. ESP32 juga dapat digunakan untuk menjalankan proses awal seperti filtering data, kalibrasi, atau komunikasi melalui protokol seperti Modbus RS-485.

Secara keseluruhan, ESP32 adalah pilihan populer dalam pengembangan perangkat IoT karena biaya yang terjangkau, kemampuan konektivitas yang andal, serta dukungan luas dari komunitas dan perangkat lunak seperti Arduino IDE dan PlatformIO.



Gambar 2. 9 ESP 32

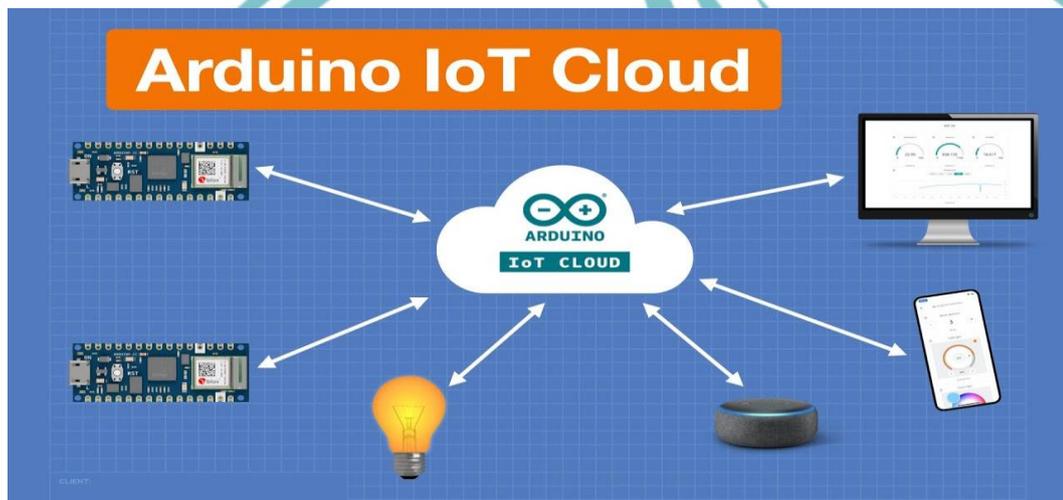
(Sumber: <https://www.lazada.co.id/products/esp32-esp-32-wroom-32-esp32-wifi-ble-development-board-dengan-arduino-ide-i4988296102.html>)

## 2.8. Arduino IoT Cloud

Arduino IoT Cloud adalah platform berbasis internet yang dikembangkan oleh Arduino untuk memudahkan pengguna dalam merancang, memantau, dan mengelola perangkat Internet of Things (IoT) secara daring. Dengan menggunakan platform ini, pengguna dapat menghubungkan berbagai perangkat mikrokontroler seperti ESP32 dan papan Arduino lainnya untuk mengirim serta menerima data sensor melalui koneksi internet secara real-time. Arduino IoT Cloud menyediakan antarmuka dashboard interaktif yang memungkinkan pengguna menampilkan data dalam bentuk grafik, indikator, maupun kontrol seperti tombol dan penggeser, sehingga monitoring sistem menjadi lebih mudah dan intuitif.

Platform ini juga memiliki sistem variabel cloud yang disebut sebagai thing properties. Variabel ini digunakan untuk menyimpan dan menyinkronkan data antara perangkat fisik dan dashboard, serta dapat diatur langsung melalui browser. Selain itu, Arduino IoT Cloud menyediakan file otomatis bernama thingProperties.h yang berisi deklarasi variabel dan pengaturan koneksi antara

perangkat dan cloud, sehingga mempermudah proses pemrograman di Arduino IDE. Arduino IoT Cloud juga mendukung protokol komunikasi seperti MQTT dan REST API, yang memungkinkan integrasi dengan layanan lain seperti Google Sheets atau IFTTT. Dalam konteks penelitian ini, Arduino IoT Cloud digunakan sebagai media pemantauan sistem klasifikasi cuaca secara online, di mana data suhu, tekanan, kelembapan, dan kecepatan angin dari sensor dikirim oleh mikrokontroler untuk divisualisasikan dan dianalisis secara real-time.



Gambar 2. 10 Arduino IOT Cloud

(Sumber: <https://www.youtube.com/watch?v=UFCmTZUoZ1M&pp=0gcJCf0Ao7VqN5tD>)

## 2.9. Kabel USB

Kabel USB (Universal Serial Bus) adalah kabel standar yang digunakan untuk menghubungkan perangkat elektronik dengan komputer atau sumber daya lainnya. Dalam konteks sistem berbasis mikrokontroler seperti ESP32 atau Arduino, kabel USB berfungsi sebagai media komunikasi data sekaligus sumber daya listrik.

Fungsi utama kabel USB dalam proyek IoT adalah untuk menghubungkan mikrokontroler ke laptop atau komputer guna melakukan pemrograman, debugging, dan pengiriman data. Saat pengembangan sistem, kabel ini memungkinkan pengguna mengunggah (upload) kode program dari Arduino IDE ke mikrokontroler. Selain itu, kabel USB juga dapat menyuplai tegangan 5V ke mikrokontroler selama proses pemrograman atau pengujian awal sebelum dipasang sumber daya eksternal seperti power bank.

### Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :

a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.

b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta

2. Dilarang mengemukakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta



Jenis konektor pada kabel USB yang umum digunakan dalam proyek IoT adalah USB Type-A (ke komputer) dan Micro-USB atau USB Type-C (ke mikrokontroler), tergantung jenis board yang digunakan. Kabel USB yang berkualitas baik akan memastikan komunikasi data yang stabil dan pengiriman daya yang optimal, sehingga sangat penting untuk proses pengembangan sistem berbasis IoT.



*Gambar 2. 11 Kabel USB*

(Sumber: <https://www.jakartanotebook.com/p/kabel-data-usb-type-c-multifungsi-fast-charging-6a-2m-tc2-white>)

## © Hak Cipta milik Politeknik Negeri Jakarta

### Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

## BAB III

### PERENCANAAN DAN REALISASI

#### 3.1 Metode Penelitian

Metode penelitian yang digunakan dalam studi ini adalah metode *laboratorium eksperimental*. Metode ini merupakan pendekatan ilmiah yang dilakukan dalam lingkungan laboratorium atau kondisi yang terkendali, dengan tujuan utama untuk menguji hipotesis atau rancangan sistem berdasarkan perlakuan yang diberikan secara langsung terhadap variabel-variabel yang telah ditentukan sebelumnya. Peneliti memiliki kontrol penuh terhadap alat, bahan, dan variabel eksperimen sehingga memungkinkan dilakukan pengamatan secara akurat dan berulang-ulang terhadap fenomena atau respon yang dihasilkan dari sistem.

Dalam konteks penelitian ini, metode *laboratorium eksperimental* digunakan untuk merancang, mengembangkan, dan menguji sistem klasifikasi cuaca berbasis metode *Adaptive Neuro-Fuzzy Inference System* (ANFIS). Sistem yang dibangun terdiri dari rangkaian perangkat keras (hardware) dan lunak (software) yang berfungsi sebagai satu kesatuan. Komponen utama dalam sistem adalah mikrokontroler ESP32 yang berperan sebagai pusat kendali, sensor BME280 untuk mengukur parameter suhu, kelembapan, dan tekanan udara, serta anemometer yang dikonversi melalui komunikasi Modbus RS485 menggunakan modul MAX485 untuk membaca kecepatan angin. Seluruh sensor dihubungkan ke ESP32 yang kemudian dikendalikan melalui pemrograman berbasis Arduino IDE dan terhubung dengan Arduino IoT Cloud sebagai platform penyimpanan dan *visualisasi* data secara daring (real-time monitoring).

Pengumpulan data dilakukan dalam rentang waktu tertentu (dalam penelitian ini selama lima hari berturut-turut) untuk memperoleh sampel data cuaca yang merepresentasikan kondisi aktual di lingkungan tempat eksperimen. Data yang diperoleh dari ESP32 selanjutnya dikirim ke MATLAB untuk diproses menggunakan metode ANFIS. Dalam MATLAB, data mentah dari sensor dijadikan input bagi sistem *fuzzy-neuro hybrid* tersebut untuk memklasifikasi kondisi cuaca berdasarkan pola yang telah dilatih sebelumnya. Proses pelatihan dilakukan menggunakan data historis sensor, dan model ANFIS yang dihasilkan akan dievaluasi berdasarkan akurasi klasifikasi terhadap kondisi cuaca seperti cerah,

berawan, atau hujan. Setelah proses inferensi dilakukan, hasil klasifikasi dari MATLAB dikirim kembali ke ESP32 dan ditampilkan pada dashboard Arduino IoT Cloud agar dapat diakses secara online melalui perangkat seperti laptop atau smartphone.

Keunggulan penggunaan metode laboratorium eksperimental dalam penelitian ini adalah memungkinkan seluruh proses — mulai dari desain sistem, implementasi perangkat keras, pengujian fungsional sensor, pemrosesan data dengan ANFIS, hingga validasi hasil klasifikasi — dilakukan secara menyeluruh dan berulang dalam kondisi yang terkontrol. Hal ini menjadikan hasil yang diperoleh bersifat objektif dan ilmiah, serta dapat digunakan untuk evaluasi dan pengembangan lebih lanjut. Dengan demikian, metode ini sangat relevan untuk digunakan dalam penelitian berbasis sistem embedded dan kecerdasan buatan karena mampu mengakomodasi kompleksitas dan kebutuhan pengujian yang spesifik terhadap interaksi antar komponen sistem secara integratif.

### 3.2 Perencanaan Alat

Sebagai langkah awal dalam pengembangan sistem klasifikasi kondisi cuaca, digunakan mikrokontroler ESP32 yang berperan dalam membaca data sensor secara real-time, menjalankan algoritma klasifikasi berbasis ANFIS secara lokal, serta mengirimkan hasilnya ke platform cloud. Sensor BME280 digunakan untuk mengukur suhu udara, kelembapan relatif, dan tekanan atmosfer, sedangkan sensor anemometer digunakan untuk mendeteksi kecepatan angin melalui sinyal analog proporsional dalam rentang 0–5 V. Karena batas maksimum input ADC ESP32 adalah 3,3 V, sinyal dari anemometer terlebih dahulu dilemahkan menggunakan rangkaian pembagi tegangan. Sensor BME280 terhubung ke ESP32 melalui antarmuka I2C dengan koneksi SDA ke GPIO21 dan SCL ke GPIO22.

