

BAB IV HASIL DAN ANALISIS

4.1 Pendahuluan

Bagian ini memberikan gambaran umum mengenai hasil penerapan dan pengujian sistem yang dibahas pada Bab IV. Pembahasan pada bab ini mencakup hasil pengujian sensor, penerapan sistem pada setiap *node*, komunikasi data, serta kinerja algoritma XGBoost dalam melakukan prediksi kondisi hujan. Hasil yang diperoleh dari setiap pengujian kemudian dianalisis berdasarkan parameter dan metode pengukuran yang telah dijelaskan pada bab sebelumnya.

Dalam penyajian hasil, *node* merupakan perangkat akuisisi data yang terdiri atas mikrokontroler dan sensor untuk melakukan pengukuran parameter cuaca pada lokasi pengamatan. Pusat pemrosesan merupakan Raspberry Pi 4 Model B yang berfungsi menerima dan mengelola data dari *node* melalui protokol MQTT. Data yang telah diterima kemudian diproses menggunakan Node-RED dan diteruskan ke *database* dan *hosting* Firebase untuk kemudian dapat ditampilkan pada sebuah *website*.

Pengujian sistem dibagi menjadi beberapa bagian, yaitu pengujian sensor, pengujian komunikasi berdasarkan parameter *Quality of Service* (QoS), pengujian prediksi menggunakan algoritma XGBoost, serta penerapan keseluruhan sistem. Pada pengujian sensor, sensor yang diuji dibandingkan dengan alat ukur referensi pada pengujian komunikasi, parameter yang digunakan meliputi *Round-Trip Time* (RTT), *jitter*, *packet loss*, dan *throughput*. Sementara itu, pengujian prediksi dilakukan dengan membandingkan hasil prediksi algoritma terhadap data *test* dan kondisi actual untuk memperoleh parameter evaluasi kinerja model.

4.2 Node 1 (Rektorat)

4.2.1 Sensor-Sensor

Dikarenakan salah satu alat ukur referensi yang tersedia tidak memiliki informasi yang cukup tentang akurasi, sedangkan alat ukur lainnya memiliki akurasi lebih rendah daripada sensor yang diuji, maka hasil pengujian tidak dapat digunakan sebagai dasar kalibrasi sensor. Sebagai gantinya, hasil pengujian akan digunakan sebagai dasar perbandingan (*comparison*), yaitu dengan membandingkan hasil pembacaan sensor terhadap instrumen referensi pada kondisi pengukuran yang sama.

a. Sensor DHT22 dan BMP280

Untuk temperatur dan tekanan udara, pengujian sensor dilakukan dengan bantuan alat ukur Altimeter Digital 8 in 1.



Gambar 4. 1 Alat Ukur Referensi untuk Temperatur dan Tekanan Udara

Berikut adalah spesifikasi lebih lengkap dari alat tersebut:

Tabel 4. 1 Spesifikasi Teknis Alat Ukur Altimeter Digital 8 in 1 [91]

Parameter	Spesifikasi
Fitur umum	<i>Altimeter, Barometer, Digital Compass, Thermometer, Weather Forecast, Clock, Calendar, LED Backlight</i>
Rentang pengukuran altimeter	-700 hingga 9.000 meter (-2.300 hingga 29.500 kaki)
Resolusi altimeter	± 1 meter ($\approx 3,3$ kaki)
Rentang pengukuran barometer	300–1100 mbar (hPa)
Resolusi barometer	0,1 mbar
Akurasi Barometer	-
Rentang pengukuran temperatur udara	-10 hingga 60°C
Akurasi termometer	-
Akurasi kompas	$\pm 3^\circ$

Parameter	Spesifikasi
Resolusi kompas	1°

Namun, *datasheet* alat ukur tersebut tidak mencantumkan informasi tentang akurasi pengukuran temperatur udara dan tekanan udara. Selain itu, alat ukur tersebut juga tidak mempunyai fitur pengukuran kelembapan udara. Oleh karena itu, digunakanlah alat kedua yang memiliki kemampuan pengukuran kelembapan udara, yaitu Thermohygrometer HTC-2.



Gambar 4. 2 Alat Ukur Referensi untuk Kelembapan Udara

Berikut adalah spesifikasi lebih lengkap tentang Thermohygrometer HTC-2 tersebut:

Tabel 4. 2 Spesifikasi Alat Ukur Thermohygrometer HTC-2 [92]

Parameter	Spesifikasi
<i>Temperature range</i>	<i>Indoor and Outdoor -50 ~ +70 degree (-58 ~ +158°F)</i>
<i>Humidity range</i>	10% ~ 99% RH
<i>Temperature accuracy</i>	± 2 °C
<i>Humidity accuracy</i>	±10% RH

Dapat dilihat pada tabel, akurasi pengukuran kelembapan udara dari Thermohygrometer HTC-2 lebih rendah daripada DHT22, sehingga menjadikannya dasar kalibrasi akan tidak dapat membuat pembacaan

sensor menjadi lebih baik.

Pengujian dilakukan di ruangan tertutup, karena ruangan yang demikian dapat memastikan stabilitas termal dan keseragaman kondisi lingkungan.



Gambar 4. 3 Rangkaian untuk Pengujian Sensor

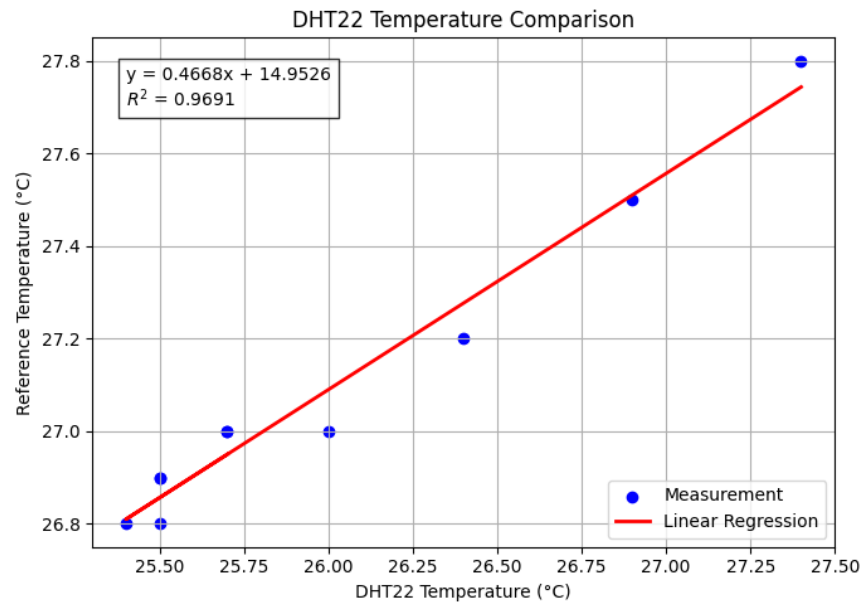


Gambar 4. 4 Proses Pengujian Sensor pada Ruangan Tertutup

Adapun hasil pengujian sensor dapat dilihat pada tabel berikut:

Tabel 4. 2 Data Temperatur Udara Referensi dan Temperatur Udara DHT22

<i>Timestamp</i>	Temperatur Referensi (°C)	Temperatur DHT22 (°C)	Δtemperatur (°C)
17:22	27	25,7	1,3
17:24	27	25,7	1,3
17:26	26,9	25,5	1,4
17:28	26,9	25,5	1,4
17:30	26,8	25,4	1,4
17:32	26,8	25,5	1,3
17:34	27	26	1
17:36	27,2	26,4	0,8
17:38	27,5	26,9	0,6
17:40	27,8	27,4	0,4



Gambar 4. 4 Grafik Regresi Linear Temperatur Udara Referensi dan Temperatur Udara DHT22

Perbandingan yang dilakukan menggunakan metode *direct multi-point* dengan regresi linear:

$$y = ax + b \quad (4.1)$$

dengan:

$$a = \frac{n(\sum xy) - (\sum x)(\sum y)}{n(\sum x^2) - (\sum x)^2} \quad (4.2)$$

$$b = \frac{\sum y - m(\sum x)}{n} \quad (4.3)$$

Keterangan:

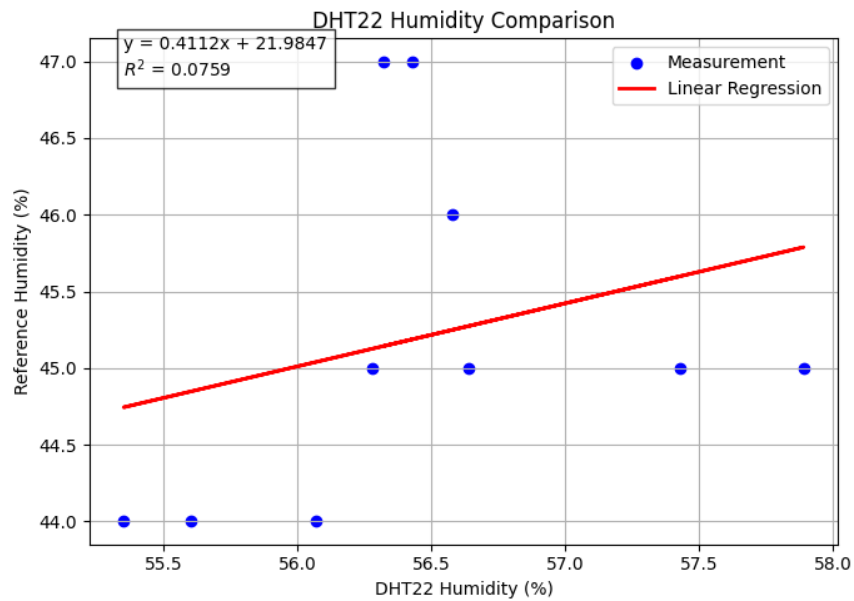
- a : kemiringan atau *slope*
- b : *intercept*
- n : jumlah data
- y : temperatur udara referensi
- x : temperatur udara DHT22

Dari perhitungan, didapatkan nilai $a = 0.466825$ dan $b = 14.952559$, dan grafiknya dapat dilihat pada Gambar 4.4.

