

BAB IV

PERANCANGAN

Bagian ini menjelaskan mengenai rencana penelitian yang akan dilakukan oleh peneliti. Dalam pengembangan sistem, diperlukan tahapan yang sistematis untuk memastikan sistem yang dikembangkan mampu beroperasi dengan baik dan optimal. Berikut ini adalah tahapan perancangan yang dilakukan oleh peneliti.

4.1 Analisis Penelitian Terdahulu

Peneliti melakukan analisis terhadap penelitian sebelumnya untuk mengevaluasi sistem yang telah ada dengan tujuan mengidentifikasi aspek-aspek seperti keunggulan, kelemahan, dan kebutuhan perubahan sistem. Proses analisis ini dilakukan melalui berbagai metode, seperti studi literatur dan analisis dokumen. Hasil dari analisis ini berupa kesimpulan mengenai sistem yang sudah ada, yang selanjutnya dimanfaatkan untuk memperbaiki atau mengembangkan sistem guna mengatasi kelemahan yang ditemukan pada sistem sebelumnya.

Penelitian yang dilakukan oleh Rudi Hermawan, Dewanto Rosian Adhy, Arip, Siti Maesaroh, dan Akpil Mauhib, dengan judul “Pemanfaatan Sensor Curah Hujan Dan Debit Air Sungai Untuk Monitoring Banjir Berbasis Internet Of Things” menggunakan beberapa alat utama untuk mendukung pengumpulan dan pengolahan data. Dalam penelitian ini, Arduino Uno berfungsi sebagai mikrokontroler utama yang membaca dan mengolah data dari sensor debit air sebelum mengirimkannya ke ESP8266 melalui komunikasi serial. Selanjutnya, ESP8266 berfungsi sebagai penghubung ke internet. Alat ini akan menerima data dari Arduino Uno dan sensor curah hujan, lalu mengirimkannya ke server untuk disimpan di database. Sensor Debit Air (Water Flow Sensor HF-S201) digunakan untuk mengukur debit air secara real-time, sementara Sensor Curah Hujan (Rainfall Tipping Bucket) akan mencatat intensitas hujan menggunakan sistem jungkat-jungkit. Semua data yang dikumpulkan dari kedua sensor dikirim ke server melalui ESP8266 untuk disimpan dan diakses melalui API. Selanjutnya, data yang tersimpan divisualisasikan pada dashboard berbasis website, yang menampilkan informasi secara real-time mengenai debit air, curah hujan, dan kondisi ketinggian

air, serta menyediakan fitur historis yang berguna untuk analisis lebih lanjut dan prediksi banjir.

Sistem ini bekerja melalui beberapa tahap utama. Pertama, pengumpulan data dilakukan oleh Water Flow Sensor HF-S201 untuk mengukur debit air sungai, dan Rainfall Tipping Bucket digunakan untuk mencatat intensitas curah hujan. Selanjutnya, data dari kedua sensor diterima oleh Arduino Uno dan ESP8266. Arduino Uno akan memproses data debit air dan mengirimkannya ke ESP8266 melalui komunikasi serial, sedangkan data curah hujan akan langsung diproses oleh ESP8266. Setelah itu, ESP8266 akan mengirimkan data debit air dan curah hujan ke server melalui koneksi internet agar dapat disimpan dalam database. Kemudian, data yang tersimpan akan diolah dan ditampilkan melalui dashboard berbasis website secara real-time, meliputi informasi curah hujan per menit, debit air per detik, serta kondisi ketinggian air, dengan tambahan fitur historis untuk mendukung analisis dan prediksi banjir.

Penggunaan Rainfall Tipping Bucket dan Water Flow Sensor HF-S201 menjadi dasar dari peneliti menggunakan alat tersebut. Sensor Tipping Bucket bekerja dengan akurasi tinggi melalui mekanisme jungkat-jungkit untuk mencatat intensitas curah hujan secara real-time, dan dapat memberikan data yang relevan untuk memantau pola hujan. Di sisi lain, Sensor Debit Air HF-S201 mampu mengukur aliran air dengan tingkat akurasi hingga 99%, sehingga dapat mendeteksi perubahan debit air sungai yang signifikan. Kombinasi kedua sensor ini pastinya akan memungkinkan pengumpulan data yang saling melengkapi, menyediakan informasi tentang cuaca dari curah hujan dan kondisi air dari debit sungai.

4.2 Spesifikasi Kebutuhan Sistem Baru

Sistem Prediksi Banjir Real-Time Berbasis IoT menggunakan algoritma SVM merupakan sistem yang dikembangkan untuk memperkirakan potensi banjir di daerah aliran sungai. Sistem ini memanfaatkan berbagai perangkat dan sensor serta integrasi dengan algoritma yang memungkinkan untuk mengukur dan mengolah parameter penting dalam melakukan prediksi banjir. Untuk membangun sistem ini, diperlukan perangkat keras yang sesuai, khususnya mikrokontroler yang

mendukung tujuan penelitian. Spesifikasi perangkat keras tersebut akan dijelaskan lebih lanjut pada subbab berikut.

4.2.1 Spesifikasi Kebutuhan Perangkat Keras

Spesifikasi kebutuhan perangkat keras yang digunakan oleh peneliti dalam pengembangan Sistem Prediksi Banjir Real-Time Berbasis Iot menggunakan SVM dapat dilihat pada tabel berikut.

Tabel 4.1 Spesifikasi Kebutuhan Perangkat Keras

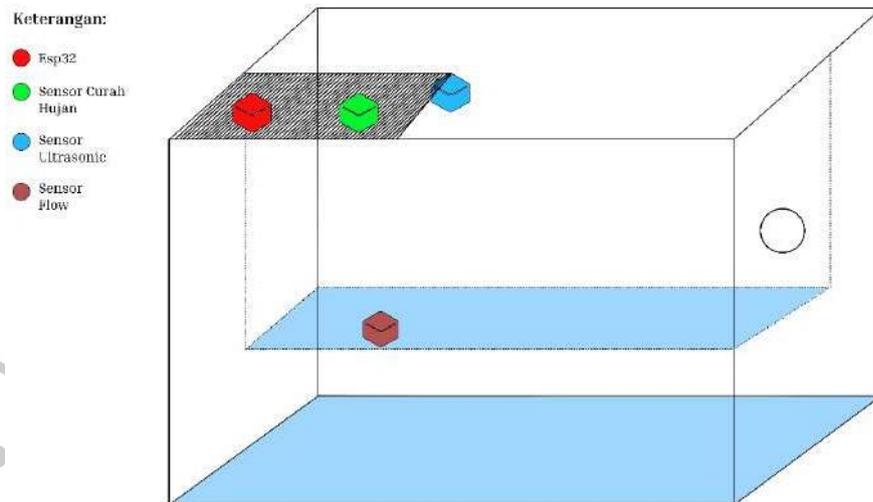
No.	Nama Perangkat	Jumlah	Kebutuhan
1.	ESP32	1	Microcontroller utama yang digunakan untuk menerima data sensor dan mengirimkan ke server dengan jaringan WI-FI.
2.	Sensor Ultrasonic	1	Perangkat yang digunakan untuk membaca nilai ketinggian dari air sungai.
3.	Sensor Curah Hujan (Tipping Bucket)	1	Perangkat yang digunakan untuk mengukur intensitas curah hujan pada saat kondisi cuaca hujan.
4.	Sensor Flow	1	Perangkat yang digunakan untuk membaca nilai debit air yang mengalir pada sungai.
5.	Pompa Air	1	Perangkat yang digunakan untuk mengontrol sirkulasi atau pengaliran air untuk simulasi pengelolaan banjir.