ESP32 mendapatkan catu daya dari USB 5 V, sedangkan anemometer diberi suplai 12 V sesuai kebutuhannya, dengan ground yang disatukan agar pembacaan sinyal tetap stabil. ESP32 terhubung ke jaringan Wi-Fi lokal dan dikonfigurasi sebagai *Thing* pada Arduino IoT Cloud, dengan variabel-variabel untuk suhu, kelembapan, tekanan udara, kecepatan angin, dan hasil klasifikasi ANFIS. Setiap siklus pengukuran, ESP32 membaca data dari sensor, mengonversi sinyal analog menjadi kecepatan angin dalam satuan m/s, lalu menjalankan fungsi ANFIS yang

#### Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkannya dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta



**Hak Cipta :**

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, - penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

telah ditanamkan langsung di dalam program untuk menghasilkan klasifikasi kondisi cuaca secara real-time. Seluruh data kemudian dikirimkan ke dashboard Arduino IoT Cloud agar dapat dipantau oleh pengguna dari jarak jauh.

Pengembangan perangkat lunak dilakukan melalui penulisan sketch Arduino, mencakup inisialisasi koneksi Wi-Fi, konfigurasi sensor, dan pembacaan data secara berkala. Di dalam *loop* utama, ESP32 membaca seluruh parameter input, menjalankan algoritma ANFIS untuk memprosesnya, lalu mengunggah data sensor dan hasil klasifikasi ke cloud. Proses verifikasi dilakukan melalui pemantauan langsung pada monitor serial dan dashboard cloud, serta dapat diperkuat dengan membandingkan hasil sistem dengan data cuaca dari sumber terpercaya seperti BMKG.

### 3.2.1 Deskripsi Alat

Nama sistem : Sistem pemantauan dan klasifikasi kondisi cuaca berbasis IoT

Fungsi sistem : Sistem ini dirancang untuk mengukur suhu, kelembapan, tekanan, dan kecepatan angin dalam sistem pemantauan dan klasifikasi cuaca secara real time. Prancangan sistem ini dapat memberi peringatan dini tentang kondisi cuaca melalui sistem klasifikasi menggunakan metode ANFIS.

Nama sub sistem 1 : Implementasi sensor anemometer dan sistem monitoring pemantaun kondisi cuaca berbasis Internet of Things (IoT) secara real time menggunakan Webstite

Fungsi sub sistem 1 : menampilkan data cuaca yang diperoleh dari sensor dalam waktu nyata melalui platform berbasis website.

Data yang dikumpulkan dari sensor, seperti suhu, kelembaban, tekanan udara, dan kecepatan angin, dikirim ke server dan dapat diakses oleh pengguna kapan saja dan di mana saja. Website ini memungkinkan pengguna untuk memantau kondisi cuaca secara langsung serta melihat histori data yang telah terekam.

**Hak Cipta :**

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :

a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.

b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta

2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

Nama sub sistem 2 : Implementasi sensor bme280 pada sistem pemantauan dan klasifikasi cuaca berbasis Internet of things (IoT)

Fungsi sub sistem 2 : Sub sistem ini mengimplementasikan sensor BME280 yang berfungsi untuk mengukur suhu, kelembapan, dan tekanan menggunakan mikrokontroler ESP32. Dari hasil pembacaan sensor BME280 akan diproses dalam sistem pemantauan dan klasifikasi cuaca.

Nama sub sistem 3 : Analisis sistem klasifikasi kondisi cuaca menggunakan metode Adaptive Neural-Fuzzy Inference System (ANFIS)

Fungsi sub sistem 3 : Sistem klasifikasi cuaca berbasis ANFIS ini terdiri dari beberapa sub-sistem utama, yaitu akuisisi data, pemrosesan klasifikasi, dan komunikasi cloud. Sub-sistem akuisisi menggunakan sensor BME280 untuk membaca suhu, kelembapan, dan tekanan udara, serta anemometer untuk mengukur kecepatan angin, dengan sinyal analog yang dikonversi agar sesuai dengan input ADC ESP32. Keempat data input tersebut kemudian diproses langsung secara real-time oleh ESP32 menggunakan fungsi ANFIS yang telah ditanamkan di dalam program untuk menghasilkan klasifikasi kondisi cuaca. Seluruh data sensor dan hasil klasifikasi dikirim ke Arduino IoT Cloud melalui koneksi Wi-Fi, sehingga pengguna dapat memantau kondisi cuaca secara langsung melalui dashboard daring.

Untuk mendukung kinerja dari Sistem Pemantauan dan klasifikasi cuaca berbasis IoT ini, digunakan beberapa komponen utama dengan fungsi masing-masing yang saling terintegrasi. Adapun rincian spesifikasi alat dan peran fungsional tiap komponen adalah sebagai berikut.

1. Alat ini menggunakan 1 buah ESP32 yang digunakan sebagai controller untuk menginisialisasi dan pembacaan data sensor.

**Hak Cipta :**

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

2. Menggunakan 1 buah sensor BME280 yang berfungsi untuk mengukur suhu, kelembapan, dan tekanan.
3. Menggunakan 1 buah sensor Anemometer yang berfungsi untuk mengukur kecepatan angin dalam satuan m/s.
4. Menggunakan 1 buah MAX485 yang digunakan sebagai pengubah sinyal komunikasi differential RS485 dari sensor Anemometer menjadi level TTL (0-3,3 V, atau 5V) yang bisa dibaca oleh mikrokontroler ESP32.
5. Menggunakan 1 buah MINI DC UPS DC12V 2A yang berfungsi untuk mensuplai daya ke dalam sistem.

### 3.2.2 Realisasi Alat

Realisasi alat dilakukan dengan perancangan alat yang telah dibuat menggunakan *hardware* dan *software*, setelah melakukan perancangan alat akan dilakukan pemrograman pada *Arduino ide* untuk selanjutnya di monitoring dan analisis klasifikasi menggunakan metode *ANFIS*

### 3.2.3 Gambaran Umum Rancang Bangun Alat

Pada bagian ini akan dijelaskan secara garis besar komponen dan alur kerja sistem perangkat klasifikasi cuaca berbasis *IoT* dengan metode *ANFIS*. Rancang bangun alat mencakup pemilihan dan pemasangan sensor (BME280 untuk suhu, kelembapan, dan tekanan udara; anemometer untuk kecepatan angin), *mikrokontroler ESP32* sebagai otak pengolah data, serta koneksi dan pengaturan komunikasi dengan *Arduino IoT Cloud*. Data mentah dari sensor dikumpulkan secara periodik oleh ESP32, kemudian dikirim ke modul pemrosesan *ANFIS* yang dijalankan di *MATLAB* atau diimplementasikan *onboard*, sebelum hasil klasifikasi dikembalikan dan ditampilkan secara *real-time* melalui *dashboard cloud*. Desain perangkat mempertimbangkan kebutuhan stabilitas pembacaan sensor, kecepatan transmisi data, dan kemudahan perawatan, sehingga seluruh rangkaian dapat beroperasi secara andal dalam berbagai kondisi lingkungan.

## © Hak Cipta milik Politeknik Negeri Jakarta

### Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, - penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta



Gambar 3. 1 Bentuk Alat



Gambar 3. 2 Bentuk Alat

Berikut daftar realisasi perangkat hardware yang digunakan pada sistem pemantauan dan klasifikasi cuaca berbasis *internet of things* (IoT):

Tabel 3. 1 Spesifikasi Mekanik Alat

No	Nama Alat	Spesifikasi	Fungsi
1	ESP32	<ul style="list-style-type: none"> <li>– CPU Tensilica Xtensa dual-core 240 MHz</li> <li>–Wi-Fi 802.11 b/g/n,</li> <li>Bluetooth BLE</li> </ul>	Sebagai otak (microcontroller) untuk mengumpulkan data sensor, pemrosesan ringan, dan mengirim data ke cloud via Wi-Fi.
2	Sensor BME280 + Radiation Shield	<ul style="list-style-type: none"> <li>– Tekanan: 300–1100 hPa</li> <li>Suhu:40...+85 °C</li> <li>– Kelembaban: 0–100 %RH</li> <li>–Pelindung plastik putih multilayer</li> </ul>	Mengukur suhu, kelembaban, dan tekanan udara dengan akurasi tinggi; dipasang di dalam pelindung radiasi (shield) untuk mencegah radiasi matahari langsung.
3	Anemometer RS-485	<ul style="list-style-type: none"> <li>– Kisaran: 0–45 m/s</li> <li>– Output Modbus RTU RS-485</li> </ul>	Mengukur kecepatan angin; terhubung ke ESP32 melalui modul transceiver MAX485.
4	Modul MAX485 (RS-485 ↔ TTL)	Tegangan kerja 5 V	Mengkonversi sinyal RS-485 anemometer menjadi TTL yang dapat dibaca oleh ESP32.
5	Tripod & Mounting Bracket	<ul style="list-style-type: none"> <li>–Tinggi adjustable hingga ~1,5 m</li> <li>–Material aluminium</li> </ul>	Struktur penopang semua sensor dan enclosure, mudah dipindah-pindah dan diarahkan.
6	Kabel USB & Jumper Wires	<ul style="list-style-type: none"> <li>–USB-A to micro-USB</li> <li>–Dupont male-female</li> </ul>	Konektor daya dan pengkabelan antara ESP32, modul, dan sensor.

#### Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkannya dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

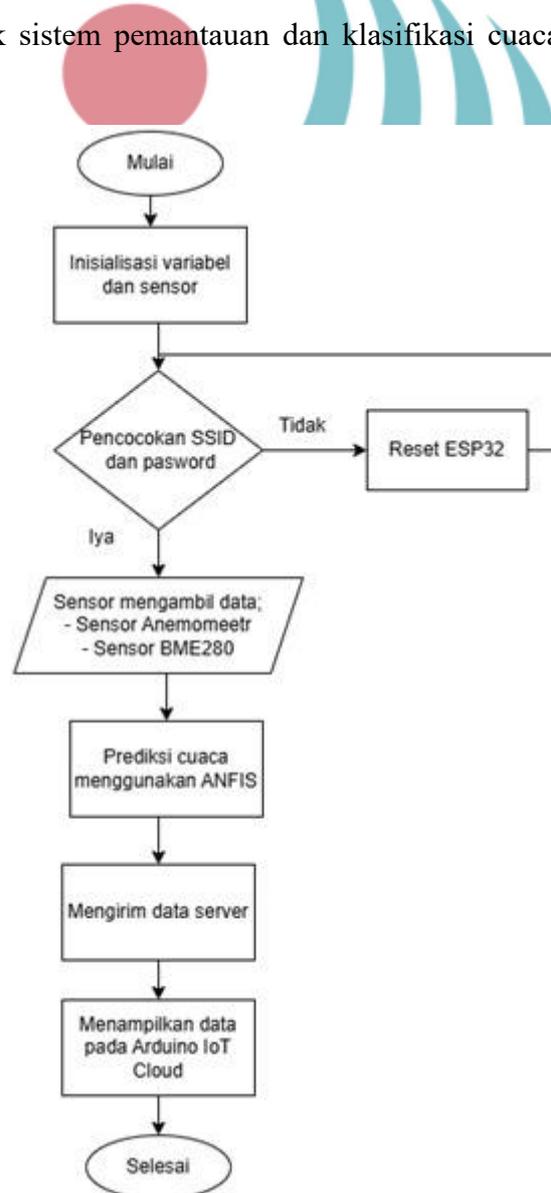
**Hak Cipta :**

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

7	Enclosure Box (IP65)	–Ukuran ~15×10×5 cm – Waterproof	Melindungi ESP32, modul MAX485, RTC, dan SD card dari cuaca (hujan/debu).
8	Power Bank	Output USB 12V 2A	Menyediakan catu daya portabel untuk ESP32 dan modul-modul terkait.

