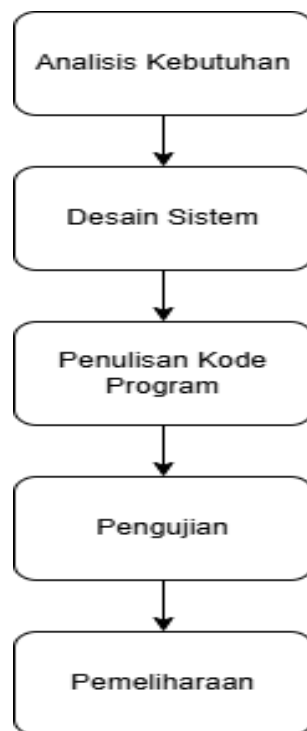


BAB III

METODE PENELITIAN

3.1 Alur Penelitian

Dalam penelitian ini, terdapat alur yang menjadi acuan dalam melaksanakan kegiatan penelitian. Penjabaran alur penelitian selanjutnya dapat dilihat pada gambar 3.1



Gambar 3.1 Waterfall

3.2 Metode Pengembangan Sistem

Metode pengembangan sistem dalam rekayasa sistem dan perangkat lunak merupakan proses pembuatan serta modifikasi sistem, yang melibatkan model dan metodologi yang digunakan dalam pengembangan perangkat lunak. Salah satu metodologi yang diterapkan dalam penelitian ini adalah Waterfall, yang terdiri dari beberapa tahapan, yaitu: Analisis Kebutuhan, Desain Sistem, Penulisan kode program, Pengujian (Testing), dan Pemeliharaan.

3.2.1 Analisis Kebutuhan

Tahapan ini bertujuan untuk mengumpulkan dan menganalisis kebutuhan pengguna terhadap sistem yang akan dikembangkan.

1. Kebutuhan Software

Perangkat lunak yang dibutuhkan untuk membangun dan menjalankan sistem :

- a. Sistem Operasi *Microsoft Windows 11*
- b. Visual Studio Code
- c. TensorFlow Keras
- d. Python

2. Kebutuhan Hardware

Perangkat keras yang dibutuhkan tergantung pada skala sistem dan kompleksitas data. Berikut ini adalah spesifikasi minimum:

Laptop

- a. RAM 8GB
- b. Intel Core i5-10300H

3. Kebutuhan Data

Data merupakan komponen krusial untuk membangun model prediksi. Sumber data yang digunakan meliputi: Google Maps API (Traffic Layer/Distance Matrix) dan Pemerintah/Dinas Perhubungan (jika tersedia) (Ibad et al., 2020) (Iskandar et al., n.d.).

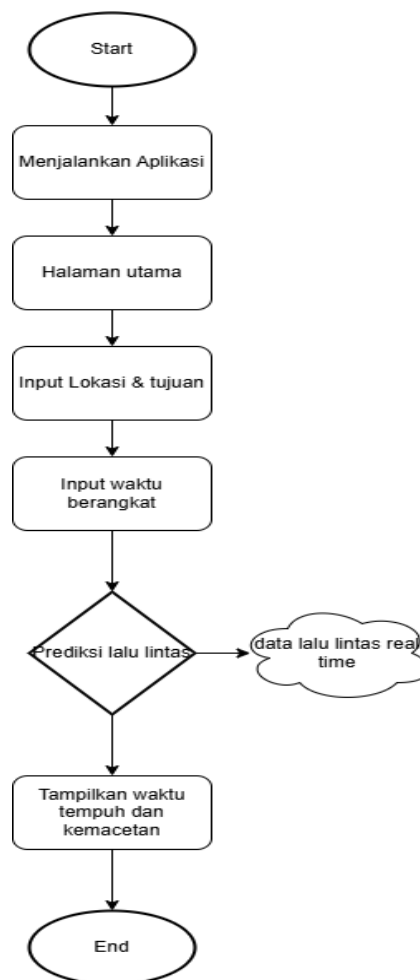
3.2.2 Desain Sistem

Tahap desain merupakan proses penting dalam pengembangan sistem, di mana pengembang merinci bagaimana sistem akan dibangun dan diimplementasikan berdasarkan hasil analisis kebutuhan. Tujuan dari tahap ini adalah untuk menghasilkan spesifikasi teknis dan rancangan sistem yang jelas sebagai panduan dalam proses implementasi. Dalam

proyek ini, desain sistem mencakup beberapa aspek utama, yaitu arsitektur sistem, desain aliran data, desain basis data, dan desain antarmuka pengguna. Pada proyek ini, desain sistem dalam kerangka *Waterfall* mencakup beberapa komponen utama yaitu: *Flowchart*, *Use Case Diagram* dan *Activity Diagram*.