Sedangkan untuk kelembapan udara:

Tabel 4. 3 Data Kelembapan Udara Referensi dan Kelembapan Udara DHT22

<i>Timestamp</i>	Kelembapan Referensi (%)	Kelembapan DHT22 (%)	Δkelembapan (%)
17:22	45	57,43	-12,43
17:24	45	56,64	-11,64
17:26	44	56,07	-12,07
17:28	44	55,6	-11,6
17:30	44	55,35	-11,35
17:32	45	56,28	-11,28
17:34	45	57,89	-12,89
17:36	46	56,58	-10,58
17:38	47	56,32	-9,32
17:40	47	56,43	-9,43



Gambar 4.5 Grafik Regresi Linear Kelembapan Udara Referensi dan Kelembapan Udara DHT22

Jika dianalisis secara visual melalui grafik pada Gambar 4.5, sebaran data tidak menunjukkan hubungan linear yang kuat antara kelembapan udara referensi dan kelembapan udara DHT22, seperti layaknya grafik pada Gambar 4.4. Pada metode regresi linear, terdapat suatu parameter yang menunjukkan tingkat linearitas pada data, yaitu R^2 atau koefisien determinasi.

$$R^2 = 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2} \quad (4.4)$$

Keterangan:

- n : jumlah data
- y_i : nilai kelembapan udara referensi
- $\bar{y}$: rata-rata nilai kelembapan udara referensi
- $\hat{y}_i$: nilai hasil regresi linear

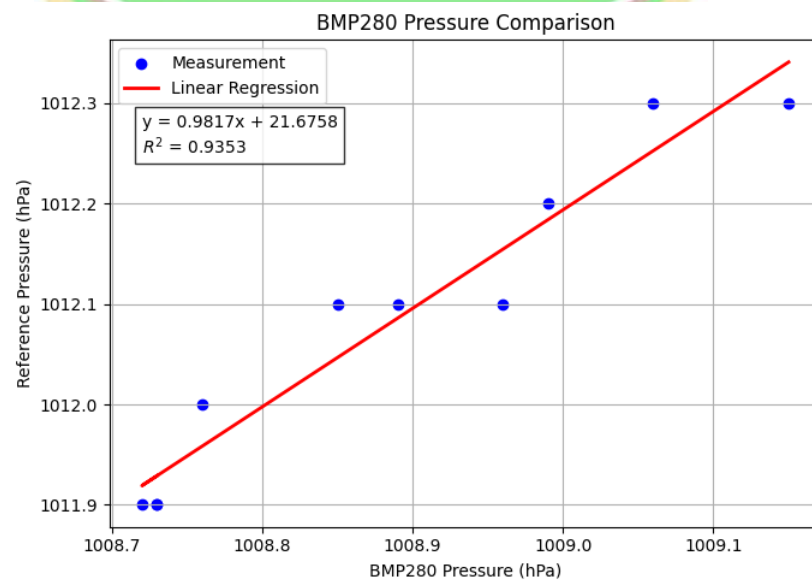
Setelah kalkulasi, hasilnya adalah nilai $R^2 = 0.075928$. Nilai ini sangat kecil jika dibandingkan dengan nilai R^2 untuk temperatur udara, yaitu 0.969067. Namun, berdasarkan analisis selisih pengukuran, sensor DHT22 menunjukkan bias positif yang relatif konstan sekitar 11% RH (lihat Tabel 4.3). Ketidaklinearan pada pengujian kelembapan udara ini dapat terjadi

karena beberapa hal. Pertama, adalah akurasi alat ukur referensi yang lebih rendah daripada sensor DHT22 (Thermohygrometer HTC-2: $\pm 10\%$ RH, DHT22: $\pm 2\%$ RH), kedua, waktu tunggu atau *delay* yang tidak cukup dalam pengambilan data (2 menit), sehingga perubahan pada salah satu sensor belum terjadi saat data diambil, dan ketiga, kondisi lingkungan tempat pengujian yang masih tidak homogen atau stabil.

Untuk tekanan udara, nilai antara tekanan udara referensi dan tekanan udara dari sensor kembali menunjukkan tren linear:

Tabel 4. 4 Data Tekanan Udara Referensi dan Tekanan Udara BMP280

<i>Timestamp</i>	Tekanan Udara Referensi (hPa)	Tekanan Udara BMP280 (hPa)	Δtekanan Udara (hPa)
17:22	1011.9	1008.73	3.17
17:24	1011.9	1008.73	3.17
17:26	1011.9	1008.72	3.18
17:28	1012.0	1008.76	3.24
17:30	1012.1	1008.85	3.25
17:32	1012.1	1008.89	3.21
17:34	1012.1	1008.96	3.14
17:36	1012.2	1008.99	3.21
17:38	1012.3	1009.06	3.24
17:40	1012.3	1009.15	3.15

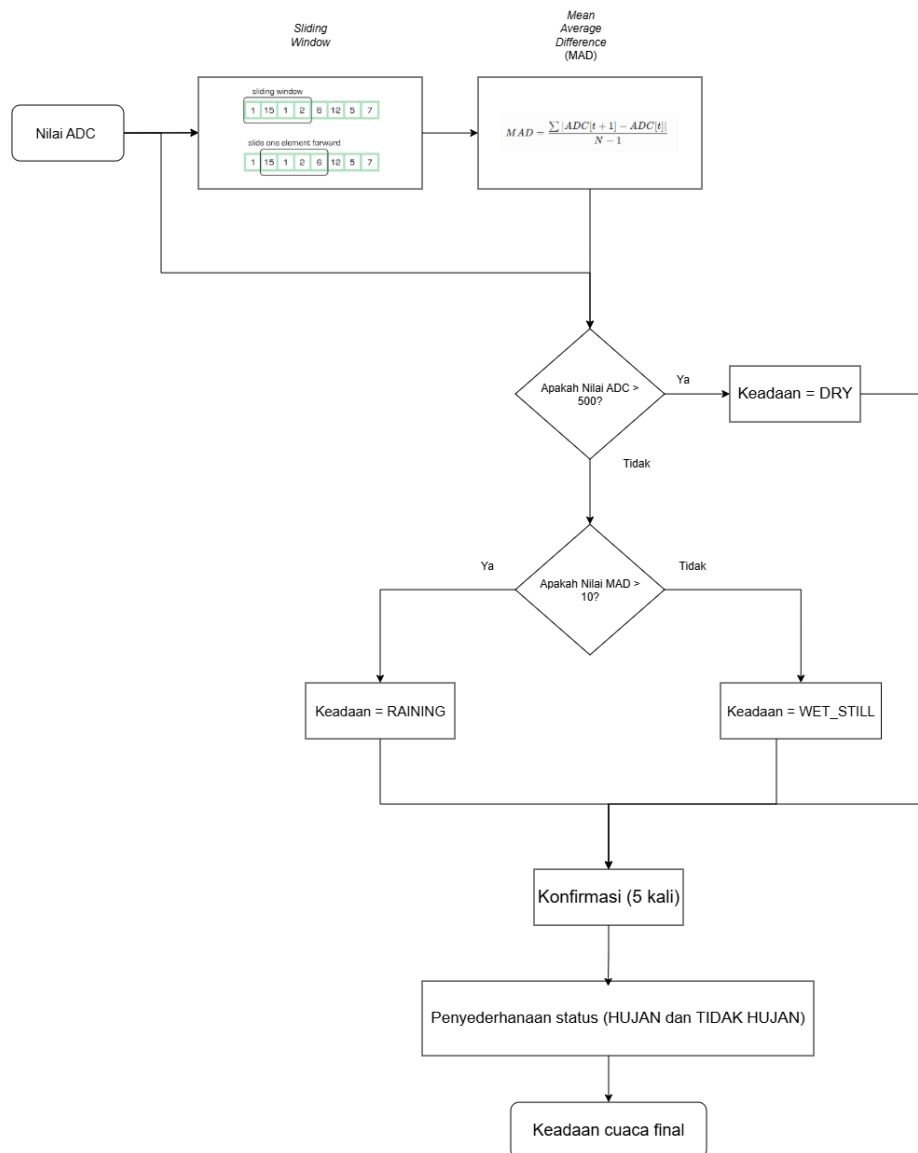


Gambar 4. 6 Grafik Regresi Linear Tekanan Udara Referensi dan Tekanan Udara BMP280

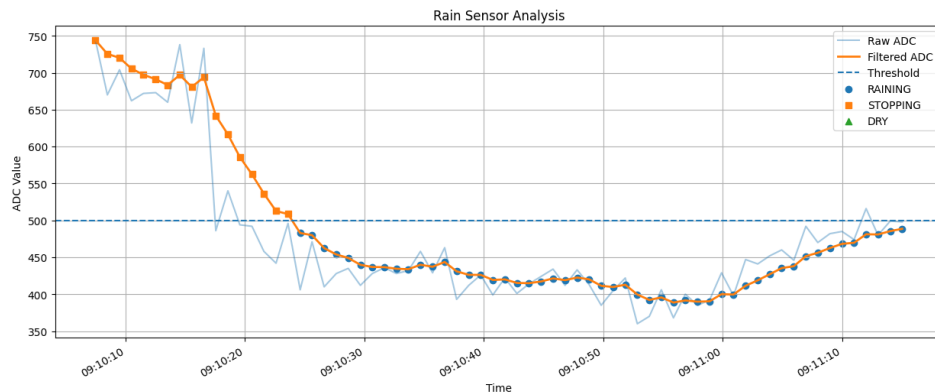
Pada pengujian pengukuran tekanan udara, jika dikalkulasi menggunakan metode regresi linear, maka nilai $a = 0.981683$ dan $b = 21.675844$.

b. Sensor Hujan (*Rain Sensor*)

Kemudian, untuk sensor hujan, pengujian awalnya ialah dengan menggunakan siraman air *sprayer*, untuk menemukan titik *threshold* nilai ADC yang pas. Siraman air *sprayer* dalam hal ini mensimulasikan air hujan. Adapun prinsip kerja lebih lengkap tentang sensor hujan ialah seperti yang dijelaskan oleh Gambar 4.7.