### 3.2.3 Cara kerja Alat

Cara kerja alat untuk sistem pemantauan dan klasifikasi cuaca berbasis IoT sebagai berikut:



Gambar 3. 3 Flowchart

**Hak Cipta :**

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

*Flowchart* tersebut menggambarkan alur proses sistem pemantauan dan klasifikasi cuaca. Proses dimulai saat ESP32 dalam keadaan power ON dan sketch program arduino dijalankan. Pada tahap ini, ESP32 akan menginisialisasi hardware dan software. ESP32 akan mengaktifkan modul wifi, antarmuka I2C untuk BME280, dan pin ADC untuk sensor Anemometer, Serta memuar model klasifikasi ANFIS yang sudah diprogram.

Selanjutnya akuisisi data cuaca, ESP32 akan membaca data dari sensor BME280 yang memberikan tiga jenis data sekaligus: suhu udara, kelembapan relatif, dan tekanan atmosfer. Secara bersamaan, ESP32 juga membaca sinyal dari anemometer, baik analog (tegangan dari kecepatan angin) atau digital (melalui konversi RS485 menggunakan modul MAX485). Nilai-nilai ini akan dikalibrasi secara matematis untuk menghasilkan satuan fisik seperti °C, %RH, hPa, dan m/s.

Setelah semua data sensor diperoleh, nilai-nilai ini akan dijadikan input untuk model ANFIS (Adaptive Neuro-Fuzzy Inference System) yang telah diprogram ke dalam ESP32. Model ini sebelumnya telah dilatih menggunakan data historis cuaca di MATLAB. ANFIS kemudian akan memproses input dan memberikan hasil klasifikasi, seperti klasifikasi suhu atau klasifikasi cuaca (cerah, berawan, hujan) dalam waktu dekat.

Data hasil pembacaan sensor dan hasil klasifikasi ANFIS akan dikirim ke Arduino IoT Cloud melalui koneksi Wi-Fi. ESP32 akan melakukan update variabel-variabel IoT yang telah ditentukan di dashboard cloud. Data dikirim menggunakan protokol MQTT atau HTTP yang mendukung pengiriman data secara ringan dan cepat.

Setelah sampai di Arduino IoT Cloud, data akan ditampilkan dalam bentuk numerik dan grafik pada dashboard. Pengguna dapat melihat suhu, kelembapan, tekanan udara, kecepatan angin, serta hasil klasifikasi cuaca secara real-time melalui web atau aplikasi mobile. Sistem juga dapat menambahkan fitur seperti peringatan dini jika parameter cuaca melewati batas tertentu.

**Hak Cipta :**

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

Setelah semua proses selesai, ESP32 akan menunggu jeda waktu tertentu (2 menit), kemudian kembali ke proses awal untuk membaca sensor, melakukan klasifikasi baru, dan memperbarui data ke cloud. Maka, sistem ini bekerja otomatis dan terus menerus selama alat mendapat pasokan daya dan koneksi Wi-Fi.

### 3.2.4 Spesifikasi Alat

Alat sistem pemantauan dan klasifikasi cuaca berbasis IoT ini memiliki beberapa spesifikasi sebagai berikut:

1. Mampu mendeteksi suhu, kelembapan, dan tekanan menggunakan sensor BME280
2. Mampu mendeteksi kecepatan angin menggunakan sensor Anemometer
3. Mampu memonitoring hasil pembacaan sensor dari mikrokontroler ESP32 pada Arduino IoT Cloud
4. Mampu menganalisis data untuk klasifikasi cuaca menggunakan metode ANFIS.

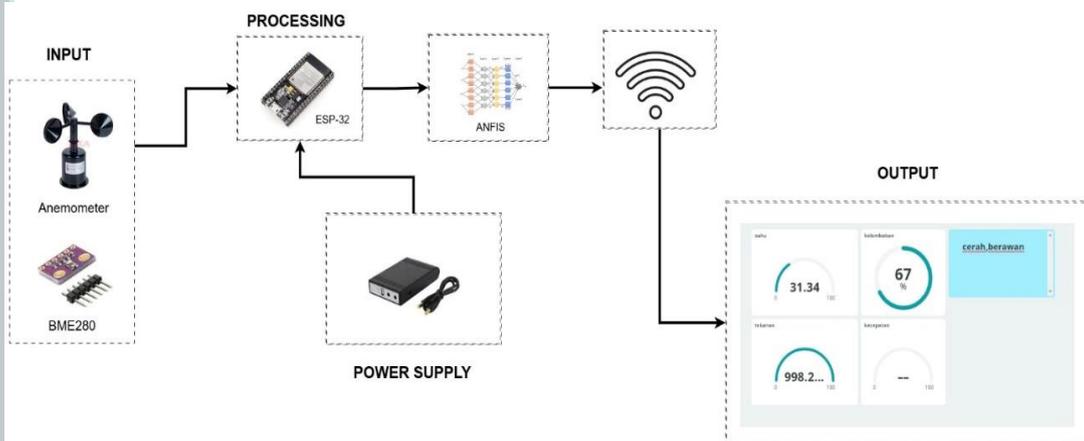
### 3.3 Merancang sistem klasifikasi cuaca

Merancang sistem klasifikasi cuaca merupakan langkah penting dalam menghadirkan solusi teknologi yang mampu memantau dan memklasifikasi kondisi atmosfer secara real-time dan akurat. Sistem ini dibangun dengan mengintegrasikan perangkat keras berupa sensor-sensor lingkungan dan mikrokontroler, serta perangkat lunak yang mendukung analisis data dan pengambilan keputusan berbasis kecerdasan buatan. Dalam penelitian ini, proses perancangan dimulai dari pemilihan parameter cuaca utama seperti suhu, kelembapan, tekanan udara, dan kecepatan angin, kemudian dilanjutkan dengan pemilihan sensor yang sesuai dan pengembangan algoritma klasifikasi menggunakan metode *Adaptive Neuro-Fuzzy Inference System (ANFIS)*. Tahapan ini mencakup desain alur sistem, pemrograman perangkat, hingga integrasi data dengan platform *IoT* untuk memantau hasil klasifikasi secara langsung. Perancangan yang matang menjadi dasar keberhasilan sistem dalam menghasilkan klasifikasi cuaca yang andal dan responsif terhadap perubahan lingkungan.

### Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
- a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
- b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkannya dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

### 3.3.1 Diagram Blok



Gambar 3. 4 Diagram Blok Sistem

### 3.3.2 Akuisisi Data Sensor (Input)

Sistem ini menggunakan dua jenis sensor untuk membaca parameter lingkungan:

- **Sensor BME280**

Berfungsi untuk mengukur:

- Suhu udara (dalam °C)
- Tekanan udara (dalam hPa)
- Kelembaban udara (dalam %)

Sensor ini berkomunikasi dengan ESP32 melalui protokol I2C, dan dapat mengirimkan data secara presisi dengan interval tertentu (misalnya setiap 2 menit).

- **Sensor Anemometer (Kecepatan Angin)**

Digunakan untuk mengukur:

- Kecepatan angin (dalam m/s)

Sensor ini bekerja melalui protokol Modbus RS485, sehingga dalam sistem digunakan modul konversi MAX485 untuk menghubungkan sensor dengan mikrokontroler ESP32.

### 3.3.3 Mikrokontroler ESP32 sebagai Unit Pemroses

ESP32 berperan sebagai pusat pemrosesan utama yang memiliki beberapa tugas penting:

**Hak Cipta :**

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkannya dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

- Membaca data dari sensor BME280 dan anemometer.
- Mengubah data analog/digital sensor menjadi nilai terukur seperti suhu, tekanan, kelembapan, dan kecepatan angin.
- Mengirimkan data tersebut ke Arduino IoT Cloud melalui koneksi WiFi secara real-time.

Selain itu, ESP32 juga bertugas mengirimkan data ke MATLAB untuk dianalisis dengan metode ANFIS

### 3.3.4 Platform Cloud: Arduino IoT Cloud (Monitoring & Kontrol)

ESP32 terhubung ke Arduino IoT Cloud untuk memungkinkan:

- Monitoring data sensor secara real-time melalui dashboard web.
- Melihat kondisi suhu, tekanan, kelembapan, dan kecepatan angin dari mana saja menggunakan perangkat yang terhubung internet.
- Menyimpan data ke cloud sebagai histori.

### 3.3.5 Pemrosesan : Adaptive Neuro-Fuzzy Inference System (ANFIS)

Di tahap ini, data dari sensor yang telah dikumpulkan dan dikirimkan ke komputer akan dianalisis menggunakan metode ANFIS yang dijalankan melalui software MATLAB.

#### 3.3.5.1 Persiapan Data

Data dari sensor dikumpulkan dalam format tabel yang berisi:

- Suhu (°C)
- Tekanan (hPa)
- Kelembapan (%)
- Kecepatan angin (m/s)
- Label cuaca (misalnya: cerah, berawan, hujan)

Data dilabeli berdasarkan aturan logika atau referensi data cuaca BMKG. Dataset ini kemudian dibagi menjadi dua bagian:

- Data training: untuk melatih model ANFIS.
- Data pengujian: untuk menguji keakuratan model.

#### 3.3.5.2 Struktur ANFIS

ANFIS terdiri dari 5 layer:

1. Fuzzifikasi (fungsi keanggotaan seperti Gaussian atau Triangular)
2. Aturan fuzzy (kombinasi himpunan input)

**Hak Cipta :**

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkannya dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

3. Normalisasi bobot
4. Konsekuen (fungsi linier  $p \cdot x + q \cdot y + r$ )
5. Output (penjumlahan seluruh aturan aktif)

### 3.3.5.3 Pelatihan Model

Model ANFIS dilatih menggunakan metode Hybrid Learning:

- Forward Pass: Parameter linier dihitung menggunakan Least Squares Estimation.
- Backward Pass: Parameter non-linier (fungsi keanggotaan) diperbarui dengan Gradient Descent.

### 3.3.5.4 Klasifikasi Output Cuaca

Setelah model dilatih, data real-time dari sensor bisa dimasukkan ke dalam model, dan akan menghasilkan nilai output antara 1 hingga 3:

- Nilai mendekati 1 → Cuaca Cerah
- Nilai mendekati 2 → Cuaca Berawan
- Nilai mendekati 3 → Cuaca Hujan

### 3.3.5.5 Visualisasi Hasil

Output ditampilkan kembali di dashboard atau console MATLAB dan bisa dikirimkan kembali ke ESP32 untuk ditampilkan di Arduino IoT Cloud Dashboard.