4.3 Perancangan Sistem

Perancangan sistem mencakup serangkaian tahapan yang dirancang untuk menjelaskan struktur sistem yang akan dikembangkan dan cara kerjanya secara

rinci. Proses perancangan sistem diawali dengan analisis prinsip kerja, yang bertujuan untuk mengidentifikasi alur kerja sistem secara mendalam. Penjelasan mengenai alur pengembangan sistem disajikan sebagai berikut.

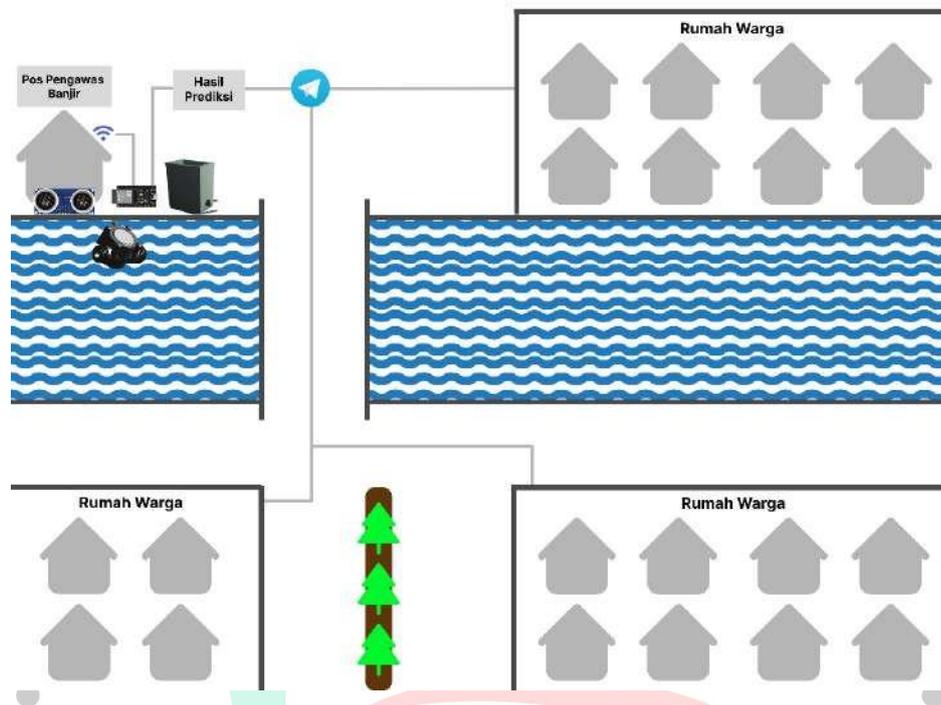
4.3.1 Perancangan Fisik Sistem



Gambar 4.1 Perancangan Fisik Sistem

Gambar 4.1 menampilkan rancangan fisik sistem dari sistem prediksi banjir berbasis IoT, yang dibuat dalam lingkungan buatan menggunakan akrilik berbentuk persegi panjang. Lingkungan ini dirancang untuk meniru kondisi asli Sungai Payung Mas, yang menjadi lokasi penelitian. Dimensi asli sungai adalah panjang 1 km (1.000 m), lebar 2 m, dan tinggi 2,7 m. Untuk keperluan pengujian, dimensi tersebut diskalakan menjadi panjang 50 cm, lebar 20 cm, dan tinggi 27 cm. Dengan demikian, lingkungan buatan ini menggunakan skala 1:2000 untuk panjang, 1:100 untuk lebar, dan 1:100 untuk tinggi. Sensor Ultrasonic dipasang di bagian atas untuk mengukur ketinggian air. Sensor Flow diposisikan sejajar dengan arah aliran air untuk membaca debit air. Sensor Curah Hujan diletakkan di area terbuka pada bagian atas lingkungan untuk menangkap intensitas curah hujan secara langsung. Semua sensor dihubungkan ke ESP32 sebagai microcontroller.

4.3.1 Perancangan Instalasi

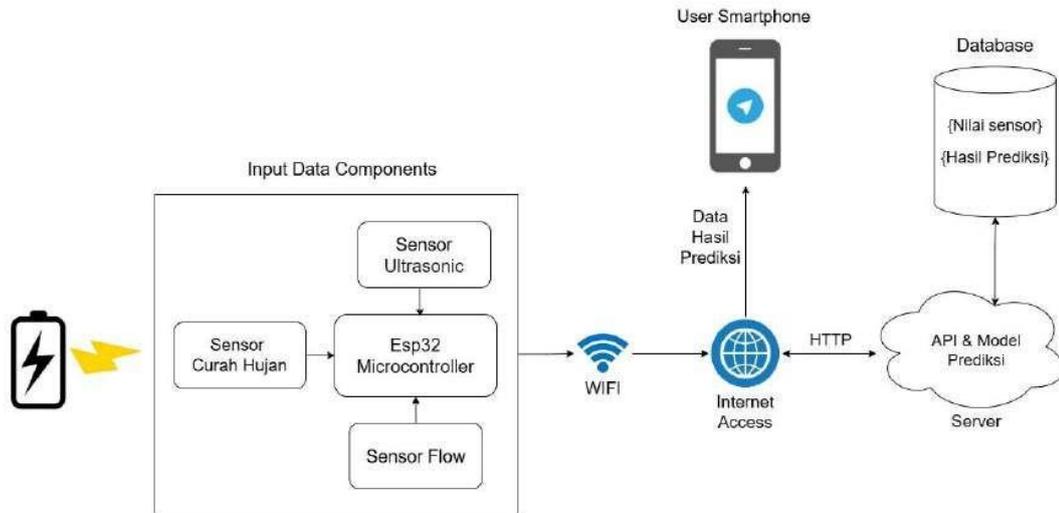


Gambar 4.2 Perancangan Instalasi

Rancangan instalasi pada Gambar 4.2 menjelaskan cara kerja sistem prediksi banjir berbasis IoT yang terhubung langsung di lapangan, yaitu sungai yang terletak pada perumahan Pondok Payung Mas, Ciputat. Pos Pengawas Banjir menjadi pusat kontrol, yang dilengkapi dengan perangkat seperti ESP32 dan sensor (Ultrasonic, Flow, dan Curah Hujan). Sensor-sensor yang dipasang di sungai secara otomatis menangkap parameter penting seperti ketinggian air, debit aliran, dan intensitas curah hujan secara real-time. Selanjutnya, data yang terkumpul diproses dengan algoritma prediksi SVM, dan hasilnya akan dikirim melalui internet ke Bot Telegram untuk memberikan notifikasi peringatan dini kepada warga di daerah sekitar.

4.3.2 Diagram Blok IoT

Gambar di bawah ini merupakan diagram blok dari sistem yang dikembangkan oleh peneliti. Gambar ini meliputi semua komponen yang berperan dalam menjalankan proses pada system prediksi banjir berbasis IoT yang dikembangkan oleh peneliti.



Gambar 4.3 Diagram Blok IoT

Diagram blok dirancang untuk menggambarkan struktur komponen yang mendukung pengembangan sistem. Arsitektur ini menjelaskan bagaimana setiap komponen saling terhubung secara menyeluruh agar dapat menjalankan proses kerja sesuai tujuan yang diinginkan. Gambar 4.3 menampilkan diagram blok sistem prediksi banjir yang dikembangkan oleh peneliti. Proses dimulai dari sensor ultrasonic, sensor flow, dan sensor curah hujan, yang menangkap data lingkungan. Data tersebut dikirim ke ESP32 Microcontroller untuk dikumpulkan dan diproses awal. ESP32 mengirimkan data ke server melalui Wi-Fi yang terhubung ke Internet Access. Di server, data diterima oleh API dan disimpan ke dalam database. Model prediksi membaca data dari database untuk menganalisis dan memprediksi kondisi banjir. Hasil prediksi disimpan kembali ke database. Tahap terakhir adalah pengiriman hasil prediksi kepada pengguna melalui aplikasi Telegram. Proses ini dilakukan melalui Internet Access, sehingga pengguna dapat menerima informasi prediksi secara real-time.

