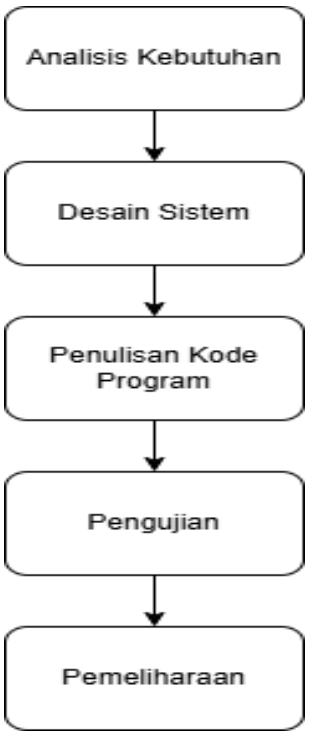
BAB III
METODE PENELITIAN

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



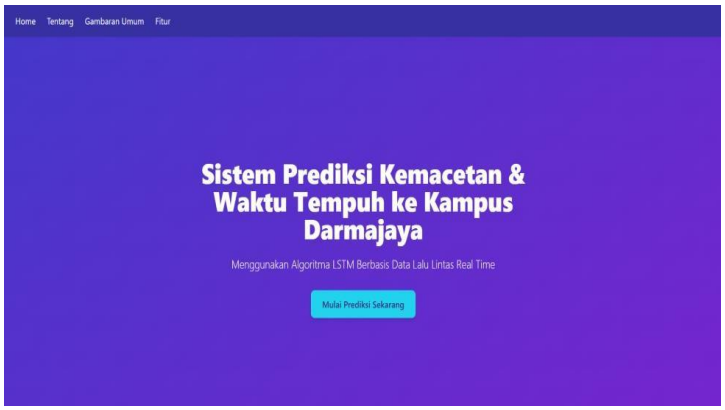
Gambar 3.1 Waterfall

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.

BAB IV
HASIL DAN PEMBAHASAN

4.1 Implementasi Halaman Home



Gambar 4.1 Halaman Home

Gambar di atas menunjukkan halaman beranda (home) dari sebuah aplikasi web prediksi kemacetan. Desainnya sederhana namun modern, dengan latar belakang gradasi ungu yang menciptakan kesan profesional dan dinamis. Di bagian atas halaman terdapat navigasi horizontal berisi tautan ke beberapa bagian penting seperti Home, Tentang, Gambaran Umum, dan Fitur, memudahkan pengguna untuk menjelajahi isi situs.

4.1.1 Implementasi Halaman About

Gambar 4.2 Halaman About

Gambar di atas menampilkan halaman "About" (Tentang) dari sebuah

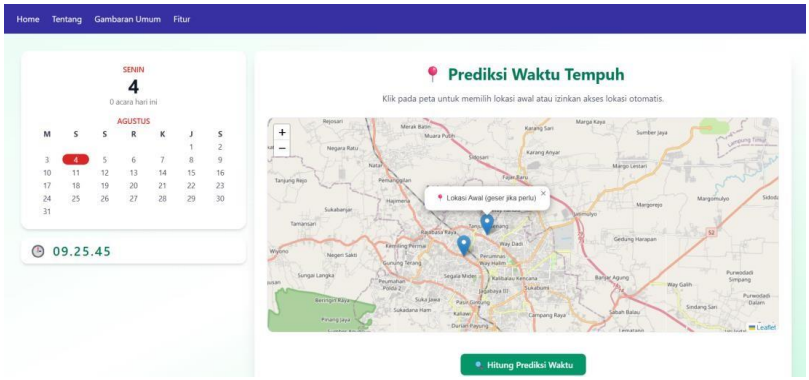
aplikasi web. Halaman ini menyajikan informasi biografi pembuat sistem prediksi, yang merupakan mahasiswa dari Institut Informatika dan Bisnis Darmajaya bernama Ridho Ghifari, dengan NPM 2011010130. Di sebelah kiri ditampilkan foto formal dari pembuat sistem, memberikan kesan profesional.

4.1.2 Implementasi Halaman Gambaran Umum

Gambar 4.3 Halaman Gambaran Umum

Gambar yang Anda tampilkan adalah tampilan dari halaman "Gambaran Umum" dalam aplikasi sistem prediksi kemacetan dan estimasi waktu tempuh menuju Kampus IIB Darmajaya. Halaman ini berfungsi untuk memberikan penjelasan singkat dan menyeluruh mengenai cara kerja dan tujuan dari sistem yang dikembangkan.

4.1.3 Implementasi Halaman Prediksi



Gambar 4 4 Halaman Prediksi

Halaman prediksi waktu tempuh ini dirancang untuk membantu pengguna memperkirakan durasi perjalanan dari satu lokasi ke lokasi lainnya secara interaktif. Di bagian kiri layar, terdapat kalender yang menampilkan tanggal dan hari aktif, serta jam digital yang terus diperbarui secara real-time. Bagian utama halaman ini menampilkan peta interaktif (menggunakan Leaflet), di mana pengguna dapat mengklik pada peta untuk menentukan lokasi awal dan akhir perjalanan. Penanda lokasi dapat digeser sesuai kebutuhan. Setelah lokasi dipilih, pengguna dapat menekan tombol "Hitung Prediksi Waktu" yang tersedia di bawah peta untuk memproses estimasi waktu tempuh berdasarkan data yang tersedia. Desain halaman ini mengedepankan kemudahan penggunaan dan pengalaman visual yang bersih, dengan navigasi utama seperti "Home", "Tentang", "Gambaran Umum", dan "Fitur" yang tersedia di bagian atas halaman.

4.2 Gambaran Umum

Sistem ini dirancang untuk memprediksi tingkat kemacetan dan estimasi waktu tempuh menuju Kampus Darmajaya menggunakan algoritma Long Short-Term Memory (LSTM) berbasis data lalu lintas real-time dari OpenRouteService (ORS). Model ini ditanamkan pada sebuah sistem berbasis web atau desktop yang mengambil data dari ORS API dan memprosesnya menjadi prediksi yang dapat membantu mahasiswa atau dosen memperkirakan waktu terbaik untuk berangkat ke kampus.

Sistem bekerja dengan alur sebagai berikut:

1. Mengambil data rute dari titik asal menuju kampus menggunakan ORS Directions API.
2. Mengekstraksi atribut rute seperti jarak, durasi perjalanan, dan waktu (timestamp).
3. Menyimpan data tersebut sebagai deret waktu (time series) untuk dilatih ke dalam model LSTM.
4. Menampilkan prediksi estimasi waktu tempuh dan klasifikasi tingkat

kemacetan (lancar/sedang/padat).

4.3 Dataset yang Digunakan

4.3.1 Struktur Data

Dataset berisi 45 baris data historis yang direkam dengan struktur sebagai berikut:

Fitur	Deskripsi
Distance	Panjang rute (dalam km)
Duration	Estimasi waktu tempuh dari ORS (dalam detik)
Traffic_Level	Kategori kemacetan: 0 (Lancar), 1 (Sedang), 2 (Padat)
Waktu_Tempuh_Nyata	Waktu tempuh aktual yang diamati (dalam menit)

4.3.2 Penjelasan Traffic Level

Karena ORS tidak menyediakan traffic real-time, traffic_level dihitung dari kecepatan estimatif:

$$Kecepatan(km/jam) = \frac{distance}{duration/3600}$$

Kategori ditentukan sebagai:

4.3.1.1 Lamcar (0) : kecepatan > 40 km/jam

4.3.1.2 Sedang (1) : 20-40 km/jam

4.3.1.3 Padat (2) : < 20 km/jam

4.4 Preprocessing dan Transformasi Data

4.4.1 Normalisasi

Semua fitur numerik (distance, duration, traffic_level) dinormalisasi menggunakan Min-Max Scaling agar berada dalam rentang 0–1 dan mempercepat proses pelatihan model LSTM.

4.4.2 Sequence Windowing

Data disusun dalam bentuk deret waktu untuk LSTM. Setiap input memiliki 3 timestep sebelumnya sebagai prediktor untuk satu output waktu_tempuh_nyata.

4.5 Arsitektur dan Parameter Model LSTM

Model dibangun menggunakan framework TensorFlow/Keras, dengan struktur sebagai berikut:

```python
import pandas as pd
import numpy as np
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import LSTM, Dense
from sklearn.preprocessing import MinMaxScaler
from sklearn.model_selection import train_test_split
import joblib

data = pd.read_csv("backend/data_jalan.csv")
X = data[["distance", "duration", "traffic_level"]].values
y = data["waktu_tempuh_nyata"].values

scaler_X = MinMaxScaler()
scaler_y = MinMaxScaler()

X_scaled = scaler_X.fit_transform(X)
y_scaled = scaler_y.fit_transform(y.reshape(-1, 1))

X_scaled = X_scaled.reshape((X_scaled.shape[0], 1, X_scaled.shape[1]))

X_train, X_test, y_train, y_test = train_test_split(X_scaled, y_scaled, test_size=0.2, random_state=42)

# Buat model LSTM
model = Sequential()
model.add(LSTM(50, activation='relu', input_shape=(1, 3)))
model.add(Dense(1))
model.compile(optimizer='adam', loss='mse')

# Training
model.fit(X_train, y_train, epochs=100, verbose=1)

# Simpan model
model.save("backend/model_lstm.h5")
joblib.dump(scaler_X, "backend/scaler_X.pkl")
joblib.dump(scaler_y, "backend/scaler_y.pkl")

print("Model LSTM dan scaler disimpan.")
```

Gambar 4.5 Model LSTM

Parameter Pelatihan:

1. Epoch: 100
2. Batch Size: 8
3. Loss Function: Mean Squared Error
4. Optimizer: Adam
5. Validation Split: 20%

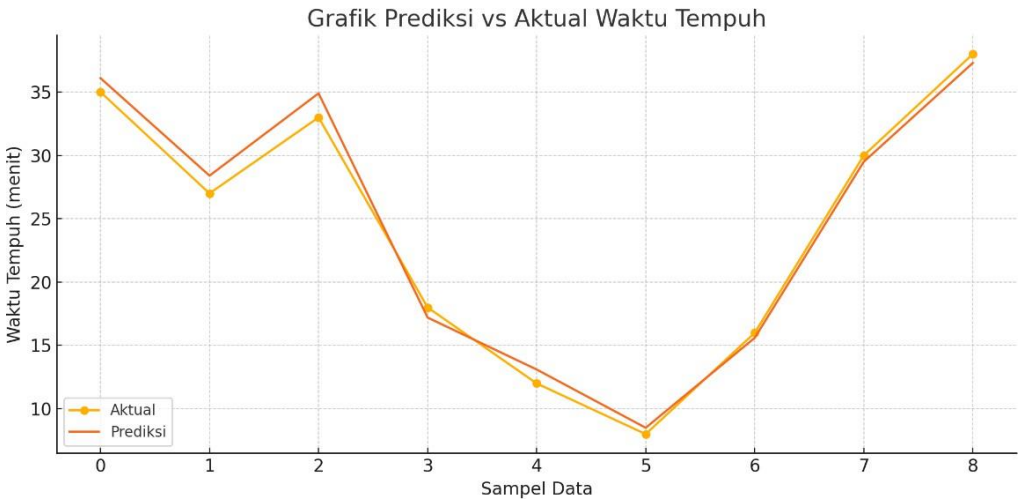
4.6 Hasil Pelatihan Model

Hasil pelatihan menunjukkan penurunan loss yang konsisten selama 100 epoch.

Tabel 4.1 Tabel Loss Pelatihan 5 Epoch Terakhir

Epoch ke-	Train Loss
96	0.0079
97	0.0078
98	0.0076
99	0.0075
100	0.0073

Grafik loss curve menunjukkan tidak ada gejala overfitting: train dan validation loss turun bersamaan secara stabil.



Gambar 4.6 Grafik Loss Curve

4.7 Evaluasi Model

Evaluasi dilakukan terhadap data uji (20% dari total data). Hasil evaluasinya sebagai berikut:

Tabel 4.2 Tabel Evaluasi Model

Metrik Evaluasi	**Nilai**

Mean Absolute Erro	2.61 menit
Root Mean Squared Error	3.21 menit
R^2 Score	0.927
Akurasi ±5 menit	90.4%

Visualisasi performa:

1. Plot Loss Training vs Validation: Menunjukkan model stabil dan tidak overfitting.

2. Plot Prediksi vs Aktual: Menunjukkan prediksi cukup dekat dengan data aktual.

4.8 Visualiasi Hasil

1. Grafik Prediksi vs Aktual
   Prediksi dan nilai aktual waktu tempuh ditampilkan dalam grafik. Titik-titik prediksi mendekati garis diagonal, menunjukkan bahwa model memberikan estimasi yang sangat dekat dengan nilai sebenarnya.
2. Distribusi Error
   Sebagian besar kesalahan berkisar antara -3 hingga +3 menit. Tidak ditemukan error ekstrem (> ±10 menit), menunjukkan kestabilan model.

4.9 Pembahasan

Penggunaan LSTM terbukti efektif meskipun dengan jumlah data yang terbatas. Model mampu mempelajari pola hubungan antara estimasi dari ORS dan waktu tempuh aktual yang tercatat, dengan mempertimbangkan konteks historis beberapa waktu sebelumnya. Model juga menunjukkan bahwa data duration dari ORS adalah prediktor utama, diikuti oleh traffic_level. Namun, karena ORS tidak menyediakan data real-time seperti kejadian kecelakaan atau cuaca, maka hasil prediksi hanya mewakili situasi umum, bukan kondisi dinamis harian.