### 3.6.3.6 Power Supply

- Powerbank 12V 2A digunakan untuk menyuplai daya ke ESP32 dan sensor, sehingga alat bisa bekerja secara portabel dan tanpa ketergantungan ke sumber listrik tetap.

### 3.3.5.7 Output Sistem

Data dan hasil klasifikasi dari model ANFIS dikirim ke dashboard Arduino IoT Cloud dalam bentuk:

- Tampilan suhu, tekanan, kelembapan, dan kecepatan angin
  - Klasifikasi kondisi cuaca (cerah, berawan, hujan)
- Tampilan ini dapat diakses melalui PC atau smartphone.

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :

a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, - penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.

b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta

2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

## BAB IV PEMBAHASAN

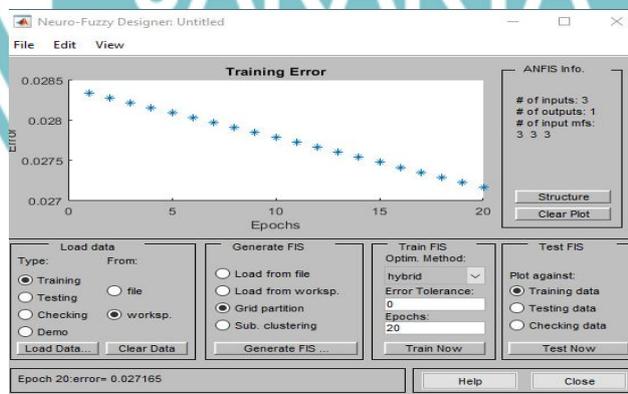
### 4.1 Hasil Klasifikasi Cuaca Menggunakan Metode ANFIS

Setelah data sensor yang terdiri dari suhu, tekanan, kelembapan, dan kecepatan angin diperoleh dari perangkat berbasis IoT, proses selanjutnya adalah melakukan klasifikasi cuaca dengan menggunakan metode Adaptive Neuro-Fuzzy Inference System (ANFIS). Metode ini menggabungkan keunggulan dari logika fuzzy dan jaringan saraf tiruan untuk membentuk model klasifikasi yang adaptif dan mampu menangani ketidakpastian data.

#### 4.1.1 Pembuatan Membership Function Input dan Output

Pada sistem logika fuzzy yang dirancang, terdapat parameter input berupa suhu, tekanan udara, kelembapan, dan kecepatan angin yang diperoleh dari sensor BME280 dan anemometer. Output dari sistem ini berupa klasifikasi kondisi cuaca menjadi Cerah, Berawan, atau Hujan. Proses pembuatan membership function untuk masing-masing parameter input dapat divisualisasikan menggunakan perangkat lunak seperti MATLAB. Visualisasi ini digunakan untuk mengamati bentuk keanggotaan setiap parameter dan memastikan bahwa sistem fuzzy mampu menangkap pola data cuaca secara akurat sebelum dilakukan pelatihan menggunakan metode Adaptive Neuro-Fuzzy Inference System (ANFIS).

##### 4.1.1.1 Training Error



Gambar 4. 1 *Taining Error*  
(Sumber: Dokume Pribadi)

### Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :

a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, -penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.

b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta

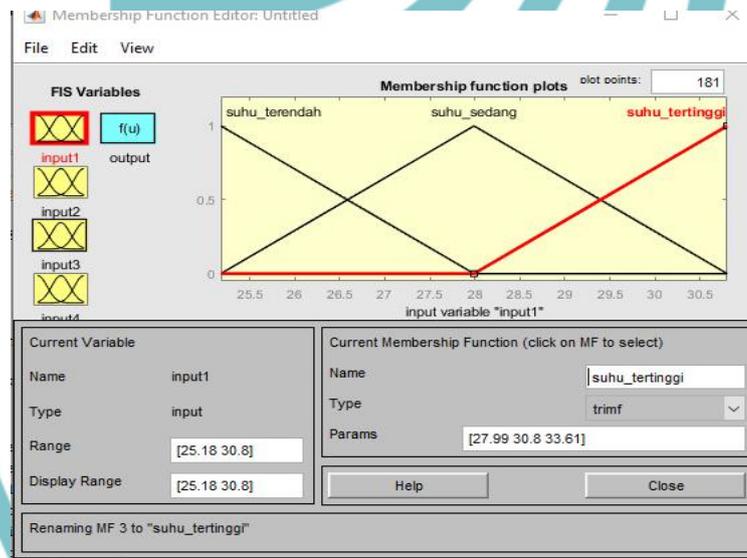
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

- Ini adalah tampilan *Neuro-Fuzzy Designer* pada MATLAB.
- Grafik menunjukkan penurunan *Training Error* terhadap jumlah *Epoch*
- Terlihat *Training Error* turun dari sekitar 0.0285 ke 0.027165 pada epoch ke-20, yang menandakan proses training berjalan baik dengan error yang terus menurun.

Pengaturan:

- Jumlah input: 4
- Output: 1
- Membership Function tiap input: 3
- Metode training: **Hybrid**
- Error tolerance: **0**

#### 4.1.1.2 Suhu



Gambar 4. 2 MF Suhu  
(Sumber: Dokume Pribadi)

#### A. Suhu Terendah

- **Rentang nilai:** Sekitar 25 – 28.8°C
- **Parameter:** [25, 25, 28.8]
- **Bentuk:** Segitiga dengan puncak pada nilai suhu yang lebih rendah ( $\mu = 1$  pada suhu 25°C).
- Fungsi ini merepresentasikan suhu yang tergolong rendah atau sejuk.
- Semakin jauh dari 25 menuju 28.8, derajat keanggotaannya menurun.

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :

a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.

b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta

2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

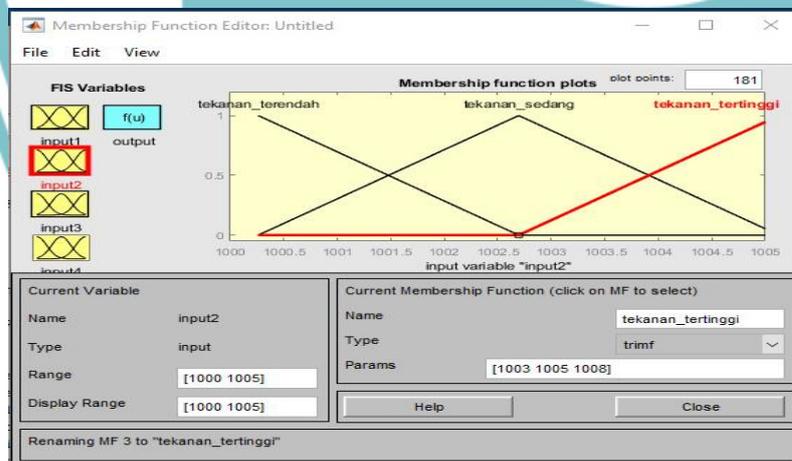
## B. Suhu Sedang

- **Rentang nilai:** Sekitar **28.8 – 30.8°C**
- **Parameter:** [27.99, 30.8, 33.61] (*mengacu dari gambar*)
- **Bentuk:** Segitiga dengan puncak ( $\mu = 1$ ) pada suhu **30.8°C**.
- Menunjukkan kondisi suhu sedang, tidak terlalu panas maupun dingin.
- Nilai keanggotaan tertinggi dicapai di sekitar suhu 30.8, lalu menurun ke kiri (27.99) dan ke kanan (33.61).

## C. Suhu Tertinggi

- **Rentang nilai:** **30.8 – 33.61°C**
- **Parameter:** [30.8, 33.61, 33.61]
- **Bentuk:** Segitiga dengan puncak ( $\mu = 1$ ) pada suhu **33.61°C**.
- Menggambarkan kondisi suhu tinggi atau panas.
- Semakin mendekati 33.61, nilai derajat keanggotaan menuju kategori suhu tinggi semakin besar.

### 4.1.1.3 Tekanan Udara



Gambar 4. 3 MF Tekanan Udara  
(Sumber: Dokume Pribadi)

## A. Tekanan Terendah

- **Rentang nilai:** **1000 – 1003 hPa**
- **Parameter:** [1000, 1000, 1003]
- **Bentuk:** Segitiga dengan nilai keanggotaan maksimum ( $\mu = 1$ ) pada tekanan **1000 hPa**.

**Hak Cipta :**

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :

a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.

b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta

2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

- Menandakan tekanan udara yang rendah.
- Derajat keanggotaan akan menurun seiring mendekati nilai 1003.

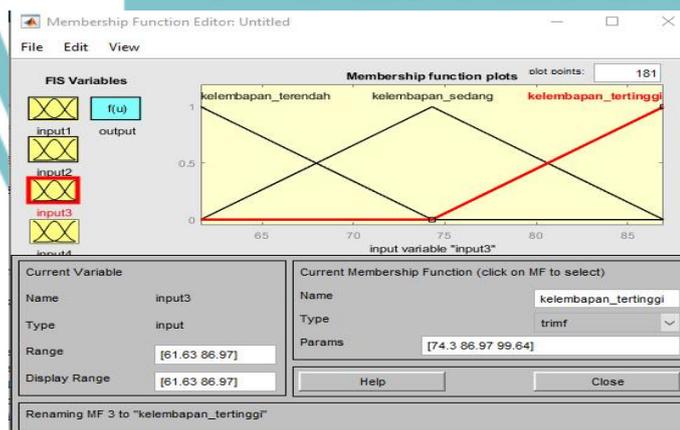
**B. Tekanan Sedang**

- **Rentang nilai: 1003 – 1005 hPa**
- **Parameter:** [1000, 1003, 1005]
- **Bentuk:** Segitiga dengan puncak ( $\mu = 1$ ) pada tekanan **1003 hPa**.
- Mewakili kondisi tekanan udara normal atau sedang.
- Keanggotaan maksimum pada nilai 1003, lalu turun ke kiri (1000) dan kanan (1005).

**C. Tekanan Tertinggi**

- **Rentang nilai: 1003 – 1005 hPa**
- **Parameter:** [1003, 1005, 1008]
- **Bentuk:** Segitiga dengan puncak ( $\mu = 1$ ) pada tekanan **1005 hPa**.
- Menandakan tekanan udara tinggi.
- Semakin mendekati nilai 1008, keanggotaan menuju "tekanan tinggi" semakin besar.

**4.1.1.4 Kelembapan Udara**



Gambar 4. 4 MF Kelembapan Udara  
(Sumber: Dokume Pribadi)

**A. Kelembapan Terendah**

- **Rentang nilai: 61 – 74.3%**
- **Parameter:** [61, 61, 74.3]

### Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :

a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.

b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta

2. Dilarang mengumumkannya dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

- **Bentuk:** Segitiga dengan puncak ( $\mu = 1$ ) pada kelembapan **61%**.
- Menggambarkan kondisi kelembapan rendah atau kering.
- Keanggotaan menurun ketika mendekati nilai 74.3.

