BAB V
HASIL DAN PEMBAHASAN

Bab ini memuat uraian tentang hasil penelitian yang telah dilakukan. Setiap pencapaian dari komponen-komponen yang mendukung pembuatan sistem ini akan diuraikan secara menyeluruh. Bab ini terdiri dari dua subbab, yaitu hasil dan pembahasan. Penjelasan dari masing-masing subbab dijabarkan sebagai berikut.

5.1 Hasil

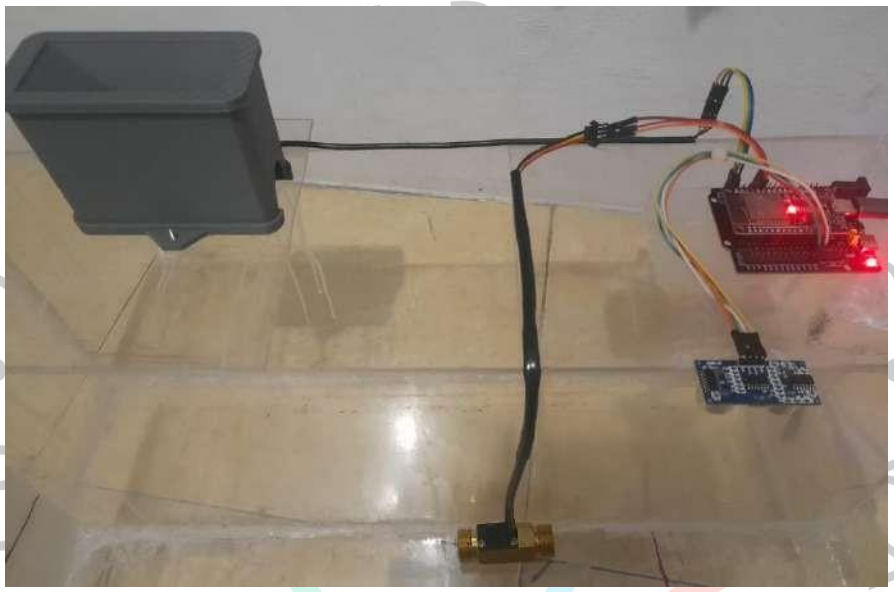
Pada subbab ini akan diuraikan hasil implementasi sistem prediksi banjir yang telah disusun berdasarkan kebutuhan dan perancangan yang dilakukan oleh peneliti. Hasil implementasi ini terbagi menjadi tiga bagian utama, yaitu pengembangan perangkat keras (sistem tertanam) untuk pengambilan data, hasil prediksi SVM (Support Vector Machine), dan pengembangan perangkat lunak (kode program). Ketiga bagian tersebut bertujuan untuk memastikan bahwa sistem prediksi banjir dapat berfungsi sesuai dengan prinsip kerja yang telah dijelaskan pada subbab sebelumnya. Berikut merupakan hasil dari penelitian yang dilakukan.

5.1.1 Perakitan

Tahap perakitan merupakan proses implementasi seluruh komponen yang mendukung pengembangan Sistem Prediksi Banjir Berbasis IoT. Pada tahap ini, proses perakitan mencakup tiga elemen utama, yaitu pembuatan prototipe lingkungan, integrasi sensor untuk pengumpulan data, dan pengiriman hasil prediksi ke bot Telegram sebagai notifikasi pengguna. Prototipe lingkungan dibuat menggunakan akrilik yang dirancang menyerupai model sungai mini, dilengkapi dengan pompa air untuk menciptakan aliran dinamis yang memungkinkan simulasi kondisi nyata. Struktur akrilik ini mendukung pengukuran parameter seperti tinggi muka air, debit aliran, dan curah hujan dalam berbagai skenario. Setelah prototipe selesai, integrasi sensor dilakukan dengan memasang sensor ultrasonik untuk mengukur tinggi muka air, sensor flow meter untuk memantau debit aliran air, dan sensor curah hujan tipe tipping bucket untuk mencatat volume curah hujan. Data dari sensor-sensor ini digunakan sebagai input utama dalam sistem prediksi banjir

berbasis algoritma Support Vector Machine (SVM). Hasil prediksi sistem kemudian dikirimkan ke bot Telegram untuk memberikan notifikasi kepada pengguna secara real-time, sehingga pengguna dapat memperoleh informasi terkini mengenai potensi banjir. Penjelasan lebih rinci akan disampaikan pada subbab-subbab berikutnya.

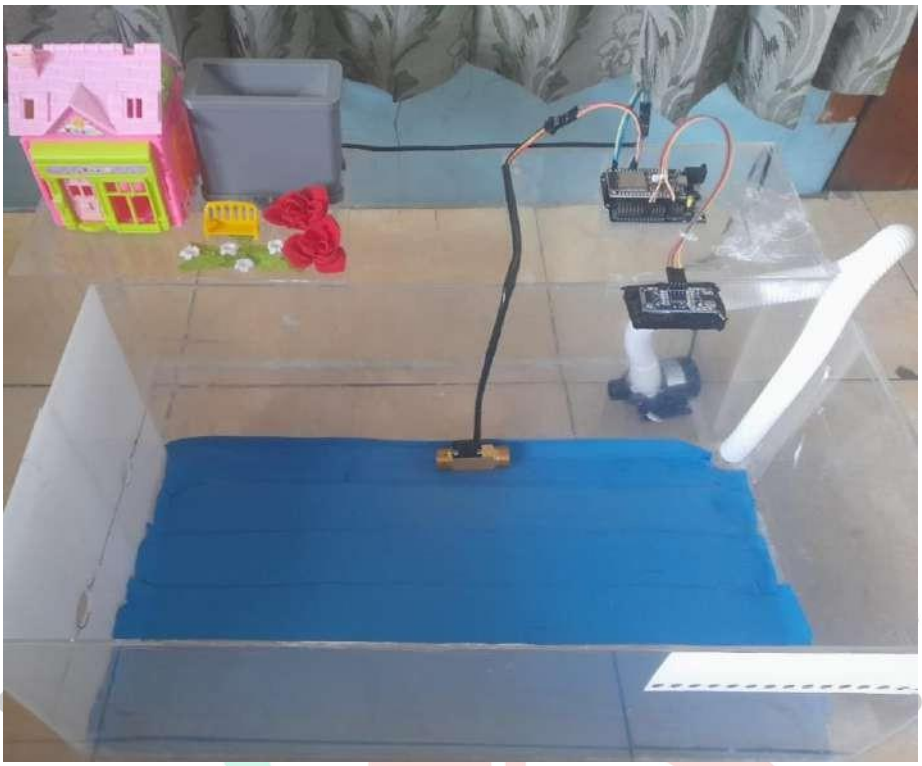
5.1.1.1 Tampilan Rangkaian Sistem



Gambar 5.1 Rangkaian Sistem

Gambar 5.1 menunjukkan rangkaian sistem prediksi banjir berbasis IoT yang dirancang untuk mengukur parameter lingkungan secara real-time. Sistem ini terdiri dari sensor ultrasonik untuk mengukur ketinggian air, sensor flow meter untuk mendeteksi laju aliran air, serta mikrokontroler ESP32 yang berfungsi sebagai pusat pengolahan data dan terhubung dengan sumber daya listrik untuk operasionalnya. Sebagai komponen utama dalam proses prediksi banjir, sistem ini berperan penting dalam pengambilan data lingkungan yang menjadi dasar untuk menghasilkan prediksi kondisi banjir secara akurat. Agar sistem ini dapat berfungsi dengan baik, seluruh komponen harus selalu dalam kondisi optimal, terutama mikrokontroler dan sensor ultrasonik, yang harus dirancang untuk bertahan terhadap paparan air hujan. Perlindungan tambahan, seperti pelindung tahan air atau penempatan di area yang terlindung, diperlukan untuk memastikan perangkat ini tetap stabil dan operasional dalam kondisi lingkungan yang ekstrem.

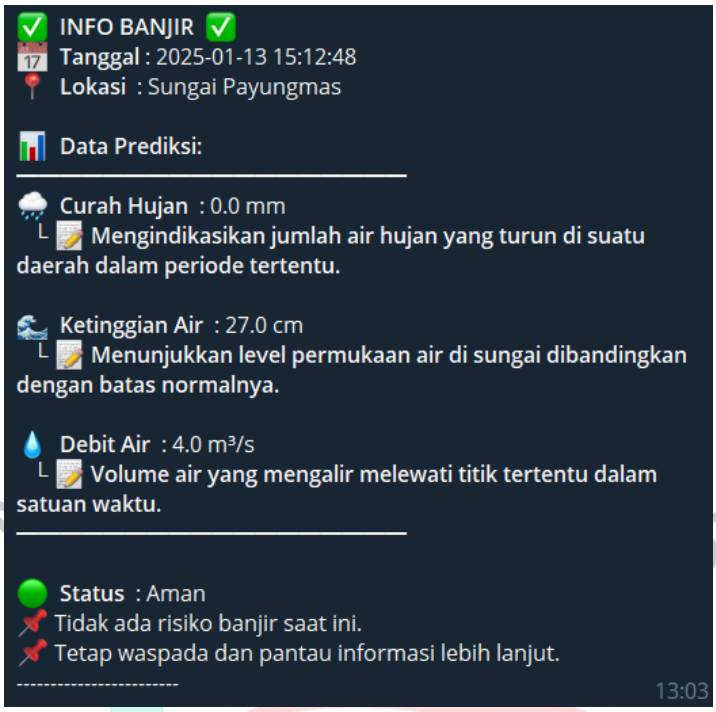
5.1.1.2 Tampilan Integrasi Sistem Pada Prototype Lingkungan



Gambar 5.2 Integrasi Sistem pada Prototype Lingkungan

Gambar 5.2 menjelaskan hasil integrasi sistem dengan prototype lingkungan yang telah dirancang. Pada gambar, sistem telah dipasangkan dengan prototype lingkungan yang menyerupai kondisi aktual. Komponen sensor, seperti sensor ultrasonik, flow, dan curah hujan, telah dihubungkan dengan mikrokontroler ESP32 dan diposisikan di lokasi strategis untuk membaca parameter lingkungan. Aliran air dalam prototype ini dihasilkan menggunakan sebuah pompa yang menyedot suplai air untuk menciptakan simulasi aliran. Hal ini bertujuan untuk memberikan kondisi yang mendekati nyata. Prototype ini juga dilengkapi dengan sebuah lubang pada sisi dinding untuk menjaga ketinggian air tetap dalam kondisi normal. Prototype ini telah dibangun sesuai dengan hasil perancangan sebelumnya, dengan dimensi ketinggian sungai sebesar 25 cm, lebar 20 cm, dan panjang 50 cm, sehingga mampu merepresentasikan kondisi sungai dalam skala kecil dengan akurasi yang memadai.