3.2.2.1 Flowchart

Flowchart Merupakan representasi grafis dari alur proses sistem. Flowchart digunakan untuk menggambarkan langkah-langkah yang terjadi dalam suatu proses atau sistem secara berurutan. Alur ini akan menggambarkan logika dasar bagaimana data dan kontrol mengalir dalam sistem.

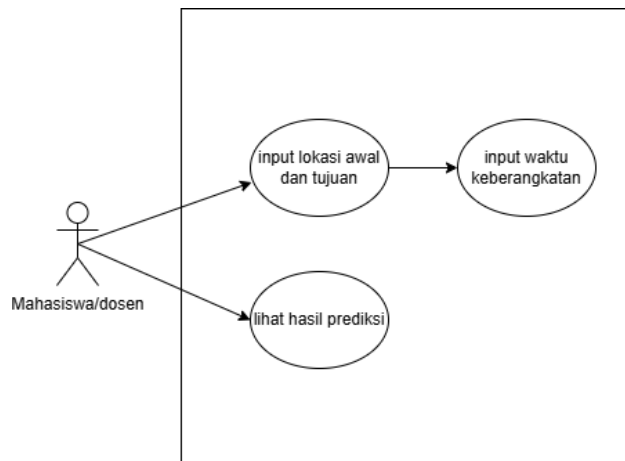


Gambar 3.2 Flowchart

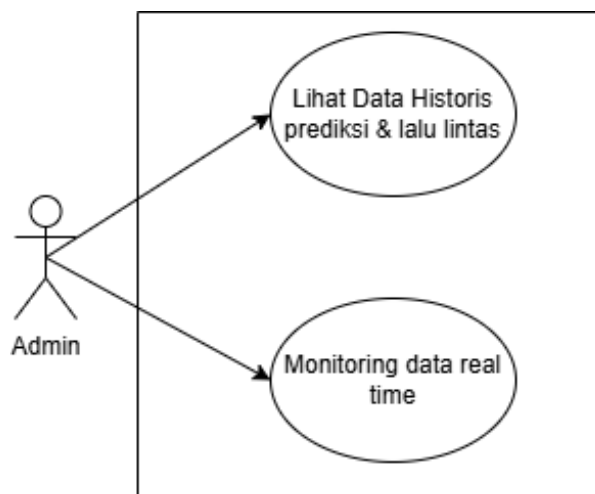
3.2.2.2 Use Case Diagram

Dalam perancangan sistem prediksi kemacetan dan waktu tempuh ke Kampus Darmajaya, telah dirancang use case diagram yang menggambarkan fungsi- fungsi yang dimiliki oleh masing-masing role pengguna. Rancangan use case diagram untuk Mahasiswa (Pengguna) dan Admin disusun untuk memvisualisasikan interaksi antara aktor dengan sistem secara jelas dan terstruktur.

Use case diagram pengguna (Mahasiswa/Dosen) digambarkan pada Gambar 3.3, sedangkan use case diagram untuk Admin dapat dilihat pada Gambar 3.4.



Gambar 3.3 Use Case Diagram Pengguna

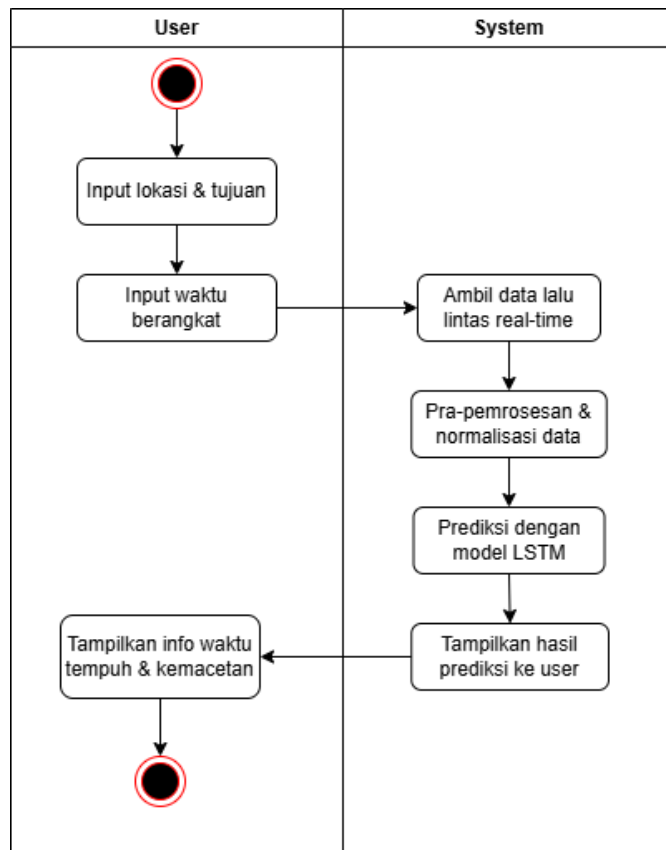


Gambar 3.4 Use Case Diagram Admin

3.2.2.3 Activity Diagram

Diagram aktivitas adalah rencana mengenai urutan langkah atau rangkaian kerja dalam suatu sistem yang akan digunakan. Activity diagram juga berperan dalam memberikan rincian serta mengelompokkan jalur visual dari sistem yang dirancang. Komponen dalam diagram aktivitas memiliki bentuk khusus dan terhubung satu sama lain melalui panah sebagai penanda alur proses.