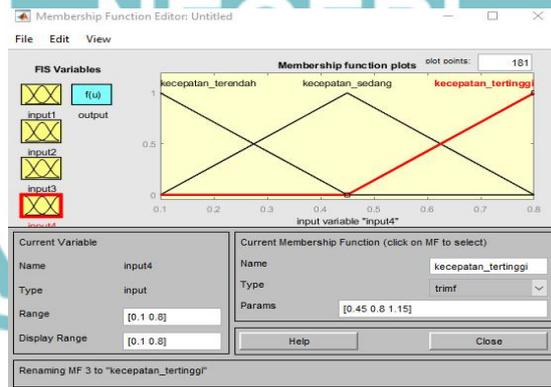
### B. Kelembapan Sedang

- **Rentang nilai:** 74.3 – 86.97%
- **Parameter:** [74.3, 86.97, 99.64]
- **Bentuk:** Segitiga dengan puncak ( $\mu = 1$ ) pada kelembapan **86.97%**.
- Mewakili kelembapan dalam kondisi sedang.
- Derajat keanggotaan tertinggi pada 86.97, lalu menurun ke kiri dan kanan.

### C. Kelembapan Tertinggi

- **Rentang nilai:** 86.97 – 99.64%
- **Parameter:** [86.97, 99.64, 99.64]
- **Bentuk:** Segitiga dengan puncak ( $\mu = 1$ ) pada **99.64%**.
- Menggambarkan kondisi kelembapan sangat tinggi atau lembap sekali.
- Semakin mendekati 99.64, derajat keanggotaan ke kategori ini semakin besar.

#### 4.1.1.5 Kecepatan angin



Gambar 4. 5 MF Kecepatan Angin  
(Sumber: Dokume Pribadi)

### A. Kecepatan Terendah

- **Rentang nilai:** 0.1 – 0.45 m/s
- **Parameter:** [0.1, 0.1, 0.45]

**Hak Cipta :**

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :

a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.

b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta

2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

- **Bentuk:** Segitiga dengan puncak ( $\mu = 1$ ) pada kecepatan **0.1 m/s**.
- Mewakili kondisi kecepatan angin rendah atau tenang.
- Keanggotaan menurun seiring mendekati 0.45 m/s.

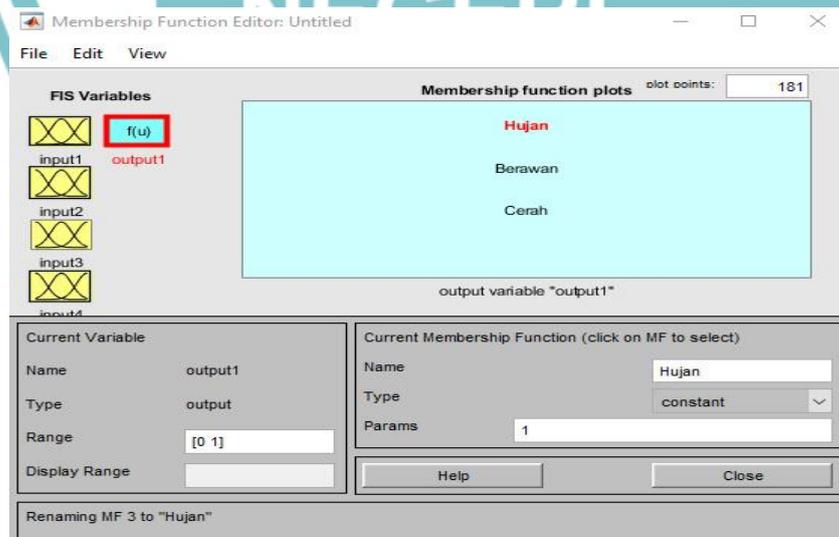
**B. Kecepatan Sedang**

- **Rentang nilai:** **0.45 – 0.8 m/s**
- **Parameter:** [0.45, 0.8, 1.15]
- **Bentuk:** Segitiga dengan puncak ( $\mu = 1$ ) pada **0.8 m/s**.
- Menggambarkan kondisi angin sedang.
- Derajat keanggotaan tertinggi pada 0.8, kemudian menurun ke kiri dan kanan.

**C. Kecepatan Tertinggi**

- **Rentang nilai:** **0.8 – 1.15 m/s**
- **Parameter:** [0.8, 1.15, 1.15]
- **Bentuk:** Segitiga dengan puncak ( $\mu = 1$ ) pada kecepatan **1.15 m/s**.
- Menandakan kondisi kecepatan angin yang tinggi atau kencang.
- Semakin mendekati 1.15, nilai keanggotaan ke kategori ini semakin besar.

**4.1.1.6 Output**



Gambar 4. 6 MF Output  
(Sumber: Dokume Pribadi)

**Hak Cipta :**

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

Pada implementasi ANFIS (Adaptive Neuro-Fuzzy Inference System) menggunakan MATLAB R2021a, fungsi keanggotaan (membership functions) untuk variabel keluaran diatur melalui Membership Function Editor dari Fuzzy Logic Toolbox. Setelah memasukkan keempat variabel masukan—yaitu suhu (input1), tekanan (input2), kelembapan (input3), dan kecepatan angin (input4)—selanjutnya kita menambahkan satu variabel keluaran (output1) yang akan merepresentasikan kategori cuaca.

Dalam sistem Sugeno yang digunakan, variabel output1 didefinisikan memiliki tiga fungsi keanggotaan tipe constant, yaitu:

- Hujan dengan parameter konstanta 1,
- Berawan dengan parameter konstanta 0.5,
- Cerah dengan parameter konstanta 0.

Setiap fungsi keanggotaan ini merepresentasikan nilai output implikasi fuzzy pada tahap inferensi. Nilai konstanta tersebut akan menjadi consequent (aturan if-then) ketika suatu aturan fuzzy terpenuhi. Rentang nilai output (Display Range) disetel pada interval [0, 1] untuk memudahkan proses normalisasi dan interpretasi hasil akhir ANFIS.

Pada jendela Membership Function Editor:

1. Panel kiri menampilkan daftar variabel FIS, dimana output1 sedang dipilih.
2. Panel kanan atas memperlihatkan plot fungsi keanggotaan “Hujan”, “Berawan”, dan “Cerah”.
3. Panel kanan bawah (“Current Membership Function”) memungkinkan penamaan ulang, pemilihan tipe MF (constant, linear, dll.), serta pengisian parameter MF sesuai kebutuhan model.

Dengan konfigurasi ini, setiap aturan fuzzy yang dilatih ANFIS akan menghitung output sebagai salah satu dari ketiga nilai konstanta di atas, yang nantinya diuji terhadap data historis BMKG untuk memperoleh akurasi klasifikasi kondisi cuaca (cerah, berawan, atau hujan).

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :

a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, - penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.

b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta

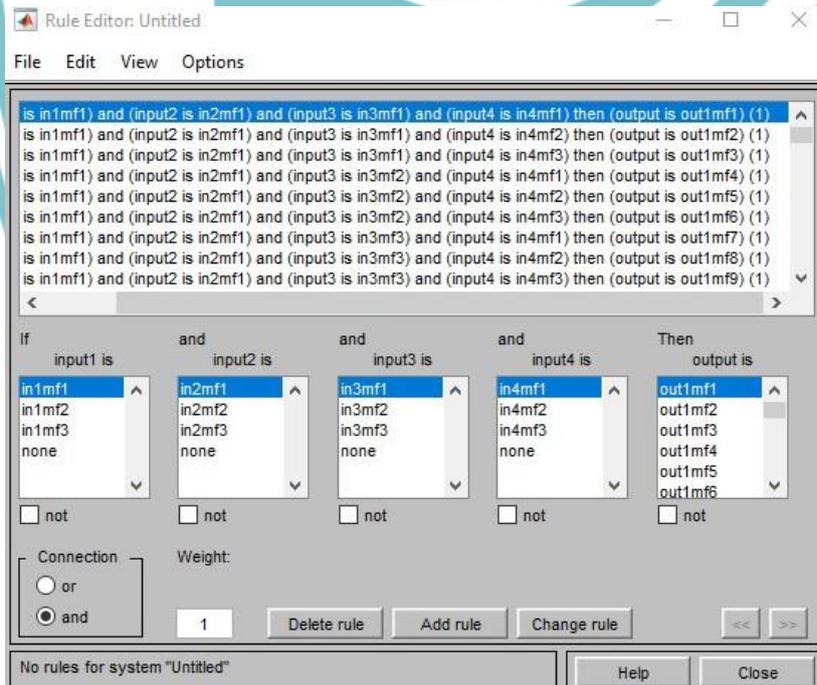
2. Dilarang mengumumkannya dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

4.1.2 Pembuatan Kaidah Aturan ANFIS

Tabel 4. 1 Aturan ANFIS

No	Suhu	Tekanan	Kelembapan	Angin	Output Cuaca
1	Panas	Tinggi	Rendah	Pelan	Cerah
2	Sedang	Normal	Sedang	Sedang	Berawan
3	Dingin	Rendah	Tinggi	Kencang	Hujan
4	Panas	Normal	Rendah	Sedang	Cerah
5	Sedang	Tinggi	Sedang	Kencang	Berawan
6	Dingin	Normal	Tinggi	Sedang	Hujan
7	Panas	Tinggi	Sedang	Pelan	Cerah
8	Sedang	Rendah	Sedang	Pelan	Berawan
9	Dingin	Normal	Tinggi	Kencang	Hujan

Berdasarkan aturan yang telah dibuat dalam tabel MacVicar Whelan diatas maka akan disimulasikan dengan software Matlab 2021



Gambar 4. 7 Rule Editor (Sumber: Dokume Pribadi)

4.1.3 Pengujian Simulasi Rules Fuzzy

Pengujian simulasi *rules fuzzy* dilakukan untuk mengevaluasi bagaimana sistem *Fuzzy Inference System (FIS)* bekerja dalam merespons data input cuaca dan menghasilkan klasifikasi berdasarkan aturan-aturan fuzzy yang telah dibentuk.

Proses ini dilakukan menggunakan fitur *Rule Viewer* yang tersedia dalam perangkat lunak *MATLAB*. *Rule Viewer* merupakan alat bantu visual yang memungkinkan pengguna untuk melihat secara langsung bagaimana setiap nilai input dipetakan ke dalam fungsi keanggotaan (*membership function*), bagaimana aturan-aturan (*rules*) fuzzy diaktifkan, serta bagaimana sistem menentukan output akhir berdasarkan kombinasi dari semua aturan tersebut.

Pada sistem yang diuji, terdapat empat parameter input utama yang digunakan sebagai variabel masukan, yaitu suhu udara, tekanan udara, kelembapan relatif, dan kecepatan angin. Parameter-parameter ini diperoleh dari sensor *BME280* untuk suhu, kelembapan, dan tekanan, serta *anemometer* untuk mengukur kecepatan angin. Nilai-nilai input tersebut kemudian dimasukkan ke dalam sistem *FIS* yang telah dilatih menggunakan metode *Adaptive Neuro-Fuzzy Inference System (ANFIS)* sebelumnya.