Gambar 4. 7 Algoritma Penentuan Keadaan Cuaca oleh Sensor Hujan



Gambar 4.8 Grafik Uji *Threshold* Nilai ADC Sensor Hujan

Nilai ADC mempunyai rentang dari 0 hingga 1023 (dari ADC 10 bit). Berdasarkan percobaan, saat sensor disiram oleh air *sprayer*, nilai ADC cenderung berada di bawah 500. Oleh karena itu, 500 digunakan sebagai *threshold* dasar penentuan keadaan hujan atau tidak hujan.

Namun penentuan keadaan tersebut masih mempunyai kelemahan, yaitu keterlambatan atau *lag* nilai ADC sensor untuk kembali atau berubah ke nilai angka tertentu saat keadaan berubah. *Noise* atau ketidakstabilan, misalnya saat nilai ADC berubah-ubah di sekitar batas *threshold* dengan cepat, juga turut andil menjadi masalah dalam operasional sensor ini.

Oleh karena itu, dibutuhkan suatu algoritma pemrosesan data sehingga keadaan hujan dan tidak hujan dapat dibedakan. Algoritma tersebut, seperti yang dijelaskan dalam Gambar 4.7, diawali dengan pembacaan nilai analog (ADC) dari sensor hujan setiap 1 detik. Nilai pembacaan tersebut kemudian disimpan ke dalam *sliding window* berukuran 10 data untuk menghitung nilai *Mean Absolute Difference* (MAD), yaitu rata-rata selisih absolut antar pembacaan berturut-turut. Selanjutnya, kondisi sensor diklasifikasikan menjadi tiga kandidat keadaan berdasarkan nilai ADC dan MAD. Apabila nilai ADC lebih besar dari ambang batas (500), maka kondisi dikategorikan sebagai DRY. Sebaliknya, jika nilai ADC berada di bawah ambang batas, maka nilai MAD digunakan untuk membedakan kondisi RAINING dan WET_STILL. Nilai MAD yang melebihi ambang batas 10 menunjukkan adanya perubahan sinyal yang signifikan akibat tetesan hujan sehingga diklasifikasikan sebagai RAINING, sedangkan nilai MAD di bawah ambang batas menunjukkan permukaan sensor masih basah tetapi tidak lagi menerima tetesan hujan, sehingga diklasifikasikan sebagai WET_STILL. Untuk meningkatkan kestabilan hasil klasifikasi,

setiap perubahan status harus terdeteksi secara berturut-turut sebanyak lima kali (*confirmation count*) sebelum status akhir diperbarui. Pendekatan ini bertujuan untuk mengurangi kesalahan klasifikasi akibat fluktuasi sesaat atau *noise* pada pembacaan sensor. Pada saat data status hujan atau tidak hujan tersebut dikirimkan ke pusat pemrosesan, klasifikasinya disederhanakan menjadi HUJAN dan TIDAK HUJAN (DRY dan WET_STILL menjadi TIDAK HUJAN, RAINING menjadi HUJAN).

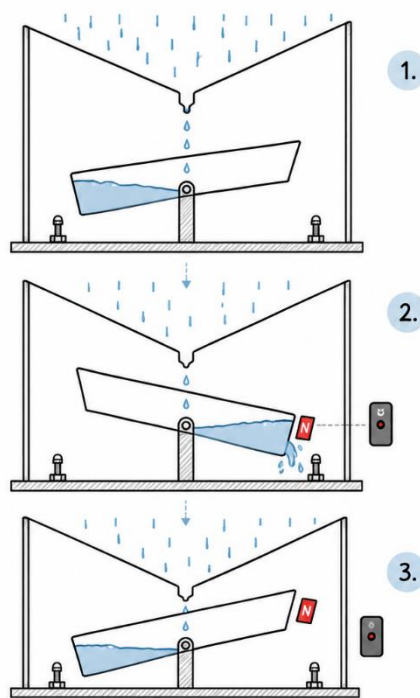
$$MAD = \frac{\sum(ADC_{[t+1]} - ADC_{[t]})}{N-1} \quad (4.5)$$

Dengan N sebagai ukuran window, dalam kasus ini N = 10



Gambar 4. 9 Sensor Hujan pada Node 1

Sedangkan untuk alat ukur *tipping bucket*, pergerakan ember dideteksi menggunakan *Hall effect sensor* KY-003 dengan memanfaatkan magnet yang dipasang pada ember. Sensor hanya dipasang pada satu sisi mekanisme sehingga satu pulsa yang dihasilkan sensor merepresentasikan satu siklus penuh ember, yaitu dua kali gerakan jungkit mekanis.



Gambar 4. 10 Cara Kerja *Tipping Bucket*

Untuk memperoleh konstanta kalibrasi, dilakukan pengukuran massa air yang diperlukan untuk menghasilkan satu pulsa sensor sebanyak sepuluh kali percobaan.

Tabel 4. 5 Data Kalibrasi *Tipping Bucket* Node 1

Percobaan ke-	Jungkit ke-	Banyak air tertampung (gram atau mL)
1	2	8
2	4	9
3	6	7
4	8	6
5	10	6
6	12	6
7	14	6
8	16	6
9	18	6
10	20	6
Rata-rata		6.6

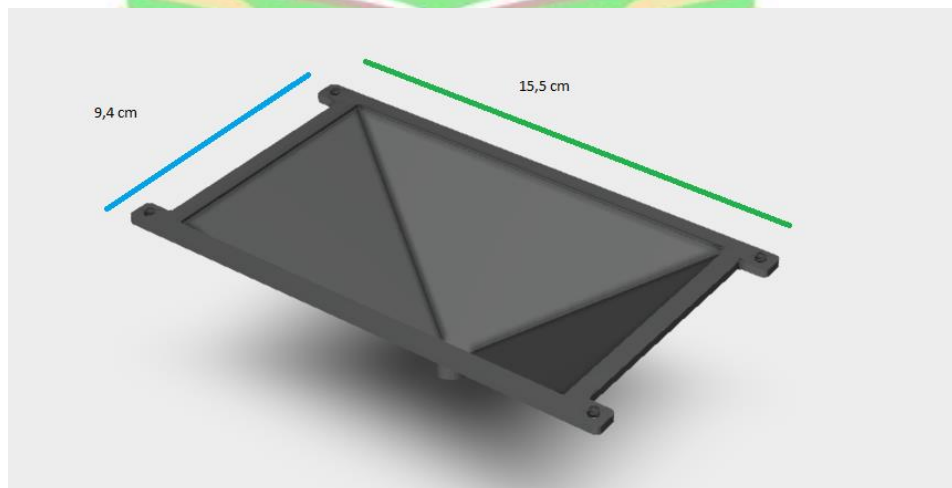


Gambar 4. 11 Proses Kalibrasi *Tipping Bucket*

Setelah percobaan dilakukan, didapati bahwa rata-rata tiap 1 siklus jungkit atau 2 kali jungkit menampung air sebanyak 6,6 mL. Nilai tersebut kemudian dikonversi menjadi tinggi curah hujan berdasarkan luas bukaan corong sebesar 145,7 cm² menggunakan Persamaan 4.6.

$$h = \frac{V}{A} \quad (4.6)$$

Dengan h sebagai tinggi, V sebagai volume rata-rata, dan A sebagai area penampang pada *tipping bucket*.



Gambar 4. 12 Ukuran Penampang pada *Tipping Bucket*

Perhitungan lengkapnya ialah sebagai berikut:

$$V = 6,6 \text{ mL} \quad (4.7)$$

$$A = 15,5 \times 9,4 \text{ cm} = 145,7 \text{ cm}^2 \quad (4.8)$$

$$h = \frac{6,6}{145,7} = 0,0453 \text{ cm} = 0,453 \text{ mm} \quad (4.9)$$

Dengan demikian, setiap satu pulsa *Hall effect sensor* merepresentasikan curah hujan sebesar 0,453 mm, yang selanjutnya digunakan sebagai faktor konversi dalam perhitungan intensitas hujan pada sistem.



Gambar 4. 13 Tipping Bucket pada Node 1

4.2.2 Quality of Service (QoS)

Seperti yang dijelaskan pada Bab 3, bahwa parameter-parameter QoS yang akan diuji ialah *delay (latency)*, *jitter*, *packet loss*, dan *throughput*. Adapun dalam penelitian ini, pengukuran parameter QoS dibagi menjadi dua bagian, mengikuti arsitektur sistem yang dirancang, yaitu pengukuran antara node dan pusat pemrosesan, serta pusat pemrosesan dan server Firebase.

a. Tentang *delay*

Hal pertama yang perlu diperhatikan dalam pengukuran waktu komunikasi adalah pewaktuan atau *timing*. Pada pengukuran *one-way delay* (salah satu jenis pengukuran *delay*) antara dua perangkat yang saling terhubung dan bertukar data, waktu pada kedua perangkat perlu disinkronkan agar selisih *timestamp* pengiriman dan penerimaan dapat merepresentasikan waktu tempuh data secara akurat. Sebagai ilustrasi, apabila perangkat A mengirim data pada pukul 12:00:00.000 dan data diterima perangkat B 500 ms kemudian, maka waktu penerimaan sebenarnya adalah 12:00:00.500. Namun, apabila clock perangkat B memiliki perbedaan waktu sebesar 200 ms terhadap perangkat A, *timestamp* yang dicatat oleh perangkat B dapat

menghasilkan waktu penerimaan yang berbeda sehingga berpotensi menyebabkan kesalahan dalam perhitungan *one-way delay*.