5.1.1.3 Tampilan Halaman Telegram Untuk Notifikasi



Gambar 5.3 Hasil Prediksi Telegram

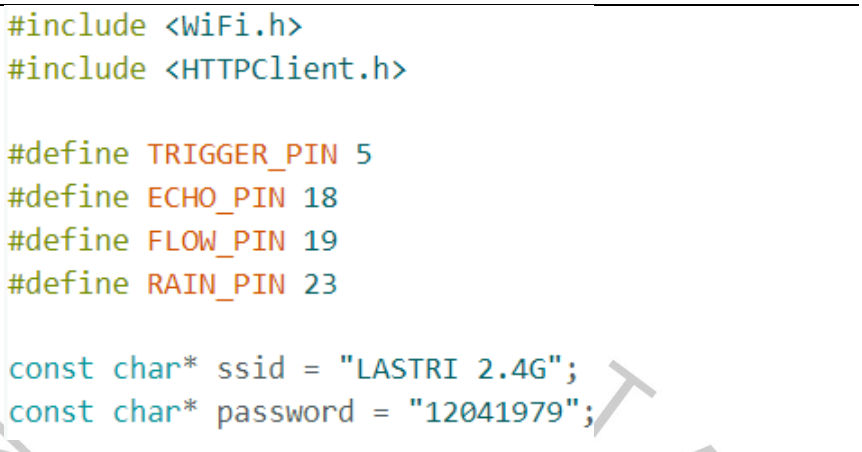
Gambar 5.3 memperlihatkan tampilan halaman Telegram yang digunakan untuk notifikasi hasil prediksi banjir pada sistem yang dirancang. Dalam gambar tersebut, bot Telegram dengan nama "Prediksi Banjir Sungai Cipayung Mas" secara otomatis mengirimkan notifikasi kepada pengguna setiap kali prediksi banjir dilakukan. Informasi yang disampaikan meliputi jarak (ketinggian air), laju aliran air, curah hujan, kondisi prediksi (Aman, Waspada, atau Bahaya), serta waktu prediksi dilakukan. Contoh notifikasi menunjukkan bahwa bot dapat mendeteksi perubahan parameter lingkungan dan memberikan status kondisi terbaru.

5.1.2 Kode Program

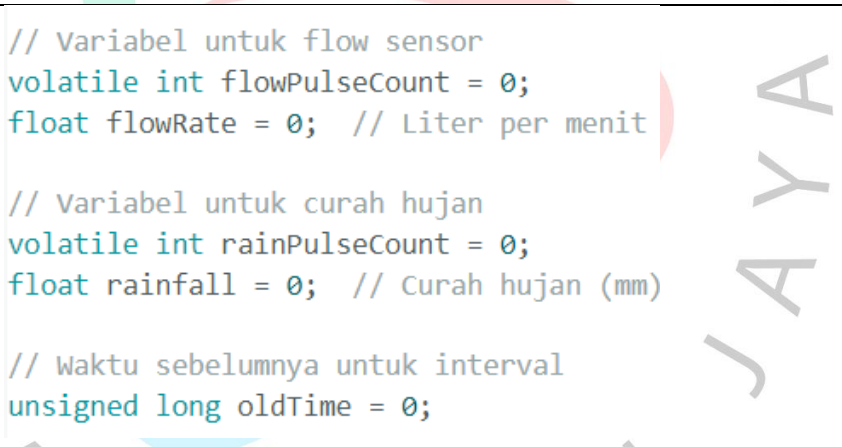
Kode program merupakan elemen penting dalam pengembangan sebuah sistem, karena kode tersebut berfungsi untuk mengontrol perangkat keras (hardware) dan sensor-sensor yang digunakan. Oleh karena itu, diperlukan kode program yang dirancang khusus untuk memenuhi kebutuhan sistem. Berikut ini adalah kode program yang digunakan dalam proyek ini, yang mencakup kontrol perangkat keras dan sensor.

Tabel 5.1 Kode Program Sistem Tertanam

Potongan Kode Program ke-1

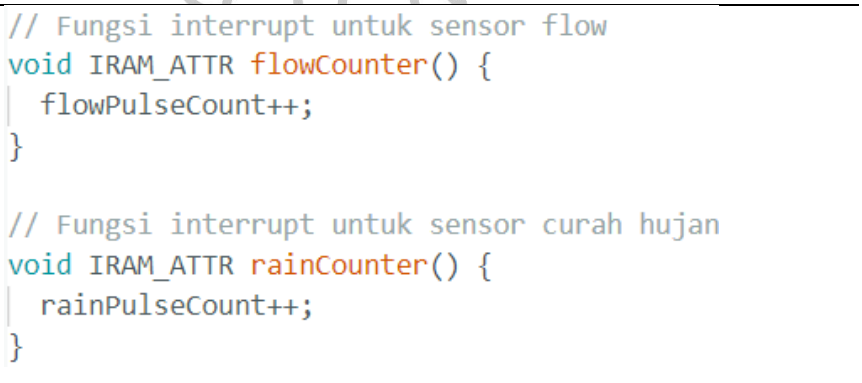
Gambar	 #include <WiFi.h> #include <HTTPClient.h> #define TRIGGER_PIN 5 #define ECHO_PIN 18 #define FLOW_PIN 19 #define RAIN_PIN 23 const char* ssid = "LASTRI 2.4G"; const char* password = "12041979";
Keterangan	Pendefinisian library yang digunakan, konfigurasi pin untuk sensor ultrasonik, flow, dan curah hujan, serta pengaturan koneksi Wi-Fi untuk pengiriman data ke server.

Potongan Kode Program ke-2

Gambar	
	// Variabel untuk flow sensor
	volatile int flowPulseCount = 0;
	float flowRate = 0; // Liter per menit
	
	// Variabel untuk curah hujan
	volatile int rainPulseCount = 0;
	float rainfall = 0; // Curah hujan (mm)
	
	// Waktu sebelumnya untuk interval
	unsigned long oldTime = 0;

Keterangan	Deklarasi variabel untuk menyimpan data pulsa dari sensor flow dan curah hujan, serta variabel waktu untuk pengiriman data secara berkala.

Potongan Kode Program ke-3

Gambar	
	// Fungsi interrupt untuk sensor flow
	void IRAM_ATTR flowCounter() {
	  flowPulseCount++;
	}
	
	// Fungsi interrupt untuk sensor curah hujan
	void IRAM_ATTR rainCounter() {
	  rainPulseCount++;
	}

Keterangan	Fungsi interrupt untuk menghitung pulsa dari sensor flow dan curah hujan secara real-time.

Potongan Kode Program ke-4

Gambar

```
void setup() {
  Serial.begin(9600);

  // Konfigurasi pin
  pinMode(TRIGGER_PIN, OUTPUT);
  pinMode(ECHO_PIN, INPUT);
  pinMode(FLOW_PIN, INPUT_PULLUP);
  pinMode(RAIN_PIN, INPUT_PULLUP);

  // Attach interrupt
  attachInterrupt(digitalPinToInterrupt(FLOW_PIN), flowCounter, FALLING);
  attachInterrupt(digitalPinToInterrupt(RAIN_PIN), rainCounter, FALLING);

  // Menghubungkan ke WiFi
  WiFi.begin(ssid, password);
  Serial.print("Menghubungkan ke WiFi");
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.println("\nTerhubung ke WiFi!");
  Serial.print("Alamat IP: ");
  Serial.println(WiFi.localIP());
}
```

Keterangan	Inisialisasi pin, setup fungsi interrupt untuk sensor, konfigurasi koneksi ke jaringan Wi-Fi, serta penyiapan komunikasi untuk pengiriman data dari sensor ke server.