4.3.3 Perancangan Pin Sistem

Peneliti menggunakan skema perancangan pin sebagai panduan untuk melihat keterkaitan antara mikrokontroler dengan masing-masing sensor yang akan diintegrasikan ke dalam sistem prediksi banjir real-time berbasis IoT. Tabel 4.2 menyajikan skema perancangan pin yang digunakan dalam penelitian ini.

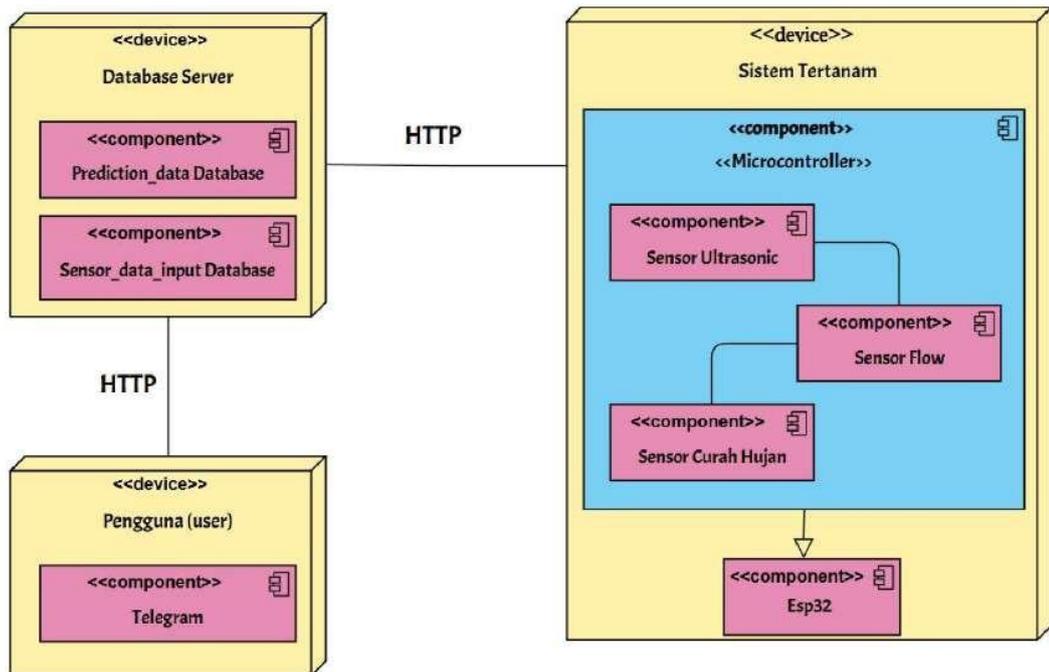
Tabel 4.2 Rancangan Pin ESP32

No.	Pin ESP32 (GPIO)	Kebutuhan
1.	GPIO 5	Pin Trig Sensor Ultrasonic
2.	GPIO 18	Pin Echo Sensor Ultrasonic
3.	GPIO 23	Pin Data Sensor Curah Hujan Tipping Bucket
4.	GPIO 19	Pin Data Sensor Flow
5.	GND	Ground untuk semua perangkat
6.	5V	Power untuk perangkat seperti sensor

Tabel 4.2 ini menggambarkan rancangan konfigurasi pin GPIO pada ESP32 untuk menghubungkan sensor-sensor dalam sistem prediksi banjir berbasis IoT. Pin GPIO 5 dan GPIO 18 digunakan untuk mengontrol trigger dan echo dari Sensor Ultrasonic. Pin GPIO 23 menghubungkan data dari Sensor Curah Hujan Tipping Bucket, dan Pin GPIO 19 untuk data dari Sensor Flow. Pin GND berfungsi sebagai ground untuk semua perangkat, sedangkan 5V menyediakan daya bagi sensor-sensor yang terhubung.

4.3.4 Deployment Diagram

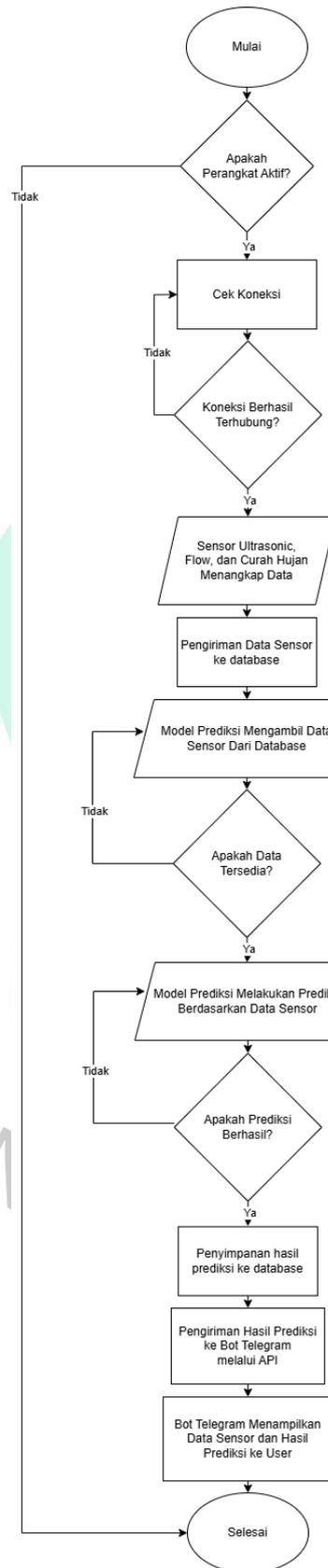
Deployment diagram digunakan untuk menggambarkan bagaimana komponen sistem diimplementasikan pada perangkat keras. Diagram ini digunakan untuk memperlihatkan lokasi fisik setiap komponen, koneksi antar perangkat, serta hubungan antara perangkat keras dan perangkat lunak yang digunakan. Dalam sistem prediksi banjir berbasis IoT, deployment diagram diperlukan untuk memberikan gambaran proses kerja setiap komponen. Proses ini mencakup bagaimana sensor dan mikrokontroler terhubung ke IP gateway, bagaimana IP gateway berinteraksi dengan backend, hingga bagaimana hasil prediksi yang tervalidasi disimpan di database dan dikirimkan kepada pengguna.



Gambar 4.4 Deployment Diagram

Deployment diagram di atas menggambarkan bagaimana hubungan antar komponen dalam sistem prediksi banjir berbasis IoT yang dikembangkan. Sistem ini terdiri dari tiga komponen utama, yaitu Sistem Tertanam, Database Server, dan Pengguna (User). Pada Sistem Tertanam, mikrokontroler ESP32 digunakan sebagai pusat pengendali, yang dihubungkan dengan tiga sensor utama: Sensor Ultrasonic untuk mengukur ketinggian air, Sensor Flow untuk membaca debit air, dan Sensor Curah Hujan untuk mencatat intensitas hujan. Data yang dihasilkan oleh sensor-sensor ini diproses oleh ESP32 sebelum dikirimkan ke server melalui koneksi HTTP. Database Server dirancang untuk menyimpan data yang diterima dari ESP32. Server ini terdiri dari dua komponen utama, yaitu Prediction_data Database, yang digunakan untuk menyimpan hasil prediksi sistem, dan Sensor_data_input Database, yang berfungsi untuk menyimpan data mentah dari sensor. Hasil prediksi kemudian dikirimkan kepada Pengguna (User) melalui platform Telegram, yang digunakan untuk menerima notifikasi real-time terkait potensi banjir. Seluruh proses ini dilakukan dengan memanfaatkan jaringan HTTP sebagai penghubung antar komponen.