Dalam tampilan *Rule Viewer*, setiap kolom input menunjukkan grafik fungsi keanggotaan fuzzy yang merepresentasikan kategori nilai tertentu, seperti “rendah”, “sedang”, atau “tinggi”. Area yang diarsir berwarna kuning menunjukkan derajat keanggotaan (*degree of membership*) dari nilai input terhadap masing-masing fungsi fuzzy. Setelah itu, sistem memproses informasi ini melalui aturan-aturan fuzzy yang telah ditetapkan. Dalam sistem ini terdapat sebanyak 81 aturan (*rules*), yang merupakan hasil kombinasi dari semua kemungkinan nilai input berdasarkan jumlah fungsi keanggotaan yang digunakan. Setiap aturan akan menghasilkan kontribusi tertentu terhadap output akhir, tergantung dari tingkat aktivasi (*activation level*) masing-masing aturan.

Kolom output pada *Rule Viewer* menunjukkan hasil akhir dari proses inferensi fuzzy, yaitu berupa satu nilai numerik yang mewakili kategori cuaca. Sebagai contoh, output dengan nilai mendekati 1 mengindikasikan cuaca cerah, nilai sekitar 2 mengindikasikan cuaca berawan, dan nilai mendekati 3 menunjukkan potensi cuaca hujan. Proses ini mencakup lima tahapan utama dalam sistem *ANFIS*, yaitu *fuzzification*, perhitungan kekuatan aturan, normalisasi, perhitungan konsekuen, dan *defuzzification*. Semua proses tersebut berjalan secara otomatis dan divisualisasikan secara interaktif melalui antarmuka *Rule Viewer*.

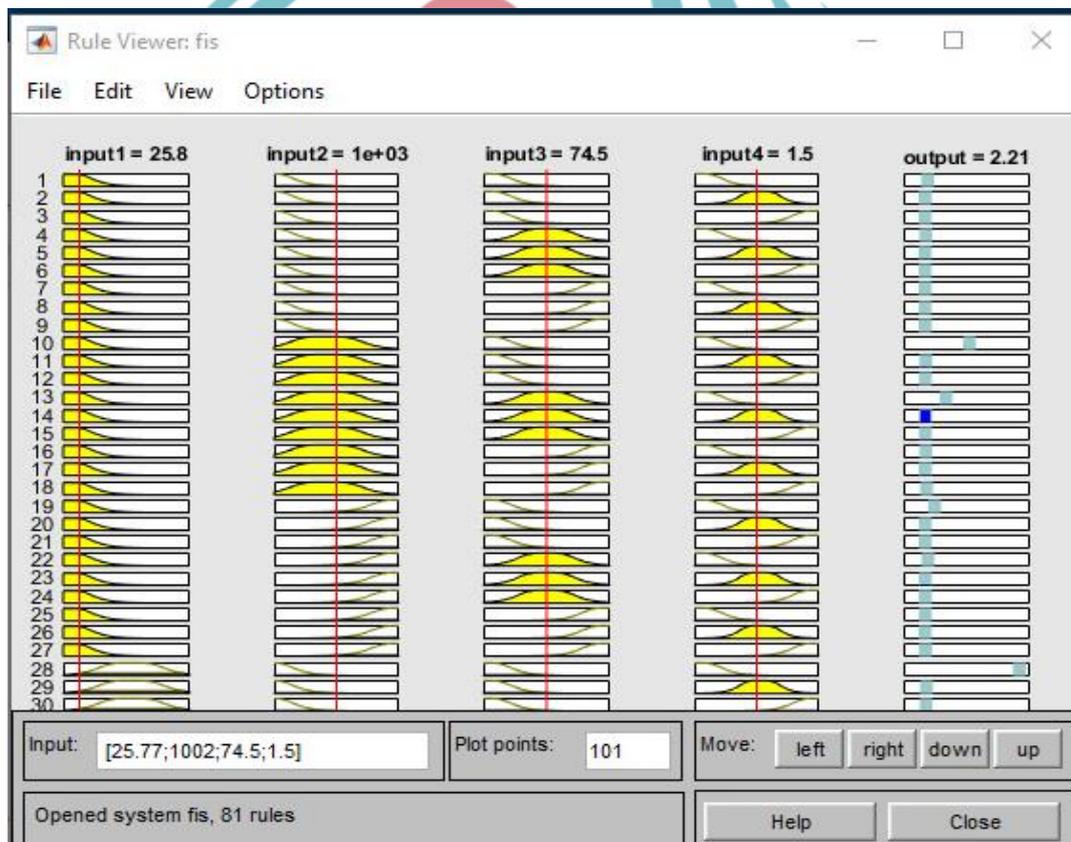
#### Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkannya dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

### Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

Pengujian ini sangat penting karena memberikan gambaran menyeluruh tentang bagaimana sistem memproses informasi dan menghasilkan klasifikasi. Selain itu, dengan menggunakan *Rule Viewer*, pengguna dapat mengevaluasi apakah aturan fuzzy yang dibentuk telah berfungsi sesuai harapan, dan apakah sistem memberikan respons yang logis terhadap variasi input. Oleh karena itu, tahap ini menjadi bagian krusial dalam validasi desain sistem klasifikasi cuaca berbasis *ANFIS* yang dirancang dalam penelitian ini.



Gambar 4. 8 *Rule Viewer*  
(Sumber: Dokume Pribadi)

#### 4.1.4 Realisasi Pemrograman

```
>> data1=[];
>> data2=[];
>> data3=[];
>> data4=[];
>> data5=[];
>> allData = [data1; data2; data3;data4;data5];
```

Gambar 4. 9 Program Penerimaan Data  
(Sumber: Dokume Pribadi)

**Hak Cipta :**

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :

a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.

b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta

2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

- data1 = [];

Menginisialisasi variabel data1 sebagai array kosong. Variabel ini digunakan untuk menyimpan data sensor dari hari pertama pengambilan data cuaca.

- data2 = [];

Menginisialisasi variabel data2 sebagai array kosong untuk menyimpan data sensor dari hari kedua.

- data3 = [];

Menginisialisasi variabel data3 sebagai array kosong untuk menyimpan data sensor dari hari ketiga.

- data4 = [];

Menginisialisasi variabel data4 sebagai array kosong untuk menyimpan data sensor dari hari keempat.

- data5 = [];

Menginisialisasi variabel data5 sebagai array kosong untuk menyimpan data sensor dari hari kelima.

- allData = [data1; data2; data3; data4; data5];

Menggabungkan seluruh data dari data1 hingga data5 menjadi satu matriks vertikal allData. Penggabungan dilakukan menggunakan tanda titik koma ; yang berarti baris demi baris. Variabel ini nantinya digunakan sebagai data pelatihan (training data) untuk model ANFIS dalam memklasifikasi cuaca berdasarkan parameter suhu, tekanan, kelembapan, dan kecepatan angin.

```

Command Window
>> if suhu > 30 && kelembapan < 70 && tekanan > 1010 && angin < 10
    output(i) = 1; % Cerah
elseif suhu >= 27 && suhu <= 30 && kelembapan >= 70 && kelembapan <= 85 && tekanan >= 1005 && tekanan <=
    output(i) = 2; % Berawan
else
    output(i) = 3; % Hujan
end
>> trainingData = [allData output];
input = trainingData(:,1:4);
target = trainingData(:,5);
>> ruleList = [
    3 3 1 1 1 1 1 % Cerah
    2 2 2 2 1 1 1 % Berawan
    1 1 3 3 1 1 1 % Hujan
];

ruleList = [
    3 2 2 1 1 1 1; % Suhu panas, kelembapan sedang → Cerah
    2 2 3 1 3 1 1 % Suhu sedang, kelembapan tinggi → Hujan
];

```

Gambar 4. 10 Program Aturan Klasifikasi Cuaca  
(Sumber: Dokume Pribadi)

**Hak Cipta :**

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :

a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.

b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta

2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

cuacaFIS = newfis('KlasifikasiCuaca');

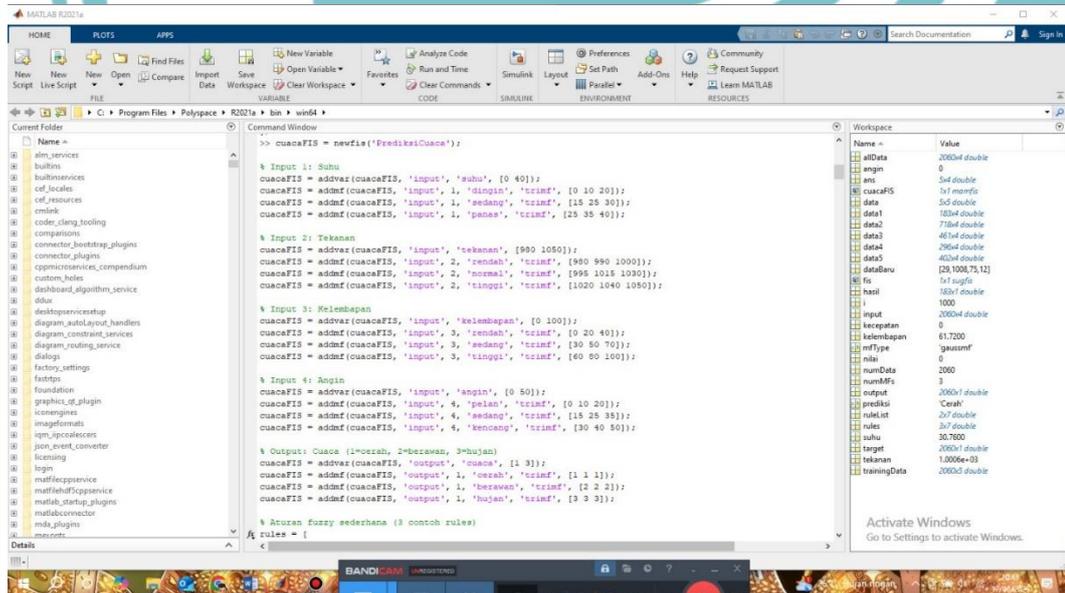
Digunakan untuk membuat sistem inferensi fuzzy baru bernama "KlasifikasiCuaca" yang menjadi dasar logika fuzzy sebelum digunakan pada pelatihan ANFIS.

- addvar dan addmf untuk setiap variabel:

Fungsi addvar digunakan untuk mendefinisikan variabel input dan output dalam sistem fuzzy, seperti suhu, tekanan, kelembapan, dan angin. Selanjutnya, addmf menambahkan membership function (fungsi keanggotaan) untuk masing-masing variabel menggunakan fungsi trimf (triangular membership function). Setiap input diberi 3 kategori: rendah, sedang, tinggi, dan output diberi 3 kategori: cerah (1), berawan (2), hujan (3).