Potongan Kode Program ke-5

Gambar

```
void loop() {
  // --- Sensor Ultrasonik ---
  digitalWrite(TRIGGER_PIN, LOW);
  delayMicroseconds(2);
  digitalWrite(TRIGGER_PIN, HIGH);
  delayMicroseconds(10);
  digitalWrite(TRIGGER_PIN, LOW);

  long duration = pulseIn(ECHO_PIN, HIGH);
  long distance = duration * 0.034 / 2;
```

Keterangan	Berfungsi untuk membaca data dari sensor ultrasonik. Pin TRIGGER_PIN dikontrol untuk mengirim sinyal pulsa, dan pin ECHO_PIN membaca pantulan pulsa tersebut untuk menghitung jarak dalam satuan sentimeter (cm).

Potongan Kode Program ke-6

Gambar	```// --- Sensor Flow dan Curah Hujan ---
	if (millis() - oldTime > 3000) { // Interval pengiriman setiap 3 detik
	  oldTime = millis();
	
	  flowRate = flowPulseCount / 7.5; // L/min
	  flowPulseCount = 0; // Reset pulsa flow
	
	  rainfall = rainPulseCount * 0.2794; // Konversi pulsa ke mm
	  rainPulseCount = 0; // Reset pulsa hujan
	
	  // --- Tampilkan Data ---
	  Serial.println("----- Data Sensor -----");
	  Serial.print("Jarak (Ultrasonik): ");
	  Serial.print(distance);
	  Serial.println(" cm");
	  Serial.print("Laju Aliran Air: ");
	  Serial.print(flowRate);
	  Serial.println(" L/min");
	  Serial.print("Curah Hujan: ");
	  Serial.print(rainfall);
	  Serial.println(" mm");```

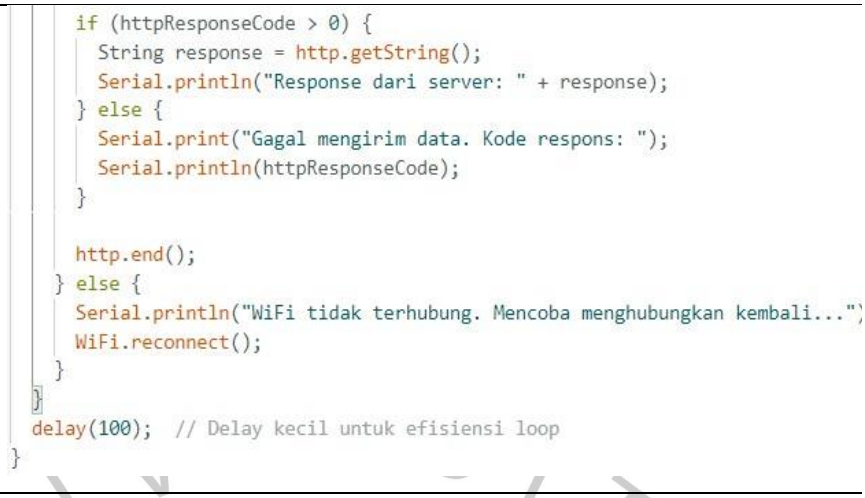
Keterangan	Kode ini menghitung laju aliran air dan curah hujan setiap 3 detik menggunakan nilai pulsa dari sensor. Pulsa flow dikonversi ke L/min, sedangkan pulsa hujan dikonversi ke mm. Data hasil perhitungan, termasuk jarak dari sensor ultrasonik, laju aliran air, dan curah hujan, ditampilkan pada Serial Monitor.

Potongan Kode Program ke-7

Gambar	```// --- Kirim Data ke Server ---
	if (WiFi.status() == WL_CONNECTED) {
	  HTTPClient http;
	  String serverName = "http://10.0.0.40/sensor/sensor_data.php";
	  http.begin(serverName);
	  http.addHeader("Content-Type", "application/x-www-form-urlencoded");
	
	  String httpRequestData = "distance=" + String(distance) + "&flowRate='
	  + String(flowRate) + "&rainfall=" + String(rainfall);
	  Serial.println("Mengirim data ke server...");
	
	  int httpResponseCode = http.POST(httpRequestData);```

Keterangan	Kode ini memeriksa koneksi WiFi sebelum memulai komunikasi HTTP. Jika terhubung, kode membuat koneksi ke server melalui URL yang ditentukan dan mengatur header konten. Data berupa jarak (distance), laju aliran (flowRate), dan curah hujan (rainfall) dikemas dalam format URL- encoded dan dikirim ke server menggunakan metode HTTP POST.

Potongan Kode Program ke-8

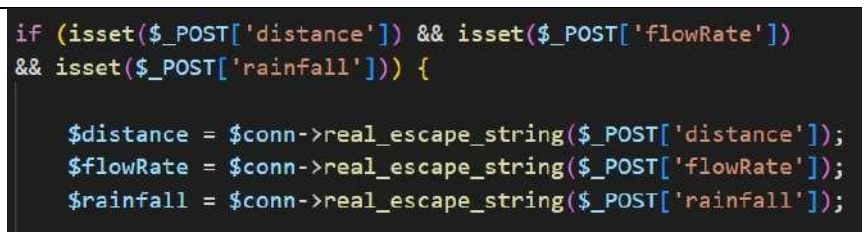
Gambar	
Keterangan	Kode ini memeriksa respons dari server setelah pengiriman data. Jika respons berhasil (kode respons > 0), pesan respons ditampilkan di Serial Monitor. Jika gagal, kode mencetak pesan kesalahan dengan kode respons. Koneksi HTTP diakhiri dengan http.end(), dan delay singkat diberikan sebelum iterasi berikutnya.

Tabel 5.2 Kode Program Pengambilan Data Sensor

Potongan Kode Program ke-1

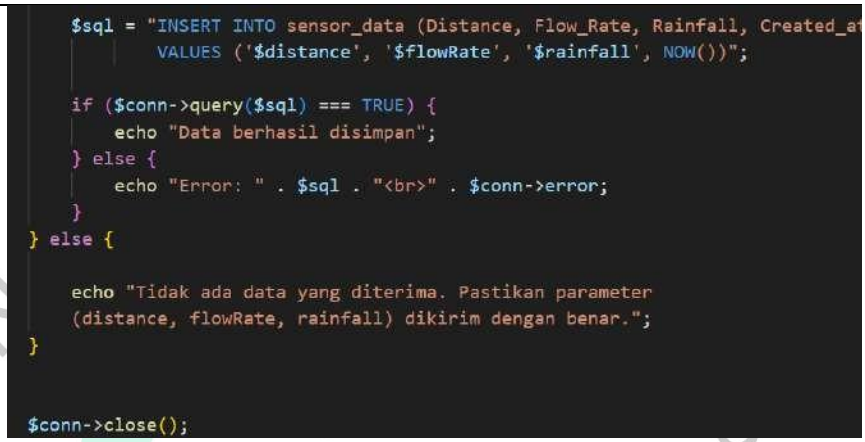
Gambar	
Keterangan	Bagian ini berfungsi untuk mengatur koneksi ke database MySQL. Jika koneksi gagal, program akan berhenti dan menampilkan pesan kesalahan.

Potongan Kode Program ke-2

Gambar	

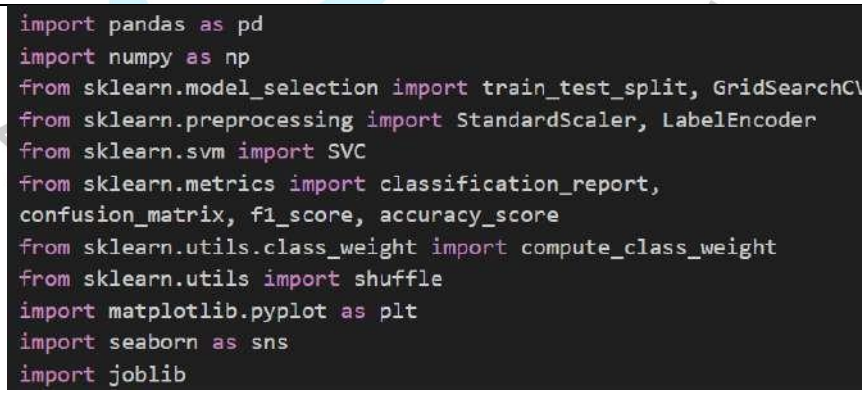
Keterangan	Bagian ini memeriksa apakah parameter distance, flowRate, dan rainfall dikirim melalui metode POST. Jika ada, data diambil dan dibersihkan menggunakan real_escape_string untuk mencegah serangan SQL injection.

Potongan Kode Program ke-3

Gambar	
Keterangan	Bagian ini menyimpan data sensor (Distance, Flow_Rate, dan Rainfall) ke dalam tabel sensor_data. Jika proses berhasil, akan ditampilkan pesan Data berhasil disimpan, jika gagal, akan menampilkan pesan kesalahan. Jika data POST tidak ditemukan, sistem menampilkan pesan peringatan agar memastikan parameter dikirim dengan benar.

Tabel 5.3 Kode Program Pembuatan Model Prediksi

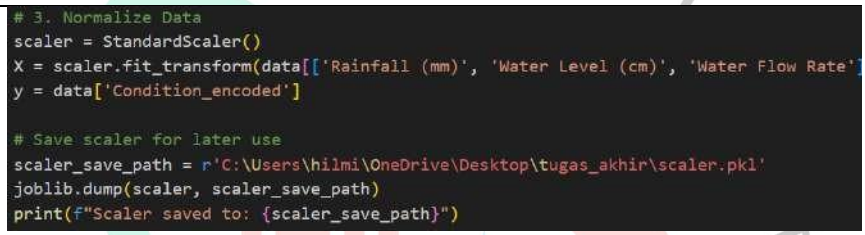
Potongan Kode Program ke-1

Gambar	
Keterangan	Pemanggilan library yang digunakan dalam melakukan pelatihan model.

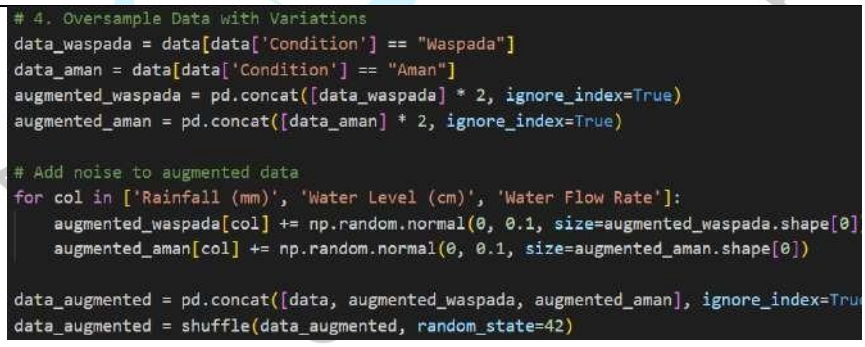
Potongan Kode Program ke-2

Gambar	
Keterangan	Membaca dataset dari file CSV yang berisi data kondisi banjir, seperti curah hujan, ketinggian air, dan laju aliran air. Mengubah label kondisi (Aman, Waspada, Bahaya) menjadi nilai numerik (0, 1, 2) untuk pemrosesan model. Label encoder disimpan untuk penggunaan di masa depan.

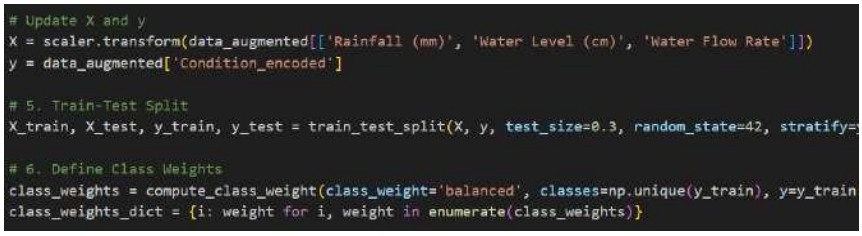
Potongan Kode Program ke-3

Gambar	
Keterangan	Melakukan normalisasi pada fitur input agar data memiliki skala yang seragam. Normalizer disimpan untuk memastikan konsistensi saat melakukan prediksi.

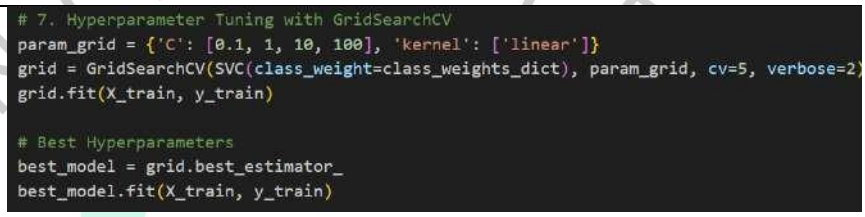
Potongan Kode Program ke-4

Gambar	
Keterangan	Melakukan oversampling untuk menyeimbangkan dataset, khususnya pada kategori minoritas seperti Waspada dan Aman, dengan menambahkan variasi noise kecil.

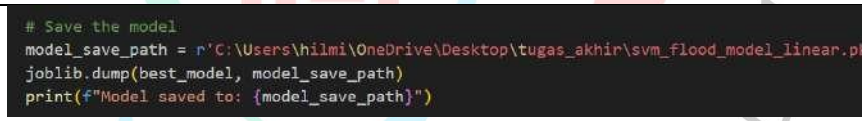
Potongan Kode Program ke-5

Gambar	
Keterangan	Membagi dataset menjadi data latih dan data uji dengan rasio 70:30, sambil menjaga proporsi label menggunakan metode stratify. Menghitung bobot kelas secara otomatis untuk menangani dataset yang tidak seimbang.

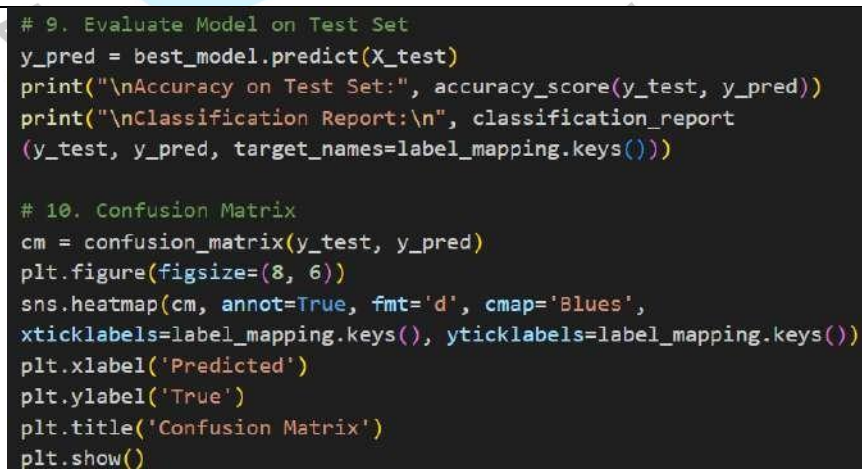
Potongan Kode Program ke-6

Gambar	
Keterangan	Melakukan tuning parameter untuk Support Vector Machine (SVM) menggunakan GridSearchCV dan validasi silang 5-fold untuk menemukan kombinasi parameter terbaik.

Potongan Kode Program ke-7

Gambar	
Keterangan	Menyimpan model SVM terbaik yang telah dilatih untuk digunakan pada prediksi.

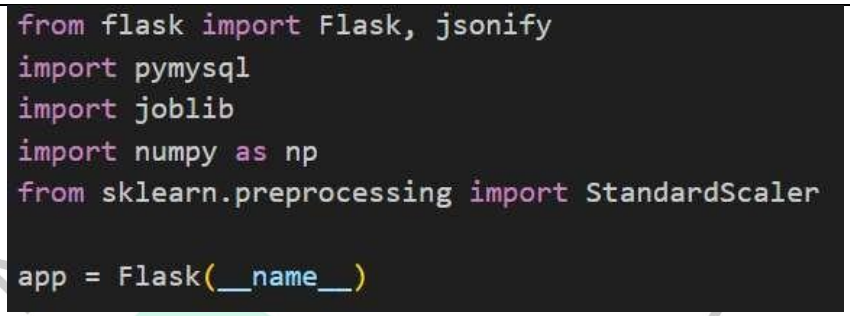
Potongan Kode Program ke-8

Gambar	
Keterangan	Mengevaluasi kinerja model pada data uji menggunakan metrik akurasi dan laporan klasifikasi (precision, recall, f1-score). Menampilkan

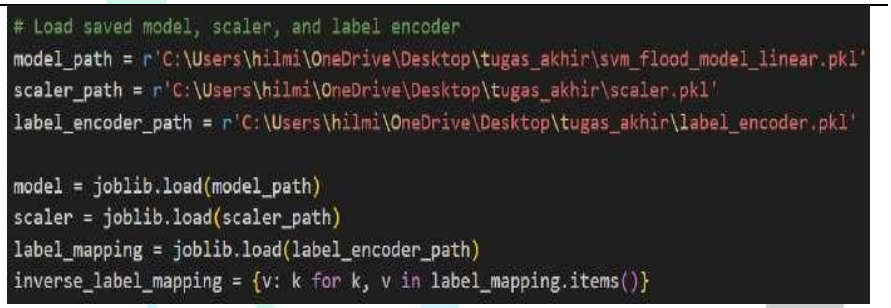
confusion matrix untuk memvisualisasikan distribusi prediksi model, menunjukkan perbandingan antara prediksi benar dan salah untuk setiap kelas.

Tabel 5.4 Kode Program Melakukan Proses Prediksi

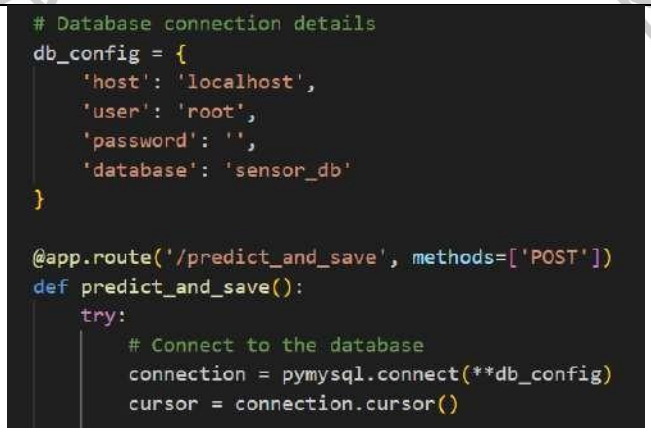
Potongan Kode Program ke-1

Gambar	
Keterangan	Pemanggilan library yang digunakan dalam melakukan proses prediksi.

Potongan Kode Program ke-2

Gambar	
Keterangan	Pemanggilan file model yang disimpan, Scaler untuk normalisasi data dan label encoder untuk mengonversi hasil prediksi dari angka menjadi nama kondisi.

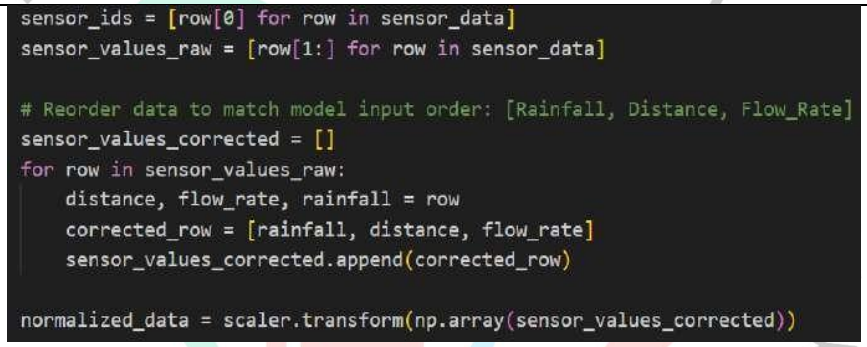
Potongan Kode Program ke-3

Gambar	
Keterangan	Menyimpan konfigurasi koneksi ke database MySQL. Mendefinisikan endpoint /predict_and_save yang menerima HTTP POST Request.

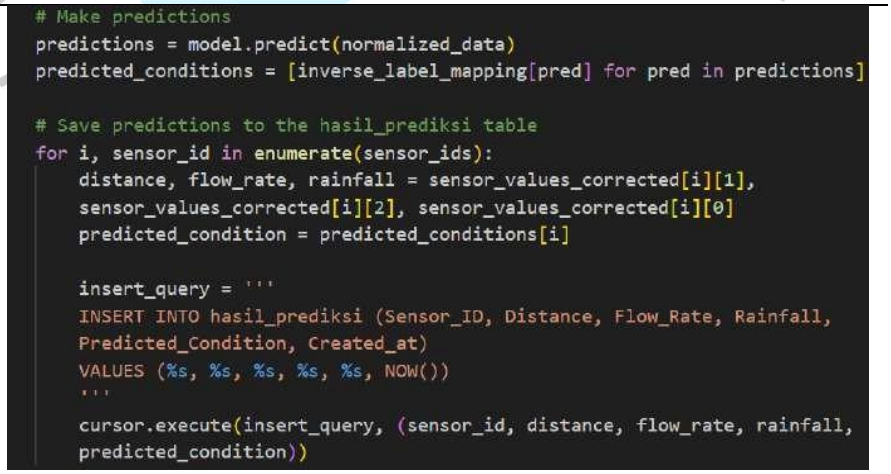
Potongan Kode Program ke-4

Gambar	
Keterangan	Query ini mengambil data sensor dari tabel sensor_data. Mengabaikan data yang sudah memiliki prediksi di tabel hasil_prediksi.

Potongan Kode Program ke-5

Gambar	
Keterangan	Data sensor diproses agar sesuai dengan urutan input [Rainfall, Distance, Flow_Rate] untuk model. Data dinormalisasi menggunakan StandardScaler.

Potongan Kode Program ke-6

Gambar	
Keterangan	Model memprediksi data sensor yang telah dinormalisasi. Hasil prediksi dikonversi ke label aslinya (Aman, Waspada, Bahaya) menggunakan

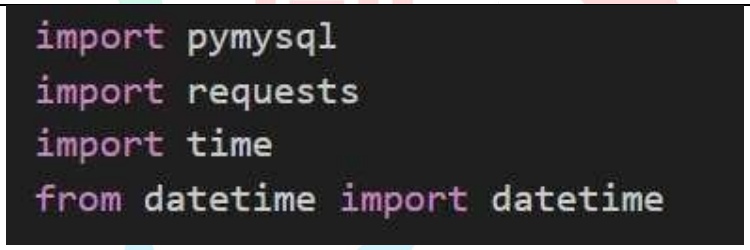
	inverse label mapping. Data hasil prediksi, termasuk Sensor_ID, Distance, Flow Rate, Rainfall, dan Predicted Condition, disimpan ke tabel hasil_prediksi. Waktu penyimpanan dicatat menggunakan fungsi NOW().

Potongan Kode Program ke-7

Gambar	
Keterangan	API mengembalikan respon JSON yang menyatakan jumlah data yang diproses dan berhasil disimpan.

Tabel 5.5 Kode Program Pengiriman Hasil Prediksi ke Telegram

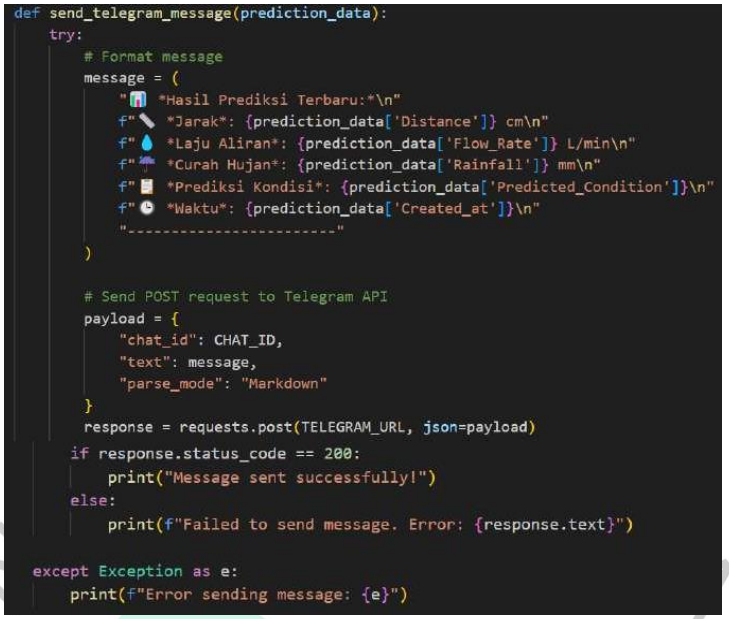
Potongan Kode Program ke-1

Gambar	
Keterangan	Pemanggilan library yang digunakan dalam melakukan proses pengiriman hasil prediksi.

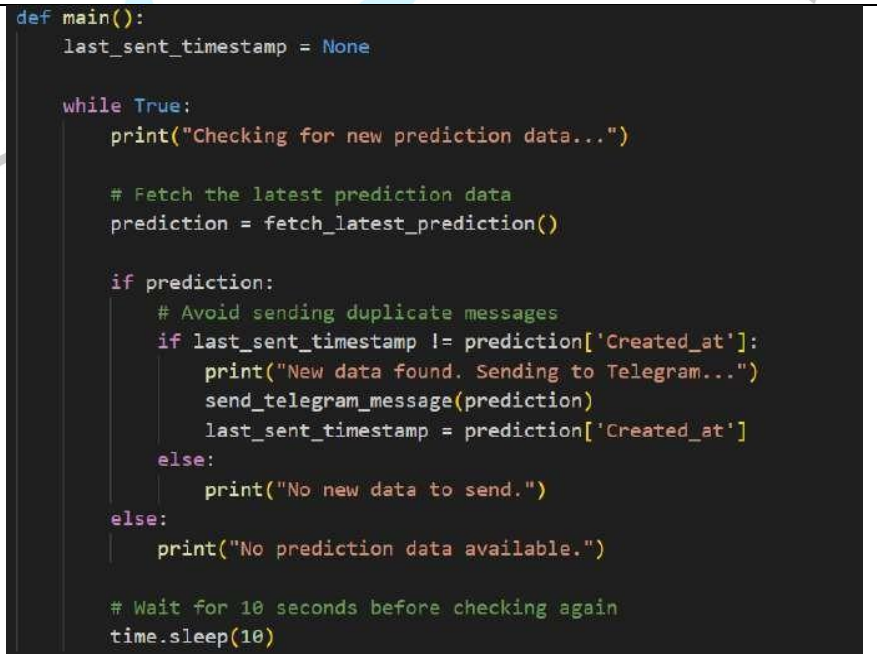
Potongan Kode Program ke-2

Gambar	
Keterangan	Fungsi fetch_latest_prediction mengambil data prediksi terbaru dari tabel hasil_prediksi di database. Koneksi ke database dibuat menggunakan konfigurasi db_config, lalu query SQL dijalankan untuk mendapatkan satu data terbaru berdasarkan waktu (Created_at) dalam urutan menurun. Data hasil query diformat menjadi dictionary dengan informasi seperti Sensor_ID, Distance, Flow_Rate, Rainfall, Predicted_Condition, dan Created_at, atau None jika tidak ada data ditemukan. Fungsi ini juga menangani kesalahan dan memastikan koneksi ke database ditutup setelah selesai.

Potongan Kode Program ke-3

Gambar	
Keterangan	Fungsi send_telegram_message mengirimkan hasil prediksi ke bot Telegram. Data prediksi dari parameter prediction_data diformat menjadi pesan teks berisi informasi jarak, laju aliran, curah hujan, kondisi prediksi, dan waktu. Pesan ini dikirim melalui API Telegram menggunakan metode POST dengan chat_id, teks, dan format Markdown. Jika pengiriman berhasil, fungsi mencetak konfirmasi; jika gagal, menampilkan pesan kesalahan. Fungsi ini juga menangani error selama proses pengiriman pesan.