Permasalahan tersebut dapat diatasi dengan menggunakan pendekatan pengukuran *Round-Trip Time* (RTT), yang mengukur waktu komunikasi data secara pulang-pergi, yaitu ketika data dikirim dari perangkat pengirim menuju perangkat penerima dan respons dikirim kembali menuju perangkat pengirim. Dengan pendekatan ini, waktu awal ketika data dikirim dan waktu akhir ketika respons diterima kembali dapat dicatat menggunakan clock pada perangkat yang sama, sehingga pengukuran tidak bergantung pada kesamaan clock antara kedua perangkat. Hal ini membuat penulis memilih pendekatan pengukuran RTT tersebut.

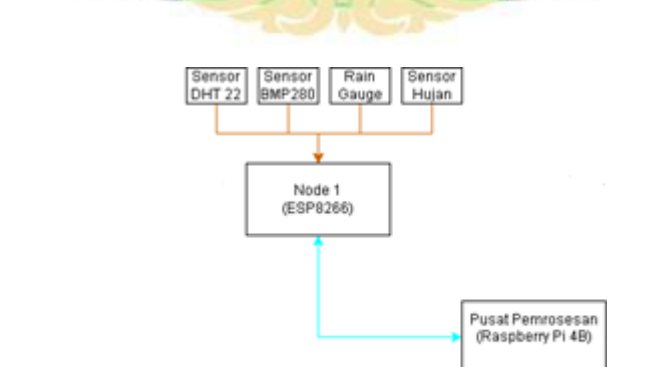
b. Tentang *Packet Loss*

Packet loss diperoleh dengan membandingkan jumlah paket yang dikirim dengan jumlah paket yang berhasil diterima atau memperoleh respons dari perangkat penerima selama pengujian. Setiap paket yang telah dikirim tetapi tidak berhasil diterima atau tidak memperoleh respons dalam waktu yang telah ditentukan dianggap sebagai paket yang hilang. Selanjutnya, persentase *packet loss* dihitung dengan membandingkan jumlah paket yang hilang terhadap total paket yang dikirim menggunakan persamaan:

$$Packet\ Loss = \frac{Jumlah\ Paket\ Hilang}{Jumlah\ Paket\ Dikirim} \times 100\% \quad (4.10)$$

Kondisi ini juga dapat memengaruhi pengukuran parameter QoS lainnya, karena paket yang hilang tidak menghasilkan waktu komunikasi yang dapat digunakan dalam perhitungan RTT dan *jitter*, begitupun dengan perhitungan *throughput*, yang tidak mengikutkan paket yang hilang dalam perhitungannya.

1. Node dan Pusat Pemrosesan



Gambar 4. 14 Diagram Hubungan Node 1 dan Pusat Pemrosesan

- Pusat Pemrosesan (Raspberry Pi 4 Model B) mengirim perintah START ke *node* untuk memulai pengukuran.
- *Node* mencatat waktu awal (*start time*) tepat sebelum paket data dikirim.
- *Node* mengirim paket data melalui MQTT menuju pusat pemrosesan.
- Pusat pemrosesan menerima paket data dan segera mengirimkan pesan ACK sebagai respons atau umpan balik (*feedback*).
- *Node* menerima pesan ACK dari pusat pemrosesan.
- *Node* mencatat waktu akhir (*end time*) ketika ACK diterima.
- *Node* menghitung RTT dari selisih waktu akhir dan waktu awal:

$$RTT = Waktu\ Awal - Waktu\ Akhir \quad (4.11)$$

- *Node* mencatat nilai RTT setiap siklus secara berurutan, nilai tersebut kemudian digunakan untuk menghitung jitter.

$$Jitter = \frac{\sum_{i=2}^n |RTT_i - RTT_{i-1}|}{n-1} \quad (4.12)$$

- Apabila paket yang dikirim tidak memperoleh respons dalam batas waktu yang telah ditentukan, paket tersebut dicatat sebagai paket hilang (*lost packet*). Persentase *packet loss* kemudian dihitung dengan Persamaan 4.10.
- Jumlah data yang berhasil diterima dan waktu pengujian dicatat untuk menghitung *throughput*.

$$Throughput = \frac{Jumlah\ Data\ Diterima}{Waktu\ Tes} \quad (4.13)$$

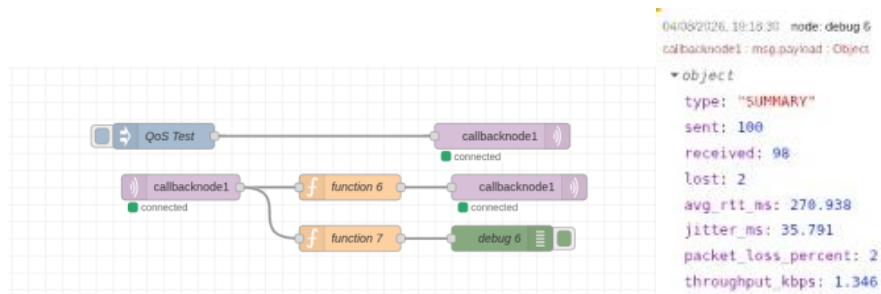
- Proses pengiriman, penerimaan respons, pencatatan RTT, dan pencatatan paket hilang diulangi hingga mencapai 100 siklus pengiriman.

Berikut adalah data hasil pengukuran parameter QoS pada Node 1:

Tabel 4. 6 Rangkuman Hasil Pengukuran Parameter QoS Komunikasi Node 1 dan Pusat Pemrosesan

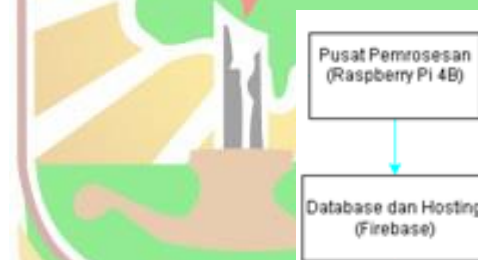
Parameter (rata-rata)	Nilai
RTT (ms)	270,938
Jitter (ms)	35,791
Packet Loss (%)	2
Throughput (kbps)	1,346

Hasil pengujian menunjukkan bahwa komunikasi antara *node* dan pusat pemrosesan memiliki rata-rata RTT sebesar 270,938 ms dengan *jitter* sebesar 35,791 ms, yang menunjukkan adanya variasi waktu komunikasi antar-paket. Nilai *packet loss* sebesar 2% menunjukkan bahwa sebagian besar paket berhasil diterima, sedangkan *throughput* sebesar 1,346 kbps menunjukkan jumlah data yang berhasil ditransmisikan selama pengujian.



Gambar 4. 15 Mekanisme Perhitungan Parameter QoS Node 1 pada Pusat Pemrosesan dan Hasilnya

2. Pusat Pemrosesan dan Server Firebase



Gambar 4. 16 Diagram Hubungan Pusat Pemrosesan dan Server Firebase

Konsep atau metode yang digunakan dalam pengukuran parameter QoS antara pusat pemrosesan dan server Firebase sama dengan yang digunakan antara *node* dan pusat pemrosesan. Pada pengukuran ini, pusat pemrosesan bertugas sebagai perangkat yang menghitung waktu, seperti yang dilakukan oleh *node* pada pengukuran parameter QoS *node* dan pusat pemrosesan. Data hasil pengukuran tersebut dapat dilihat pada Tabel 4.7.

Tabel 4. 7 Rangkuman Hasil Pengukuran Parameter QoS Komunikasi Pusat Pemrosesan dan Server Firebase

Parameter (rata-rata)	Nilai
RTT (ms)	57,938
Jitter (ms)	69,977
Packet Loss (%)	0
Throughput (kbps)	16,390

Jika dilihat dari data yang dihasilkan, berdasarkan perbandingan dengan rekomendasi ITU-T Y.1541 (salah satu referensi standar kualitas jaringan untuk jaringan berbasis IP) sistem menunjukkan kualitas komunikasi yang baik, terutama pada komunikasi antara pusat pemrosesan dan server Firebase. Pada komunikasi Node 1 menuju pusat pemrosesan, diperoleh rata-rata RTT sebesar 270,938 ms, jitter 35,791 ms, packet loss 2%, dan throughput 1,346 kbps. Sementara itu, pada komunikasi pusat pemrosesan menuju server Firebase, diperoleh rata-rata RTT sebesar 57,938 ms, jitter 69,977 ms, packet loss 0%, dan throughput 16,390 kbps. Standar ITU-T Y.1541 (*class 4*) mempunyai kriteria *delay* ≤ 1 detik dan *packet loss* $\leq 0,1\%$ [93-94]. Berikut adalah tabel yang menjelaskan capaian hasil pengukuran parameter QoS pada sistem terhadap standar ITU-T Y1541 (*class 4*):

Tabel 4. 8 Perbandingan Standar ITU-T Y.1541 (*class 4*) dengan Nilai Parameter QoS Komunikasi Node 1 dan Pusat Pemrosesan

Parameter (rata-rata)	Sistem	Standar ITU-T Y.1541 (<i>class 4</i>)	Status
RTT (ms)	270,938	≤ 2000	Memenuhi
Jitter (ms)	35,791	Tidak didefinisikan	-
Packet Loss (%)	2	$\leq 0,1$	Tidak memenuhi
Throughput (kbps)	1,346	Tidak didefinisikan	-

Tabel 4. 9 Perbandingan Standar ITU-T Y.1541 (*class 4*) dengan Nilai Parameter QoS Komunikasi Pusat Pemrosesan dan Server Firebase

Parameter (rata-rata)	Sistem	Standar ITU-T Y.1541 (<i>class 4</i>)	Status
RTT (ms)	57,938	≤ 2000	Memenuhi
Jitter (ms)	69,977	Tidak didefinisikan	-
Packet Loss (%)	0	$\leq 0,1$	Memenuhi
Throughput (kbps)	16,390	Tidak didefinisikan	-

Hasil pengujian menunjukkan bahwa komunikasi Node 1 dan pusat pemrosesan memiliki RTT lebih tinggi, yaitu sebesar 270,938 ms, *packet loss* 2%, dan *throughput* lebih rendah sebesar 1,346 kbps yang dapat dipengaruhi oleh kondisi koneksi Wi-Fi dan kualitas sinyal. Sementara itu, komunikasi pusat pemrosesan dan server Firebase memiliki RTT lebih rendah sebesar 57,938 ms dan *packet loss* 0% yang menunjukkan komunikasi lebih stabil, meskipun *jitter* sebesar 69,977 ms menunjukkan adanya fluktuasi waktu pengiriman paket selama komunikasi.

4.3 Node 2 (*Workshop AFRG*)

4.3.1 Sensor-Sensor

Sensor yang digunakan untuk mendeteksi turunnya hujan, mengukur temperatur, kelembapan, serta tekanan udara pada Node 2 adalah sama dengan yang digunakan pada Node 1. Hal ini membuat sensor-sensor tersebut memiliki spesifikasi yang sama, seperti akurasi atau resolusi, sehingga proses perbandingan (komparasi) terhadap alat ukur referensi (altimeter 8 in 1 dan hygrometer HTC-2) dan penentuan parameter seperti nilai *threshold* (pada sensor hujan, cukup dilakukan pada Node 1).

Adapun *tipping bucket*, yang merupakan alat ukur yang cara kerjanya terdiri dari gabungan antara pendeteksian sensor (*Hall effect sensor*), dan mekanisme mekanik berupa ember jungkit (*tipping bucket mechanism*), tetap memerlukan proses kalibrasi pada setiap unit. Hal ini dikarenakan karakteristik mekanik, seperti volume tampung ember, titik keseimbangan, posisi poros, serta toleransi proses manufaktur dan perakitan, dapat berbeda antar unit sehingga memengaruhi besarnya curah hujan yang direpresentasikan oleh setiap satu kali jungkitan. Tabel 4.10 berikut memuat hasil kalibrasi *tipping bucket* pada Node 2.

Tabel 4. 10 Data Kalibrasi *Tipping Bucket* Node 2

Percobaan ke-	Jungkit ke-	Banyak air tertampung (gram atau mL)
1	2	9
2	4	10
3	6	8
4	8	8
5	10	10
6	12	8
7	14	10
8	16	8
9	18	10
10	20	9
Rata-rata		9.0

Jika dikalkulasi lebih lanjut:

$$V = 9,0 \text{ mL} \quad (4.10)$$

$$A = 15,5 \times 9,4 \text{ cm} = 145,7 \text{ cm}^2 \quad (4.11)$$

$$h = \frac{9,0}{145,7} = 0.06177 \text{ cm} = 0.6177 \text{ mm} \quad (4.12)$$

Maka tiap satu pulsa *Hall effect sensor* (2 kali jungkit) merepresentasikan curah hujan sebesar 0,618 mm, yang selanjutnya digunakan sebagai faktor konversi dalam perhitungan intensitas hujan pada sistem.

4.3.2 *Quality of Service (QoS)*

Metode pengukuran parameter-parameter QoS pada Node 2 ialah sama dengan yang digunakan pada Node 1. Adapun hasil pengukuran parameter tersebut dapat dilihat pada bagian berikut:

1. Node 2 dan Pusat Pemrosesan

Tabel 4. 11 Rangkuman Hasil Pengukuran Parameter QoS Komunikasi Node 2 dan Pusat Pemrosesan

Parameter (rata-rata)	Nilai
RTT (ms)	465.392
Jitter (ms)	47.953

Parameter (rata-rata)	Nilai
<i>Packet Loss (%)</i>	0
<i>Throughput</i> (kbps)	1.033

2. Pusat Pemrosesan dan Server Firebase

Node 2 dan Node 1 memiliki pusat pemrosesan yang sama, dengan kata lain, pusat pemrosesan dan server Firebase pada sistem memiliki mekanisme kerja yang sama untuk setiap *node*, sehingga hasil penerapan pada Node 1 (Tabel 4.7) dapat digunakan sebagai perwakilan.

Tabel 4. 12 Perbandingan Standar ITU-T Y.1541 (*class 4*) dengan Nilai Parameter QoS Komunikasi Node 2 dan Pusat Pemrosesan

Parameter (rata-rata)	Sistem	Standar ITU-T Y.1541 (<i>class 4</i>)	Status
RTT (ms)	465.392	≤ 2000	Memenuhi
Jitter (ms)	47.953	Tidak didefinisikan	-
<i>Packet Loss (%)</i>	0	$\leq 0,1$	Memenuhi
<i>Throughput</i> (kbps)	1.033	Tidak didefinisikan	-

Jika dibandingkan antara komunikasi Node 1 dan pusat pemrosesan dan Node 2 dan pusat pemrosesan, Node 1 memiliki waktu komunikasi dan variasi *delay* yang lebih rendah serta *throughput* sedikit lebih tinggi, tetapi mengalami *packet loss* sebesar 2%, sementara Node 2 tidak mengalami *packet loss* meskipun RTT dan *jitter*-nya lebih tinggi serta *throughput* lebih rendah.

4.4 Evaluasi Model XGBoost (Prediksi)

Dataset yang digunakan dalam penelitian ini adalah *dataset* “Indonesia Hourly Weather for Rain Prediction” yang tersedia pada platform Kaggle, yang digunakan sebagai dasar dalam proses prediksi hujan untuk enam jam ke depan [95]. *Dataset* dibagi menjadi dua bagian, yaitu data latih (*training*) dan data uji (*testing*). Data latih terdiri atas 147,240 (82,36%) baris data, sedangkan data uji terdiri atas 31,536 (17,64%) baris data.

Tabel 4. 13 Lima Baris Pertama *Dataset*

temperature_2m	relative_humidity_2m	surface_pressure	timestamp	rain_next_6h
19.8	82	932.4	2022-01-01 05:00:00	1
18.9	89	932.8	2022-01-01 06:00:00	0
21.6	78	933.8	2022-01-01 07:00:00	0
22.7	78	934.6	2022-01-01 08:00:00	0
23.7	74	935.1	2022-01-01 09:00:00	0

Pada tahap awal, *dataset* terdiri atas lima variabel, yaitu *temperature_2m*, *relative_humidity_2m*, *surface_pressure*, *timestamp*, dan *rain_next_6h*. Tiga variabel pertama merupakan parameter meteorologi yang menggambarkan kondisi atmosfer pada waktu pengamatan, sedangkan *timestamp* menunjukkan waktu terjadinya pengamatan. Variabel *rain_next_6h* digunakan sebagai variabel target dalam proses klasifikasi.

Setelah fitur utama ditentukan, dilakukan *feature engineering* untuk memperoleh informasi tambahan dari data meteorologi. Fitur-fitur tambahan yang dibentuk adalah:

1. Fitur waktu (*Timestamp*)

Mengekstraksi jam dan bulan dari *timestamp*, kemudian menggunakan representasi siklik (sin dan cos) agar pola waktu yang berulang dapat dikenali oleh model.

$$hour_{sin} = \sin\left(\frac{2\pi h}{24}\right) \quad (4.13)$$

$$hour_{cos} = \cos\left(\frac{2\pi h}{24}\right) \quad (4.14)$$

$$month_{sin} = \sin\left(\frac{2\pi m}{12}\right) \quad (4.15)$$

$$month_{cos} = \cos\left(\frac{2\pi m}{12}\right) \quad (4.16)$$

2. Fitur tren

Menghitung perubahan temperatur, kelembapan, dan tekanan udara terhadap kondisi 1, 3, dan 6 jam sebelumnya untuk menangkap kecenderungan perubahan kondisi atmosfer.

$$\Delta X_t^{(k)} = X_t - X_{t-k} \quad (4.17)$$

Dengan:

- X_t = nilai variabel meteorologi saat ini
- X_{t-k} = nilai jam sebelumnya
- $k = 1, 3, \text{ atau } 6 \text{ jam}$

3. Rolling Mean

Menghitung rata-rata temperatur, kelembapan, dan tekanan udara selama 1, 3, dan 6 jam untuk menggambarkan kondisi atmosfer dalam rentang waktu tertentu dan mengurangi pengaruh fluktuasi sesaat.

$$\bar{X}_t^{(k)} = \frac{1}{k} \sum_{i=0}^{k-1} X_{t-i} \quad (4.18)$$

4. Rolling Standard Deviation

Mengukur tingkat variabilitas temperatur, kelembapan, dan tekanan udara selama 3 dan 6 jam untuk mengetahui kestabilan atau perubahan kondisi atmosfer.

$$\sigma_X^{(k)} = \sqrt{\frac{1}{k} \sum_{i=0}^{k-1} \left(X_{t-i} - \bar{X}_t^{(k)} \right)^2} \quad (4.19)$$

Dengan:

- $\sigma_X^{(k)}$ = standar deviasi parameter X pada jendela k

5. Rentang nilai (Range)

Menghitung selisih antara nilai maksimum dan minimum temperatur serta kelembapan dalam periode 3 dan 6 jam, sehingga perubahan kondisi atmosfer dapat direpresentasikan.

$$R_X^{(k)} = \max(X_{t-k+1}, \dots, X_t) - \min(X_{t-k+1}, \dots, X_t) \quad (4.20)$$

Dengan:

- $R_X^{(k)}$ = rentang nilai parameter X

- $\max()$ = nilai maksimum dalam jendela
- $\min()$ = nilai minimum dalam jendela

6. Fitur interaksi

Membentuk kombinasi antara temperatur, kelembapan, dan tekanan untuk menangkap hubungan antarvariabel meteorologi yang mungkin tidak terlihat jika setiap variabel digunakan secara terpisah.

$$X_{T,H} = T_t H_t \quad (4.21)$$

$$X_{H/T} = \frac{H_t}{T_t + \epsilon} \quad (4.22)$$

$$X_{H,\Delta P} = H_t \Delta P_t^{(6)} \quad (4.23)$$

$$X_{T,\Delta P} = T_t \Delta P_t^{(6)} \quad (4.24)$$

Dengan:

- T_t = temperatur pada waktu t
- H_t = kelembapan pada waktu t
- $\Delta P_t^{(6)}$ = perubahan tekanan udara selama enam jam

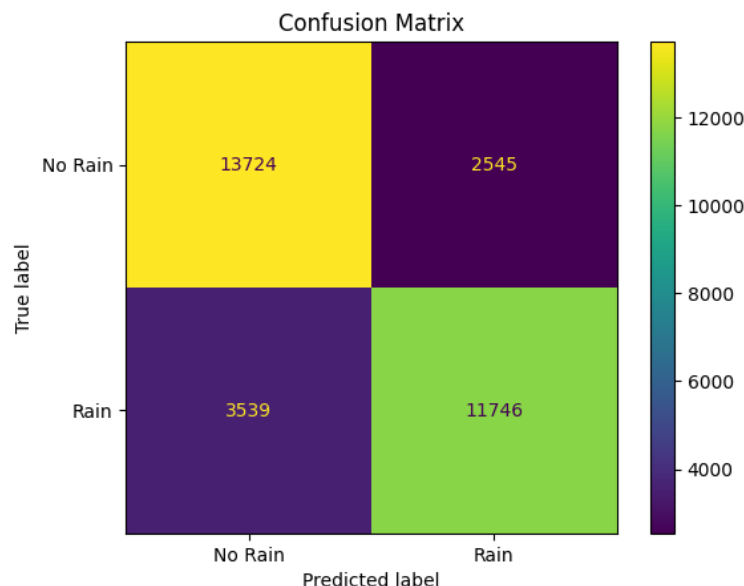
Pada tahap pelatihan, model XGBoost di-*tuning* menggunakan beberapa *hyperparameter* (istilah untuk parameter pelatihan seperti *learning_rate* dan *max_depth*) untuk memperoleh kombinasi yang memberikan performa terbaik pada data *test*. Proses *tuning* dilakukan secara iteratif (200 kali) dengan mencoba berbagai kombinasi nilai *hyperparameter* dan mengevaluasinya berdasarkan akurasi serta F1-score kelas hujan (kelas 1). Kombinasi *hyperparameter* terbaik yang diperoleh dapat dilihat pada Tabel 4.14.

Tabel 4. 14 Rangkuman Parameter XGBoost yang Menghasilkan Performa Terbaik

Parameter	Nilai
n_estimators	219
max_depth	6
learning_rate	0.0875
subsample	0.8902
colsample_bytree	0.7493

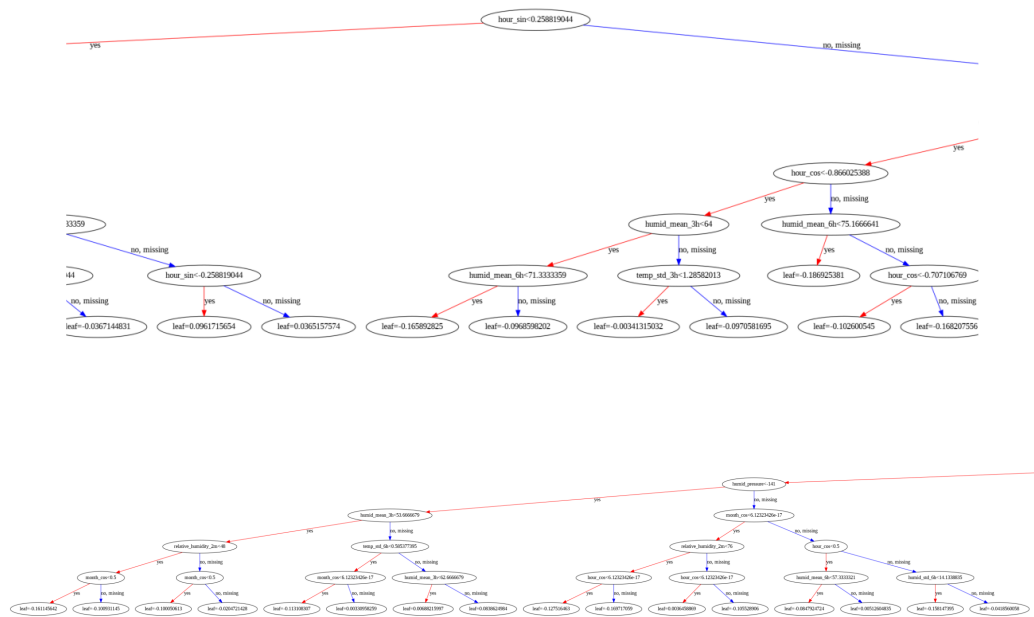
Parameter	Nilai
min_child_weight	4
gamma	4.1858
reg_alpha	0.4546
reg_lambda	3.2594
eval_metric	logloss
random_state	42

Parameter-parameter tersebut menghasilkan performa akurasi sebesar 80,72% dan F1-score hujan sebesar 79,43%. Nilai akurasi menunjukkan persentase seluruh data yang berhasil diklasifikasikan dengan benar oleh model, baik berpotensi hujan maupun tidak berpotensi hujan, sedangkan nilai F1-score hujan menunjukkan kemampuan model dalam memberikan prediksi hujan yang tepat sekaligus mendeteksi kejadian hujan dengan baik.



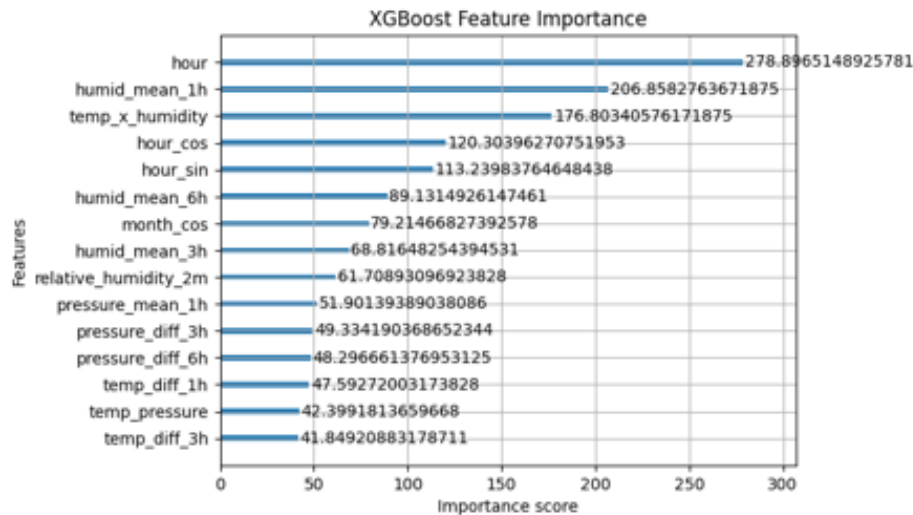
Gambar 4. 17 *Confusion Matrix* Model yang Dihasilkan

Kemudian, berdasarkan *confusion matrix* pada Gambar 4.17, model menghasilkan 13.724 *True Negative* (TN), yaitu data *No Rain* yang berhasil diprediksi sebagai *No Rain*, dan 11.746 *True Positive* (TP), yaitu data *Rain* yang berhasil diprediksi sebagai *Rain*. Sementara itu, terdapat 2.545 *False Positive* (FP), yaitu kondisi sebenarnya *No Rain* tetapi diprediksi *Rain*, serta 3.539 *False Negative* (FN), yaitu kondisi sebenarnya *Rain* tetapi diprediksi *No Rain*. Hasil ini menunjukkan bahwa model mampu mengenali kedua kelas, tetapi masih terdapat kesalahan prediksi pada kondisi hujan maupun tidak hujan.



Gambar 4. 18 Bagian dari *Decision Tree* Ke-0 pada Model XGBoost yang Dilatih

Pada Gambar 4.18, ditampilkan potongan bagian dari *decision tree* pertama (ke-0) pada model XGBoost yang dilatih. *Decision tree* inilah yang pertama kali mengeluarkan kontribusi untuk nilai probabilitas yang dihasilkan oleh algoritma nantinya. Pada pohon pertama ini, algoritma memilih fitur *hour_sin* (cek Persamaan 4.13) sebagai pemisah pertama. Jika nilai *hour_sin* yang dimasukkan bernilai lebih kecil daripada 0,258819044, maka alur prediksi akan mengarah ke sebelah kiri ("yes") dan selanjutnya hingga sampai ke daun (*leaf*). Jika sebaliknya, maka alur prediksi akan mengarah ke sebelah kanan ("no, missing", "missing" berarti nilai tidak diinputkan atau NaN). Nilai *leaf* yang didapatkan akan diakumulasikan, lalu dikonversi ke nilai probabilitas. Nilai probabilitas ini kemudian dibandingkan dengan sebuah *threshold*. *Threshold* yang digunakan dalam sistem ini adalah 0,5. Maksudnya adalah jika nilai probabilitas keluaran *decision tree* sama dengan atau lebih tinggi daripada 0,5, maka hasil prediksi ialah "hujan". Sebaliknya, jika nilai tersebut lebih kecil daripada 0,5, maka hasil prediksi ialah "tidak hujan". Adapun masing-masing fitur pada model memiliki tingkat *importance* masing-masing, seperti yang ditunjukkan pada Gambar 4.19.

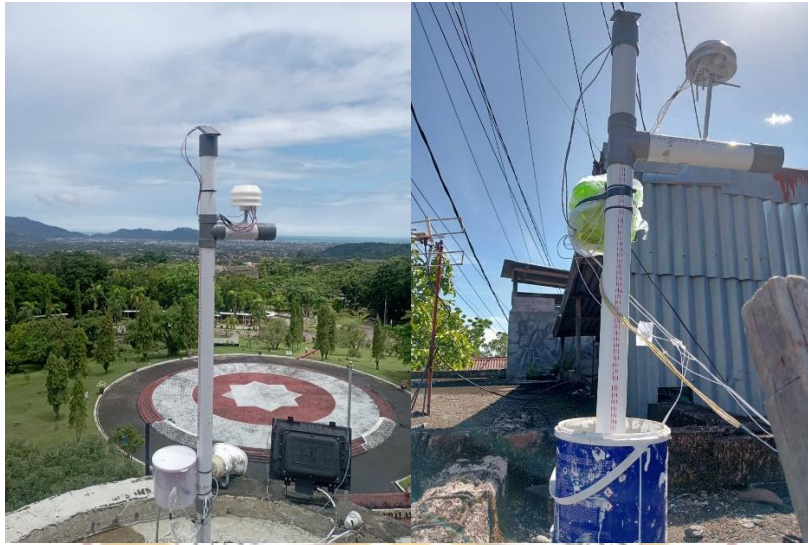


Gambar 4. 19 *Importance* Tiap Fitur

Tingkat *importance* diambil dari *gain* yang dimiliki tiap fitur. Dengan bahasa umum, *gain* Adalah Tingkat seberapa baik sebuah fitur menghasilkan prediksi saat digunakan untuk *split* (memisahkan, seperti kelembapan < 5). Pada Gambar 4.19, fitur *hour* menempati posisi paling atas, artinya fitur *hour* paling baik dalam memisahkan keadaan hujan atau tidak hujan pada 6 jam ke depan, atau dengan kata lain, paling banyak berkontribusi dalam proses prediksi yang dilakukan oleh model.

4.5 Penerapan Sistem (Keseluruhan)

Penempatan Node 1 dan Node 2 dilakukan pada lokasi yang telah ditentukan dengan mempertimbangkan kondisi lingkungan di sekitar titik pengamatan. Setiap *node* diusahakan ditempatkan pada posisi yang minim gangguan dari objek atau kondisi lingkungan yang dapat memengaruhi hasil pengukuran parameter cuaca, sehingga data yang diperoleh dapat merepresentasikan kondisi di lokasi pengamatan. Sumber listrik dan koneksi Wi-Fi diperoleh dari fasilitas terdekat pada masing-masing lokasi. Meskipun demikian, penerapan perangkat di lingkungan luar ruangan masih memiliki beberapa risiko, terutama terkait ketahanan perangkat terhadap kondisi lingkungan, sehingga aspek perlindungan dan ketahanan alat masih perlu ditingkatkan pada pengembangan selanjutnya.



Gambar 4. 20 Node 1 dan Node 2 yang Telah Terpasang pada Lokasinya Masing-Masing

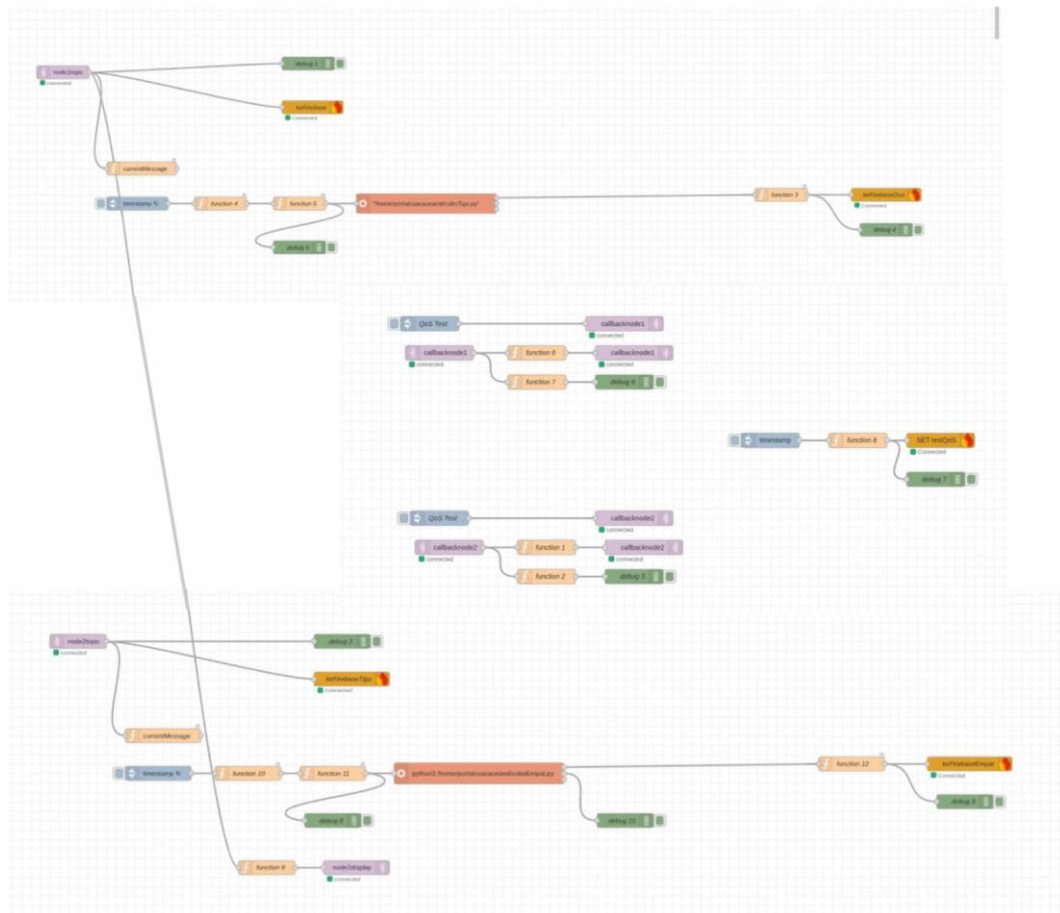


Gambar 4. 21 Pusat Pemrosesan

Pusat pemrosesan yang menggunakan Raspberry Pi 4 Model B dapat beroperasi secara kontinu selama didukung oleh sirkulasi udara yang baik dan koneksi Wi-Fi yang memadai. Sirkulasi udara yang baik dapat menjaga suhu kerja dan mencegah terjadinya peningkatan panas yang berlebihan. Dengan kondisi tersebut, pusat pemrosesan dapat menjalankan proses penerimaan, pengolahan, prediksi, dan pengiriman data secara terus-menerus sesuai kebutuhan sistem.

4.5.1 Workflow (Node-RED)

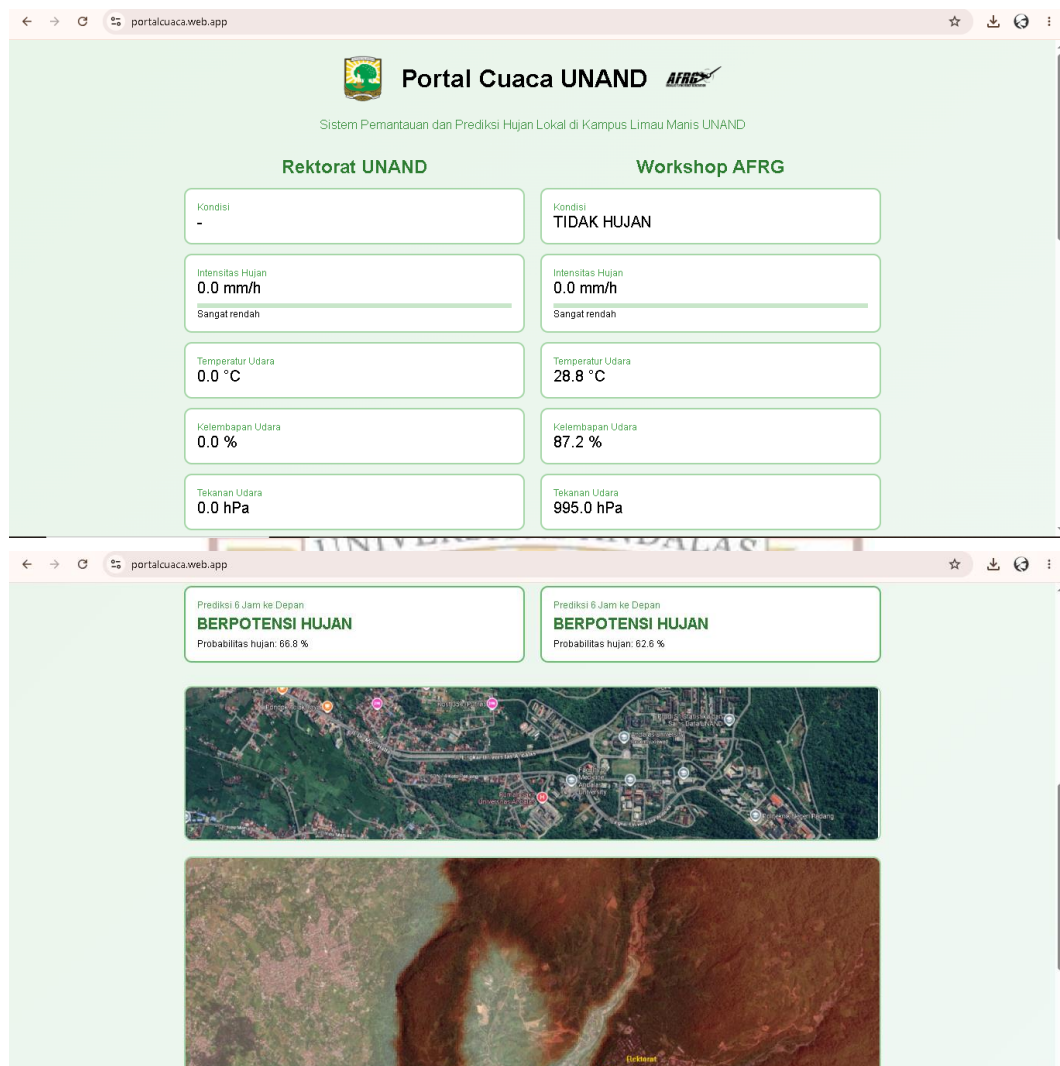
Konfigurasi Node-RED yang digunakan dalam penerapan sistem ditunjukkan oleh Gambar 4.22. Konfigurasi tersebut memperlihatkan alur (*workflow*) pengolahan data mulai dari penerimaan data sensor melalui MQTT, pemrosesan data, hingga penerusan data ke layanan penyimpanan dan kebutuhan sistem lainnya.



Gambar 4. 22 *Workflow* pada Node-RED

4.5.2 Website

Gambar 4.23 menunjukkan tampilan *website* yang digunakan sebagai antarmuka pemantauan dan prediksi keadaan cuaca. *Website* tersebut menampilkan data hasil pengukuran parameter cuaca dari tiap *node*, prediksi keadaan cuaca untuk enam jam ke depan, dan informasi-informasi lainnya, seperti peta hasil kalkulasi IDW tentang area yang diwakili keadaan cuacanya oleh tiap *node*.



Gambar 4. 23 Tampilan Website “portalcuaca.web.app”

4.5.3 Data Prediksi *Real-Time*

a. Node 1

Tabel 4. 15 Catatan Data Prediksi Node 1

Waktu Prediksi	Probabilitas Hujan	Prediksi	Keadaan Aktual	Status
01:11	32.32%	Tidak hujan	Tidak hujan	Benar
02:11	54.49%	Hujan	Tidak hujan	Salah

Waktu Prediksi	Probabilitas Hujan	Prediksi	Keadaan Aktual	Status
03:11	43.22%	Tidak hujan	Tidak hujan	Benar
04:48	57.83%	Hujan	Tidak hujan	Salah
05:48	46.97%	Tidak hujan	Tidak hujan	Benar
06:48	43.90%	Tidak hujan	Tidak hujan	Benar
07:48	37.56%	Tidak hujan	Tidak hujan	Benar
08:48	18.67%	Tidak hujan	Tidak hujan	Benar
09:48	9.69%	Tidak hujan	Tidak hujan	Benar
10:48	71.51%	Hujan	Tidak hujan	Salah
11:48	53.29%	Hujan	Tidak hujan	Salah
12:48	90.78%	Hujan	Tidak hujan	Salah
13:48	90.18%	Hujan	Tidak hujan	Salah

Tabel 4. 16 Catatan Data Tambahan Prediksi Node 1 (pada hari lain)

Waktu Prediksi	Probabilitas Hujan	Prediksi	Keadaan Aktual	Status
09:55	0,5087	Hujan	Tidak hujan	Salah
10:55	0,4757	Tidak hujan	Tidak hujan	Benar

Waktu Prediksi	Probabilitas Hujan	Prediksi	Keadaan Aktual	Status
11:57	66,45%	Hujan	Hujan	Benar
12:57	65,56%	Hujan	Hujan	Benar
13:57	95,31%	Hujan	Tidak hujan	Salah
14:57	91,77%	Hujan	Tidak hujan	Salah
15:57	97,46%	Hujan	Tidak hujan	Salah
16:57	94,96%	Hujan	Tidak hujan	Salah

b. Node 2

Tabel 4. 17 Catatan Data Prediksi Node 2

Waktu Prediksi	Probabilitas Hujan	Prediksi	Keadaan Aktual	Status
01:11	64.23%	Hujan	Tidak hujan	Salah
02:11	46.87%	Tidak hujan	Tidak hujan	Benar
03:11	37.18%	Tidak hujan	Tidak hujan	Benar
04:48	50.78%	Hujan	Tidak hujan	Salah
05:48	31.18%	Tidak hujan	Tidak hujan	Benar
06:48	24.67%	Tidak hujan	Tidak hujan	Benar
07:48	40.69%	Tidak hujan	Tidak hujan	Benar

Waktu Prediksi	Probabilitas Hujan	Prediksi	Keadaan Aktual	Status
08:48	29.10%	Tidak hujan	Tidak hujan	Benar
09:48	15.13%	Tidak hujan	Tidak hujan	Benar
10:48	48.96%	Tidak hujan	Tidak hujan	Benar
11:48	77.62%	Hujan	Tidak hujan	Salah
12:48	76.18%	Hujan	Tidak hujan	Salah
13:48	91.55%	Hujan	Tidak hujan	Salah

Berdasarkan hasil pengujian, algoritma XGBoost yang diterapkan pada sistem menghasilkan akurasi sebesar 80,72% dan F1-score sebesar 79,43% pada kelas hujan. Jika dibandingkan dengan penelitian S. Nuthalapati dan A. Nuthalapati yang memperoleh akurasi sebesar 80,95% menggunakan algoritma XGBoost untuk prediksi keadaan cuaca, hasil penelitian ini memiliki selisih sebesar 0,23 poin persentase. Sementara itu, Yasper et al. melaporkan bahwa XGBoost mampu mencapai akurasi hingga 95% dalam mengestimasi kondisi hujan dan tidak hujan. Dengan demikian, akurasi yang diperoleh pada penelitian ini masih berada di bawah hasil yang dilaporkan oleh kedua penelitian tersebut, tetapi memiliki hasil yang relatif mendekati penelitian S. Nuthalapati dan A. Nuthalapati. Perbedaan hasil tersebut dapat dipengaruhi oleh karakteristik lokasi pengamatan, periode dan jumlah data, fitur masukan, serta konfigurasi model yang digunakan pada masing-masing penelitian.

Berdasarkan pengujian *real-time* tambahan pada kedua *node*, diperoleh 34 prediksi dengan 18 prediksi sesuai dan 16 prediksi tidak sesuai, sehingga akurasi gabungan sebesar 52,94%. Seluruh kesalahan prediksi berupa *false positive*, yaitu sistem memprediksi “Hujan”, tetapi tidak terjadi hujan dalam periode enam jam berikutnya. Hasil ini merupakan pengujian lapangan tambahan dengan jumlah data terbatas dan tidak menggantikan hasil evaluasi utama model XGBoost sebesar 80,72% akurasi dan 79,43% F1-score. Pengujian *real-time* menunjukkan bahwa model masih dapat menghasilkan *false alarm* pada kondisi lapangan tertentu. Adapun *false alarm* pada pengujian *real-time* kemungkinan dipengaruhi oleh beberapa faktor:

1. Pembacaan Sensor

Faktor berikutnya adalah keterbatasan pembacaan sensor, keterbatasan tersebut dapat menghasilkan fluktuasi nilai parameter masukan, yang kemudian memengaruhi hasil prediksi model.

2. Kondisi Lingkungan Lokal

Perubahan kondisi atmosfer dan pengaruh lingkungan lokal di sekitar *node* dapat menghasilkan pola parameter yang menyerupai kondisi sebelum hujan tanpa selalu diikuti oleh kejadian hujan. Contohnya adalah temperatur yang dipengaruhi oleh atap seng atau beton di sekitar tempat pemasangan *node*.

3. Fitur XGBoost yang Kurang

False alarm juga dapat dipengaruhi oleh keterbatasan fitur meteorologi yang digunakan, karena beberapa faktor seperti tutupan awan dan karakteristik angin belum direpresentasikan secara langsung.

4. Nilai *Threshold* yang Belum Maksimal

Selain itu, penentuan *classification threshold* juga dapat memengaruhi jumlah *false positive* yang dihasilkan. *Threshold* yang digunakan menentukan batas probabilitas untuk mengklasifikasikan suatu kondisi sebagai hujan. *Threshold* yang lebih rendah cenderung membuat model lebih sensitif dalam mendeteksi hujan, tetapi dapat meningkatkan *false positive*, sedangkan peningkatan *threshold* dapat mengurangi *false positive* dengan konsekuensi meningkatnya kemungkinan *false negative*. Terakhir, horizon prediksi enam jam juga dapat menyebabkan ketidakpastian yang lebih besar karena kondisi atmosfer masih dapat mengalami perubahan setelah prediksi dilakukan.

Selama pengambilan data yang dilakukan kurang lebih 2 minggu, masih terdapat kendala pada proses akuisisi data yang diakibatkan dari kurangnya reliabilitas sistem, seperti rusaknya data yang diakibatkan oleh masalah pada catu daya sistem atau kabel yang longgar, sehingga data yang dapat digunakan menjadi lebih sedikit. Hal ini menjadi saran perbaikan yang dibahas pada Bab 5.