- rules = [...];

Bagian ini berisi aturan fuzzy yang ditulis dalam bentuk matriks. Setiap baris merepresentasikan satu aturan logika fuzzy. Misalnya: 3 3 1 1 1 1 berarti jika suhu panas (3), tekanan tinggi (3), kelembapan rendah (1), dan angin pelan (1) maka output cuaca adalah cerah (1).



```

>> cuacaFIS = newfis('PrediksiCuaca');

% Input 1: Suhu
cuacaFIS = addvar(cuacaFIS, 'input', 'suhu', [0 40]);
cuacaFIS = addmf(cuacaFIS, 'input', 1, 'dingin', 'trimf', [0 10 20]);
cuacaFIS = addmf(cuacaFIS, 'input', 1, 'sedang', 'trimf', [15 25 30]);
cuacaFIS = addmf(cuacaFIS, 'input', 1, 'panas', 'trimf', [25 35 40]);

% Input 2: Tekanan
cuacaFIS = addvar(cuacaFIS, 'input', 'tekanan', [990 1050]);
cuacaFIS = addmf(cuacaFIS, 'input', 2, 'rendah', 'trimf', [990 990 1000]);
cuacaFIS = addmf(cuacaFIS, 'input', 2, 'normal', 'trimf', [995 1015 1030]);
cuacaFIS = addmf(cuacaFIS, 'input', 2, 'tinggi', 'trimf', [1020 1040 1050]);

% Input 3: Kelembapan
cuacaFIS = addvar(cuacaFIS, 'input', 'kelembapan', [0 100]);
cuacaFIS = addmf(cuacaFIS, 'input', 3, 'rendah', 'trimf', [0 20 40]);
cuacaFIS = addmf(cuacaFIS, 'input', 3, 'sedang', 'trimf', [30 50 70]);
cuacaFIS = addmf(cuacaFIS, 'input', 3, 'tinggi', 'trimf', [60 80 100]);

% Input 4: Angin
cuacaFIS = addvar(cuacaFIS, 'input', 'angin', [0 50]);
cuacaFIS = addmf(cuacaFIS, 'input', 4, 'pelan', 'trimf', [0 10 20]);
cuacaFIS = addmf(cuacaFIS, 'input', 4, 'sedang', 'trimf', [15 25 35]);
cuacaFIS = addmf(cuacaFIS, 'input', 4, 'kencang', 'trimf', [30 40 50]);

% Output: Cuaca (1=cerah, 2=berawan, 3=hujan)
cuacaFIS = addvar(cuacaFIS, 'output', 'cuaca', [1 3]);
cuacaFIS = addmf(cuacaFIS, 'output', 1, 'cerah', 'trimf', [1 1 1]);
cuacaFIS = addmf(cuacaFIS, 'output', 1, 'berawan', 'trimf', [2 2 2]);
cuacaFIS = addmf(cuacaFIS, 'output', 1, 'hujan', 'trimf', [3 3 3]);

% Aturan fuzzy sederhana (3 contoh rules)
rules = [
    3 3 1 1 1 1;
    3 3 1 1 1 1;
    3 3 1 1 1 1;
];

```

Gambar 4. 11 Klasifikasi Klasifikasi Cuaca  
(Sumber: Dokume Pribadi)

cuacaFIS = addrule(cuacaFIS, rules);

Fungsi ini menambahkan aturan fuzzy ke dalam sistem FIS yang telah dibuat.

- hasil = evalfis(data, cuacaFIS);

Digunakan untuk melakukan evaluasi klasifikasi cuaca dari data input

**Hak Cipta :**

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :

a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.

b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta

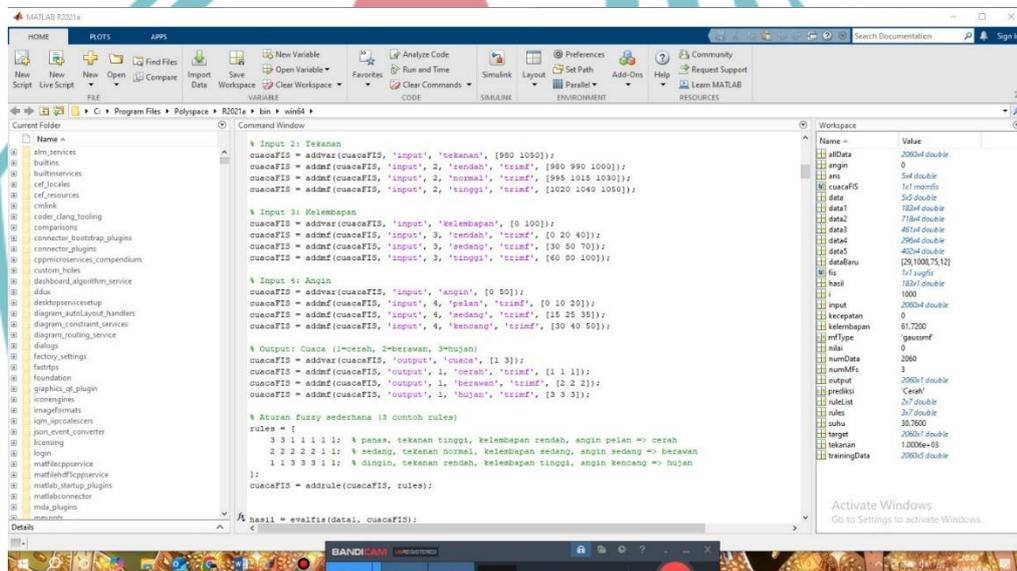
2. Dilarang mengumumkannya dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

menggunakan sistem inferensi fuzzy cuacaFIS. Output-nya berupa angka (1 untuk cerah, 2 untuk berawan, 3 untuk hujan).

• for i = 1:length(hasil)...

Perulangan ini digunakan untuk menampilkan hasil klasifikasi cuaca dari setiap baris data input. Output angka akan diklasifikasikan menjadi kategori cuaca dengan logika:

- Jika nilai output  $\leq 1.5 \rightarrow$  Cerah
- Jika nilai output  $> 1.5$  dan  $\leq 2.5 \rightarrow$  Berawan
- Jika nilai output  $> 2.5 \rightarrow$  Hujan



```

% Input 1: Tekanan
cuacaFIS = addvar(cuacaFIS, 'input', 'tekanan', [980 1050]);
cuacaFIS = addmf(cuacaFIS, 'input', 2, 'rendah', 'trimf', [980 990 1000]);
cuacaFIS = addmf(cuacaFIS, 'input', 2, 'normal', 'trimf', [995 1015 1030]);
cuacaFIS = addmf(cuacaFIS, 'input', 2, 'tinggi', 'trimf', [1020 1040 1050]);

% Input 2: Kelembapan
cuacaFIS = addvar(cuacaFIS, 'input', 'kelembapan', [0 100]);
cuacaFIS = addmf(cuacaFIS, 'input', 3, 'rendah', 'trimf', [0 20 40]);
cuacaFIS = addmf(cuacaFIS, 'input', 3, 'sedang', 'trimf', [30 50 70]);
cuacaFIS = addmf(cuacaFIS, 'input', 3, 'tinggi', 'trimf', [60 80 100]);

% Input 3: Angin
cuacaFIS = addvar(cuacaFIS, 'input', 'angin', [0 50]);
cuacaFIS = addmf(cuacaFIS, 'input', 4, 'pelan', 'trimf', [0 10 20]);
cuacaFIS = addmf(cuacaFIS, 'input', 4, 'sedang', 'trimf', [15 25 35]);
cuacaFIS = addmf(cuacaFIS, 'input', 4, ' kencang', 'trimf', [30 40 50]);

% Output: Cuaca (1=cerah, 2=berawan, 3=hujan)
cuacaFIS = addvar(cuacaFIS, 'output', 'cuaca', [1 3]);
cuacaFIS = addmf(cuacaFIS, 'output', 1, 'cerah', 'trimf', [1 1 1]);
cuacaFIS = addmf(cuacaFIS, 'output', 2, 'berawan', 'trimf', [2 2 2]);
cuacaFIS = addmf(cuacaFIS, 'output', 3, 'hujan', 'trimf', [3 3 3]);

% Aturan fuzzy sederhana (3 contoh rules)
rules = 1
    3 3 1 1 1 1 : % panas, tekanan tinggi, kelembapan rendah, angin pelan => cerah
    2 2 2 2 1 1 : % sedang, tekanan normal, kelembapan sedang, angin sedang => berawan
    1 1 3 3 1 1 : % dingin, tekanan rendah, kelembapan tinggi, angin kencang => hujan

cuacaFIS = addrule(cuacaFIS, rules);

hasil = evalfis(data, cuacaFIS);

```

Gambar 4. 12 Program Pelatihan Data  
(Sumber: Dokume Pribadi)

trainingData = [allData output];

Menggabungkan seluruh data input (allData) dan output label (output) ke dalam satu matriks sebagai data pelatihan untuk metode ANFIS.

### Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```
>> [trainedFis, trainError] = anfis(trainingData, fis, 100);
Warning: Specifying a FIS structure will be removed in a future release. Update your code to use FIS objec
> In anfis>anfisWithOptionalInputs (line 310)
In anfis (line 69)

ANFIS info:
  Number of nodes: 193
  Number of linear parameters: 405
  Number of nonlinear parameters: 24
  Total number of parameters: 429
  Number of training data pairs: 2060
  Number of checking data pairs: 0
  Number of fuzzy rules: 81

Start training ANFIS ...

 1  0.21529
 2  0.215207
 3  0.215125
 4  0.215043
Step size increases to 0.011000 after epoch 5.
 5  0.214961
 6  0.21488
 7  0.214791
 8  0.214702
Step size increases to 0.012100 after epoch 9.
 9  0.214613
10  0.214524
11  0.214425
12  0.214326
Step size increases to 0.013310 after epoch 13.
13  0.214227
14  0.214127
fx
```

Gambar 4. 13 Program Training Anfis dan Output nya  
(Sumber: Dokume Pribadi)

```
[trainedFis, trainError] = anfis(trainingData, fis, 100);
```

Perintah ini menjalankan pelatihan sistem ANFIS (Adaptive Neuro Fuzzy Inference System) menggunakan data pelatihan sebanyak 100 epoch. Model fis yang telah dibuat digunakan sebagai struktur awal ANFIS.

- Output hasil pelatihan ANFIS:

Informasi yang ditampilkan seperti jumlah node, parameter, dan fuzzy rules membantu memantau proses pelatihan. Bila RMSE (Root Mean Square Error) menurun, berarti model semakin baik dalam mempelajari pola data.

```
>> hasil = evalfis(datal, cuacaFIS);

for i = 1:length(hasil)
  if hasil(i) < 2
    prediksi = 'Cerah';
  elseif hasil(i) < 2.5
    prediksi = 'Berawan';
  elseif hasil(i) < 3
    prediksi = 'Hujan';
  end
  disp(['Baris ' num2str(i) ': ' prediksi]);
end
```

Gambar 4. 14 Program Hasil Klasifikasi Cuaca  
(Sumber: Dokumen Pribadi)

**Hak Cipta :**

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkannya dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

evalfis(data, trainedFis);

Setelah ANFIS dilatih, perintah ini digunakan untuk melakukan klasifikasi berdasarkan model yang telah dilatih (trainedFis) terhadap data baru.

- fprintf('%0.2f\t%s\n', nilai, klasifikasi);

Menampilkan hasil klasifikasi secara rapi dalam format nilai dan label klasifikasi.