Potongan Kode Program ke-4

Gambar	

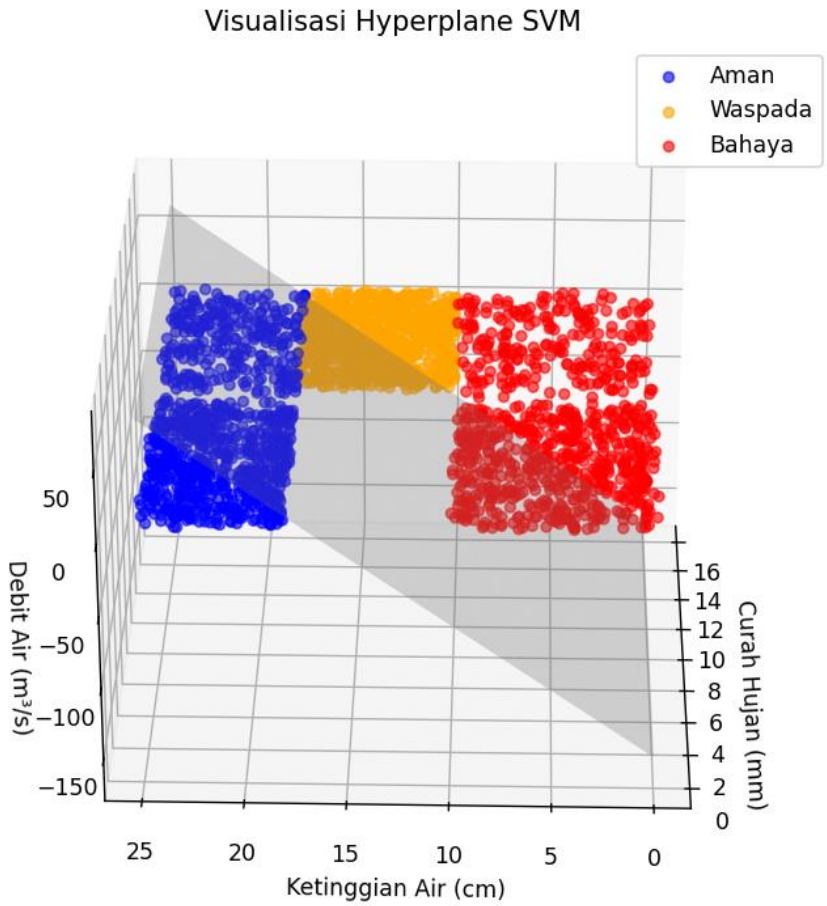
Keterangan	Fungsi main adalah program utama yang berjalan secara terus-menerus untuk memantau data prediksi terbaru di database. Fungsi ini memanggil fetch_latest_prediction untuk mengambil data terbaru dan memeriksa apakah data tersebut belum pernah dikirim sebelumnya. Jika ada data baru, fungsi akan memanggil send_telegram_message untuk mengirim hasil prediksi ke bot Telegram, dan memperbarui timestamp terakhir yang dikirim. Program ini melakukan pengecekan setiap 10 detik, memastikan tidak ada duplikasi pengiriman data.

5.2 Pembahasan

Sub bab ini menjelaskan hasil prediksi sistem, di mana setiap nilai parameter input diolah menggunakan rumus algoritma, pengujian sistem dengan metode blackbox, dan performa model dievaluasi berdasarkan hasil pengujian tersebut.

5.2.1 Hasil Perhitungan Prediksi Banjir

Pada tahap ini, peneliti telah berhasil membangun Hyperplane pada model berdasarkan dataset yang digunakan dalam sistem prediksi banjir menggunakan SVM. Hyperplane yang dihasilkan berfungsi sebagai batas keputusan yang memisahkan kategori kondisi sungai berdasarkan tiga parameter utama, yaitu Curah Hujan (mm), Ketinggian Air (cm), dan Debit Air (m^3/s). Model ini dilatih untuk membedakan tiga kondisi utama sungai, yaitu Aman (Biru), Waspada (Oranye), dan Bahaya (Merah). Berikut merupakan distribusi data pada hyperplane.



Gambar 5.4 Hasil Pembuatan Hyperplane

Hyperplane yang ditampilkan dalam bidang abu-abu berfungsi sebagai batas klasifikasi untuk memisahkan setiap kondisi berdasarkan data yang diberikan. Semakin jauh titik data dari hyperplane, semakin kuat kepastian model dalam menentukan kategorinya.

5.2.1.2 Hasil Penerapan Marginasi

Peneliti telah berhasil menerapkan marginisasi pada data hasil pelatihan model SVM untuk prediksi banjir. Proses marginisasi dilakukan dengan tujuan memaksimalkan jarak antar kategori (Aman, Waspada, dan Bahaya) dalam ruang fitur, sehingga model dapat membedakan setiap kategori dengan lebih optimal. Data yang digunakan berasal dari hasil pelatihan sebelum dan setelah marginisasi. Berikut merupakan tabel perbandingan akurasi model sebelum dan sesudah menggunakan proses marginisasi.

Tabel 5.6 Performa Model dengan Marginasi

Metrik	Sebelum Marginasi	Setelah Marginasi
Akurasi Model (%)	98.20000%	99.06667%
Mean Absolute Error	0.008000	0.009333
Root Mean Squared Error	0.0894	0.0966

Setelah menerapkan marginisasi, akurasi model tetap stabil di atas 99%, dengan hanya sedikit penurunan dari 99.20% menjadi 99.07%, menunjukkan bahwa marginisasi tidak menyebabkan overfitting atau degradasi performa. Meskipun nilai Mean Absolute Error (MAE) dan Root Mean Squared Error (RMSE) sedikit meningkat, hal ini mencerminkan peningkatan fleksibilitas model dalam menangani variasi data, sehingga prediksi lebih adaptif terhadap distribusi yang lebih seimbang. Selain itu, marginisasi memperlebar jarak antar kategori dalam ruang fitur, memungkinkan model mengklasifikasikan data baru dengan lebih baik dan mengurangi kesalahan prediksi pada data yang berada di sekitar batas keputusan (hyperplane).

5.2.1.3 Hasil Prediksi Data Uji Lapangan

Hasil dari percobaan prediksi dapat dilihat pada Tabel 5.7.

Tabel 5.7 Data Percobaan Prediksi Banjir

No.	Curah Hujan	Ketinggian Air	Debit Air
1	3.36 m/m	27 cm	4.24 m/m
2	14.06 m/m	9.21 cm	10.76 m/m
3	5.32 m/m	14.21 cm	4.54 m/m

Untuk menghitung nilai prediksi dapat dilakukan dengan memasukkan rumus fungsi keputusan yang digunakan dalam model SVM untuk menghitung prediksi.

$$\boldsymbol{f(x) = (-0.5886 \times x1) + (1.2226 \times x2) + (-0.7289 \times x3) + (-1.4692)}$$

Nilai vektor bobot [-0.5886, 1.2226, -0.7289] dan bias -1.4692 diperoleh melalui proses pelatihan model SVM menggunakan data latih yang telah dikumpulkan. Dalam proses ini, algoritma SVM mengoptimalkan bobot dengan

mencari hyperplane terbaik yang dapat memisahkan kategori Aman, Waspada, dan Bahaya berdasarkan parameter curah hujan, ketinggian air, dan debit air. Nilai-nilai tersebut merepresentasikan kontribusi masing-masing fitur dalam menentukan hasil klasifikasi, di mana bobot yang lebih besar menunjukkan pengaruh yang lebih dominan terhadap keputusan prediksi. Hasil perhitungan untuk setiap data uji adalah sebagai berikut:

- Data 1:

$$\boldsymbol{f(x)} = (\mathbf{-0.5886} \times 3.36) + (\mathbf{1.2226} \times 27) + (\mathbf{-0.7289} \times 4.24) + (\mathbf{-1.4692}) = 26.462 \rightarrow \textbf{Kategori Aman (0)}$$

- Data 2:

$$\boldsymbol{f(x)} = (\mathbf{-0.5886} \times 14.06) + (\mathbf{1.2226} \times 9.21) + (\mathbf{-0.7289} \times 10.76) + (\mathbf{-1.4692}) = \text{-}6.326 \rightarrow \textbf{Kategori Bahaya (2)}$$

- Data 3:

$$\boldsymbol{f(x)} = (\mathbf{-0.5886} \times 5.32) + (\mathbf{1.2226} \times 14.21) + (\mathbf{-0.7289} \times 4.54) + (\mathbf{-1.4692}) = 5.215 \rightarrow \textbf{Kategori Waspada (1)}$$

Berdasarkan perhitungan data menggunakan fungsi Keputusan, Data dengan nilai f(x) yang positif tinggi cenderung masuk dalam kategori Aman, karena kondisi sungai masih dalam batas normal dan tidak menunjukkan tanda-tanda peningkatan risiko banjir. Sementara itu, data dengan f(x) yang mendekati nol atau sedikit positif dikategorikan sebagai Waspada, yang menunjukkan adanya potensi peningkatan risiko banjir meskipun belum mencapai kondisi kritis. Berikut tampilan data prediksi yang telah diolah menggunakan model.

Tabel 5.8 Data Hasil Uji Lapangan Prediksi Banjir

No.	Curah Hujan	Ketinggian Air	Debit Air	Hasil Prediksi
1	3.36 m/m	27 cm	4.24 m/m	Aman (0)
2	14.06 m/m	9.21 cm	10.76 m/m	Bahaya (2)
3	5.32 m/m	14.21 cm	4.54 m/m	Waspada (1)

5.2.2 Hasil Pengujian Black Box Sistem

Pengujian sistem ini menggunakan pendekatan black box, dengan berbagai variasi input yang diuji untuk memastikan setiap fungsi dalam sistem beroperasi

sesuai dengan kebutuhan yang telah direncanakan. Hasil yang diperoleh dari pengujian ini akan dievaluasi dengan membandingkannya terhadap spesifikasi yang telah ditentukan sebelumnya. Hasil pengujian black box dijelaskan pada tabel berikut.