4.3.5 Diagram Alir Sistem



Gambar 4.5 Diagram Alir Sistem

Berdasarkan gambar diatas, sistem dimulai dengan pemeriksaan apakah perangkat aktif. Jika perangkat tidak aktif, maka sistem berhenti, tetapi jika perangkat aktif, sistem akan melanjutkan untuk memeriksa koneksi. Jika koneksi tidak berhasil, sistem akan mencoba kembali. Setelah koneksi berhasil, sensor ultrasonic, flow, dan curah hujan mulai mengumpulkan data real-time tentang ketinggian air, debit air, dan curah hujan. Data yang dihasilkan oleh sensor kemudian dikirimkan ke server melalui API untuk penyimpanan dan pemrosesan lebih lanjut. Setelah data tersedia di server, model prediksi akan mengambil data tersebut untuk dilakukan analisis. Model prediksi menggunakan algoritma SVM untuk memproses data dan menghasilkan prediksi banjir berdasarkan input yang diterima dari sensor. Hasil prediksi ini akan dikirimkan kembali ke server melalui API dan diteruskan ke bot Telegram untuk disampaikan ke pengguna. Bot Telegram menampilkan data sensor dan hasil prediksi kepada pengguna secara langsung, dengan memberi notifikasi tentang potensi banjir. Proses berakhir setelah data dikirimkan dan diterima oleh pengguna untuk kesiapsiagaan.

4.3.6 Perancangan Basis Data

Untuk menciptakan hubungan antar data yang saling berkorelasi, diperlukan perancangan basis data yang terstruktur dengan baik. Perancangan ini bertujuan untuk menyimpan data pendukung yang dihasilkan oleh sensor-sensor yang digunakan dalam penelitian, sehingga proses operasional sistem dapat berjalan dengan lancar.

Tabel 4.3 Tabel Sensor Data

No.	Field	Type	Key Type	Length	Keterangan
1.	ID	INT	PRIMARY	4	ID data auto increment
2.	Rainfall_mm	FLOAT	-	-	Curah hujan dalam mm
3.	Water_level_cm	FLOAT	-	-	Tinggi air dalam cm
4.	Water_flow_rate	FLOAT	-	-	Kecepatan aliran air
5.	created_at	TIMESTAMP	-	-	Waktu input data secara otomatis

Tabel ini mendeskripsikan struktur data yang digunakan untuk menyimpan informasi dari sensor dalam sistem prediksi banjir. Field ID sebagai primary key berfungsi untuk mencatat jumlah data yang telah tersimpan di dalam tabel. Rainfall_mm, Water_Level_cm, dan Water_Flow_rate masing-masing menyimpan data curah hujan, tinggi air, dan kecepatan aliran air dalam format float. Field created_at mencatat waktu input data secara otomatis menggunakan tipe timestamp, yang berguna untuk analisis berdasarkan waktu. Struktur ini dirancang untuk mendukung pengumpulan dan pengolahan data secara efisien.

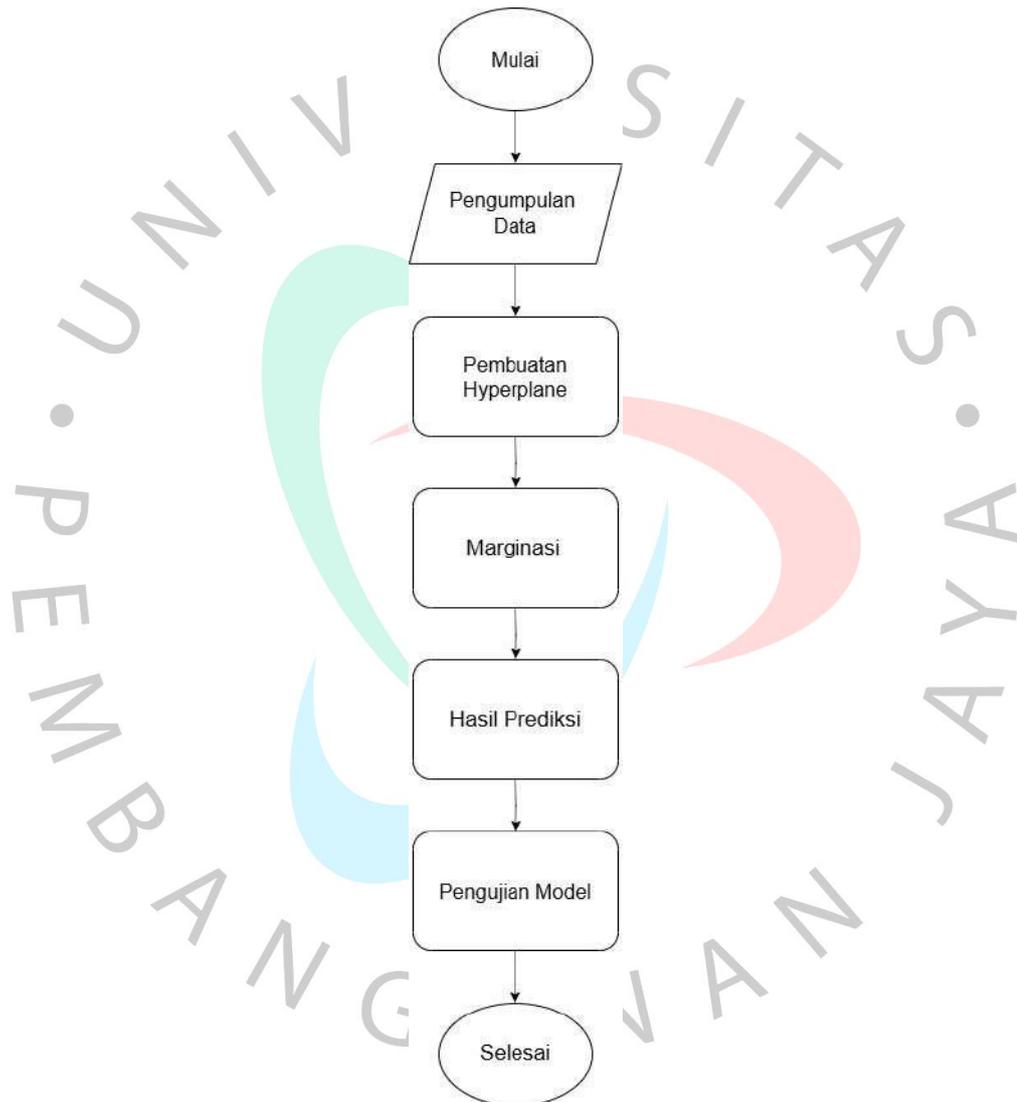
Tabel 4.4 Tabel Hasil Prediksi

No.	Field	Type	Key Type	Length	Keterangan
1.	ID	INT	PRIMARY	4	ID data auto increment
2.	Sensor_ID	INT	FOREIGN	-	ID sensor dari table sensor data
3.	Rainfall_mm	FLOAT	-	-	Curah hujan dalam mm
4.	Water_level_cm	FLOAT	-	-	Tinggi air dalam cm
5.	Water_flow_rate	FLOAT	-	-	Kecepatan air
6.	Predicted_Condition	VARCHAR	-	-	Hasil prediksi kondisi
7.	Created_at	TIMESTAMP	-	-	Waktu input data secara otomatis

Tabel ini digunakan untuk menyimpan hasil prediksi sistem berdasarkan data dari sensor. Field ID akan menjadi primary key, sementara Sensor ID mereferensikan ID dari tabel sebelumnya sebagai foreign key, yang digunakan untuk menghubungkan data prediksi dengan data sensor yang relevan. Data curah hujan (Rainfall_cm), tinggi air (Water_Level_cm), dan kecepatan aliran air (Water_Flow_rate) disimpan dalam format float. Field Predicted_Condition menyimpan hasil prediksi model SVM, seperti "Aman" atau "Waspada" atau "Bahaya". Selanjutnya, Created_at mencatat waktu input data secara otomatis.

4.4 Perancangan Algoritma *Support Vector Machine (SVM)*

Perancangan algoritma SVM digunakan peneliti dalam menjelaskan seluruh tahapan algoritma SVM dalam melakukan prediksi berdasarkan parameter yang digunakan. Parameter yang digunakan yaitu curah hujan, debit air, dan ketinggian air, yang didapat dari sensor-sensor sistem tertanam. Berikut merupakan tahapan algoritma SVM sehingga menghasilkan suatu prediksi yang valid untuk pengguna.



Gambar 4.6 Diagram Alir Perancangan Algoritma SVM

4.4.1 Pengumpulan Data

Pengumpulan data merupakan tahapan awal dalam proses prediksi banjir menggunakan algoritma Support Vector Machine (SVM). Dataset yang digunakan dalam penelitian ini diperoleh dari periode pengukuran 1 September – 31 Desember

2024, dengan sumber data yang berbeda berdasarkan jenis parameter yang diukur. Data curah hujan diperoleh dari BMKG, dengan pencatatan yang dilakukan berdasarkan periode bulanan, sedangkan data ketinggian air dan debit air dikumpulkan melalui pengukuran langsung di Sungai Payungmas menggunakan sensor yang terhubung ke sistem tertanam. Data yang dikumpulkan dikirimkan oleh ESP32 ke server dan disimpan dalam database untuk pemrosesan lebih lanjut. Dataset yang berhasil dikumpulkan berjumlah 3.550 data, yang mencakup parameter curah hujan, ketinggian air, dan debit air. Untuk memastikan model dapat belajar secara optimal, dataset ini dibagi menjadi 70% sebagai data latih, yang digunakan dalam proses pelatihan model, dan 30% sebagai data uji, yang digunakan untuk mengevaluasi performa model dalam mengklasifikasikan prediksi banjir.

Tabel 4.5 Contoh Kelas Prediksi dan Threshold

No.	Kondisi	Ketinggian Air (cm)	Laju Air (mm/s)	Curah Hujan (mm/h)	Keterangan
1.	Aman	> 18 cm	0.4 mm/s	0 - 8 mm/h	Kondisi normal tidak ada risiko banjir
2.	Waspada	10 – 18 cm	5 - 8 mm/s	9 - 16 mm/h	Tingkat kewaspadaan meningkat, perhatikan curah hujan dan laju air.
3.	Bahaya	< 10 cm	> 9 mm/s	> 16 mm/h	Risiko banjir tinggi, Langkah darurat perlu dilakukan.

Pada Tabel 4.5 ditampilkan kondisi sungai yang ditentukan berdasarkan threshold dari parameter utama, yaitu ketinggian air, laju air, dan curah hujan. Kondisi ini terbagi menjadi tiga kategori: Aman, Waspada, dan Bahaya, yang masing-masing memiliki batasan nilai tertentu. Kondisi Aman menunjukkan situasi normal tanpa risiko banjir, sementara Waspada menandakan peningkatan kewaspadaan terhadap potensi banjir. Kondisi Bahaya menggambarkan risiko banjir tinggi yang memerlukan tindakan darurat.

Tabel 4.6 Contoh Fitur dan Kelas Prediksi

No.	Rainfall (mm)	Water Level (mm)	Water Flow Rate	Condition
1.	8.78	14.312	1.174	Aman
2.	17.816	9.444	6.002	Waspada
3.	21.834	4.593	13.898	Bahaya

Tabel 4.6 mencakup penjabaran data yang diperoleh dari pembacaan sensor-sensor pada sistem yang dirancang untuk memantau parameter curah hujan, ketinggian air, dan debit air. Baris pertama menggambarkan kondisi di mana curah hujan tercatat sebesar 8.78 mm, ketinggian air mencapai 14.312 cm, dan debit air berada pada 1.174 m³/s. Berdasarkan parameter-parameter tersebut, kondisi dikategorikan sebagai Aman, yang menunjukkan bahwa tidak ada potensi ancaman banjir pada saat data tersebut diambil. Baris kedua memberikan contoh pembacaan sensor dengan curah hujan sebesar 17.816 mm, ketinggian air 9.444 cm, dan debit air 6.002 m³/s. Data ini menghasilkan kondisi yang dikategorikan sebagai Waspada, mengindikasikan bahwa meskipun banjir belum terjadi, terdapat potensi ancaman yang perlu diantisipasi. Selanjutnya, baris ketiga menunjukkan kondisi curah hujan sebesar 21.834 mm, ketinggian air 4.593 cm, dan debit air 13.898 m³/s, yang diklasifikasikan sebagai Bahaya. Kategori ini menunjukkan adanya risiko tinggi terjadinya banjir berdasarkan parameter yang tercatat.

4.4.2 SMOTE Oversampling

SMOTE (Synthetic Minority Oversampling Technique) digunakan untuk menyeimbangkan distribusi data dalam proses pelatihan model prediksi banjir. Teknik ini diterapkan karena adanya ketidakseimbangan jumlah data antar kategori dalam dataset yang dikumpulkan. Secara khusus, parameter ketinggian air dan debit air cenderung lebih sulit diperoleh dalam jumlah besar dibandingkan curah hujan, yang lebih mudah diakses melalui sumber resmi seperti BMKG. Ketidakseimbangan data ini dapat berdampak pada performa model, terutama dalam memprediksi kategori yang memiliki jumlah data lebih sedikit, seperti kategori Bahaya dan Waspada. Sebelumnya, total 3.550 data telah berhasil dikumpulkan selama periode 1 September – 31 Desember 2024, yang mencakup parameter utama seperti curah hujan, ketinggian air, dan debit air. Namun, jumlah

data pada kategori Bahaya dan Waspada masih jauh lebih sedikit dibandingkan kategori Aman, sehingga diperlukan teknik oversampling untuk meningkatkan representasi data dalam pelatihan model. Dengan menerapkan SMOTE, dataset mengalami peningkatan menjadi 5.659 data, yang terdiri dari distribusi yang lebih seimbang di antara ketiga kategori. Teknik ini bekerja dengan menghasilkan data sintetis berdasarkan sampel minoritas, sehingga tidak hanya meningkatkan jumlah data tetapi juga mempertahankan pola distribusi yang relevan dalam proses pelatihan model. Data yang telah diseimbangkan ini akan dibagi untuk proses pelatihan dan pengujian, dengan 70% (3.961 data) digunakan sebagai data latih dan 30% (1.698 data) sebagai data uji.

Tabel 4.7 Distribusi Dataset SMOTE

Kondisi	Sebelum SMOTE	Setelah SMOTE
Aman	1740 Data	2.500 Data
Waspada	1.050 Data	1.900 Data
Bahaya	760 Data	1.259 Data
Total	3.550 Data	5.659 data

Pada tabel 4.7 menjelaskan penyebaran dataset setelah menggunakan teknik SMOTE terlihat bahwa jumlah data untuk kategori Waspada dan Bahaya mengalami peningkatan signifikan setelah penerapan SMOTE. Hal ini bertujuan untuk memastikan bahwa model SVM dapat belajar dari distribusi data yang lebih seimbang, sehingga meningkatkan kemampuan model. Berikut merupakan langkah SMOTE dalam membuat sampel baru:

1. Memilih sampel acak dari data minoritas.
2. Mengidentifikasi sampel terdekat (k-nearest neighbors).
3. Menginterpolasi nilai-nilai fitur antara sampel terpilih dan tetangga terdekatnya untuk menghasilkan data baru.
4. Menghasilkan data baru dengan jumlah sebanding atau mendekati data mayoritas.

4.4.3 Pembuatan Hyperplane

Hyperplane adalah konsep penting dalam metode seperti Support Vector Machines (SVM) atau klasifikasi berbasis ruang dimensi tinggi. Secara sederhana,

hyperplane adalah sebuah bidang (atau garis dalam dimensi 2D) yang membagi ruang fitur menjadi beberapa bagian berdasarkan kelas-kelas data. Rumus SVM dapat dijabarkan sebagai berikut.

Deskripsi: Data baru dimasukkan ke fungsi hyperplane untuk menghitung hasil fungsi keputusan.

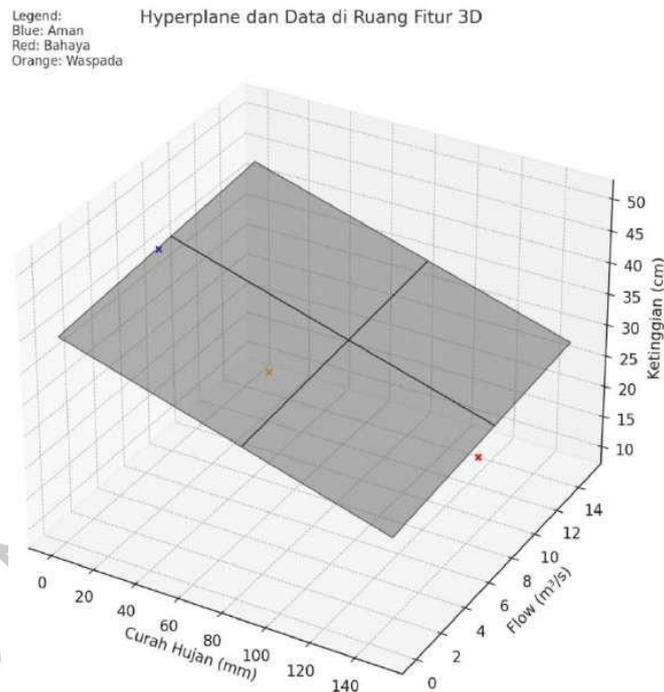
$$f(x) = w \cdot x + b$$

w : Vektor bobot dari model SVM (dihasilkan saat pelatihan).

x : Data baru yang masuk (misalnya: [80, 8.5, 20]).

b : Bias (offset dari hyperplane).

Rumus tersebut digunakan untuk menentukan posisi sebuah titik data baru (x) relatif terhadap hyperplane dalam model SVM. Di sini, w adalah vektor bobot yang menentukan orientasi hyperplane, w dan b adalah bias yang menentukan posisi hyperplane terhadap asal ruang koordinat. Selama pelatihan, w dan b dihitung untuk memaksimalkan margin antara dua kelas data. Nilai $f(x)$ menunjukkan sisi mana dari hyperplane data tersebut berada: jika $f(x) > 0$, maka data berada di satu sisi (aman), sedangkan jika $f(x) < 0$, data berada di sisi lain (bahaya). Jika $f(x) = 0$, data tepat atau berada hyperplane (waspada). Berikut merupakan gambaran dari tahapan pembuatan hyperplane dari sistem prediksi banjir yang dikembangkan oleh peneliti.



Gambar 4.7 Hyperplane

Gambar 4.7 menggambarkan hyperplane yang memisahkan data dalam ruang fitur tiga dimensi, yang terdiri dari parameter Curah Hujan (mm), Flow (m^3/s), dan Ketinggian (mm). Hyperplane ditampilkan sebagai bidang abu-abu, yang bertindak sebagai batas pemisah untuk menentukan kategori data, berdasarkan parameter yang dimiliki. Dalam gambar, data diklasifikasikan ke dalam tiga kategori: Aman (biru), Bahaya (merah), dan Waspada (oranye). Setiap kategori ditentukan berdasarkan posisi relatif data terhadap hyperplane—apakah data berada di atas, di bawah, atau pada bidang hyperplane. Parameter Curah Hujan merepresentasikan intensitas hujan, Flow menunjukkan kecepatan aliran air, dan Ketinggian mengukur elevasi lokasi, semuanya berkontribusi pada pengambilan keputusan model.

4.4.4 Marginasi

Setelah pembuatan hyperplane dalam ruang fitur, langkah selanjutnya adalah menghitung margin. Margin adalah jarak paling dekat antara hyperplane dan titik data dari setiap kelas. Margin dihitung untuk memastikan bahwa hyperplane memiliki pemisahan maksimal antara kelas-kelas yang berbeda, sehingga menghasilkan klasifikasi yang lebih baik.

Margin dihitung dengan jarak dari data ke hyperplane, dinormalisasi dengan bobot hyperplane ($\|w\|$):

$$\text{Margin} = \frac{|f(x)|}{\|w\|}$$

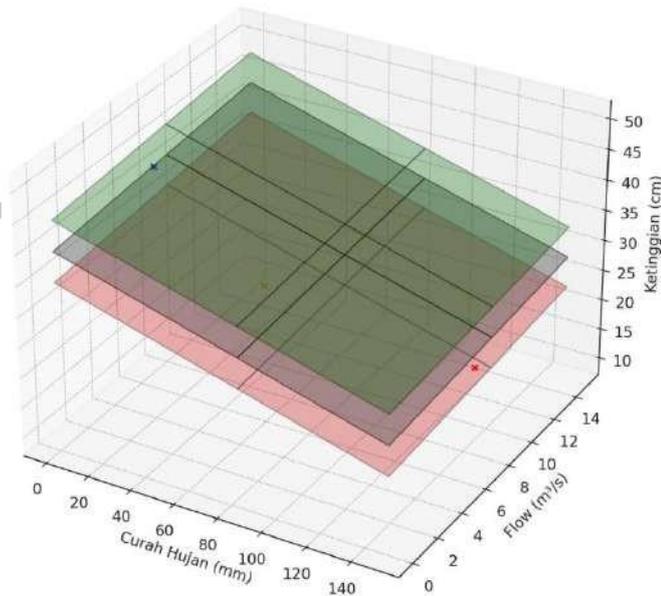
$|f(x)|$: Hasil fungsi keputusan (nilai absolut memastikan jarak).

$\|w\|$: Panjang (norma) vektor bobot:

$$\|w\| = \sqrt{w_1^2 + w_2^2 + w_3^2}$$

Rumus margin digunakan untuk menghitung jarak normalisasi antara titik data tertentu dan hyperplane dalam ruang fitur. Komponen $f(x)$ merepresentasikan hasil dari fungsi hyperplane sebelumnya. Nilai $f(x)$ menunjukkan posisi relatif data terhadap hyperplane, sedangkan nilai absolut $|f(x)|$ memastikan jarak dihitung tanpa memperhatikan arah titik terhadap hyperplane. Bagian denominator, $\|w\|$, adalah panjang (norma) vektor bobot w , yang dihitung menggunakan rumus Euclidean. Norma ini berfungsi untuk mengukur bobot secara keseluruhan dan digunakan untuk menormalkan jarak $|f(x)|$, sehingga margin menjadi ukuran yang konsisten.

Penentuan Margin dan Data Baru



Gambar 4.8 Marginasi

Gambar 4.8 menjelaskan terkait margin hyperplane pada ruang fitur 3D, yang digunakan untuk memisahkan kategori Aman, Waspada, dan Bahaya berdasarkan parameter Curah Hujan, Ketinggian Air, dan Debit Air. Hyperplane berfungsi sebagai batas keputusan, sedangkan margin, yang diwakili oleh area berwarna hijau dan merah, menunjukkan jarak terdekat antara hyperplane dengan titik data dari setiap kategori. Margin ini dirancang agar pemisahan antar kelas dapat dilakukan secara optimal. Titik-titik pada gambar menunjukkan data yang dikelompokkan berdasarkan atribut Curah Hujan, Ketinggian Air, dan Debit Air, dengan warna sesuai kategori. Posisi data dibandingkan dengan hyperplane untuk menentukan kelasnya, di mana hasil prediksi didasarkan pada nilai fungsi keputusan $f(x)$. Data baru akan dikelompokkan sesuai posisinya terhadap hyperplane dan margin, memastikan prediksi yang lebih akurat.

4.4.5 Perancangan Data Prediksi

Hasil prediksi diperoleh berdasarkan marginisasi dan hyperplane yang dilakukan dengan menentukan posisi data baru dalam ruang fitur relatif terhadap hyperplane dan margin. Bentuk data terkait hasil prediksi sistem yang dikembangkan oleh peneliti dapat dilihat pada tabel dibawah.

Tabel 4.8 Contoh Tabel Hasil Prediksi

No.	Rainfall (mm)	Water Level (cm)	Water Flow Rate	Condition	Predicted_Condition
1.	11.427	12.17	4.301	Aman	Aman
2.	16.976	9.677	9.963	Waspada	Waspada
3.	25.442	5.381	11.068	Bahaya	Bahaya
4.	15.352	9.147	9.299	Waspada	Waspada
5.	16.355	10.383	9.032	Waspada	Waspada
6.	17.661	10.637	9.081	Waspada	Waspada
7.	11.427	12.17	4.301	Aman	Aman
8.	26.604	5.358	10.993	Bahaya	Bahaya
9.	16.604	9.262	9.06	Waspada	Waspada
10.	15.617	9.244	8.824	Waspada	Waspada
11.	17.891	9.126	8.175	Waspada	Waspada

12	17.768	9.682	8.791	Waspada	Waspada
13	10.73	10.583	3.501	Aman	Aman
14	17.363	10.155	9.645	Waspada	Waspada
15	12.944	10.828	3.523	Aman	Aman
16	19.177	5.288	10.621	Bahaya	Bahaya
17	13.602	10.217	4.415	Aman	Aman
18	17.253	10.892	9.507	Waspada	Waspada
19	11.024	11.735	3.543	Aman	Aman
20	16.474	9.463	9.331	Waspada	Waspada
21	12.263	10.135	4.24	Aman	Aman
22	19.428	4.597	10.614	Bahaya	Bahaya
23	14.928	10.594	9.929	Waspada	Waspada
24	19.173	5.866	10.343	Bahaya	Bahaya
25	16.694	9.685	8.568	Waspada	Waspada
26	6.618	11.815	3.847	Aman	Aman
27	28.091	4.154	10.064	Bahaya	Bahaya
28	15.531	10.824	9.089	Waspada	Waspada
29	26.458	5.356	11.613	Bahaya	Bahaya
30	11.304	12.406	4.186	Aman	Aman
31	12.141	13.45	3.802	Aman	Aman
32	14.672	10.81	8.729	Waspada	Waspada
33	6.313	14.256	4.329	Aman	Aman
34	11.084	13.009	3.598	Aman	Aman
35	10.788	11.839	4.042	Aman	Aman
36	11.78	14.54	3.572	Aman	Aman
37	29.913	5.202	11.415	Bahaya	Bahaya
38	11.235	13.509	4.245	Aman	Aman
39	23.238	4.802	10.533	Bahaya	Bahaya
40	17.635	10.132	8.531	Waspada	Waspada
41	28.652	4.705	10.312	Bahaya	Bahaya
42	13.797	12.834	3.894	Aman	Aman
43	23.889	5.907	11.588	Bahaya	Bahaya
44	22.253	4.335	11.735	Bahaya	Bahaya
45	16.607	9.593	8.216	Waspada	Waspada

46	15.867	9.544	8.543	Waspada	Waspada
47	14.477	10.535	8.028	Waspada	Waspada
48	8.894	10.61	4.418	Aman	Aman
49	23.004	4.133	11.309	Bahaya	Bahaya
50	27.177	4.369	11.128	Bahaya	Bahaya
51	14.37,	13.538	3.908	Aman	Aman
52	10.295	11.047	3.622	Aman	Aman
53	29.175	4.126	11.273	Bahaya	Bahaya
54	19.094	5.582	10.133	Bahaya	Bahaya
55	6.984	13.486	3.989	Aman	Aman
56	16.138	9.504	8.238	Waspada	Waspada
57	11.797	10.357	3.996	Aman	Aman
58	29.951	5.368	11.178	Bahaya	Bahaya
59	24.327	5.258	11.664	Bahaya	Bahaya
60	19.981	4.528	11.672	Bahaya	Bahaya
61	17.936	10.558	8.23	Waspada	Waspada
62	16.352	9.718	9.96	Waspada	Waspada
63	15.406	10.002	9.699	Waspada	Waspada
64	5.542	12.018	3.864	Aman	Aman
65	25.211	5.734	11.976	Bahaya	Bahaya
66	17.025	9.038	8.759	Waspada	Waspada
67	25.105	5.63	10.773	Bahaya	Bahaya
68	14.807	10.99	8.822	Waspada	Waspada
69	26.061	5.69	10.74	Bahaya	Bahaya
70	14.325	10.097	9.375	Waspada	Waspada
71	21.283	5.814	10.481	Bahaya	Bahaya
72	8.214	10.63	4.108	Aman	Aman
73	14.878	10.677	9.322	Waspada	Waspada
74	12.528	13.133	4.006	Aman	Aman
75	28.377	5.756	11.069	Bahaya	Bahaya
76	15.253	10.691	8.134	Waspada	Waspada
77	14.556	9.175	9.387	Waspada	Waspada
78	20.432	4.444	10.793	Bahaya	Bahaya
79	9.284	14.815	3.856	Aman	Aman

80	21.904	5.532	10.652	Bahaya	Bahaya
81	16.785	10.955	9.357	Waspada	Waspada
82	5.126	14.286	4.21	Aman	Aman
83	22.293	4.449	11.634	Bahaya	Bahaya
84	27.784	5.555	10.222	Bahaya	Bahaya
85	19.322	4.067	10.446	Bahaya	Bahaya
86	9.786	10.623	3.638	Aman	Aman
87	7.594	10.753	3.788	Aman	Aman
88	7.46	11.831	3.862	Aman	Aman
89	15.292	9.858	9.185	Waspada	Waspada
90	5.76	10.542	3.685	Aman	Aman
91	13.074	14.57	3.743	Aman	Aman
92	15.593	9.446	9.403	Waspada	Waspada
93	6.552	10.599	3.802	Aman	Aman
94	5.435	9.929	8.914	Waspada	Waspada
95	14.292	9.119	9.575	Waspada	Waspada
96	6.945	10.708	4.027	Aman	Aman
97	21.621	5.276	11.383	Bahaya	Bahaya
98	20.083	5.33	10.164	Bahaya	Bahaya
99	15.003	9.096	9.489	Waspada	Waspada
100	5.001	14.718	4.197	Aman	Aman
101	14.61	10.233	8.25	Waspada	Waspada
102	16.28	9.253	8.275	Waspada	Waspada
103	11.078	10.086	3.988	Aman	Aman

Tabel di atas merupakan contoh hasil prediksi yang diperoleh dari model yang telah dibangun menggunakan pendekatan marginisasi dan hyperplane. Tabel ini menampilkan beberapa parameter input, yaitu Rainfall (mm), Water Level (cm), dan Water Flow Rate sebagai fitur yang digunakan untuk menentukan kondisi (Condition) aktual dan kondisi prediksi (Predicted Condition). Setiap baris pada tabel merepresentasikan satu data observasi. Kolom "Condition" menunjukkan kondisi aktual yang diberikan berdasarkan label dataset awal, sementara kolom "Predicted Condition" adalah hasil klasifikasi dari model SVM. Jika nilai pada kolom "Condition" dan "Predicted Condition" sama, ini berarti model berhasil

memprediksi dengan benar untuk data tersebut. Sebaliknya, jika terdapat perbedaan, hal itu menunjukkan kesalahan prediksi oleh model.

4.5 Perancangan Notifikasi Sistem

Dalam sistem prediksi banjir yang dikembangkan, notifikasi sistem berperan penting dalam memberikan peringatan dini kepada pengguna berdasarkan hasil prediksi model Support Vector Machine (SVM). Notifikasi ini dikirimkan secara otomatis menggunakan Telegram Bot, yang dipilih sebagai media utama karena memiliki berbagai keunggulan. Telegram memiliki API yang terbuka dan gratis, sehingga dapat diintegrasikan langsung dengan sistem tanpa biaya tambahan. Selain itu, Telegram Bot memungkinkan pengiriman pesan otomatis ke pengguna atau grup tanpa memerlukan nomor telepon. Berikut perbandingan telegram dengan platform notifikasi lainnya.

Tabel 4.9 Tabel Perbandingan Platform Notifikasi

Fitur	Telegram	WhatsApp	SMS
Kecepatan	Cepat (real-time)	Cepat (real-time)	Tergantung operator
Dukungan Bot	Ya	Tidak	Tidak
Biaya	Gratis	Berbayar	Bebayar per sms
Format Pesan	Teks, gambar, grafik	Teks, gambar	Teks
Dukungan Grup	Ya	Ya	Tidak
Akses Data	ID telegram	Nomor Hp	Nomor Hp

Dengan keunggulan tersebut, Telegram Bot menjadi solusi yang lebih fleksibel dan efisien dibandingkan WhatsApp atau SMS dalam implementasi notifikasi prediksi banjir.

Tabel 4.10 Format Penulisan Notifikasi

No.	Format	Alasan
1.	Struktur pesan yang konsisten	Agar mudah dibaca dan dipahami oleh pengguna
2.	Menampilkan parameter utama kondisi sungai	Agar pengguna mengetahui kondisi sungai yang sebenarnya
3.	Menggunakan symbol atau emoji	Agar tampilan lebih menarik dan memudahkan pengguna dalam mengidentifikasi Tingkat kondisi.
4.	Menjelaskan Langkah Tindakan selanjutnya	Agar pengguna mengetahui langkah yang harus dilakukan selanjutnya.

Tabel 4.9 menunjukkan perbandingan elemen-elemen penting dalam perancangan format pesan notifikasi yang dikirimkan melalui sistem. Format pesan harus diatur dengan baik agar notifikasi yang diterima pengguna mudah dipahami dan memberikan informasi yang jelas. Keempat faktor tersebut merupakan pertimbangan peneliti agar tercipta sebuah format yang informatif dan mudah dipahami oleh pengguna, sehingga pengguna dapat segera mengambil tindakan yang sesuai berdasarkan informasi yang disampaikan.

4.6 Perancangan Pengujian

Perancangan pengujian merupakan metode yang digunakan oleh peneliti untuk menguji metode yang direncanakan dalam sistem. Metode pengujian yang digunakan dalam penelitian ini adalah Black Box.

4.6.1 Perancangan Pengujian Black Box Sistem

Pengujian prototipe dilakukan untuk mengevaluasi kinerja dan fungsi sistem setelah tahap pengembangan. Rincian pengujian prototipe yang digunakan dalam penelitian ini ditampilkan pada Tabel 4.11.

Tabel 4.11 Perancangan 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.
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.
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.
Pengujian Black Box ke-4	
Skenario Pengujian	Rangkaian dihidupkan.
Hasil yang Diharapkan	ESP32 terhubung ke jaringan Wi-Fi dan mendapatkan alamat IP yang valid.
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.
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.
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.
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.
Pengujian Black Box ke-9	
Skenario Pengujian	Model mengirimkan hasil prediksi ke Telegram.
Hasil yang Diharapkan	Model mengirim pesan prediksi ke Telegram yang mencakup waktu, hasil prediksi, dan detail data sensor.
Pengujian Black Box ke-10	
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.
Pengujian Black Box ke-11	
Skenario Pengujian	Aplikasi Telegram menampilkan hasil prediksi ke user.
Hasil yang Diharapkan	API telegram berhasil menampilkan pesan prediksi ke user yang mencakup waktu, hasil prediksi, dan detail data sensor.

4.6.2 Perancangan Pengujian Peforma Model

Pengujian performa dilakukan untuk mengevaluasi akurasi prediksi model terhadap data aktual yang telah diolah. Peneliti akan melakukan evaluasi performa model menggunakan Confusion Matrix untuk menilai akurasi klasifikasi model dalam membedakan kondisi Aman, Waspada, dan Bahaya. Confusion Matrix membagi hasil prediksi ke dalam True Positive (TP), True Negative (TN), False Positive (FP), dan False Negative (FN), yang digunakan untuk mengidentifikasi kesalahan dalam prediksi. Setelah mendapatkan Confusion Matrix, beberapa metrik utama dihitung untuk mengevaluasi performa model. Akurasi mengukur persentase prediksi yang benar, Precision menilai ketepatan prediksi positif, Recall mengukur sejauh mana model dapat mendeteksi kondisi yang sebenarnya terjadi, dan F1-Score memberikan keseimbangan antara Precision dan Recall. Selain itu, model dievaluasi menggunakan Mean Absolute Error (MAE) dan Root Mean Squared Error (RMSE) untuk mengukur tingkat kesalahan prediksi. MAE menghitung rata-

rata selisih absolut antara prediksi dan nilai aktual, sementara RMSE memberikan gambaran lebih sensitif terhadap error besar. Berikut perhitungan dari setiap evaluasi yang dijelaskan dalam tabel 4.12.

Tabel 4.12 Perancangan Pengujian Performa

No.	Metode	Pengukuran Performa	Perhitungan	Hasil yang Diharapkan
1.	Confusion Matrix	Akurasi, Precision, F1.	Berdasarkan TP, TN, FP, FN	Acc>80%
2.	Mean Absolute Error	Akurasi error.	$(MAE = \frac{1}{n} \sum_{i=1}^n)$	0% - 10%
3.	Root Mean Squared Error	Rata-rata error kuadrat.	$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (y' - y)^2}$	0% - 10%
4.	Akurasi	Persentase prediksi yang benar.	$Akurasi = \frac{TP + TN}{TP + TN + FP + FN}$	>80%
5.	Precision	Prediksi positif yang benar.	$Presisi = \frac{TP}{TP + FP}$	>80%
6.	Recall	Mendeteksi seluruh data positif.	$Recall = \frac{TP}{TP + FN}$	>80%
7.	F1-Score	Keseimbangan model.	$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall}$	>80%

Kriteria keberhasilan model ditentukan berdasarkan akurasi serta tingkat kesalahan prediksi yang dihasilkan. Agar model dikategorikan sebagai optimal, hasil pengujian harus menunjukkan Akurasi, Precision, Recall, dan F1-Score di atas 80%, serta memiliki Mean Absolute Error (MAE) dan Root Mean Squared Error (RMSE) dalam rentang 0% - 10%. Jika hasil pengujian menunjukkan akurasi di bawah ambang batas tersebut atau MAE/RMSE lebih dari 10%, maka model

memerlukan perbaikan lebih lanjut. Apabila model memiliki akurasi yang rendah, peneliti akan melakukan penyesuaian parameter model (hyperparameter tuning).