Berikut ini adalah contoh activity diagram dari sistem prediksi kemacetan dan waktu tempuh ke Kampus Darmajaya berbasis machine learning.



Gambar 3.5 Activity Diagram

3.2.3 Penulisan Kode Program

Pada bagian frontend, sistem ini dibangun menggunakan bahasa pemrograman HTML untuk merancang antarmuka pengguna yang

sederhana namun fungsional. Tujuan dari penulisan kode HTML ini adalah agar pengguna dapat dengan mudah melakukan input lokasi asal mereka dan mendapatkan hasil prediksi kemacetan serta estimasi waktu tempuh menuju kampus Darmajaya.

3.3 Pendekatan Penelitian

Penelitian ini menggunakan pendekatan kuantitatif eksperimental dengan memanfaatkan data lalu lintas real-time sebagai input utama dalam proses pemodelan. Metode ini digunakan karena penelitian ini bertujuan untuk membangun sistem prediksi berbasis machine learning dan mengevaluasi akurasi secara kuantitatif. Teknik *data-driven* digunakan dalam proses pembelajaran model, khususnya menggunakan metode Long Short-Term Memory (LSTM) untuk prediksi kemacetan dan waktu tempuh.

3.4 Tahapan Penelitian

Tahapan penelitian ini disusun secara sistematis agar proses pengembangan sistem prediksi kemacetan dan waktu tempuh ke kampus Darmajaya berbasis data lalu lintas real-time dapat dilakukan secara terstruktur dan terarah. Berikut adalah uraian tahapan penelitian yang dilakukan:

3.4.1 Pengumpulan Data

Pengumpulan data dilakukan melalui integrasi API seperti Google Maps API, OpenTraffic, atau data milik Dinas Perhubungan Kota Bandar Lampung. Data yang dikumpulkan meliputi informasi waktu tempuh, kecepatan rata-rata kendaraan, tingkat kepadatan lalu lintas, dan timestamp (waktu pencatatan). Data dikumpulkan selama periode waktu tertentu untuk memastikan keragaman dan kualitas dataset.

3.4.2 Pra-Pemrosesan Data

Tahapan ini mencakup proses pembersihan data (data cleaning), transformasi data, normalisasi, serta pembuatan fitur yang relevan untuk pelatihan model. Selain itu, data dibagi menjadi data latih (training data)

dan data uji (testing data) dengan perbandingan tertentu (misalnya 80:20). Tujuannya adalah agar data yang dimasukkan ke dalam model sudah siap untuk diolah oleh algoritma machine learning.

3.4.3 Perancangan dan Pelatihan Model Machine Learning

Setelah data siap, dilakukan proses pemodelan menggunakan algoritma LSTM. Model ini dirancang untuk mempelajari pola historis dari data lalu lintas guna menghasilkan prediksi waktu tempuh dan tingkat kemacetan di masa mendatang. Arsitektur model meliputi beberapa lapisan LSTM, dropout, dan dense layer dengan fungsi aktivasi tertentu yang dioptimasi melalui proses pelatihan (training).

3.4.4 Evaluasi Model

Untuk menilai performa model LSTM yang dibangun, digunakan beberapa metrik evaluasi seperti Mean Absolute Error, Root Mean Square Error, dan Mean Absolute Percentage Error. Metrik ini akan menunjukkan seberapa akurat model dalam memprediksi kondisi lalu lintas dan estimasi waktu tempuh dibandingkan dengan data aktual.

3.4.5 Implementasi Sistem

Model yang sudah dilatih dan divalidasi kemudian diimplementasikan ke dalam sistem berbasis web atau aplikasi dashboard. Sistem ini memungkinkan pengguna untuk melakukan input titik asal dan tujuan, lalu menampilkan hasil prediksi secara real-time. Backend sistem dibangun menggunakan bahasa pemrograman Python atau Node.js, sementara frontend menggunakan teknologi seperti HTML, JavaScript, dan CSS/Framework.

3.5 Perancangan Model LSTM

3.5.1 Arsitektur LSTM

Model LSTM dirancang dengan struktur sebagai berikut:

1. Input Layer: urutan data (misalnya 3 timestep)

2. LSTM layer: 64 unit
3. Dropout: 0.2
4. Dense layer: 1 neuron (output prediksi waktu tempuh nyata)

3.5.2 Parameter Pelatihan

1. Optimizer: Adam
2. Loss function: Mean Squared Error
3. Epoch: 100
4. Batch size: 8