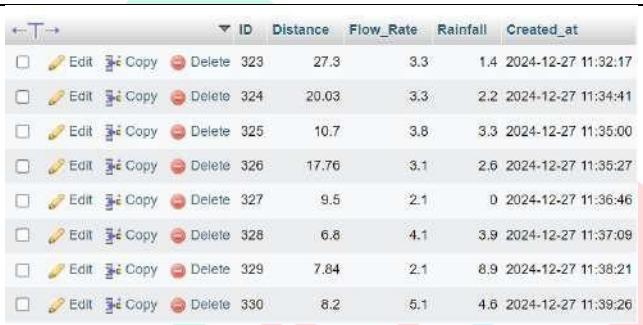
Tabel 5.9 Hasil Pengujian Black Box

Pengujian Black Box ke-1	
Skenario Pengujian	Sensor flow mengukur nilai debit air sungai
Hasil yang diharapkan	Sensor flow dapat menangkap nilai debit air dan mengirimkan data ke ESP32 melalui pin digital.
Hasil Pengujian	11:06:37.451 -> Laju Aliran Air: 55.47 L/min
Pengujian Black Box ke-2	
Skenario Pengujian	Sensor ultrasonic mengukur ketinggian air sungai
Hasil yang diharapkan	Sensor ultrasonic dapat mengukur jarak permukaan air dan mengirimkan data ke ESP32.
Hasil Pengujian	03:19:39.993 -> ----- Data Sensor Ultrasonik ----- 03:19:39.993 -> Jarak (Ultrasonik): 24.82 cm
Pengujian Black Box ke-3	
Skenario Pengujian	Sensor curah hujan (tipping bucket) menghitung curah hujan
Hasil yang diharapkan	Sensor tipping bucket dapat mendeteksi pergerakan bucket (pulsa) dan mengirimkan data ke ESP32.
Hasil Pengujian	11:06:33.973 -> Curah Hujan: 4.47 mm
Pengujian Black Box ke-4	
Skenario Pengujian	Rangkaian dihidupkan
Hasil yang diharapkan	ESP32 terhubung ke jaringan Wi-Fi dan mendapatkan alamat IP yang valid.
Hasil Pengujian	11:08:15.099 -> Terhubung ke WiFi! 11:08:15.099 -> Alamat IP: 10.0.0.88

Pengujian Black Box ke-5

Skenario Pengujian	ESP32 mengirimkan nilai sensor ke database PHPMyAdmin
Hasil yang diharapkan	ESP32 berhasil mengirimkan nilai sensor (flow, ultrasonic, tipping bucket) ke tabel sensor_data.
Hasil Pengujian	11:06:37.494 -> Mengirim data ke server... 11:06:37.758 -> Response dari server: Data berhasil disimpan

Pengujian Black Box ke-6

Skenario Pengujian	Database menerima data dari ESP32
Hasil yang diharapkan	Data dari ESP32 berhasil disimpan di tabel sensor_data dengan kolom yang sesuai.
Hasil Pengujian	

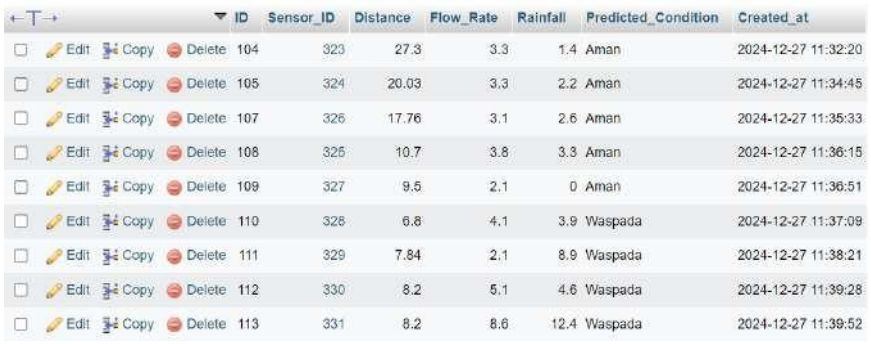
	ID	Distance	Flow_Rate	Rainfall	Created_at
Edit Copy Delete	323	27.3	3.3	1.4	2024-12-27 11:32:17
Edit Copy Delete	324	20.03	3.3	2.2	2024-12-27 11:34:41
Edit Copy Delete	325	10.7	3.8	3.3	2024-12-27 11:35:00
Edit Copy Delete	326	17.76	3.1	2.6	2024-12-27 11:35:27
Edit Copy Delete	327	9.5	2.1	0	2024-12-27 11:36:46
Edit Copy Delete	328	6.8	4.1	3.9	2024-12-27 11:37:09
Edit Copy Delete	329	7.84	2.1	8.9	2024-12-27 11:38:21
Edit Copy Delete	330	8.2	5.1	4.6	2024-12-27 11:39:26

Pengujian Black Box ke-7

Skenario Pengujian	Model mengambil, memproses data sensor dan memberikan hasil prediksi.
Hasil yang diharapkan	Model menghasilkan prediksi kondisi (Aman, Waspada, Bahaya) berdasarkan data dari tabel data_sensor.
Hasil Pengujian	PS C:\Users\hilmi\OneDrive\Desktop\tugas_akhir> python prediksi_svm.py [INFO] Memulai proses prediksi otomatis... [INFO] Berhasil memproses 1 baris data dan menyimpan hasil prediksi.

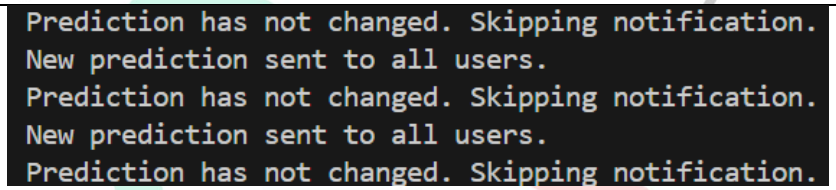
Pengujian Black Box ke-8

Skenario Pengujian	Model mengirimkan hasil prediksi ke database PHPMyAdmin
Hasil yang diharapkan	Prediksi disimpan di tabel prediksi_output dengan kolom: ID, data sensor, hasil prediksi, timestamp.

Hasil Pengujian	

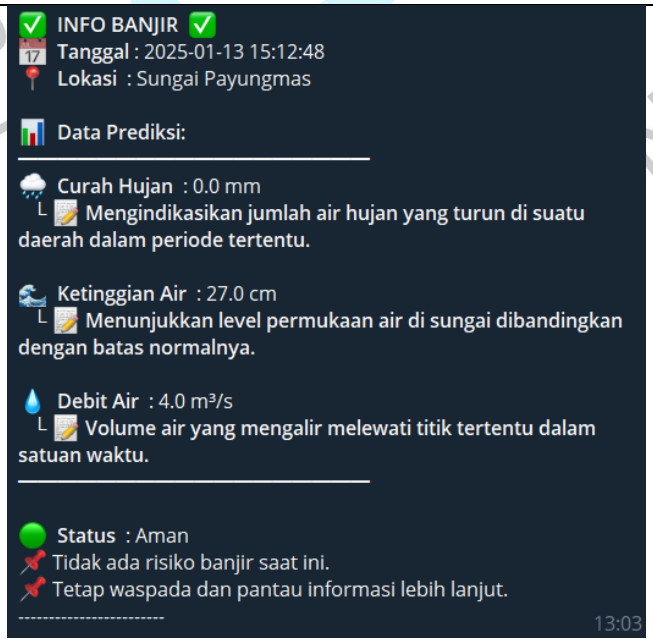
←T→		ID	Sensor_ID	Distance	Flow_Rate	Rainfall	Predicted_Condition	Created_at
☐	Edit Copy Delete	104	323	27.3	3.3	1.4	Aman	2024-12-27 11:32:20
☐	Edit Copy Delete	105	324	20.03	3.3	2.2	Aman	2024-12-27 11:34:45
☐	Edit Copy Delete	107	326	17.76	3.1	2.6	Aman	2024-12-27 11:35:33
☐	Edit Copy Delete	108	325	10.7	3.8	3.3	Aman	2024-12-27 11:36:15
☐	Edit Copy Delete	109	327	9.5	2.1	0	Aman	2024-12-27 11:36:51
☐	Edit Copy Delete	110	328	6.8	4.1	3.9	Waspada	2024-12-27 11:37:09
☐	Edit Copy Delete	111	329	7.84	2.1	8.9	Waspada	2024-12-27 11:38:21
☐	Edit Copy Delete	112	330	8.2	5.1	4.6	Waspada	2024-12-27 11:39:28
☐	Edit Copy Delete	113	331	8.2	8.6	12.4	Waspada	2024-12-27 11:39:52

Pengujian Black Box ke-9

Skenario Pengujian	API telegram mengirimkan hasil prediksi ke Telegram
Hasil yang diharapkan	API telegram mengirim pesan prediksi ke Telegram yang mencakup waktu, hasil prediksi, dan detail data sensor.
Hasil Pengujian	

```
Prediction has not changed. Skipping notification.
New prediction sent to all users.
Prediction has not changed. Skipping notification.
New prediction sent to all users.
Prediction has not changed. Skipping notification.
```

Pengujian Black Box ke-10

Skenario Pengujian	Aplikasi Telegram menampilkan hasil prediksi ke user
Hasil yang diharapkan	Aplikasi telegram berhasil menampilkan pesan prediksi ke user yang mencakup waktu, hasil prediksi, dan detail data sensor dengan format penulisan yang dirancang oleh peneliti.
Hasil Pengujian	

Hasil pengujian blackbox pada tabel menunjukkan bahwa setiap komponen dan proses yang ada dalam sistem telah berjalan selaras dengan skenario yang dirancang. Berikut adalah hasil pengujian blackbox sistem prediksi banjir:

1. Sensor tipping bucket berhasil mendeteksi curah hujan dan mengirimkan data ke ESP32.
2. Sensor ultrasonic dan aliran air juga berhasil mengukur jarak dan laju aliran air dengan akurasi sesuai.
3. ESP32 berhasil terhubung ke jaringan Wi-Fi dan mendapatkan alamat IP yang valid.
4. Data dari ESP32 berhasil disimpan di tabel sensor_data dengan format kolom yang sesuai.
5. Model berhasil mengambil data dari tabel sensor_data dan memprosesnya untuk menghasilkan prediksi kondisi (Aman, Waspada, Bahaya) lali menyimpan pada tabel hasil_prediksi.
6. Aplikasi Telegram berhasil menampilkan hasil prediksi kepada pengguna dalam format pesan yang dibuat.

Berdasarkan hasil pengujian blackbox yang telah dilakukan, terdapat beberapa kelemahan yang dapat diidentifikasi dalam sistem prediksi banjir:

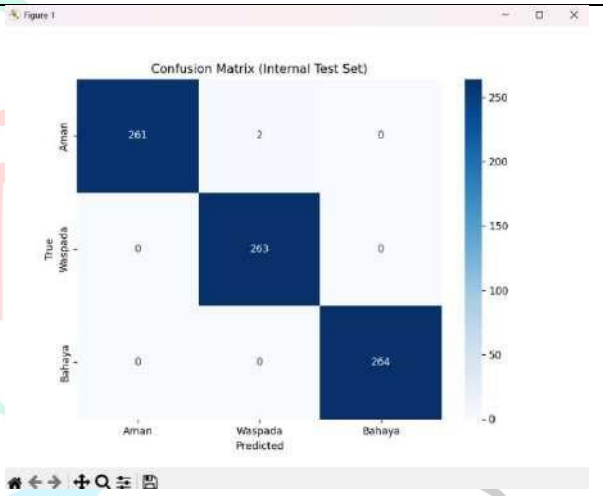
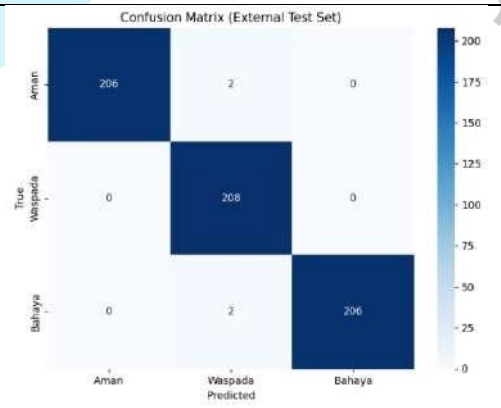
1. Ketergantungan pada Koneksi Internet: Sistem bergantung pada koneksi Wi-Fi untuk mengirimkan data dari ESP32 ke server. Jika terjadi gangguan jaringan atau koneksi internet lemah, data dapat mengalami keterlambatan atau bahkan gagal dikirim.
2. Akurasi Sensor dalam Kondisi Ekstrem: Performa sensor dapat dipengaruhi oleh kondisi ekstrem seperti hujan deras terus-menerus, turbulensi air yang tinggi, atau penghalang fisik yang mengganggu pembacaan sensor ultrasonic.
3. Kemungkinan Noise pada Data Sensor: Data yang diterima dari sensor dapat mengalami noise akibat gangguan lingkungan, seperti angin kencang atau perubahan arus yang tiba-tiba pada sensor aliran air.
4. Keterbatasan Penyimpanan Data pada ESP32: ESP32 memiliki keterbatasan dalam kapasitas penyimpanan dan pemrosesan data. Jika data

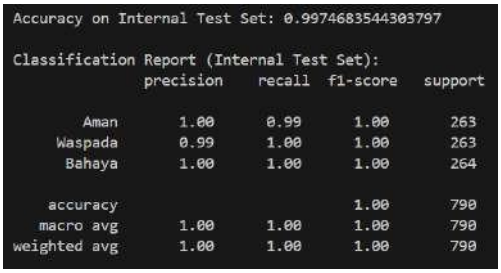
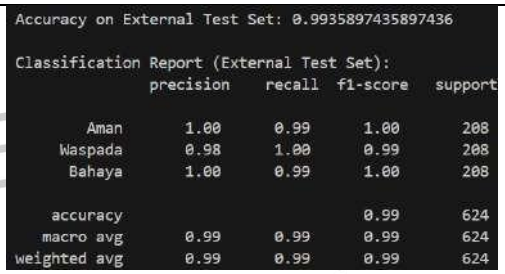
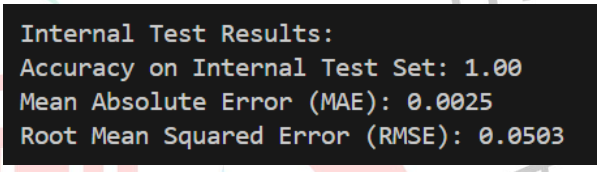
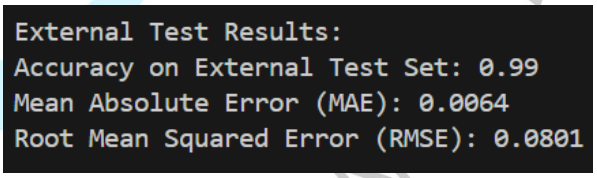
tidak segera dikirim ke server dalam jumlah besar, ada kemungkinan buffer penuh.

5.2.3 Hasil Pengujian Performa Model

Pengujian blackbox pada tabel 5.9 merupakan hasil pengujian pada model yang digunakan dalam sistem prediksi banjir. Hasil pengujian ini menunjukkan bahwa integrasi antara kode sumber berjalan sesuai dengan alur yang telah dirancang. Pengujian ini dilakukan pada seluruh aspek yang dibutuhkan oleh model agar bekerja secara optimal.

Tabel 5.10 Hasil Pengujian Black Box Model

No.	Metode	Pengukuran	Hasil Pengujian
1.	Confussion Matrix Data Latih	Akurasi Prediksi Model pada data latih	
2.	Confussion Matrix Data Uji	Akurasi Prediksi Model data uji	

3.	Confussion Matrix Data Latih Classification Report	Akurasi, Precision, Recall, F1, dan Supoort Data Latih	
4.	Confussion Matrix Data Uji Classification Report	Akurasi, Precision, Recall, F1, dan Supoort Data Uji	
5.	Mean Absolute Error (MAE) & Root Mean Squared Error (RMSE) data latih	Akurasi Error dan Akurasi Error Rata-rata Data Latih	
6.	Mean Absolute Error (MAE) & Root Mean Squared Error (RMSE) Data uji	Akurasi Error dan Akurasi Error Rata-rata Data Uji	

Hasil pengujian blackbox pada model prediksi menunjukkan bahwa model telah berfungsi dengan baik dalam melakukan prediksi kondisi dengan tingkat akurasi yang tinggi. Berdasarkan pengukuran menggunakan data latih sebanyak 3.961 data dan data uji sebanyak 1.698 data, diperoleh hasil sebagai berikut:

1. Akurasi Internal: Model berhasil mencapai akurasi sebesar 99.7% pada data latih, dengan nilai error (MAE) yang sangat rendah sebesar 0.0025 dan

RMSE sebesar 0.0503. Hasil ini menunjukkan bahwa model yang peneliti bangun mampu memahami pola data latih dengan sangat baik.

2. Akurasi Eksternal: Pada data uji eksternal. model mencatat akurasi sebesar 99.3%, dengan nilai MAE sebesar 0.0064 dan RMSE sebesar 0.0801. Hasil ini menunjukkan bahwa model mampu mempertahankan performa yang sangat baik meskipun diuji pada data baru yang belum pernah dilihat sebelumnya.
3. Confusion Matrix dan Classification Report: Hasil confusion matrix dan classification report mengindikasikan bahwa model mempunyai performa konsisten dalam memprediksi seluruh kategori (Aman, Waspada, Bahaya), dengan nilai precision, recall, dan F1-score yang sangat tinggi di semua kelas.

Berdasarkan hasil pengujian model, diperoleh juga beberapa evaluasi terkait kelemahan sistem yan kedepannya perlu diperhatikan, diantaranya sebagai berikut:

1. Peningkatan MAE dan RMSE Setelah Marginisasi: terdapat sedikit peningkatan dalam nilai error yang perlu diperhatikan dalam penggunaan lebih lanjut.
2. Keterbatasan Pada Data Ekstrem: Model mungkin memiliki kesulitan dalam menangani data dengan karakteristik yang jauh berbeda dari pola yang telah dipelajari, seperti anomali curah hujan yang sangat tinggi atau kondisi sungai yang tidak biasa.