```
Baris 1: Berawan
Baris 2: Berawan
Baris 3: Cerah
Baris 4: Berawan
Baris 5: Berawan
Baris 6: Berawan
Baris 7: Berawan
Baris 8: Berawan
Baris 9: Berawan
Baris 10: Berawan
Baris 11: Cerah
Baris 12: Cerah
Baris 13: Berawan
Baris 14: Berawan
```

Gambar 4. 15 Hasil Klasifikasi  
(Sumber: Dokumen Pribadi)

Gambar tersebut menunjukkan implementasi program MATLAB untuk mengklasifikasikan hasil output sistem ANFIS ke dalam tiga kategori kondisi cuaca, yaitu "Cerah", "Berawan", dan "Hujan". Pada bagian editor (jendela kode), terdapat skrip yang melakukan iterasi terhadap seluruh elemen dalam array hasil, yaitu kumpulan nilai klasifikasi numerik yang dihasilkan oleh model ANFIS. Setiap nilai hasil(i) disimpan dalam variabel nilai, kemudian diklasifikasikan menggunakan struktur percabangan if-elseif-else. Jika nilai klasifikasi lebih kecil dari 2, maka kondisi cuaca dianggap "Cerah"; jika nilai berada di antara 2 hingga kurang dari 2.5, maka diklasifikasikan sebagai "Berawan"; dan jika nilai sama dengan atau lebih dari 2.5, maka dikategorikan sebagai "Hujan". Hasil klasifikasi tersebut disimpan dalam variabel klasifikasi, lalu ditampilkan ke layar menggunakan perintah fprintf, yang mencetak nilai numerik dan label klasifikasinya secara berpasangan.

Pada jendela *Command Window*, ditampilkan hasil klasifikasi dari beberapa data uji. Sebagai contoh, nilai klasifikasi 1.00 dikategorikan sebagai "Cerah", nilai 2.00 sebagai "Berawan", dan nilai 3.00 sebagai "Hujan". Hasil ini menunjukkan

bahwa sistem ANFIS telah berhasil menghasilkan nilai numerik yang dapat dikonversi menjadi informasi kondisi cuaca yang dapat dimengerti oleh pengguna. Proses klasifikasi ini penting dalam sistem klasifikasi cuaca karena mengubah output model yang bersifat kontinu menjadi kategori diskrit yang mudah dianalisis, ditampilkan, atau digunakan dalam pengambilan keputusan lebih lanjut. Klasifikasi ini juga memperlihatkan bahwa sistem telah bekerja sesuai dengan skenario yang dirancang untuk membedakan kondisi cuaca berdasarkan kombinasi empat parameter masukan: suhu, tekanan udara, kelembapan, dan kecepatan angin.

#### 4.2 Evaluasi dan Validasi Hasil Klasifikasi

Pengujian sistem klasifikasi cuaca menggunakan metode Adaptive Neuro-Fuzzy Inference System (ANFIS) pada platform berbasis Internet of Things (IoT) dilakukan untuk memastikan bahwa hasil klasifikasi cuaca yang dihasilkan oleh sistem sesuai dengan kondisi aktual yang terjadi di lingkungan. Pengujian ini juga dilakukan sebagai validasi model ANFIS yang telah dilatih sebelumnya berdasarkan data sensor suhu, kelembapan, tekanan udara, dan kecepatan angin.

1. Lokasi Pengujian : Jl. Bambon 1, Beji Timur, Beji, Kota Depok, Jawa Barat
2. Waktu Pengujian : 19–20 Juni 2025
3. Pelaksana Pengujian : Tegar Tri Hartono
4. Pembimbing :
5. Tujuan Pengujian:
  - Menguji integrasi antara data sensor BME280 dan anemometer dengan platform Arduino IoT Cloud.
  - Menguji pengambilan data sensor secara real-time dan memastikan keakuratannya.
  - Memklasifikasi kondisi cuaca berdasarkan data input menggunakan model ANFIS yang telah dilatih di MATLAB.
  - Mengirim hasil klasifikasi kembali ke Arduino IoT Cloud dan menampilkannya pada dashboard secara real-time.

##### 4.2.1 Prosedur Pengujian

Berikut adalah hal-hal yang harus diperhatikan saat pengujian sistem klasifikasi cuaca berbasis ANFIS dengan sensor BME280 dan Anemometer:

#### Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, -penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkannya dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

**Hak Cipta :**

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

**4.2.2 Alat**

Tabel 4. 2 Alat

No	Alat	Jumlah	Keterangan
1	ESP32	1	Sebagai mikrokontroler untuk membaca data sensor dan mengirimkan ke cloud
2	Sensor BME280	1	Mengukur suhu, tekanan, dan kelembapan
3	Sensor Anemometer	1	Mengukur kecepatan angin
4	Modul RS485 (MAX485)	1	Digunakan untuk komunikasi sensor anemometer via Modbus
5	Power Bank 12V 2A	1	Sumber daya portable untuk menghidupkan sistem
6	Laptop HP	1	Digunakan untuk pemrograman ESP32 dan MATLAB untuk analisis klasifikasi
7	Kabel USB	1	Koneksi antara ESP32 dan laptop

**4.2.3 Langkah–Langkah Pengujian**

1. Menyiapkan semua komponen yang diperlukan seperti ESP32, sensor BME280, anemometer, kabel USB, dan power bank.
2. Melakukan pengecekan wiring antara sensor dengan ESP32, termasuk koneksi RS485 untuk anemometer.
3. Mengupload program Arduino ke ESP32 menggunakan Arduino IDE, serta mengatur koneksi ke Arduino IoT Cloud.
4. Mengaktifkan sistem dan memastikan data suhu, tekanan, kelembapan, dan kecepatan angin muncul di dashboard Arduino IoT Cloud.
5. Mengakses data real-time dari Arduino IoT Cloud ke MATLAB menggunakan API.

6. Memproses data input dengan metode Adaptive Neuro-Fuzzy Inference System (ANFIS) di MATLAB untuk memklasifikasi cuaca.
7. Mengirim kembali hasil klasifikasi (cerah, berawan, atau hujan) ke Arduino IoT Cloud agar ditampilkan di dashboard pengguna.
8. Mencatat hasil pengujian dan membandingkannya dengan referensi seperti BMKG atau Google Cuaca untuk validasi hasil.

#### 4.2.4 Tabel data sensor

Tabel 4. 3 Data pada hari pertama 19 Juni 2025

NO	Waktu (WIB)	Suhu (°C)	Tekanan Udara (hPa)	Kelembapan Udara (%)	Kecepatan Angin (m/s)
1	01.43	25,18	1002,82	86,97	0,8
2	09.32	28,92	1005,13	75,46	0,6
3	10,02	29,08	1004,89	71,64	0,2
4	10.55	29,05	1004,15	68,17	0,4
5	11.26	29,08	1003,48	66,51	0,4
6	12.00	29,62	1002,86	64,27	0,3
7	13.12	30,06	1002,11	61,63	0,2
8	09.30	29,39	1004,38	72,02	0,5
9	10.00	29,20	1004,16	72,69	0,5
10	10.40	30,00	1003,53	69,10	0,4
11	11.00	30,28	1003,46	68,25	0,4
12	11.50	30,45	1002,69	67,75	0,2
13	12.15	30,66	1002,41	67,13	0,3
14	13.00	30,80	1001,84	63,50	0,2
15	13.30	30,49	1001,37	67,34	0,2
16	14.20	30,63	1000,55	65,68	0,1
17	14.45	30,80	1000,35	65,48	0,1
18	15.00	30,77	1000,39	67,07	0,4
19	15.15	30,76	1000,30	66,86	0,3
20	15.35	30,70	1000,26	66,32	0,3

Pada subbab ini ditampilkan data hasil pengukuran parameter cuaca yang diperoleh dari sensor pada tanggal 19 Juni 2025. Data direkam secara berkala mulai pukul 01.43 WIB hingga pukul 15.35 WIB, menghasilkan total 20 entri data. Setiap entri mencakup lima parameter utama, yaitu waktu pencatatan, suhu udara (°C), tekanan udara (hPa), kelembapan udara (%), dan kecepatan angin (m/s). Sebagai contoh, pada pukul 01.43 WIB tercatat suhu sebesar 25,18 °C, tekanan udara 1002,82 hPa, kelembapan udara 86,97%, dan kecepatan angin 0,8 m/s. Data ini selanjutnya digunakan sebagai input pada proses pengujian sistem klasifikasi cuaca

#### Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :

a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.

b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta

2. Dilarang mengumumkannya dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

yang telah dibangun, yang dapat berupa metode logika fuzzy atau model ANFIS (Adaptive Neuro-Fuzzy Inference System).

#### 4.2.5 Program dan Hasil pengujian

```
>> for i = 1:length(hasil)
    nilai = hasil(i);
    if nilai < 2
        prediksi = 'Cerah';
    elseif nilai < 2.5
        prediksi = 'Berawan';
    else
        prediksi = 'Hujan';
    end
    fprintf('%0.2f\t%s\n', nilai, prediksi);
end
1.00    Cerah
3.00    Hujan
1.00    Cerah
1.00    Cerah
1.00    Cerah
3.00    Hujan
1.00    Cerah
1.00    Cerah
3.00    Hujan
1.00    Cerah
1.00    Cerah
3.00    Hujan
1.00    Cerah
3.00    Hujan
3.00    Hujan
1.00    Cerah
1.00    Cerah
1.00    Cerah
```

Gambar 4. 16 Pemrograman dan hasil  
(Sumber: Dokumen Pribadi)

Setelah proses pelatihan dan pengujian sistem dilakukan, nilai output dari model kemudian diklasifikasikan ke dalam kategori cuaca tertentu menggunakan skrip program MATLAB. Program tersebut membaca setiap nilai hasil (output) dari model dan melakukan pengelompokan berdasarkan aturan: jika nilai lebih kecil dari 2 maka diklasifikasikan sebagai "Cerah", jika nilai berada di antara 2 hingga kurang dari 2,5 maka dikategorikan sebagai "Berawan", dan jika nilai lebih besar atau sama dengan 2,5 maka dikategorikan sebagai "Hujan". Berdasarkan hasil eksekusi program, diperoleh klasifikasi kondisi cuaca untuk masing-masing data, yang sebagian besar menunjukkan hasil "Cerah" dan "Hujan". Tidak terdapat klasifikasi "Berawan" dalam hasil pengujian ini, yang mengindikasikan bahwa tidak ada nilai output yang berada di antara 2 hingga 2,5. Hasil ini memberikan gambaran bahwa model klasifikasi cuaca mampu mengklasifikasikan data secara langsung dan konsisten terhadap dua kondisi utama, yaitu cerah dan hujan.

#### Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta