



KELOMPOK I

PREDIKSI WAKTU TEMPUH SORONG & AIMAS

TUGAS KELOMPOK ALGORITMA PEMOGRAMAN II

**PREDIKSI WAKTU TEMPUH
PERJALANAN SORONG-AIMAS
BERBASIS WEB MENGGUNAKAN MODEL
XGBOOST DAN FITUR LOKASI DINAMIS**

**Implementasi Prediksi Waktu Tempuh Perjalanan
Sorong – Aimas Berbasis *Website* Menggunakan Model
Xgboost dan Fitur Lokasi Dinamis**



Disusun Oleh :

(KETUA) Marsha Laura Putty	(202455202040)
Diva Ceasary Dzuhiati	(202455202076)
Margareth V Berhitu	(202455202060)
Yosia Tangdi	(202455202068)
Fredrik A Tamunete	(202455202091)

**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS MUHAMMADIYAH SORONG
TAHUN 2025**

HALAMAN PERSETUJUAN

IMPLEMENTASI PREDIKSI WAKTU TEMPUH PERJALANAN SORONG – AIMAS BERBASIS WEB MENGGUNAKAN MODEL XGBOOST DAN FITUR LOKASI DINAMIS

**Diajukan sebagai salah satu syarat
Untuk Memperoleh Nilai UTS dan UAS
Mata Kuliah Algoritma dan Pemrograman 2
Pada Prodi Informatika Fakultas Teknik
Universitas Muhammadiyah Sorong**

**Disusun Oleh :
Kelompok 1**



Sorong, 12 juni 2025

Menyetujui dan Mengetahui
Dosen Pengganti Mata Kuliah

Menyetujui
Ketua Kelompok 1

Fajar R.B Putra.S.Kom.,M.Kom.
NIDN.1428099501

Marsha Laura Putty
NIM.202455202040

KATA PENGANTAR

Puji syukur penulis panjatkan kepada Tuhan Yang Maha Esa atas berkat, rahmat dan karunia-Nya sehingga penulis dapat menyelesaikan Tugas Besar dengan judul “*IMPLEMENTASI PREDIKSI WAKTU TEMPUH PERJALANAN SORONG-AIMAS BERBASIS WEBSITE MENGGUNAKAN MODEL XGBOOST DAN FITUR LOKASI DINAMIS*”. Penulis mengucapkan terima kepada teman – teman atas kerja sama dalam pengerjaan makalah ini sampai selesai. Adapun Tugas Besar ini disusun sebagai salah satu syarat untuk memperoleh nilai UTS dan UAS Mata Kuliah Algoritma dan Pemrograman 2, Program Studi Teknik Informatika, Fakultas Teknik,UNAMIN. Tentunya tidak lupa yang kami hormati kepada :

1. Bapak Dr. H. Muhammad Ali, M.M., M.H. selaku Rektor
Universitas Muhammadiyah Sorong
2. Bapak Fajar Rahardika Bahari Putra, S.Kom., M.Kom. Selaku
Dosen Pengampuh ALGORITMA PEMOGRAMAN II
3. Ibu Dewi Astria Faroek, S.Kom., M.Kom., Selaku KAPRODI
Teknik informatika
4. Teman – teman sekalian

Penulis menyadari bahwa penyusunan Tugas Besar ini masih banyak terdapat kekurangan, maka dari itu kelompok mengharapkan kritikan dan saran yang bersifat membangun.

Sorong, 14 Juni 2025

KELOMPOK 1

DAFTAR ISI

HALAMAN SAMPUL	i
HALAMAN PERSETUJUAN	iii
KATA PENGANTAR.....	iv
DAFTAR ISI	v
DAFTAR TABEL	Error! Bookmark not defined.
DAFTAR GAMBAR	vii
DAFTAR LAMPIRAM	1
BAB 1 PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah.....	2
1.3 Tujuan Penelitian	2
1.4 Batasan Masalah	3
1.5 Manfaat penelitian.....	3
BAB II LANDASAN TEORI.....	4
2.1 <i>State Of The Art</i>	4
2.2 Studi Literatur	4
2.3 Landasan Teori.....	18
2.3.1 Waktu	18
2.3.2 Model <i>XGboost</i>	18
2.3.3 <i>Google Colab</i>	19
2.3.4 <i>Visual Studio</i>	19
2.3.5 <i>Python</i>	20
2.3.6 <i>Hypertext Markup Language (HTML)</i>	20
2.3.7 <i>Cascading Style Sheet (CSS)</i>	20
2.3.8 <i>Flowchart</i>	21
2.3.9 Perancangan Sistem <i>UML</i>	23
2.3.10 <i>UI/UX</i>	26
2.3.11 Metode Pengembangan Sistem <i>SDLC</i>	27
2.3.12 <i>Figma</i>	29
2.3.13 <i>Black box</i>	29
2.3.14 <i>Usability Testing</i>	30
BAB III METODE PENELITIAN	31
3.1 Lokasi dan Waktu Penelitian.....	31

3.1.1 Lokasi Penelitian	31
3.1.2 Waktu Penelitian.....	32
3.2 Jenis Penelitian.....	32
3.3 Kerangka berfikir/konsep	33
3.4 Alur Penelitian	34
3.4.1 Tahap Penelitian	35
3.4.1.1 Identifikasi Masalah.....	35
3.4.1.2 Studi Literatur.....	35
3.4.1.3 Pengumpulan Data.....	35
3.4.1.4 Pengembangan <i>Sistem SDLC(Software Development Life Cycle)</i>	37
3.4.1.5 Implementasi algoritma	53
3.4.1.6 Pengujian Sistem	70
BAB IV PENUTUP	76
4.1 KESIMPULAN.....	76
4.2 SARAN	76
DAFTAR PUSTAKA.....	78
LAMPIRAN	91

DAFTAR GAMBAR

Gambar 2.1 <i>State Of The Art</i>	4
Gambar 2.2 Metode Pengembangan Sistem <i>SDLC</i>	27
Gambar 2.3 <i>Black Box Testing</i>	30
Gambar 3.1 Lokasi Dinamis.....	31
Gambar 3.2 Kerangka Berpikir	33
Gambar 3.3 Alur Penelitian.....	34
Gambar 3.4 Pengembangan <i>Model Xgboost</i>	35
Gambar 3.5 <i>SDLC (Software Development Life Cycle)</i>	37
Gambar 3.6 <i>Flowchart</i> Sistem.....	41
Gambar 3.7 <i>Use Case</i> Sistem	42
Gambar 3.8 <i>Activity Diagram</i> Deteksi.....	43
Gambar 3.9 <i>Activity Diagram</i> Tentang	44
Gambar 3.10 Tampilan Utama Beranda	45
Gambar 3.11 Tampian Prediksi Perjalanan	46
Gambar 3.12 Tampilann Hasl Prediksi	47
Gambar 3.13 Tampilan Visualisasi Rute	48
Gambar 3.14 Tampilan Riwayat Prediksi	49
Gambar 3.15 Tampilan Tentang.....	50
Gambar 3.16 Tampilan Halaman Kontak.....	51
Gambar 3.17 Mengakses <i>Google Drive di Google Colab</i>	53
Gambar 3.18 Instalasi dan Persiapan	53
Gambar 3.19 Membaca File Excel dari Google Drive	54
Gambar 3.20 Menyiapkan Data Untuk Pelatihan	54
Gambar 3.21 Melatih Model <i>XGBoost</i>	55
Gambar 3.22 Evaluasi Kinerja Model <i>XGBoost</i>	55
Gambar 3. 23 Melatih Model <i>XGBoost Regressor</i>	56
Gambar 3.24 Hitmap Korelasi Antara Vitur Numerik	57
Gambar 3.25 Visualisai Sebaran Waktu Tempuh	59
Gambar 3.26 Visualisasi Feature Importance.....	60
Gambar 3.27 Optimasi Hyperparameter XGBoost.....	61
Gambar 3.28 Visualisasi Distribusi Waktu.....	61

Gambar 3.29 Mengecek Jumlah Data Kosong	62
Gambar 3.30 Mengacak Data Duplikat.....	63
Gambar 3. 31 Menghapus Data Duplikat.....	63
Gambar 3.32 Visualisasi Distribusi Waktuh Tempuh.....	64
Gambar 3.33 Visualisasi Outlier dan Distribusi	66
Gambar 3.34 Pelatihan Ulang Model XGBoost	68
gambar 3.35 Menampilkan Hasil Waktuh Tempuh	70

DAFTAR TABEL

Tabel 2.1 Tabel Perbandingan Terkait dan Terkait.....	16
Tabel 2.2 <i>Flowchart</i>	21
Tabel 2. 3 Perancangan Sistem <i>UML</i>	24
Tabel 2. 4 <i>Activity Diagram</i>	25
Tabel 3.5 Jadwal Penelitian.....	32

DAFTAR LAMPIRAM

Lampiran 1 Evaluasi Pengerjaan Kelompok 1	92
Lampiran 2 Lembar Asistensi.....	93
Lampiran 3 Dokumentasi Kelompok.....	94
Lampiran 4 <i>Github</i>	95
Lampiran 5 Implementasi <i>Google Colab</i>	96

BAB 1

PENDAHULUAN

1.1 Latar Belakang

Sistem transportasi merupakan tulang punggung bagi kelancaran aktivitas ekonomi, sosial, dan pemerataan pembangunan wilayah. Di era modern seperti saat ini, kebutuhan akan mobilitas yang efisien, tepat waktu, dan terprediksi dengan baik menjadi semakin mendesak, terlebih dalam mendukung kelancaran logistik, aksesibilitas antarwilayah, serta kenyamanan perjalanan masyarakat. Ketepatan waktu tempuh bukan hanya sekadar persoalan kecepatan, tetapi juga menjadi indikator efektivitas perencanaan transportasi dan infrastruktur yang baik (Jalan dkk., 2024).

Namun dalam praktiknya, perjalanan antar daerah, khususnya di wilayah yang belum memiliki sistem transportasi yang sepenuhnya terintegrasi seperti Papua Barat Daya masih terkendala dengan ketidakpastian dalam waktu tempuh. Faktor-faktor seperti perubahan cuaca, waktu keberangkatan, kepadatan kendaraan, dan kondisi geografis menjadi variabel yang sangat memengaruhi kecepatan dan durasi perjalanan. Ketidakpastian ini sering kali menyebabkan keterlambatan, tidak efisiensi waktu, dan meningkatnya biaya transportasi baik bagi individu maupun institusi (Sayyendra dkk., 2024).

Tujuan penelitian ini dibuat untuk menciptakan prediksi waktu tempuh yang akurat dan adaptif menjadi kebutuhan yang mendesak. Salah satu model pembelajaran mesin yang menunjukkan performa unggul dalam prediksi waktu tempuh adalah *Extreme Gradient Boosting (XGBoost)*. Model ini dikenal karena kemampuannya dalam memproses data dalam jumlah besar, akurasi tinggi, serta efisiensi komputasi yang relatif baik dibandingkan metode lain.

XGBoost mampu menangkap hubungan antara waktu perjalanan dengan fitur-fitur seperti jam keberangkatan, hari dalam minggu, lokasi asal dan tujuan, serta kondisi eksternal lainnya. Oleh karena itu, penerapan model ini dalam konteks prediksi waktu tempuh menjadi relevan untuk dikaji lebih lanjut. Dalam konteks ini, rute antara Kota Sorong dan Kabupaten Aimas dipilih sebagai studi kasus karena merupakan salah satu jalur penting di Provinsi Papua Barat Daya (Alkaff & Rizky Baskara dkk., 2023).

Penerapan model *XGBoost* dengan fitur lokasi dinamis pada rute ini bertujuan untuk menyajikan pendekatan yang berbasis sains dan teknologi dalam menyelesaikan masalah transportasi di tingkat lokal (Astuti dkk., 2024).

Laporan ini tidak hanya bertujuan untuk menyajikan analisis dan hasil prediksi, tetapi juga akan diimplementasikan dalam bentuk praktikum. Praktikum ini akan menyerupai sistem navigasi seperti Miniatur *Google Maps* namun menggunakan model prediktif yang dibangun sendiri, lengkap dengan antarmuka visual berbasis peta. Dengan cara ini, pembaca dan peserta praktikum tidak hanya belajar teori, tetapi juga memahami bagaimana teknologi pembelajaran mesin diterapkan dalam kehidupan nyata, khususnya dalam konteks transportasi berbasis data lokal.

1.2 Rumusan Masalah

Berdasarkan latar belakang di atas, maka didapatkan rumusan masalah pada penelitian ini sebagai berikut:

1. Bagaimana memanfaatkan fitur lokasi dinamis (seperti data GPS *real-time*) untuk meningkatkan akurasi prediksi waktu tempuh perjalanan antara Kota Sorong dan Kabupaten Aimas?
2. Bagaimana implementasi sistem prediksi waktu tempuh berbasis lokasi dinamis dapat divisualisasikan dalam sistem interaktif yang mudah digunakan oleh pengguna jalan?

1.3 Tujuan Penelitian

Berdasarkan rumusan masalah yang di dapat, maka tujuan penelitian ini adalah sebagai berikut :

1. Mengidentifikasi dan memanfaatkan fitur-fitur lokasi dinamis seperti kecepatan kendaraan, jarak tempuh, kondisi lalu lintas, dan cuaca untuk meningkatkan akurasi prediksi.
2. Mengembangkan sebuah prototipe sistem praktikum visual interaktif yang memungkinkan pengguna melakukan prediksi waktu tempuh dari Kota Sorong ke Kabupaten Aimas secara *real-time*.

1.4 Batasan Masalah

Batasan masalah yang digunakan pada penelitian ini adalah sebagai berikut:

1. Ruang lingkup penelitian dibatasi hanya pada rute perjalanan beberapa titik lokasi yang terdapat di Kota Sorong dan Aimas (Kabupaten Sorong).
2. Fitur lokasi dinamis yang digunakan dalam prediksi dibatasi pada parameter tertentu seperti kecepatan kendaraan, waktu tempuh historis, kondisi cuaca, dan data lalu lintas.
3. Model prediksi yang digunakan terbatas pada algoritma *XGBoost* sebagai pendekatan utama *machine learning*
4. Penelitian tidak mencakup analisis dampak sosial, ekonomi, atau kebijakan transportasi dari hasil prediksi yang dihasilkan
5. Implementasi sistem dibatasi hanya untuk platform berbasis web.

1.5 Manfaat penelitian

Manfaat penelitian yang digunakan pada penelitian ini adalah sebagai berikut:

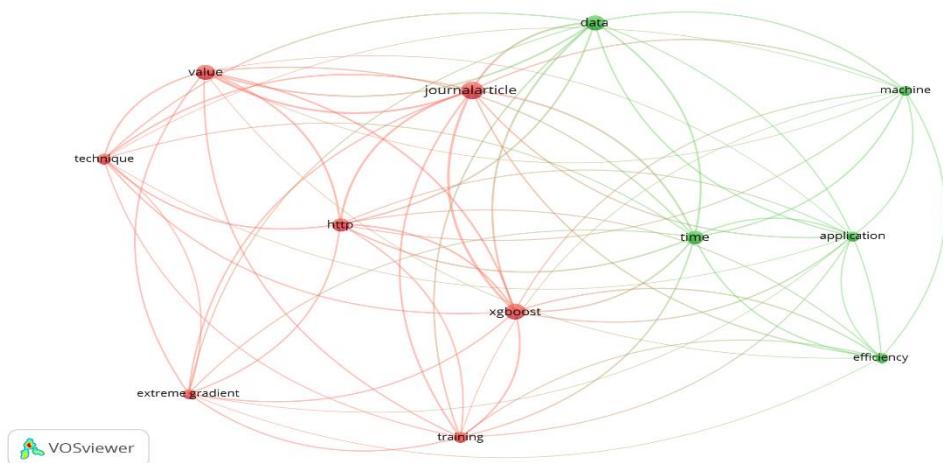
1. Menunjukkan bagaimana data lokasi yang berubah secara real-time (misalnya data GPS selama perjalanan Ber motor) dapat digunakan secara efektif dalam model pembelajaran mesin seperti *XGBoost* untuk meningkatkan akurasi prediksi waktu tempuh.
2. Mendukung pengambilan keputusan yang lebih baik oleh pengguna jalan, sopir kendaraan, layanan transportasi, atau pemerintah daerah dalam merencanakan waktu keberangkatan dan rute perjalanan.
3. Memberikan kontribusi dalam pengembangan dan penerapan algoritma *machine learning*, khususnya *XGBoost*, dalam bidang transportasi dan sistem informasi geografis.
4. Meningkatkan efisiensi operasional transportasi dengan mengantisipasi potensi keterlambatan akibat kemacetan atau kondisi cuaca yang tidak mendukung.
5. Membantu pengguna jalan, terutama masyarakat yang sering bepergian antara Sorong dan Aimas, dalam merencanakan perjalanan secara lebih efisien berdasarkan estimasi waktu tempuh yang akurat.

BAB II

LANDASAN TEORI

2.1 State Of The Art

State Of The Art adalah istilah yang merujuk pada tingkat pencapaian atau kemajuan terbaru dalam suatu bidang tertentu. Konsep ini menggambarkan sejauh mana sebuah metode, algoritma, atau teknologi telah berkembang dan digunakan secara efektif dalam penelitian-penelitian sebelumnya.



Gambar 2.1 *State Of The Art*

(Sumber: [VOSviewer](#))

State Of The Art di atas judul laporan ini menggunakan 10 teori yaitu 5 jurnal nasional dan 5 jurnal internasional yang dimana masing – masing teori memiliki keterkaitan dengan judul yang diangkat oleh penulis.

2.2 Studi Literatur

Studi literatur adalah teknik pengumpulan data atau cara untuk menyelesaikan persoalan dengan menelusuri sumber-sumber tulisan yang pernah dibuat sebelumnya. Dengan kata lain, istilah studi literatur ini juga sangat *familiar* dengan sebutan studi Pustaka. Dalam hal ini penulis mengutip beberapa Jurnal yang dijadikan acuan sebagai sumber untuk membuat sebuah *Website* Prediksi Waktu Tempuh Perjalanan Sorong-Aimas Berbasis *Website* Menggunakan Model *Xgboost* dan Fitur Lokasi Dinamis yang telah dibuat. Berikut beberapa Jurnal yang berkaitan dengan penelitian yang akan dilakukan:

1. Jurnal Nasional “ **Aplikasi Prediksi Banjir Menggunakan Algoritma XGBoost Berbasis Website**”, penelitian yang dilakukan oleh (Asnawi dkk., 2024) Bertujuan Untuk Penelitian ini berfokus pada pengembangan model prediksi risiko banjir menggunakan algoritma *XGBoost* berbasis *website*. Dataset yang digunakan berasal dari *Kaggle* dan mencakup berbagai faktor yang memengaruhi banjir, seperti kualitas bendungan, sistem drainase, longsor, dan hilangnya lahan basah.

Proses penelitian dimulai dengan pengumpulan data, dilanjutkan dengan *preprocessing* data untuk menangani *missing values*, pemilihan fitur, dan normalisasi data. Setelah data siap, model *XGBoost* dilatih dan dievaluasi menggunakan beberapa metrik, seperti *RMSE* (*Root Mean Squared Error*), *MSE* (*Mean Squared Error*), *MAE* (*Mean Absolute Error*), dan *MAPE* (*Mean Absolute Percentage Error*). Hasil evaluasi menunjukkan bahwa model memiliki performa yang sangat baik dengan nilai *RMSE* sebesar 0.001017, *MAE* sebesar 0.000801, dan *MAPE* sebesar 0.1605%, yang mengindikasikan kesalahan prediksi yang sangat kecil. Visualisasi hasil prediksi, seperti *scatter plot* dan histogram *residual*, menunjukkan bahwa model tidak memiliki bias sistematis dan kesalahan prediksi tersebar merata. Selain itu, implementasi *user interface* berbasis *website* memungkinkan pengguna untuk memasukkan parameter seperti kualitas bendungan, sistem drainase, dan faktor lainnya untuk mendapatkan prediksi risiko banjir secara *real-time*.

Penulis memilih jurnal ini karena adanya kesamaan metode dan fokus penelitian yang relevan dalam hal prediksi dengan metode yang sama dengan yang kami gunakan. Penelitian ini mengembangkan model prediksi banjir menggunakan algoritma *XGBoost* yang diintegrasikan dalam *website* interaktif. Dibandingkan penelitian sejenis yang umumnya hanya membangun model tanpa implementasi, penelitian ini unggul karena aplikatif dan responsif secara *real-time*.

2. Jurnal Nasional “**Implementasi Ensemble Learning Metode XGBoost dan Random Forest untuk Prediksi Waktu Penggantian Baterai Aki**”

Penelitian Yang di lakukan oleh (Rayadin dkk., 2024) Penelitian ini bertujuan untuk mengembangkan model prediksi waktu penggantian baterai aki pada kendaraan bermotor menggunakan pendekatan *ensemble learning* yang menggabungkan algoritma *XGBoost* dan *Random Forest*. Latar belakang penelitian ini didasarkan pada pentingnya peran baterai aki sebagai penyimpan energi listrik dan stabilisator tegangan pada kendaraan, serta tantangan dalam proses pemeriksaan dan penggantian baterai aki yang memakan waktu lama di layanan purna jual. Untuk mengatasi masalah ini, penelitian ini memanfaatkan data historis penggantian baterai aki mobil Toyota dari sebuah perusahaan otomotif di Kota Makassar, yang mencakup 7.703 baris data dengan atribut seperti model kendaraan, kilometer tempuh, jenis baterai, usia kendaraan, dan karakteristik area. Target prediksi adalah selisih waktu penggantian baterai aki dalam satuan hari. Tahapan penelitian dimulai dengan pengumpulan data, dilanjutkan dengan preprocessing data yang meliputi penanganan *outlier*, transformasi data (seperti *one-hot encoding* untuk fitur kategorikal dan *scaling* untuk fitur numerik), serta pembagian data menjadi set pelatihan (70%) dan pengujian (30%). Selanjutnya, dilakukan pemodelan dengan dua pendekatan: model tunggal (*XGBoost* dan *Random Forest*) dan model *ensemble* (*Bagging*, *Stacking*, dan *Boosting*). Proses pemodelan diawali dengan *hyperparameter tuning* untuk mengoptimalkan kinerja masing-masing algoritma. Hasil evaluasi model menggunakan metrik *MAPE* (*Mean Absolute Percentage Error*), *MAE* (*Mean Absolute Error*), *MSE* (*Mean Squared Error*), dan *RMSE* (*Root Mean Squared Error*) menunjukkan bahwa model *ensemble* memberikan kinerja yang lebih baik dibandingkan model tunggal. Secara rinci, model *bagging* (gabungan *XGBoost* dan *Random Forest*) menghasilkan nilai *MSE* terendah sebesar 145,448, sementara model *boosting* mencatat nilai error terbaik untuk *MAPE* (11,56%), *MAE* (43,80), dan *RMSE* (38,760). Hal ini mengindikasikan bahwa pendekatan *ensemble learning* mampu mengurangi kesalahan prediksi secara signifikan dan meningkatkan akurasi model.

Penulis memilih jurnal ini karena adanya kesamaan metode dan fokus penelitian yang relevan dalam hal prediksi dengan metode yang sama dengan yang kami gunakan. Penelitian ini mengembangkan model prediksi pergantian baterai akibat implementasi. Metode *ensemble learning* dalam penelitian ini terbukti lebih efektif dalam meningkatkan akurasi prediksi dibandingkan metode tunggal, menjadikannya solusi yang lebih handal untuk estimasi waktu penggantian aki kendaraan secara presisi.

3. Jurnal Nasional “***OPTIMIZATION OF XGBOOST ALGORITHM USING PARAMETER TUNNING IN RETAIL SALES PREDICTION***” Penelitian yang di lakukan oleh (Wijaya dkk., 2024) Penelitian ini bertujuan untuk mengoptimalkan algoritma *XGBoost* dalam memprediksi penjualan ritel dengan membandingkan tiga metode optimasi *parameter*: *Random Search*, *Grid Search*, dan *Bayesian Optimization*. Latar belakang penelitian ini didasarkan pada kebutuhan perusahaan ritel untuk memiliki prediksi penjualan yang akurat guna mendukung pengambilan keputusan bisnis, seperti manajemen inventaris dan perencanaan strategi pemasaran. *XGBoost* dipilih karena kemampuannya yang telah terbukti dalam menangani data kompleks dan menghasilkan prediksi yang presisi.

Dataset yang digunakan dalam penelitian ini adalah data penjualan ritel dari *Kaggle*, yang mencakup 37.427 baris data dengan 16 fitur seperti *Store*, *Date*, *Weekly_Sales*, *Temperature*, *Fuel_Price*, *CPI*, dan *Unemployment*. Data ini mencakup periode dari Februari 2019 hingga Oktober 2021. Tahapan penelitian dimulai dengan *preprocessing* data, termasuk penanganan nilai yang hilang, transformasi data kategorikal menjadi numerik, dan penghapusan *outlier* menggunakan metode *Interquartile Range (IQR)*. Selanjutnya, data dibagi menjadi set pelatihan (70%) dan pengujian (30%). Proses pemodelan dilakukan dengan melatih model *XGBoost* dasar terlebih dahulu. Hasil optimasi menunjukkan bahwa *Grid Search* memberikan kinerja terbaik dengan peningkatan nilai R^2 menjadi 98,41%, penurunan *MSE* menjadi 3.733.646,67, dan *MAE* menjadi 1.028,27. Parameter optimal yang ditemukan adalah *Learning Rate* 0,2, *Number of Trees* 300, *Tree Depth* 10, *Subsample* 1,0, dan *Sample Column* 1,0. Sementara itu, *Bayesian Optimization* juga menunjukkan hasil yang baik

dengan R^2 98,35%, tetapi sedikit lebih rendah dibandingkan *Grid Search*. Random Search justru mengalami penurunan performa dengan R^2 97,23%, menunjukkan bahwa pendekatan acak kurang efektif untuk dataset ini. Selain itu, penelitian ini juga menguji model pada dataset lain (*Walmart*) untuk memvalidasi generalisasi. Hasilnya konsisten: *Grid Search* tetap unggul dengan R^2 98,11%, mengungguli model lain seperti *Decision Tree* (90,50%), *Random Forest* (93,70%), dan *XGBoost* tanpa optimasi (93,29%). Namun, kelemahan *Grid Search* adalah waktu eksekusi yang lebih lama (1.635,78 detik) dibandingkan *Random Search* (42,97 detik) dan *Bayesian Optimization* (77,80 detik).

Penulis memilih jurnal ini karena adanya kesamaan metode dan fokus penelitian yang relevan dalam hal prediksi dengan metode yang sama dengan yang kami gunakan *Grid Search* menghasilkan model *XGBoost* paling akurat untuk prediksi penjualan ritel, namun dengan waktu komputasi lebih tinggi. *Bayesian Optimization* menawarkan akurasi mendekati dengan efisiensi waktu yang lebih baik, sedangkan *Random Search* paling cepat tetapi kurang akurat.

4. Jurnal Nasional “**Implementasi Model XGBoost untuk Prediksi Jumlah Transaksi dan Total Pendapatan di Jaringan Restoran CV Balibul**”
Penelitian yang di lakukan oleh (Garma dkk., 2024) Penelitian ini bertujuan untuk mengimplementasikan model *XGBoost* guna memprediksi jumlah transaksi dan total pendapatan di jaringan restoran *CV Balibul*. Tujuan utamanya adalah membantu manajemen restoran dalam mengoptimalkan sumber daya dan perencanaan operasional melalui prediksi yang akurat. *XGBoost* dipilih karena kemampuannya menangani data kompleks dan menghasilkan prediksi yang lebih baik dibandingkan metode lain seperti *Random Forest*. Optimasi parameter model dilakukan menggunakan *Bayesian Optimization*, sementara evaluasi kinerja model menggunakan metrik R^2 , *RMSE*, *MAPE*, dan *Pattern Similarity*.

Dataset dan *Preprocessing* Data yang digunakan berasal dari sistem Qasir, mencakup transaksi harian restoran dari tahun 2020 hingga 2024 dengan 22 atribut seperti No. Struk, Tanggal, Produk, Jumlah, dan Total Pendapatan. Data diolah menggunakan *library Pandas* untuk mengubah.

Penulis memilih jurnal ini karena adanya kesamaan metode dan fokus penelitian yang relevan dalam hal prediksi dengan metode yang sama dengan yang kami gunakan. Dengan dukungan *Bayesian Optimization* dan data transaksi riil, model *XGBoost* dalam penelitian ini berhasil memberikan prediksi yang akurat dan relevan untuk mendukung pengambilan keputusan manajemen restoran secara efisien.

5. Jurnal Nasional “**PERBANDINGAN AKURASI ALGORITMA XGBOOST DAN SVR DALAM PREDIKSI HARGA CRYPTOCURRENCY**”

Penelitian yang di lakukan oleh (Akurasi dkk., 2025) Penelitian ini bertujuan untuk membandingkan akurasi dua algoritma *machine learning*, yaitu *XGBoost* dan *Support Vector Regression (SVR)*, dalam memprediksi harga *cryptocurrency*. Latar belakang penelitian didasarkan pada volatilitas tinggi pasar *cryptocurrency* yang menyulitkan prediksi pergerakan harga secara manual. Dengan menggunakan data transaksi 10 *cryptocurrency*, penelitian ini mengevaluasi performa kedua algoritma melalui metrik seperti *Mean Absolute Error (MAE)*, *Root Mean Squared Error (RMSE)*, *Mean Squared Error (MSE)*, dan *Mean Absolute Percentage Error (MAPE)*. Hasil penelitian menunjukkan bahwa *XGBoost* secara signifikan lebih unggul dibandingkan *SVR* dalam memprediksi harga *cryptocurrency*. *XGBoost* menghasilkan nilai *MAPE* yang sangat rendah (antara 0–1%) untuk semua *cryptocurrency* yang diuji, menunjukkan kemampuannya yang baik dalam menangkap pola pergerakan harga pasar. Sebagai contoh, *XGBoost* mencapai *MAPE* sebesar 0.0697% untuk *Cardano* dan 0.1619% untuk *Bitcoin*. Sementara itu, *SVR* menunjukkan performa yang tidak konsisten. Meskipun *SVR* mampu memberikan hasil yang cukup baik untuk beberapa *cryptocurrency* seperti *Cardano* (*MAPE* 1.5114%) dan *XRP* (*MAPE* 2.28%), namun untuk *cryptocurrency* lain seperti *Dogecoin* (*MAPE* 38.70%) dan *Tether* (*MAPE* 14.47%), akurasinya jauh lebih rendah. Perbedaan performa ini disebabkan oleh karakteristik kedua algoritma. *XGBoost*, yang menggunakan teknik boosting dan regularisasi, mampu menangkap hubungan non-linear dalam data dengan lebih efektif. Selain itu, *XGBoost* secara iteratif memperbaiki kesalahan prediksi model sebelumnya, sehingga menghasilkan akurasi yang tinggi. Di sisi lain, *SVR*, yang mengandalkan kernel untuk memetakan data

ke ruang dimensi tinggi, lebih sensitif terhadap parameter dan kurang fleksibel dalam menangani volatilitas tinggi yang khas pada data *cryptocurrency*.

6. Jurnal Internasional “***Development and Validation of an XGBoost-Algorithm-Powered Survival Model for Predicting In-Hospital Mortality Based on 545,388 Isolated Severe Traumatic Brain Injury Patients from the TQIP Database***” penelitian yang dilakukan oleh (Cao dkk., 2023) Penelitian ini bertujuan untuk mengembangkan dan memvalidasi model prediksi kematian di rumah sakit pada pasien dengan cedera otak traumatis berat (TBI) menggunakan algoritma *XGBoost* yang dikombinasikan dengan regresi Cox. Data yang digunakan berasal dari basis data *Trauma Quality Improvement Program (TQIP)* milik *American College of Surgeons*, mencakup 545.388 pasien dewasa dengan TBI berat yang dirawat antara tahun 2013 hingga 2021. Data dibagi menjadi set pelatihan (80%) dan set pengujian (20%) untuk membangun dan menguji model. Metode penelitian melibatkan analisis regresi Cox dengan penalti L1 dan L2 untuk mengidentifikasi prediktor utama kematian di rumah sakit. Selanjutnya, model *XGBoost-powered Cox regression* dilatih menggunakan prediktor terpilih, seperti tingkat keparahan cedera (*Head AIS 5*), skor *Glasgow Coma Scale (GCS)* saat masuk, usia, dan variabel klinis lainnya. Kinerja model dievaluasi menggunakan indeks konkordansi *Harrell (C-index)* dan area di bawah kurva *ROC (AUC)* yang bergantung waktu. Hasil penelitian menunjukkan bahwa model *XGBoost-powered Cox regression* memiliki kinerja yang sangat baik dalam memprediksi kematian di rumah sakit, terutama pada 5 hari pertama perawatan ($AUC = 0,917$). Kinerja model tetap sangat baik hingga 20 hari perawatan ($AUC = 0,813$), yang mencakup 95,4% pasien dalam studi. Prediktor paling berpengaruh adalah skor *GCS* saat masuk, tingkat keparahan cedera (*Head AIS 5*), dan usia pasien. Namun, setelah 20 hari, kinerja model menurun ($AUC < 0,7$), menunjukkan bahwa faktor lain yang tidak tercakup dalam data mungkin menjadi penentu utama kematian pada fase ini. Analisis *SHAP (Shapley Additive Explanations)* mengungkapkan bahwa skor *GCS* memiliki dampak terbesar pada prediksi kematian, diikuti oleh *Head AIS 5* dan usia. Variabel lain seperti hipotensi, saturasi oksigen, dan intervensi

bedah saraf juga berkontribusi, tetapi dengan dampak yang lebih kecil. Model ini menunjukkan bahwa pada *fase* awal perawatan, faktor non-modifikasi seperti keparahan cedera dan usia dominan dalam menentukan hasil, sedangkan setelah 20 hari, faktor perawatan atau komplikasi mungkin lebih berpengaruh. Penelitian ini menyoroti potensi penggunaan model berbasis *XGBoost* untuk prediksi kematian di rumah sakit pada pasien TBI berat, terutama dalam fase akut.

7. Jurnal Internasional “*An implementation of XGBoost algorithm to estimate effective porosity on well log data*” penelitian yang dilakukan oleh (Fajrul Haqqi dkk., 2023) Penelitian ini bertujuan untuk Penelitian ini mengadopsi pendekatan berbasis data dengan tahapan yang sistematis. Data utama yang digunakan terdiri dari dua jenis: pertama adalah data *well log* konvensional seperti *gamma ray (GR)*, densitas batuan (*RHOB*), *neutron porosity (NPHI)*, dan *sonic travel time (DT)*; kedua adalah data porositas aktual dari pengukuran laboratorium yang berfungsi sebagai referensi kebenaran. Sebelum dimasukkan ke dalam model, data tersebut melalui proses pra-pengolahan yang mencakup normalisasi nilai-nilai log, penanganan data hilang, dan pembagian dataset menjadi bagian pelatihan dan pengujian. Algoritma *XGBoost* dipilih sebagai inti penelitian karena keunggulannya dalam menangani hubungan non-linear yang kompleks antara data log dan porositas. *XGBoost* bekerja dengan membangun serangkaian pohon keputusan secara iteratif, di mana setiap pohon baru berfokus memperbaiki kesalahan prediksi dari pohon sebelumnya. Proses ini didukung oleh teknik *optimisasi gradient descent* yang membuatnya cepat dan efisien. Untuk memastikan keandalan model, peneliti menggunakan beberapa metrik evaluasi seperti *Root Mean Square Error (RMSE)* untuk mengukur besarnya kesalahan prediksi, serta koefisien determinasi (R^2) yang menunjukkan seberapa baik variasi porositas dapat dijelaskan oleh model. Implementasi *XGBoost* pada penelitian ini menunjukkan hasil yang sangat menjanjikan. Model yang dibangun berhasil memprediksi nilai porositas efektif dengan akurasi tinggi, tercermin dari nilai R^2 yang melebihi 0.85. Ini berarti lebih dari 85% variasi porositas dalam data dapat dijelaskan oleh model berdasarkan parameter log sumur. Kesalahan prediksi yang diukur melalui *RMSE* juga

menunjukkan nilai yang relatif rendah, mengindikasikan ketepatan model. Ketika dibandingkan dengan algoritma machine learning lain seperti *Random Forest* atau *Support Vector Machine (SVM)*, *XGBoost* unggul baik dalam hal akurasi maupun konsistensi performa.

8. Jurnal Internasional “*Novel Hybrid XGBoost Model to Forecast Soil Shear Strength Based on Some Soil Index Tests*” penelitian yang dilakukan oleh (Momeni dkk., 2023) Penelitian ini bertujuan untuk mengembangkan model prediksi baru untuk memperkirakan kekuatan geser tanah (shear strength) menggunakan teknik *eXtreme gradient boosting (XGBoost)* yang dioptimalkan dengan *algoritma salp swarm (SSA)*. Kekuatan geser tanah merupakan parameter penting dalam desain struktur geoteknik seperti dinding penahan dan bendungan. Metode langsung untuk mengukur parameter ini seringkali rumit dan mahal, sehingga pendekatan tidak langsung berbasis kecerdasan buatan menjadi alternatif yang menarik. Studi ini menggunakan *database* yang terdiri dari 152 sampel tanah dari Provinsi Zanjan, Iran. Input model meliputi lima parameter indeks tanah: kadar air (W_n), berat unit kering (γ_d), persentase lolos saringan No. 4 (PPS4) dan No. 40 (PPS40), serta *indeks plastisitas* (PI). Output model adalah kekuatan geser tanah (τ), yang diukur melalui uji geser langsung dengan tegangan normal 1.5 kg/cm². Analisis korelasi menunjukkan bahwa hanya γ_d yang memiliki korelasi positif dengan τ , sedangkan parameter lainnya berkorelasi negatif. Model *XGBoost* dikembangkan dengan menggabungkan algoritma SSA untuk mengoptimalkan *hyperparameter* seperti *learning rate* dan *subsample*. Sebagai pembanding, digunakan juga metode *random search* (RS) untuk optimasi. Hasil evaluasi menunjukkan bahwa model SSA-XGBoost dengan ukuran populasi 40 menghasilkan kinerja terbaik. Pada data pelatihan, model ini mencapai nilai R^2 0.977 dan *VAF* 97.714%, dengan *RMSE* 0.026 dan *MAE* 0.019. Pada data uji, nilai R^2 0.849 dan *VAF* 84.936%, dengan *RMSE* 0.094 dan *MAE* 0.073. Model ini juga diuji dengan 30 data validasi tambahan, menghasilkan R^2 0.805 dan *VAF* 83.379%, yang menunjukkan kemampuan generalisasi yang baik meskipun terdapat indikasi *overfitting* karena ukuran dataset yang terbatas. Analisis sensitivitas mengungkapkan bahwa kadar air (W_n) merupakan parameter input yang paling berpengaruh terhadap kekuatan

geser tanah, diikuti oleh γd , PPS40, PI, dan PPS4. Penelitian ini menyimpulkan bahwa *hybrid SSA-XGBoost* merupakan alat yang efektif untuk memprediksi kekuatan geser tanah, terutama ketika pengukuran langsung tidak memungkinkan. Namun, disarankan untuk memperluas dataset di masa depan guna meningkatkan generalisasi model dan mengurangi risiko *overfitting*. Selain itu, eksplorasi teknik *machine learning* lainnya juga direkomendasikan untuk masalah serupa. Penulis memilih jurnal ini karena adanya kesamaan metode dan fokus penelitian yang relevan dalam hal prediksi dengan metode yang sama dengan yang kami gunakan Penelitian ini mengembangkan model prediksi kekuatan geser tanah menggunakan *XGBoost* yang dioptimalkan dengan *algoritma Sarp Swarm* (SSA). Sebagai pembanding, metode *Random Search* (RS) juga digunakan untuk optimasi. Hasil menunjukkan *SSA-XGBoost* unggul dengan nilai R^2 dan metrik error yang lebih baik dibanding RS. Analisis sensitivitas juga dilakukan untuk mengetahui pengaruh tiap parameter tanah terhadap prediksi.

9. Jurnal Internasional “*Data-Driven Optimised XGBoost for Predicting the Performance of Axial Load Bearing Capacity of Fully Cementitious Grouted Rock Bolting Systems*” penelitian yang dilakukan oleh (Jodeiri Shokri dkk., 2024) Penelitian ini bertujuan untuk memprediksi kapasitas beban aksial sistem baut batuan yang digROUTING penuh menggunakan teknik *eXtreme Gradient Boosting (XGBoost)* yang dioptimalkan dengan algoritma metaheuristik. Kapasitas beban aksial merupakan parameter kritis dalam desain struktur bawah tanah, dan pengukuran langsung melalui uji tarik (*pull-out test*) seringkali memakan waktu dan biaya. Oleh karena itu, pendekatan berbasis *machine learning* menjadi solusi alternatif yang efisien. Studi ini menggunakan dataset yang terdiri dari 72 sampel baut batuan yang digROUTING penuh. Parameter input meliputi rasio air terhadap grout (*W/G*), waktu pengeringan (CT), diameter konfinemen (DC), jenis grout, kandungan *fly ash* (FA), dan kekuatan tekan uniaksial (UCS) grout. *Output* model adalah nilai beban puncak (*peak load*) yang diukur melalui uji tarik. Analisis statistik menunjukkan bahwa UCS memiliki korelasi positif terkuat dengan beban puncak (koefisien Pearson 0.57), sedangkan FA memiliki pengaruh paling lemah. Lima model dikembangkan: *XGBoost* standar dan empat model *hybrid*

yang menggabungkan *XGBoost* dengan algoritma optimasi Harris *Hawk Optimization (HHO)*, *Jellyfish Search Optimizer (JSO)*, *Dragonfly Algorithm (DA)*, dan *Firefly Algorithm (FA)*. Hasil optimasi hyperparameter menunjukkan bahwa model *XGBoost-JSO* mencapai konfigurasi terbaik dengan learning rate 0.102, kedalaman maksimum 3, dan 131 estimator. Model ini mencatat kinerja terbaik dengan nilai R^2 0.988 dan *VAF* 98.737% pada data pelatihan, serta R^2 0.987 dan *VAF* 98.66% pada data uji. Error yang dihasilkan juga rendah, dengan *MAE* 0.668 dan *RMSE* 0.823 pada data uji. Analisis sensitivitas mengungkapkan bahwa *UCS* adalah parameter paling berpengaruh terhadap kapasitas beban aksial, diikuti oleh jenis grout, *DC*, rasio *W/G*, *CT*, dan *FA*. Hasil ini konsisten dengan temuan eksperimen yang menunjukkan bahwa peningkatan *UCS* dan *DC* serta penurunan rasio *W/G* dapat meningkatkan kapasitas beban aksial. Penulis memilih jurnal ini karena adanya kesamaan metode dan fokus penelitian yang relevan dalam hal prediksi dengan metode yang sama dengan yang kami gunakan. Penelitian ini membandingkan *XGBoost* standar dengan empat versi *hybrid* yang dioptimalkan menggunakan algoritma *HHO*, *JSO*, *DA*, dan *FA*. Model terbaik adalah *XGBoost-JSO*, dengan akurasi tertinggi ($R^2 = 0.987$) dan error terendah. Optimasi *metaheuristik* terbukti efektif meningkatkan kinerja prediksi kapasitas beban aksial dibandingkan *XGBoost* biasa.

10. Jurnal Internasional “*Optimizing Project Time and Cost Prediction Using a Hybrid XGBoost and Simulated Annealing Algorithm*” penelitian yang dilakukan oleh (ForouzeshNejad dkk., 2024) Penelitian ini bertujuan untuk membahas penerapan teknik *eXtreme Gradient Boosting (XGBoost)* dan optimisasi *metaheuristik* untuk memprediksi kapasitas beban aksial dari sistem penyangga batuan yang sepenuhnya digROUT dengan semen. Penelitian ini dilakukan dengan membangun dataset komprehensif yang terdiri dari 72 uji tarik (*pull-out tests*) yang mempertimbangkan berbagai parameter yang berpengaruh, seperti rasio air terhadap grout (*W/G*), waktu pemasakan (*CT*), jenis aditif grout, dan diameter penyangga (*DC*). Hasil penelitian menunjukkan bahwa nilai beban puncak (*peak load*) umumnya meningkat seiring dengan waktu pemasakan yang lebih lama, rasio *W/G* yang lebih rendah, serta nilai *UCS* (*Uniaxial Compressive Strength*) dan *DC* yang lebih

tinggi. Penelitian ini menyoroti pendekatan berbasis data yang inovatif dengan menggunakan berbagai model *XGBoost*, yang menawarkan alternatif yang lebih efisien dalam hal waktu, biaya, dan tenaga kerja dibandingkan dengan metode eksperimental tradisional untuk memprediksi kinerja baut batuan. Setelah membangun dataset dan membaginya menjadi dua bagian, yaitu dataset pelatihan dan pengujian, lima model *XGBoost* dikembangkan, termasuk model *XGBoost* mandiri dan empat model hibrida yang menggabungkan algoritma optimisasi seperti *Harris Hawk Optimisation (HHO)*, *Jellyfish Search Optimiser (JSO)*, *Dragonfly Algorithm (DA)*, dan *Firefly Algorithm (FA)*. Semua model ini dievaluasi berdasarkan kemampuannya untuk memprediksi nilai beban puncak. Dari hasil evaluasi, semua model berhasil memprediksi nilai beban puncak, namun model hibrida *XGBoost-JSO* menunjukkan kinerja yang paling unggul dengan koefisien *R-squared* tertinggi, yaitu 0.987 untuk dataset pelatihan dan 0.988 untuk dataset pengujian. Analisis sensitivitas mengungkapkan bahwa nilai *UCS* adalah parameter yang paling berpengaruh, sedangkan kandungan *FA* memiliki dampak paling kecil terhadap nilai beban puncak maksimum dari baut batuan yang sepenuhnya digROUT dengan semen. serta menggabungkan sistem pemantauan waktu nyata dengan algoritma pembelajaran mesin untuk meningkatkan efektivitas model dalam situasi praktis. Penulis memilih jurnal ini karena adanya kesamaan metode dan fokus penelitian yang relevan dalam hal prediksi dengan metode yang sama dengan yang kami gunakan, Penelitian ini membandingkan lima model prediksi kapasitas beban aksial, yaitu *XGBoost* standar dan empat versi hybrid dengan algoritma optimasi (HHO, JSO, DA, FA). *XGBoost-JSO* memberikan hasil terbaik dengan akurasi tertinggi ($R^2 = 0.988$). Pendekatan hybrid terbukti lebih unggul dibanding model *XGBoost* tunggal, terutama dalam efisiensi dan akurasi prediksi.

Tabel 2.1 Tabel Perbandingan Terkait dan Terkait

15.	HTML	✓	✓										
16.	Google Colab	✓											
17.	Google Maps	✓	✓										
18.	Draw.io	✓											

Keterangan :

PP : Peneliti Penulis

P1-P10 : Jurnal Penel

2.3 Landasan Teori

Adalah kumpulan teori, konsep, dan referensi ilmiah yang menjadi dasar pemikiran dalam suatu penelitian atau karya tulis ilmiah. Fungsinya untuk memberikan kerangka pemahaman, mendukung analisis, dan mengaitkan penelitian dengan pengetahuan yang sudah ada di bidang terkait.

2.3.1 Waktu

Manajemen waktu merupakan hal yang penting terutama bagi pelajar. Kemampuan manajemen waktu dapat membantu siswa secara teratur dan produktif dalam belajar, sehingga tidak melakukan aktivitas yang kurang bermanfaat (Harlina, Suharso, & Hartati, 2014). Manajemen waktu sebagai pembentukan *self regulated learning*, semakin tinggi pengaturan manajemen waktu akan menjadikan *self regulated learning* baik (Mulyani, 2013). Pengaturan manajemen waktu yang baik dapat membantu siswa dalam mengerjakan aktivitas sekolah dengan lancar. Siswa yang kurang mampu mengatur waktu dengan teratur akan menemukan hambatan dalam menyelesaikan aktivitas sekolahnya(Usroh dkk., 2022).

2.3.2 Model XGboost

Extreme Gradient Boosting (XGBoost) adalah algoritma *ensemble learning* berbasis metode boosting yang dikembangkan oleh Tianqi Chen pada tahun 2014. *XGBoost* bekerja dengan cara menggabungkan beberapa model pohon keputusan (*decision tree*) secara bertahap, di mana setiap pohon baru berfokus untuk memperbaiki kesalahan prediksi dari model sebelumnya.

Algoritma ini dirancang untuk memiliki efisiensi komputasi yang tinggi, fleksibilitas dalam berbagai jenis data, serta kemampuan generalisasi yang baik, sehingga sering digunakan dalam berbagai kompetisi dan aplikasi data science(Jan Melvin Ayu Soraya Dachi & Pardomuan Sitompul dkk., 2023).

2.3.3 *Google Colab*

Google Colaboratory (*Google Colab*) adalah sebuah *Integrated Development Environment* (IDE) berbasis *cloud* untuk pemrograman *Python*, di mana proses komputasi dijalankan di server milik Google yang dilengkapi dengan perangkat keras berperforma tinggi seperti CPU, GPU, dan TPU. *Google Colab* memudahkan pengguna dalam melakukan komputasi *data science*, *machine learning*, dan *deep learning* tanpa perlu memasang perangkat keras mahal di komputer pribadi. Selain itu, *Google Colab* telah menyediakan berbagai pustaka (*library*) populer seperti Keras, *TensorFlow*, *NumPy*, *Pandas*, dan *Matplotlib* secara otomatis, termasuk dukungan untuk berbagai versi *Python* dan *TensorFlow*. Di sisi penyimpanan, *Google Colab* terintegrasi langsung dengan Google Drive, sehingga memudahkan pengguna dalam menyimpan, mengakses, dan berbagi file proyek. Selama koneksi internet stabil, proses komputasi di *Google Colab* dapat berjalan lancar tanpa kendala berarti(Gelar Guntara dkk., 2023).

2.3.4 *Visual Studio*

Visual Studio Code adalah kode editor sumber yang dikembangkan oleh *Microsoft* untuk *Windows*, *Linux* dan *macOS*. *Visual Code* memudahkan dalam penulisan code yang mendukung beberapa jenis bahasa pemrograman yang digunakan dan memberi variasi warna sesuai dengan fungsi dalam rangkaian code tersebut. Selain itu, fitur lainnya adalah kemampuan untuk menambah ekstensi dimana para pengembang dapat menambah ekstensi untuk menambah fitur yang tidak ada di *Visual Studio Code*. *Visual Studio Code* bersifat *open source*, yaitu aplikasi dengan *source code* yang dapat dilihat oleh siapapun untuk berkontribusi pada pengembangan aplikasi tersebut. Code juga dapat dilihat melalui link github, menjadikan aplikasi *Visual Studio Code* memiliki banyak penggemar dalam mengembangkan aplikasi kedepannya(Wilyanto dkk., 2023).

2.3.5 *Python*

Python merupakan sebuah bahasa pemrograman yang biasanya digunakan pada saat akan membuat sebuah aplikasi berbasis *website*. Python bersifat interpratif dengan banyak fungsi yang dapat dijalankan yang memiliki filosofi perancangan yang mengacu pada level keterbacaan kode. Python sendiri juga dapat dikatakan sebagai bahasa pemrograman yang dapat menyatukan kemampuan serta kapabilitas dengan formula kode yang sangat jelas dan juga memiliki sebuah fungsionalitas pustaka standar yang cukup banyak dan bersifat menyeluruh (Jaka Naufal Semendawai dkk., 2021).

2.3.6 *Hypertext Markup Language (HTML)*

Hypertext Markup Language (HTML) adalah salah satu Bahasa pemrograman yang digunakan untuk membuat website absensi pegawai kantoran ini. *HTML* dominan dengan menggunakan tanda tag (< >) untuk menyatakan kode-kode yang akan ditafsirkan oleh browser agar halaman dapat ditampilkan dan muncul sesuai dengan posisi yang telah diatur. Bahasa *HTML* ini sendiri digunakan untuk membantu merancang struktur dasar halaman *website* atau bila dianalogikan *HTML* merupakan pondasi awal untuk Menyusun berdirinya kerangka halaman *website* secara lebih terstruktur sebelum masuk ke tahap desain dan sisi fungsionalitas. *HTML* nantinya akan dikolaborasikan dengan Bahasa pemrograman *CSS*(Sari dkk., 2022).

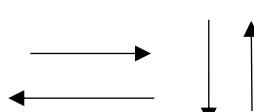
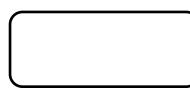
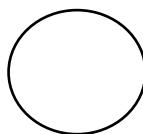
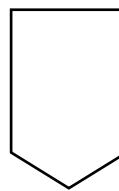
2.3.7 *Cascading Style Sheet (CSS)*

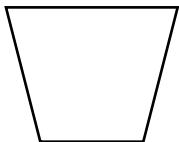
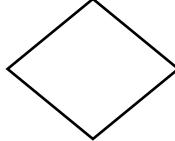
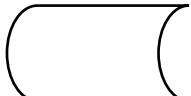
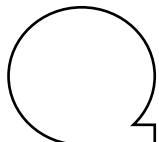
CSS (Cascading Style Sheets) adalah bahasa pemrograman yang digunakan untuk mengatur tampilan dan desain elemen-elemen pada halaman web yang ditulis menggunakan *HTML*. CSS memungkinkan pengembang web untuk mengubah warna, ukuran, posisi, font, dan tata letak halaman secara konsisten dan efisien. *CSS* ialah sebuah Bahasa yang dapat digunakan untuk mendefinisikan bagaimana dalam lima belas atau 15 bahasa mark up yang akan ditampilkan dalam suatu media yang dimana Bahasa markup ini salah satunya adalah *HTML*.(Putra dkk, 2023)

2.3.8 Flowchart

Flowchart adalah representasi grafis yang menggambarkan langkah-langkah dan urutan dalam sebuah prosedur atau program. *Flowchart* berfungsi membantu proses analisis, perancangan, dan pengkodean dengan memecah masalah menjadi bagian-bagian lebih kecil agar lebih mudah dioperasikan (Maharani dkk., 2024).

Tabel 2.2 *Flowchart*

Simbol	Keterangan
	<p>Flowchart Direction Symbol Simbol ini digunakan untuk menghubungkan antara simbol yang satu dengan simbol yang lain.</p>
	<p>Terminator Symbol <i>Symbol</i> ini digunakan untuk permulaan (<i>start</i>) Atau akhir (<i>stop</i>).</p>
	<p>Connector Symbol Simbol untuk keluar masuk atau Penyambungan proses dalam lembar/halaman yang sama.</p>
	<p>Connector Symbol Simbol untuk keluar-masuk atau Penyambungan proses dalam lembar/halaman Yang berbeda.</p>
	<p>Processing Symbol Simbol yang menunjukkan pengolahan yang Tidak dilakukan oleh komputer.</p>

	<p>Symbol Manual Operation Simbol yang menunjukkan pengolahan yang Tidak dilakukan oleh komputer.</p>
	<p>Symbol Decision Simbol pemilihan proses berdasarkan kondisi yang ada.</p>
	<p>Symbol Manual Input Simbol ini digunakan untuk pemasukan data Secara manual.</p>
	<p>Symbol Preparation Simbol ini digunakan untuk mempersiapkan penyimpanan yang akan digunakan sebagai tempat pengolahan di dalam <i>storage</i>.</p>
	<p>Symbol Predefine Proses Simbol untuk pelaksanaan suatu bagian (<i>sub program</i>)/<i>procedure</i>.</p>
	<p>Symbol Display Simbol yang menyatakan peralatan <i>output</i> yang digunakan yaitu layar, plotter, printer dan sebagainya.</p>
	<p>Symbol Disk and On-line Storage Simbol yang menyatakan input yang berasal dari disk atau disimpan ke disk.</p>
	<p>Symbol Magnetik tape unit Simbol yang menyatakan input berasal dari pita magentik atau <i>output</i> disimpan ke pita magnetik.</p>

	Symbol Punch Card Simbol yang menyatakan bahwa input berasal dari kartu atau <i>output</i> ditulis ke kartu.
	Symbol Dokumen Simbol yang menyatakan input berasal dari dokumen dalam bentuk kertas atau <i>output</i> dicetak ke kertas.

(Sumber : <https://itbox.id/blog/flowchart-adalah/>)

2.3.9 Perancangan Sistem UML

Dalam pengembangan sistem perpustakaan ini, pemodelan sistem menggunakan pendekatan *Unified Modeling Language (UML)* untuk memvisualisasikan dan mendokumentasikan aspek-aspek penting dari sistem. Pemodelan *UML* yang digunakan mencakup tiga jenis diagram utama: *Use Case Diagram*, *Activity Diagram*, dan *Sequence Diagram*. Setiap diagram memberikan perspektif yang berbeda tentang sistem yang dikembangkan *UML* (*Unified Modeling Language*) adalah sebuah bahasa standar yang digunakan untuk memvisualisasikan, merancang, dan mendokumentasikan sistem perangkat lunak. *UML* tidak hanya digunakan untuk menggambarkan struktur dan perilaku sistem, tetapi juga untuk menggambarkan hubungan antara berbagai komponen dalam sistem tersebut. *UML* digunakan oleh pengembang perangkat lunak untuk merancang, menganalisis, dan mendokumentasikan perangkat lunak secara lebih jelas dan terstruktur(Asari dkk., 2025).

1. *Use case Diagram*

Dalam pemrograman berbasis *OOP*, untuk mendeskripsikan sistem dapat menggunakan diagram *UML*. Diagram tersebut terdiri dari 13 jenis diagram yaitu *activity*, *class*, *communication*, *component*, *composite structure*, *deployment*, *interaction overview*, *object*, *package*, *squence*, *state machine*, *timing* dan *use case*. *Use case* merupakan deskripsi fungsi dari sebuah sistem dari perspektif atau sudut pandang para pengguna sistem. *Use case* mendefinisikan apa

yang akan diproses oleh sistem dan komponen –komponennya. *Use case* bekerja dengan menggunakan *scenario* yang merupakan deskripsi dari urutan atau langkah –langkah yang menjelaskan apa yang dilakukan oleh *user* terhadap sistem maupun sebaliknya. *Use case* mengidentifikasi fungsionalitas yang dipunya sistem, interaksi user dengan sistem dan keterhubungan antara *user* dengan *fungsionalitas* sistem. *DSRM* terdiri dari enam tahapan yaitu *problem identification and motivation, objective of the solution, Design and development, demonstration, evaluation dan communication*(Setiyani dkk., 2021).

Tabel 2. 3 Perancangan Sistem UML

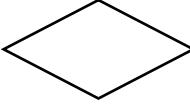
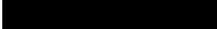
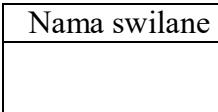
Simbol	Keterangan
	Aktor: Mewakili peran orang, <i>system</i> yang lain, atau alat ketika berkomunikasi dengan <i>use case</i>
	<i>Use case</i> : Abstraksi dan interaksi antara sistem dan aktor
	<i>Association</i> : Abstraksi dari penghubung antara aktor dengan <i>use case</i>
	Generalisasi: Menunjukkan spesialisasi aktor untuk dapat berpartisipasi dengan <i>use case</i>
	Menunjukkan bahwa suatu <i>use case</i> seluruhnya merupakan fungsionalitas dari <i>use case</i> lainnya
	Menunjukkan bahwa suatu <i>use case</i> merupakan tambahan fungsional dari <i>use case</i> lainnya jika suatu kondisi terpenuhi

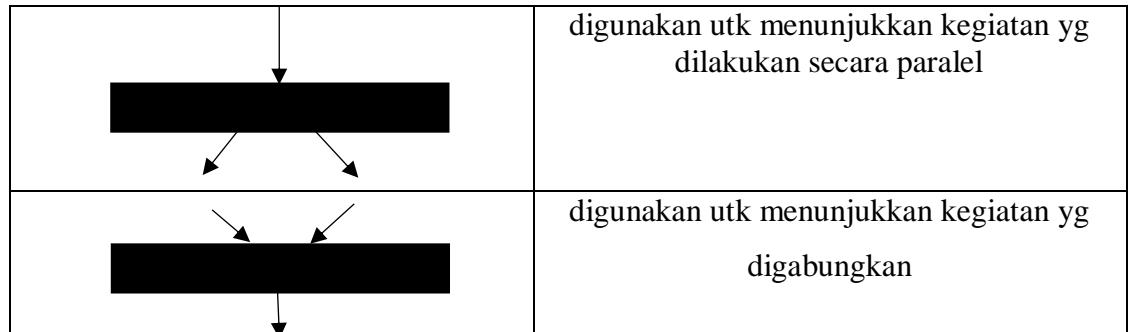
(Sumber: <https://www.dicoding.com/blog/contoh-use-case-diagram/>)

2. *Activity Diagram*

Activity diagram atau disebut juga dengan diagram aktivitas, merupakan diagram yang menggambarkan aktivitas dari sebuah sistem, bagaimana sistem melakukan suatu aktivitas dalam menjalankan fungsi tertentu. *activity diagram* pada sistem terdiri dari *activity diagram* data karyawan, *activity diagram* data jabatan, *activity diagram* data gaji, *activity diagram* manajemen user dan *activity diagram* laporan.

Tabel 2. 4 *Activity Diagram*

Simbol	Keterangan
	status awal aktivitas sistem, sebuah Diagram aktivitas memiliki sebuah status awal
	aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja
	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu
	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu
	status akhir yang dilakukan sistem, sebuah Diagram aktivitas memiliki sebuah status akhir
	memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi



(Sumber: <https://www.dicoding.com/blog/apa-itu-activity-diagram/>)

2.3.10 UI/UX

UI (User Interface) merupakan *input* dan *output* yang melibatkan interaksi langsung antara sistem dan pengguna akhir, terutama melalui media visual seperti komputer, *smartphone*, dan *tablet*. *UI* mencakup aspek desain yang memudahkan pengguna dalam mengerti dan berinteraksi dengan aplikasi. Sementara itu, *UX (User Experience)* merupakan persepsi dan respons pengguna terhadap penggunaan produk, sistem atau jasa yang mencakup penilaian kepuasan dan kenyamanan pengguna selama interaksi mereka. *Web development* didefinisikan sebagai proses dimana seorang *web developer*, yang merupakan seorang *programmer*, mengembangkan aplikasi *World Wide Web* menggunakan model *client-server*. *Web development* meliputi tugas-tugas seperti membuat dan mengembangkan *website*, termasuk menciptakan elemen-elemen seperti tombol, menu, kotak pencarian,

dan sebagainya. *UI* dan *UX* dalam pengembangan web sangat krusial dalam menentukan keberhasilan sebuah situs *website* atau aplikasi. Desain *UI* yang intuitif dan menarik tidak hanya memudahkan pengguna dalam berinteraksi dengan teknologi, tetapi juga meningkatkan keterlibatan dan kepuasan mereka. Sementara itu, pengalaman pengguna yang positif (*UX*) berkontribusi pada persepsi keseluruhan dan kepuasan pengguna, yang sangat penting dalam mempertahankan pengguna dan mendorong penggunaan berulang (Pateman & Pramudia dkk., 2024)

2.3.11 Metode Pengembangan Sistem *SDLC*

SDLC merupakan rangkaian proses untuk menggambarkan bagaimana merancang, mengembangkan, memelihara, dan meningkatkan efisiensi produk perangkat lunak (Viller & Sommerville, 2000). *SDLC* adalah proses mengembangkan atau mengubah suatu sistem perangkat lunak dengan menggunakan model-model dan metodologi yang digunakan orang untuk mengembangkan sistem-sistem perangkat lunak sebelumnya. *SDLC* memiliki beberapa Model dalam penerapan tahapan prosesnya antara lain model *Sequential Model* atau *Waterfall*, *Parallel Model*, *Iterative Model*, *Prototyping Model*,

RAD (Rapid Application Development) Model, *Spiral Model*, *V Shaped Model* dan *Agile Development*(Rahmi dkk., 2023).



Gambar 2.2 Metode Pengembangan Sistem *SDLC*

(Sumber : <https://www.dicoding.com/blog/metode-sdlc/>)

1. *Planning*

Perencanaan/*Planning* adalah mencari masa depan dan memutuskan masa depan organisasi sementara, perencanaan strategis adalah upaya yang membantu dalam membentuk tindakan organisasi yang diperlukan untuk mengambil keputusan, serta Perencanaan strategis membutuhkan setiap organisasi dalam melihat lingkungan di mana organisasi berada bekerja dan membantu memfokuskan perhatiannya pada tantangan dan masalah kritis. Perencanaan juga membantu para pemimpin dari organisasi untuk memfokuskan perhatian mereka pada masalah-masalah yang dihadapi dan mengambil keputusan solusi dari masalah yang dihadapi panduan tentang sebuah

organisasi dalam melakukan kegiatan organisasinya. Perencanaan juga bertindak sebagai peta jalan(Yenny Dwi Suharyani dkk., 2023).

2. *Design*

Design adalah aktivitas untuk menciptakan solusi atau produk baru yang memenuhi tujuan tertentu dalam situasi tertentu. Simon menggambarkan desain sebagai "ilmu yang berbasis pada desain," yaitu cabang ilmu yang membahas bagaimana kita mengembangkan solusi untuk masalah yang belum pernah ada sebelumnya. Tahapan desain dimulai dengan pengumpulan ide dan konsep, yang kemudian diwujudkan dalam sketsa-sketsa awal untuk menggambarkan bentuk, ukuran, dan fungsi produk yang diinginkan. Proses desain melibatkan tidak hanya pengembangan konsep yang inovatif tetapi juga penerapan nilai estetika yang tinggi serta pemilihan teknik finishing yang tepat untuk memastikan produk akhir memiliki tampilan yang menarik dan berkualitas tinggi(Azzahra dkk., 2025).

3. *Coding*

Coding adalah proses untuk menulis kode dalam bahasa pemrograman untuk berkomunikasi dengan komputer sehingga kita mempu memberikan perintah/tugas kepadanya secara spesifik. Proses pengkodean komputer dilakukan dengan memasukkan karakter (huruf, angka, tanda baca, dan simbol) menjadi format khusus atau menulis kode dalam bahasa pemrograman. Komputer memproses kode tersebut dan mengirimkannya kembali kepada pengguna sebagai data keluaran(Penelitian & Pendidikan dkk., 2025).

4. *Testing*

Testing adalah Pengujian mencakup input nilai unit yang valid, nilai negatif, kombinasi angka dan huruf, nilai kosong, serta angka desimal. Hasil yang diharapkan adalah penerimaan *input* valid untuk disimpan ke sistem, penolakan *input* tidak valid, dan pemberian pesan error yang sesuai. Validasi ini bertujuan memastikan bahwa data unit yang dimasukkan sesuai dengan aturan sistem yang ditetapkan (Zahra dkk., 2024).

5. Analisis

Analisis adalah proses mempelajari, menguraikan, dan memahami suatu permasalahan atau sistem secara mendalam untuk mendapatkan kesimpulan atau solusi yang tepat (Melinda dkk., 2023).

6. Perawatan

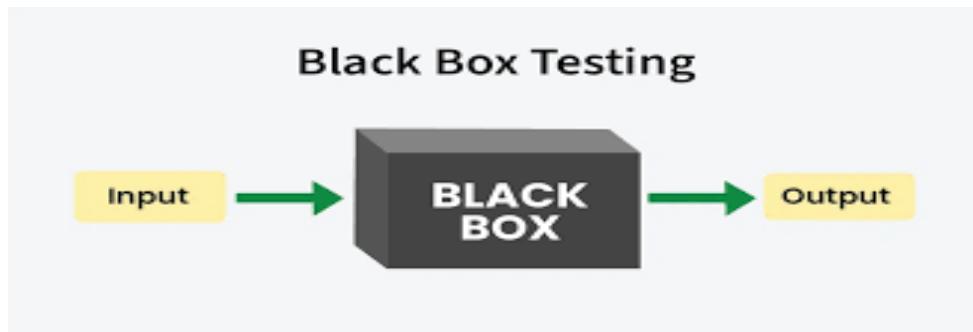
Tahap Perawatan adalah fase akhir dalam siklus hidup pengembangan sistem (SDLC) yang bertujuan untuk memastikan bahwa sistem yang telah selesai dikembangkan tetap berjalan dengan baik, aman, dan relevan dengan kebutuhan pengguna seiring waktu. Tahap ini dilakukan setelah sistem diimplementasikan dan digunakan secara aktif (Wisnu Sukma Maulana dkk., 2022).

2.3.12 *Figma*

Figma adalah sebuah web-based design tool atau alat desain berbasis web yang digunakan untuk membuat *user interface* (UI) dan *user experience* (UX), seperti desain tampilan aplikasi, *website*, hingga prototipe interaktif. Figma memungkinkan banyak pengguna untuk berkolaborasi secara *real-time*, seperti *Google Docs*, sehingga sangat efisien untuk kerja tim dalam proyek desain (Senubekti dkk., 2024).

2.3.13 *Black box*

Metode *Black Box Testing* merupakan pengujian untuk menunjukkan kesalahan pada system aplikasi seperti kesalahan pada fungsi sistem aplikasi, serta menu aplikasi yang hilang. Jadi *Black Box testing* merupakan metode uji fungsionalitas sistem aplikasi. Dalam melakukan pengujian menggunakan masukan data acak dengan tujuan untuk mendapatkan hasil yang pasti. Dikatakan pasti artinya bila salah ,maka di tolak oleh sistem informasi atau data input tersebut tidak dapat disimpan dalam data base, sedangkan bila data input benar maka dapat di terima / masuk di database sistem informasi (Uminingsih dkk., 2022).



Gambar 2.3 *Black Box Testing*

(Sumber: <https://www.geeksforgeeks.org/software-engineering-black-box-testing/>)

2.3.14 Usability Testing

Metode ini digunakan untuk menentukan keberhasilan suatu sistem. Evaluasi sangat penting dilakukan karena tingkat *usability* dari sebuah aplikasi (baik *website* maupun *mobile*) akan mempengaruhi kemudahan penggunaan. Selain itu, evaluasi juga dapat meningkatkan kepuasan pengguna, meningkatkan efisiensi, serta meminimalkan risiko kesalahan dalam penginputan data. Evaluasi ini mencakup aspek efisiensi, efektivitas, dan kepuasan pengguna(Yulianti dkk., 2024).

Keterangan :

Y = Nilai persentase yang dicari (...%).

X = jumlah nilai kategori jawaban dikalikan dengan frekuensi ($\sum = N \times R$).

N = Nilai dari setiap jawaban.

R = Frekuensi.

Skor *IDE* al = Nilai tertinggi dilakukan jumlah sampel.

BAB III

METODE PENELITIAN

3.1 Lokasi dan Waktu Penelitian

Lokasi dan waktu penelitian merupakan dua komponen penting dalam metodologi penelitian yang memberikan konteks spesifik tentang di mana dan kapan penelitian dilakukan. Lokasi Penelitian menjelaskan tempat atau wilayah pengambilan data, Sementara waktu Penelitian mencakup periode pelaksanaan, mulai dari persiapan, pengumpulan data, hingga analisis.

3.1.1 Lokasi Penelitian

Lokasi peneltian yang di gunakan dalam penelitian ini adalah hasil dari operasi waktu perjalanan yang di lakukan dari kota sorong ke aimas yang di mulai dari beberapa seperti Universitas Muhammadiyah Kota sorong, Alun – Alun Aimas, UNIMUDA, GKI Maranatha kota sorong, Tembok Berlin Kota sorong dan lain – lain.



Gambar 3.1 Lokasi Dinamis

(Sumber: *Aplikasi Inshot*)

3.1.2 Waktu Penelitian

Waktu penelitian Di perkirakan membutuhkan waktu kurang lebih tiga bulan. Waktu penelitian dapat di lihat pada tabel di bawah ini.

Berikut merupakan tabel waktu penelitian yang di lakukan :

Tabel 3.5 Jadwal Penelitian

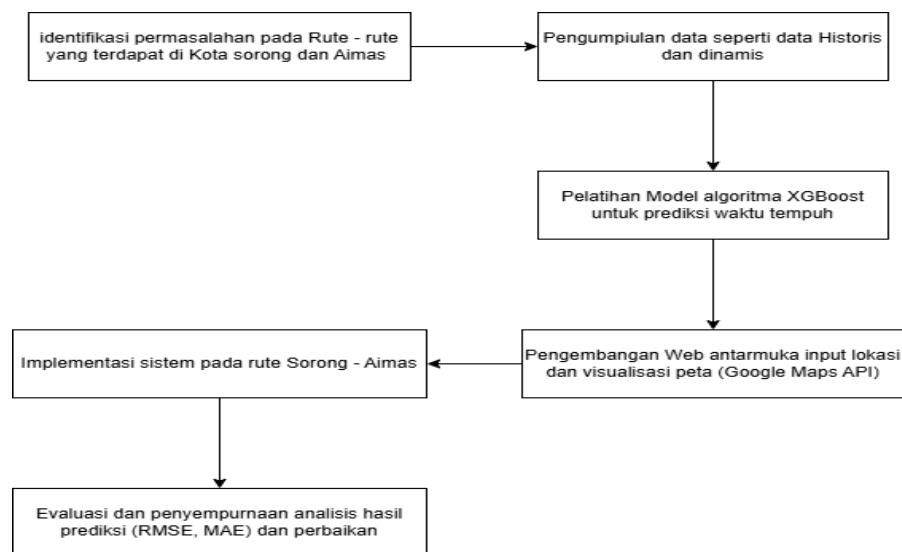
NO	Kegiatan	Waktu											
		Mei				Juni				Juli			
		Minggu Ke -		Minggu Ke -		Minggu Ke -		Minggu Ke -		Minggu Ke -		Minggu Ke -	
	Tanggal	1	2	3	4	1	2	3	4	1	2	3	4
1.	Pengajuan Judul												
2.	Penyusunan Jurnal												
3.	Pengujian Jurnal Penelitian												
4.	Studi Literatur												
5.	Survei Tempat												
6.	Observasi												
7.	Pembuatan dataset												
8.	Traninng dataset												
9.	Mendesain Kerangka Sistem												

3.2 Jenis Penelitian

Penelitian ini merupakan penelitian merupakan deskriptif kualitatif dengan pendekatan studi lapangan yang bertujuan untuk menganalisis waktu tempuh perjalanan dari kota sorong ke aimas berdasarkan beberapa titik keberangkatan dan destinasi. Metode ini dipilih untuk memberikan pemahaman mendalam tentang pola pergerakan, Faktor – faktor yang memengaruhi durasi perjalanan , serta kondisi aktual di lapangan.

3.3 Kerangka berpikir/konsep

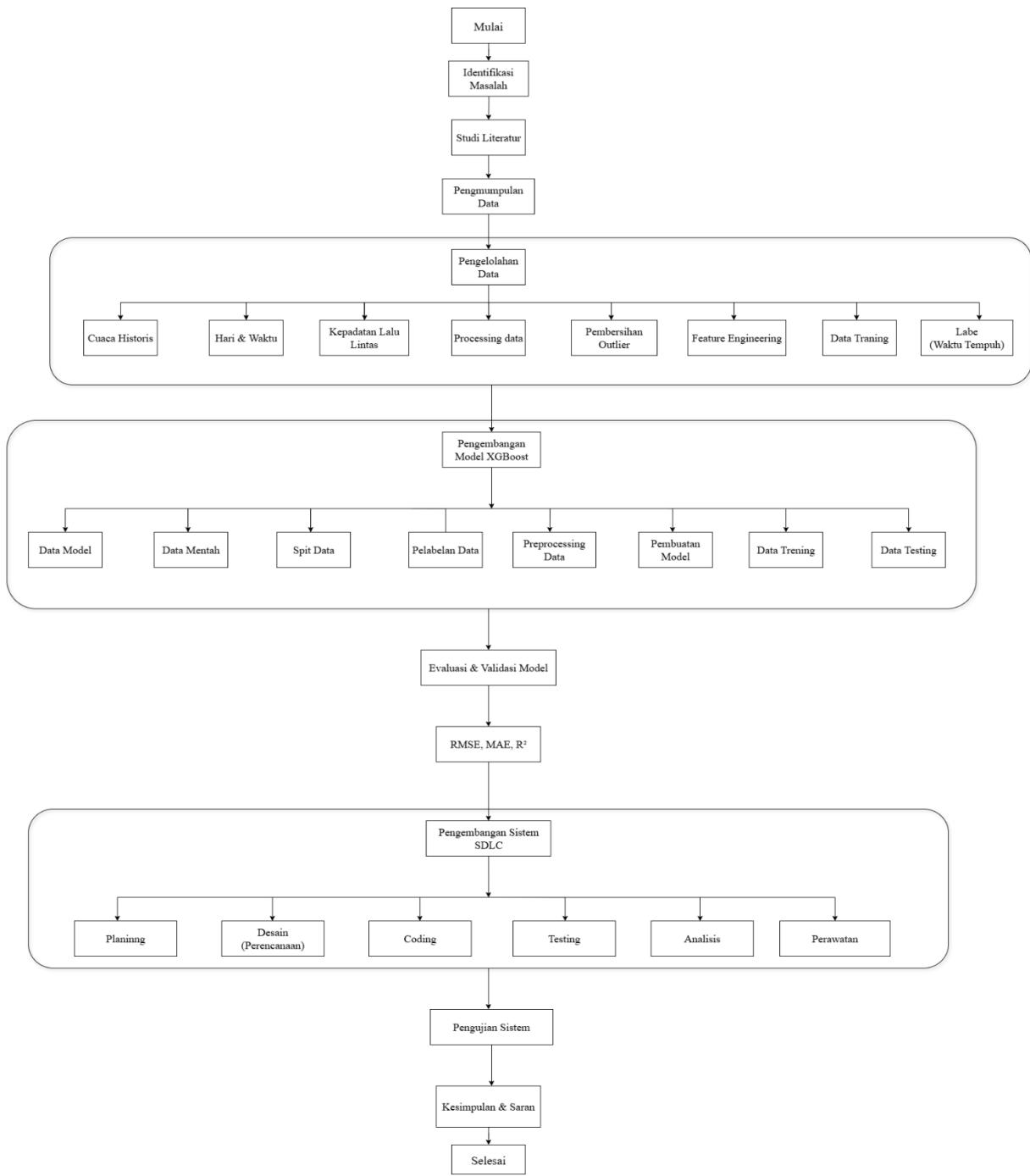
Sebelum memulai penelitian, perlu disusun kerangka berpikir yang mencangkup identifikasi masalah, penentuan tujuan dan manfaat penelitian, pengumpulan data primer dan sekunder, serta analisis data untuk mencapai kesimpulan yang relevan.



Gambar 3.2 Kerangka Berpikir
(Sumber: <https://app.diagrams.net/>)

3.4 Alur Penelitian

Alur Penelitian adalah serangkaian tahapan atau proses sistematis yang dilakukan oleh peneliti dalam melaksanakan penelitian, mulai dari perencanaan hingga pelaporan hasil penelitian.



Gambar 3.3 Alur Penelitian

(Sumber: <https://app.diagrams.net>)

3.4 Tahap Penelitian

Tahapan penelitian adalah serangkaian langkah sistematis yang dilakukan peneliti untuk menjawab pertanyaan atau masalah penelitian secara terstruktur.

3.4.1 Identifikasi Masalah

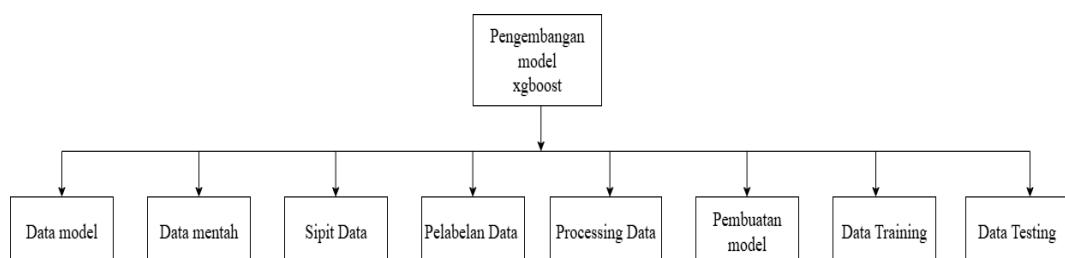
Langkah awal yang di lakukan dalam penelitian ini adalah mengidentifikasi permasalahan yang hendak diteliti. Tahap ini dilakukan setelah mendapat dan menentukan topik penelitiannya.

3.4.2 Studi Literatur

Tahap kedua yang akan di lakukan dalam penelitian ini adalah dengan mencari dan mempelajari referensi dari berbagai jurnal ataupun artikel yang berkaitan dengan judul penelitian yang kami lakukan yaitu mengenai Implementasi Prediksi Waktu Tempuh Perjalanan Sorong – Aimas Berbasis Website Menggunakan Model *Xgboost* dan Fitur Lokasi Dinamis.

3.4.3 Pengumpulan Data

Tahap Ketiga yang akan dilakukan dalam penelitian ini yaitu dengan mengumpulkan data apa saja yang dapat mempengaruhi prediksi waktu seperti Kepadatan Lalu – lintas, cuaca, Hari, dan lain – lain data ini di lakukan dengan melakukan perjalanan dari beberapa titik tertentu yang ada di Kota Sorong dan Aimas, lalu kemudian kami melakukan *tracking* seperti waktu tempuh, laju kendaraan, dan lain – lain.



Gambar 3.4 Pengembangan Model Xgboost

(Sumber: <https://app.diagrams.net/>)

Tahap kempat yang dilakukan peneliti dalam penelitian ini adalah Pengembangan model *XGBoost* berperan sebagai inti dari proses prediksi dalam penelitian ini. *XGBoost* (*Extreme Gradient Boosting*) adalah algoritma pembelajaran mesin berbasis *ensemble* yang digunakan untuk memodelkan

hubungan kompleks antara berbagai fitur (seperti cuaca, kepadatan lalu lintas, jarak, dan waktu keberangkatan) dengan target yang ingin diprediksi, yaitu waktu tempuh perjalanan. Melalui proses pengembangan ini, *XGBoost* dilatih menggunakan data historis untuk menangkap pola-pola tersembunyi dan mengoptimalkan akurasi prediksi. Pengembangan model mencakup perencanaan fitur, pemilihan parameter, hingga evaluasi hasil prediksi berdasarkan data uji.

Berdasarkan gambar di atas alur model dapat dijelaskan sebagai berikut :

1. Data Mentah

Peneliti mengumpulkan dan menyiapkan data mentah dari hasil pengumpulan awal sebagai dasar *Dataset*.

2. Split Data

Dataset dibagi menjadi beberapa kelas yang sesuai dengan kategori gejala atau jenis penyakit yang ingin diidentifikasi.

3. Pelabelan Data .

Setiap data dalam *Dataset* diberi label atau nama yang sesuai dengan elas yang telah ditentukan, guna memudahkan proses klasifikasi.

4. *Preprocessing* Data .

Gambar yang terkumpul diolah, misalnya di-resize, diputar atau di ubah ukurannya, sehingga siap untuk diinput ke dalam CNN dalam format yang sesuai.

5. Pembuatan Model

Peneliti membangun model prediksi waktu perjalanan Sorong–Aimas menggunakan algoritma *XGBoost*. Model dilatih dengan *dataset* yang mencakup fitur seperti cuaca, jam berangkat, hari, kepadatan lalu lintas, jumlah persimpangan, dan jarak tempuh. Data terlebih dahulu diproses melalui tahapan *encoding* dan pembagian menjadi data latih dan data uji. Setelah dilatih, model dievaluasi menggunakan metrik *RMSE*, *MAE*, dan *R²* untuk mengukur akurasi. Model yang telah terbentuk kemudian diimplementasikan ke dalam sistem *website* untuk memberikan prediksi waktu tempuh secara otomatis.

6. Data Training

Model *XGBoost* dilatih menggunakan data latih agar mampu mengenali pola hubungan antara faktor-faktor seperti cuaca, kepadatan, dan waktu keberangkatan dengan waktu tempuh perjalanan dari Sorong ke Aimas.

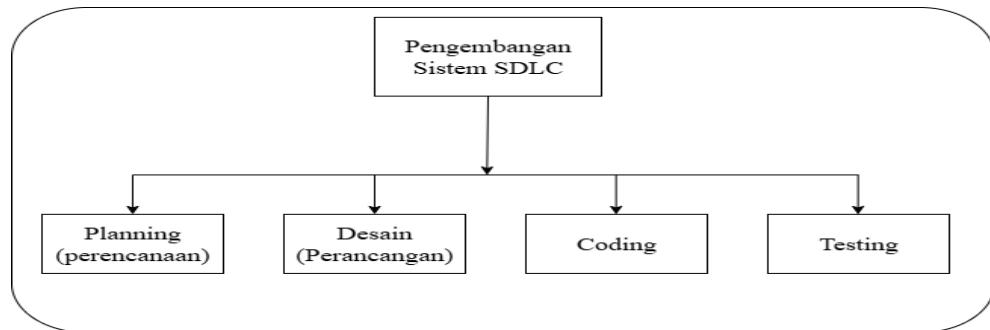
7. Data Testing

Setelah pelatihan selesai, data uji digunakan untuk mengukur kinerja model dalam mengenali atau mengklasifikasikan gambar yang belum pernah dilihat sebelumnya, mengevaluasi akurasi dan efektivitas model.

8. Data Model

Model *XGBoost* yang telah dilatih disimpan dalam format .json agar mudah digunakan kembali tanpa perlu proses pelatihan ulang. Model ini kemudian diintegrasikan ke dalam sistem berbasis web untuk memberikan prediksi waktu tempuh perjalanan Sorong–Aimas secara otomatis dan real-time kepada pengguna.

3.4.4 Pengembangan Sistem SDLC(*Software Development Life Cycle*)



Gambar 3.5 SDLC (*Software Development Life Cycle*)

(Sumber: <https://app.diagrams.net/>)

Tahap kempat yang dilakukan peneliti dalam penelitian ini adalah model pengembangan sistem. Mode pengembangan system yang di gunakan dalam penelitian ini adalah *extreme SDLC (Software Development Life Cycle)* yang mana memiliki 4 tahap sebagai berikut:

1. Planning(perencanaan)

Pada tahap *planning* dalam metode *SDLC(Software Development Life Cycle)*, peneliti menganalisis kebutuhan sistem secara menyeluruh, baik dari

sis fungsional maupun non-fungsional, untuk memastikan sistem dapat dibangun secara efektif dan sesuai dengan tujuan penelitian.

a. Analisis Kebutuhan Fungsional

Kebutuhan fungsional menjelaskan fungsi-fungsi utama yang harus dimiliki oleh sistem prediksi waktu tempuh berbasis *website*. Adapun kebutuhan fungsional yang diidentifikasi adalah:

1. Sistem berbasis *website* yang dapat diakses oleh pengguna melalui browser.
2. Pengguna dapat memasukkan data seperti lokasi awal, lokasi tujuan, dan waktu keberangkatan.
3. Sistem memproses *input* pengguna dan menghasilkan prediksi waktu tempuh menggunakan model *XGBoost*.
4. Hasil prediksi ditampilkan dalam antarmuka peta yang interaktif (menggunakan *Google Maps* atau *Leaflet*).
5. Sistem menampilkan informasi pendukung seperti estimasi jarak, rute perjalanan, dan waktu historis.
6. Pengguna dapat melihat visualisasi rute perjalanan dan titik-titik pengamatan yang digunakan dalam pelatihan model.
7. Sistem dapat memberikan estimasi waktu tempuh secara real-time atau mendekati kondisi aktual berdasarkan parameter masukan.

b. Analisis Kebutuhan Non-Fungsional

Kebutuhan non-fungsional adalah kebutuhan teknis yang berkaitan dengan performa, perangkat, dan teknologi yang digunakan untuk mendukung pengembangan aplikasi *website* ini.

1. Perangkat Keras

Laptop dengan spesifikasi sebagai berikut:

- a) Prosesor: *Intel Core i7-12700H* atau setara
- b) *RAM*: 16 GB DDR4
- c) Penyimpanan: *SSD NVMe* minimal 512 GB
- d) *GPU*: Disarankan (untuk pelatihan model)

2. Perangkat Lunak:

- a) Sistem Operasi: Windows 11

- b) *Browser: Google Chrome* (untuk mengakses aplikasi web)
- c) Platform pemrograman:
 - 1) *Google Colab* (untuk pelatihan model *XGBoost*)
 - 2) *Visual Studio Code* (untuk pengembangan *front-end* dan *back-end* aplikasi)
- 3. Bahasa dan *library*:
 - a. *Python* (untuk pengolahan data & pemodelan)
 - b. *HTML, CSS, JavaScript* (untuk antarmuka web)
 - c. *Leaflet.js* atau *Google Maps API* (untuk visualisasi peta)
- 4. Tools pendukung:
 - a. *Figma* (untuk desain *UI/UX*)
 - b. *Flask* atau *Streamlit* (jika *back-end Python* digunakan)
 - c. *Git* (untuk version control dan kolaborasi)

2. *Design* (perancangan)

Pada tahap perancangan ini akan dibuat sebuah *visualisasi Flowchart Diagram, Use Case Diagram, Activity Diagram* dan *User Interface*. Tujuan dari penjelasan ini adalah untuk memberikan gambaran yang jelas mengenai alur penelitian yang dilakukan. Gambar sistem yang digunakan dalam bentuk *Use Case Diagram* dan *Flowchart* akan membantu memperjelas proses dan langkah-langkah dalam penelitian ini.

3. Coding

Coding adalah proses penulisan serangkaian instruksi dalam bahasa pemrograman untuk mengimplementasikan algoritma atau logika tertentu agar komputer dapat menjalankan fungsi yang diinginkan. Dalam penelitian ini, coding digunakan untuk mengembangkan sistem prediksi waktu tempuh berbasis website dengan menerapkan algoritma XGBoost. Pada tahap ini, peneliti menulis kode program menggunakan Python di Google Colab untuk melatih model XGBoost dengan data perjalanan (jarak, waktu keberangkatan, kepadatan lalu lintas, dan cuaca). Setelah model selesai dilatih, kode program juga digunakan untuk memproses input dari pengguna, menghasilkan prediksi waktu tempuh, serta menampilkan hasilnya melalui antarmuka web.

yang dibangun dengan kombinasi HTML, CSS, dan JavaScript. Dengan kata lain, coding dalam laporan ini merupakan tahap implementasi dari rancangan sistem yang telah dibuat, sehingga model prediksi dapat dioperasikan secara otomatis oleh komputer. Proses coding juga mencakup optimasi model, pengolahan data, dan integrasi dengan visualisasi peta agar pengguna bisa memperoleh estimasi waktu tempuh secara real-time.

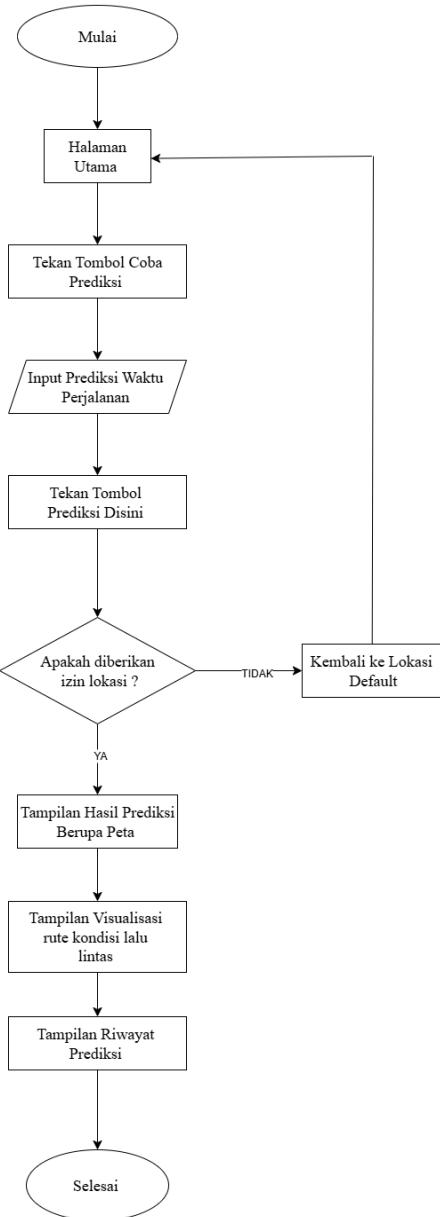
4. Testing

Testing adalah tahap pengujian sistem untuk memastikan bahwa aplikasi prediksi waktu tempuh yang dikembangkan berfungsi sesuai dengan kebutuhan dan berjalan tanpa kesalahan. Pada penelitian "Implementasi Prediksi Waktu Tempuh Perjalanan Sorong–Aimas Berbasis Website Menggunakan Model XGBoost dan Fitur Lokasi Dinamis", pengujian dilakukan untuk memverifikasi kinerja algoritma prediksi serta mengevaluasi kelayakan antarmuka website dari sisi pengguna.

Dalam laporan ini, Testing dilakukan melalui dua metode:

1. *Blackbox Testing* – Digunakan untuk menguji fungsionalitas aplikasi. Pengujian ini hanya berfokus pada input dan output sistem tanpa memperhatikan proses internal kode program. Setiap fitur diuji dengan memasukkan data tertentu (lokasi asal, tujuan, dan waktu keberangkatan), lalu diperiksa apakah output (hasil prediksi waktu tempuh) sesuai dengan yang diharapkan.
2. *Usability Testing* – Digunakan untuk menguji kemudahan penggunaan, kejelasan antarmuka, dan tingkat kepuasan pengguna. Beberapa responden diminta mencoba aplikasi dan memberikan penilaian berdasarkan pengalaman mereka.

a. *Flowchart Sistem*

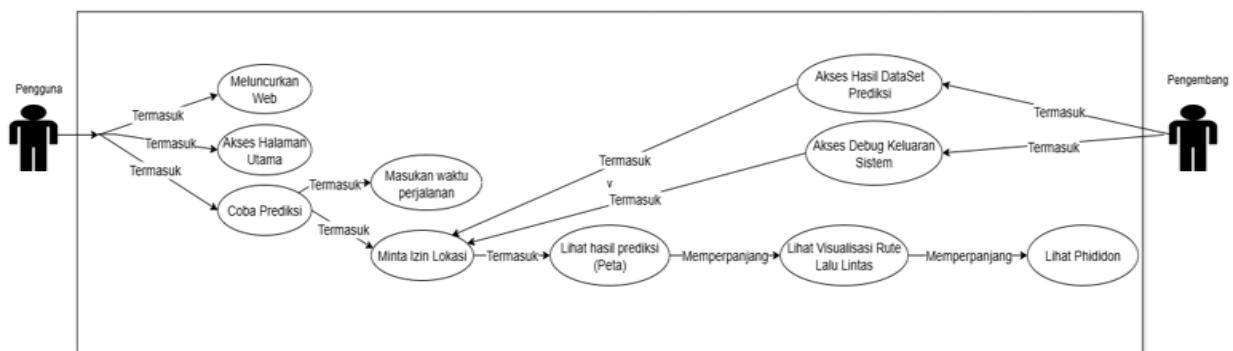


Gambar 3.6 *Flowchart Sistem*

(Sumber: <https://app.diagrams.net/>)

Pada gambar 2.10 *Flowchart* menunjukkan alur sistem prediksi waktu tempuh perjalanan berbasis web. Proses dimulai dari halaman utama, di mana pengguna menekan tombol untuk mencoba prediksi. Setelah itu, pengguna mengisi data perjalanan dan menekan tombol prediksi. Sistem akan meminta izin lokasi; jika tidak diberikan, sistem menggunakan lokasi *default*. Hasil prediksi ditampilkan dalam bentuk peta, disertai visualisasi *rute* dan kondisi lalu lintas. Terakhir, sistem menampilkan riwayat prediksi sebelum proses selesai.

b. Use Case Sistem



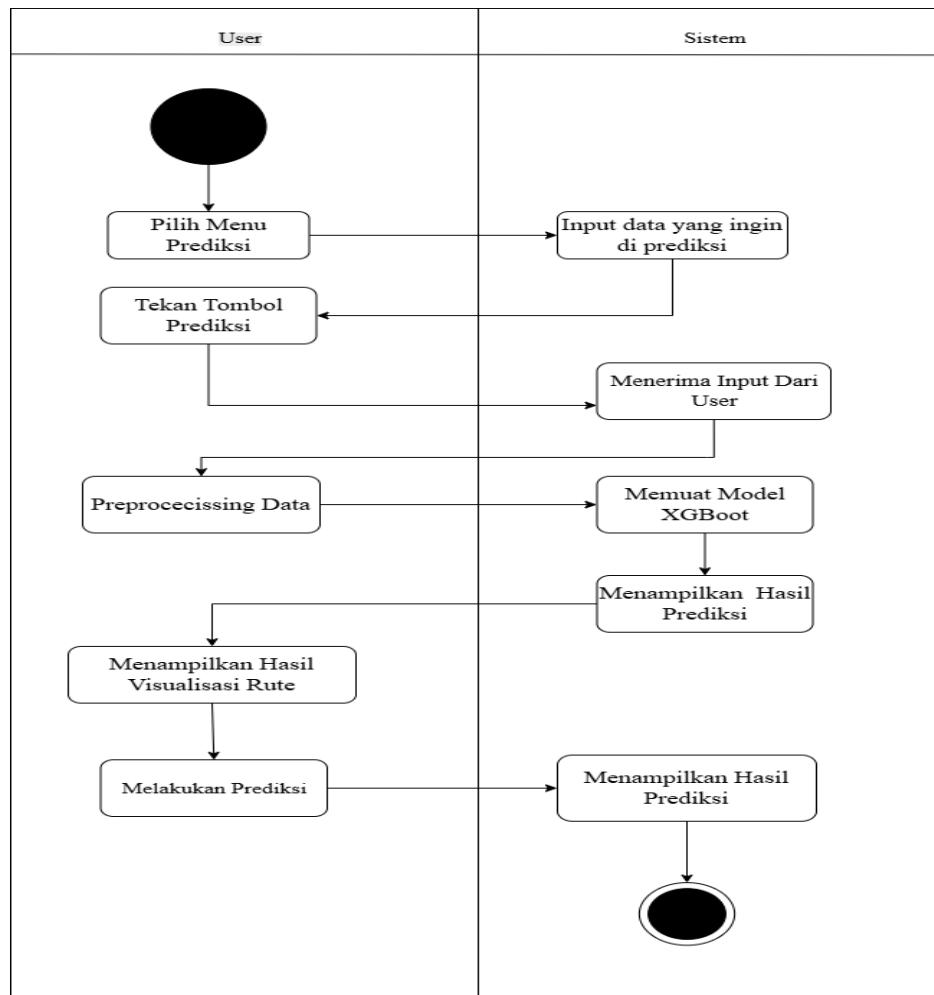
Gambar 3.7 Use Case Sistem

(Sumber: <https://app.diagrams.net/>)

Pada gambar *Use case* diagram tersebut menggambarkan interaksi antara dua aktor utama, yaitu *User* dan *Developer*, dengan sistem prediksi waktu tempuh perjalanan berbasis web. Pengguna (*User*) memulai dengan membuka aplikasi, mengakses halaman utama, dan mencoba fitur prediksi. Dalam proses prediksi, pengguna memasukkan data perjalanan dan memberikan izin lokasi. Setelah itu, pengguna dapat melihat hasil prediksi dalam bentuk peta, visualisasi rute lalu lintas, serta riwayat prediksi. Sementara itu, pengembang (*Developer*) memiliki akses khusus untuk melihat hasil dataset prediksi dan melakukan *debugging output* sistem guna memastikan akurasi dan stabilitas aplikasi. Diagram ini menunjukkan alur fungsionalitas sistem secara umum dari sisi pengguna dan pengembang dalam pengoperasian dan pengujian aplikasi.

c. *Activity Diagram*

- *Activity Diagram Deteksi*

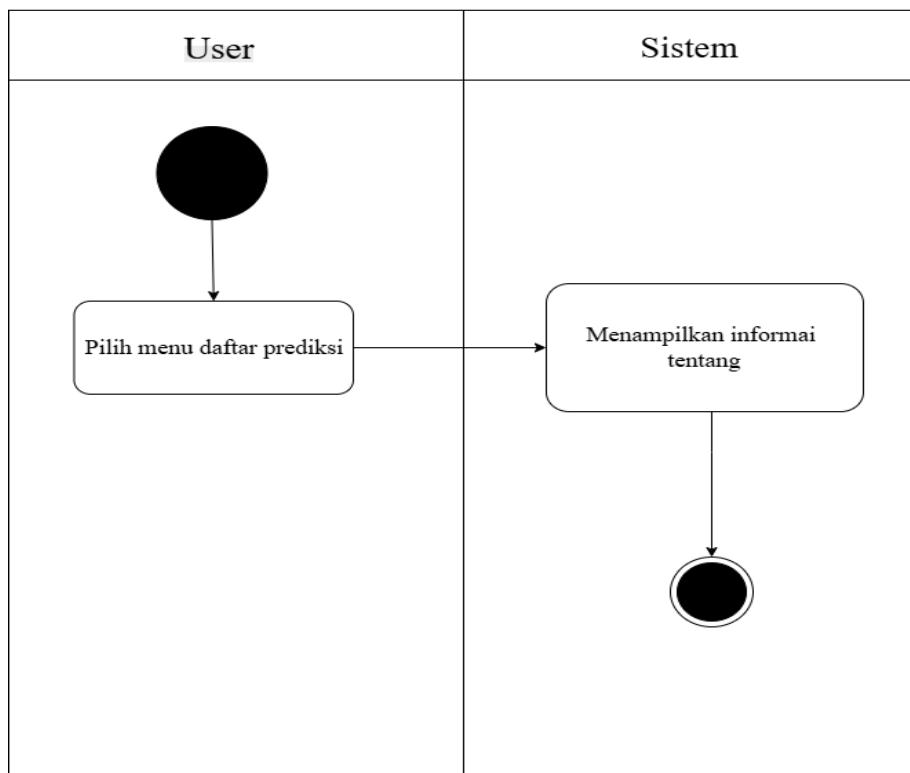


Gambar 3.8 *Activity Diagram* Deteksi

(Sumber: <https://app.diagrams.net/>)

Diagram aktivitas tersebut menggambarkan proses alur kerja prediksi antara pengguna dan sistem. Proses dimulai saat pengguna memilih menu prediksi, kemudian sistem meminta input data yang ingin diprediksi. Setelah pengguna menekan tombol prediksi dan sistem menerima *input*, data akan diproses (*preprocessing*) dan sistem memuat model *XGBoost*. Selanjutnya, hasil prediksi ditampilkan oleh sistem dan pengguna dapat melihat evaluasi rute. Setelah itu, pengguna melakukan prediksi akhir yang kemudian ditampilkan kembali oleh sistem sebagai hasil prediksi akhir. Diagram ini menggambarkan interaksi bertahap antara pengguna dan sistem dalam menghasilkan hasil prediksi berbasis model *machine learning*.

d. *Activity Diagram Tentang*



Gambar 3.9 *Activity Diagram Tentang*

(Sumber: <https://app.diagrams.net/>)

Gambar tersebut merupakan diagram aktivitas (*activity diagram*) yang menggambarkan interaksi antara pengguna (*User*) dan sistem. Proses dimulai dari pengguna yang memilih menu "daftar prediksi". Setelah itu, sistem akan merespons dengan menampilkan informasi yang relevan. Diagram ini menunjukkan alur kerja yang sederhana dan linier, dimulai dari aksi pengguna hingga sistem memberikan *output* berupa informasi yang diinginkan.

e. User Interface (UI Sistem)

User Interface bertujuan untuk memberikan *visualisasi* desain web yang akan dibuat. Tampilan Halaman Utama/Beranda akan muncul pertama kali saat web diakses. Pada menu utama/Beranda, pengguna akan menemukan beberapa pilihan, seperti Tombol untuk masuk ke bagian prediksi, Map, Riwayat Perjalanan, Tentang, Dan kontak. Pada halaman Prediksi, pengguna dapat Mengisi pilihan pilihan yang tersedia seperti input lokasi, tujuan, waktu, hari, cuaca, jenis kendaraan, dan kemudian data – data tersebut akan di kelola menjadi hasil prediksi Sedangkan halaman Tentang menampilkan informasi terkait Web yang kami buat, termasuk tujuan dan pembuatnya. Setiap elemen dalam *User Interface* dirancang untuk memberikan pengalaman pengguna yang intuitif dan informatif.

1) Tampilan Halaman Utama/Beranda



Gambar 3.10 Tampilan Utama Beranda

(Sumber: <https://www.figma.com/>)

Pada Gambar 3.11 Merupakan tampilan Halaman beranda dirancang sebagai pintu masuk utama bagi pengguna untuk memahami tujuan sistem dan fitur unggulannya. Desain ini menggunakan kombinasi warna gradasi biru ke ungu untuk menciptakan nuansa modern dan teknologi. Di bagian atas terdapat *navbar* (navigasi utama) dengan menu: Beranda, Prediksi, Map, Riwayat Perjalanan, Tentang, dan Kontak. Navigasi ini memudahkan pengguna berpindah antar fitur sistem secara cepat. Bagian utama halaman menampilkan judul besar: "Prediksi Waktu Sorong Aimas Dengan Metode

XGBoost” yang menjelaskan secara langsung fungsi utama dari sistem ini, yaitu memberikan estimasi waktu tempuh berdasarkan model pembelajaran mesin *XGBoost*. Di bawahnya terdapat deskripsi singkat yang menegaskan bahwa sistem ini menggunakan data real-time dari *Google Maps*. Terdapat juga tombol aksi utama (*call to action*) bertuliskan “*Coba Prediksi*” yang berfungsi mengarahkan pengguna untuk langsung mencoba fitur utama dari sistem. Di sisi kanan, terdapat kotak informasi fitur utama yang ditampilkan dengan latar transparan gradasi lembut. Fitur-fitur seperti integrasi kondisi cuaca, pemanfaatan data historis, visualisasi peta interaktif, dan penggunaan model *XGBoost* dijelaskan dengan ikon checklist agar mudah dipahami secara visual.

2) Tampilan Prediksi Perjalanan

Gambar 3.11 Tampian Prediksi Perjalanan

(Sumber: <https://www.figma.com/>)

Gambar 3.12 Merupakan Tampilan Halaman “Prediksi Waktu Perjalanan” merupakan inti dari sistem yang memungkinkan pengguna memperoleh estimasi waktu tempuh berdasarkan parameter yang mereka *input*. Desain antarmuka ini dibuat sederhana dan responsif agar mudah digunakan oleh siapa saja, baik di perangkat desktop maupun *mobile*. Pada bagian atas, terdapat dua kolom input lokasi yaitu “Lokasi Kamu” dan “Tujuan”, yang memungkinkan pengguna memasukkan titik keberangkatan dan titik tujuan. Fitur ini terintegrasi dengan *database*

lokasi atau peta agar akurat. Selanjutnya terdapat pilihan *dropdown* untuk memasukkan hari (misalnya Senin) dan waktu keberangkatan pengguna. Input waktu ditampilkan dalam format 24 jam dan dapat disesuaikan secara manual maupun otomatis. Di bawahnya terdapat *dropdown* untuk jenis kendaraan (misalnya: motor) dan kondisi cuaca (misalnya: cerah), yang menjadi parameter penting dalam proses prediksi waktu. Tambahan satu *dropdown* lagi yaitu kondisi waktu (seperti: pagi, siang, malam), memungkinkan sistem mengkalkulasi pengaruh waktu terhadap kondisi lalu lintas. Tombol aksi utama “Prediksi Disini” yang ditampilkan dengan warna gradasi biru-ungu dan ikon kalkulator, menandakan bahwa sistem akan memproses input tersebut menggunakan model *XGBoost* dan menampilkan estimasi waktu tempuh.

3) Tampilan Hasil Prediksi



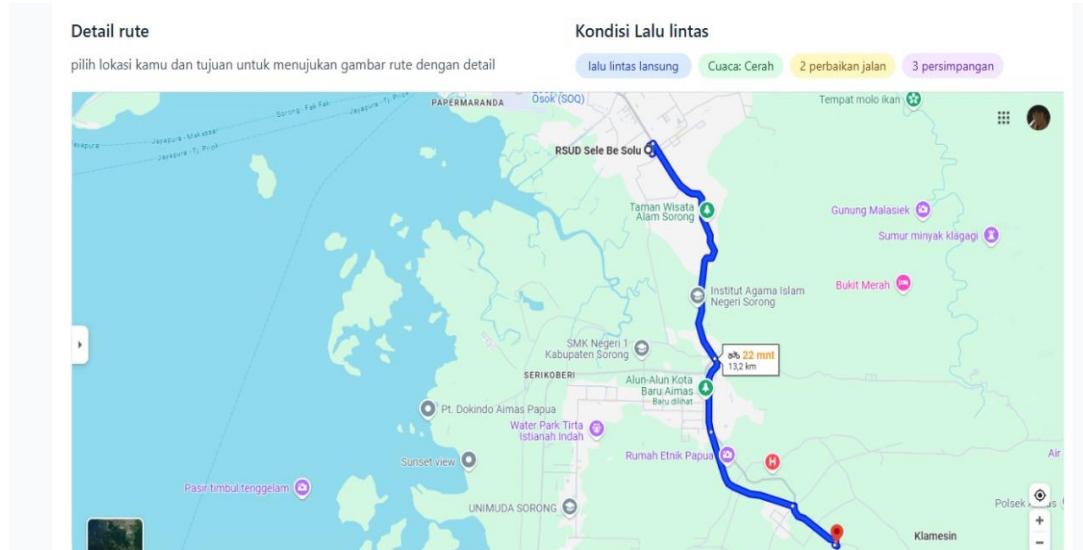
Gambar 3.12 Tampilan Hasil Prediksi

(Sumber: <https://www.figma.com/>)

Gambar 3.13 merupakan tampilan Halaman hasil predksi, halaman ini merupakan hasil akhir dari proses prediksi yang dilakukan oleh sistem setelah pengguna mengisi semua parameter perjalanan. Desain dibuat minimalis, informatif, dan fokus pada hasil utama, yaitu estimasi waktu tempuh perjalanan. Di bagian tengah terdapat elemen visual utama berupa lingkaran biru terang yang menampilkan hasil prediksi waktu tempuh, misalnya "21mnt". Angka ini merupakan hasil keluaran dari model *XGBoost* yang telah diproses berdasarkan data masukan seperti lokasi, waktu, cuaca, dan jenis kendaraan. Lingkaran biru yang bercahaya memberikan efek visual modern dan menarik perhatian, menandakan bahwa ini adalah elemen inti dari sistem. Di bawah angka prediksi terdapat

teks “Hasil Prediksi Kamu” yang memperjelas konteks informasi yang diberikan. Judul “Hasil Prediksi” di bagian kiri memberi penegasan bahwa ini adalah area keluaran. Selain itu, ikon *stopwatch* di pojok kanan atas memberi kesan bahwa sistem ini bekerja secara efisien dan berorientasi waktu.

4) Tampilan Visualisasi Rute



Gambar 3.13 Tampilan Visualisasi Rute

Sumber : *Figma*

Gambar 3.14 merupakan tampilan halaman visualisasi rute, halaman ini menampilkan visualisasi rute perjalanan secara langsung menggunakan *Google Maps API*, berdasarkan *input* lokasi awal dan tujuan yang telah dimasukkan oleh pengguna sebelumnya. Tujuannya adalah untuk memberikan gambaran nyata mengenai jalur yang akan dilalui serta faktor-faktor eksternal yang dapat memengaruhi waktu tempuh. Pada peta ditampilkan jalur biru yang merupakan rute utama dari Jalan Poncowati ke Mega Mall Sorong. Disertakan pula label jarak dan estimasi waktu tempuh menggunakan ikon sepeda atau kendaraan, yang menunjukkan bahwa sistem mendukung estimasi *multi-modality* (berdasarkan jenis kendaraan).

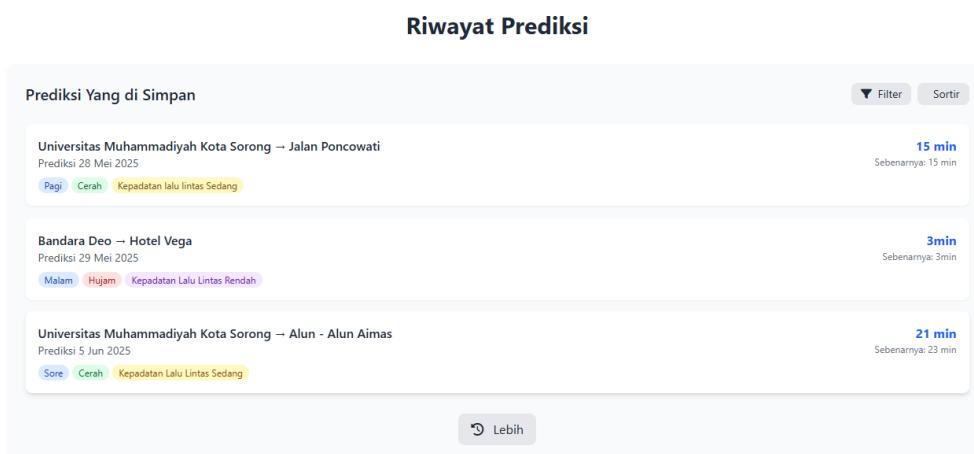
Di bagian atas kanan peta terdapat label informatif berupa:

- Status lalu lintas (misal: lalu lintas langsung)
- Kondisi cuaca (misal: cerah)

- c. Jumlah titik perbaikan jalan (misal: 2)
- d. Jumlah persimpangan (misal: 3)

Label ini membantu pengguna memahami potensi hambatan yang dapat memperlambat perjalanan, seperti adanya perbaikan jalan atau persimpangan padat. Informasi ini berasal dari parameter *input* yang dikombinasikan dengan data historis atau *real-time*, lalu digunakan sebagai fitur dalam model *XGBoost*.

5) Tampilan Riwayat Prediksi



Gambar 3.14 Tampilan Riwayat Prediksi

(Sumber : <https://www.figma.com/>)

Gambar 3.15 merupakan tampilan halaman riwayat prediksi halaman ini menampilkan daftar riwayat prediksi yang telah dilakukan oleh pengguna sebelumnya. Fitur ini dirancang untuk memberikan rekap data perjalanan dan perbandingan antara waktu tempuh hasil prediksi dengan waktu aktual, sehingga pengguna dapat menilai sejauh mana akurasi dari sistem prediksi yang dibangun.

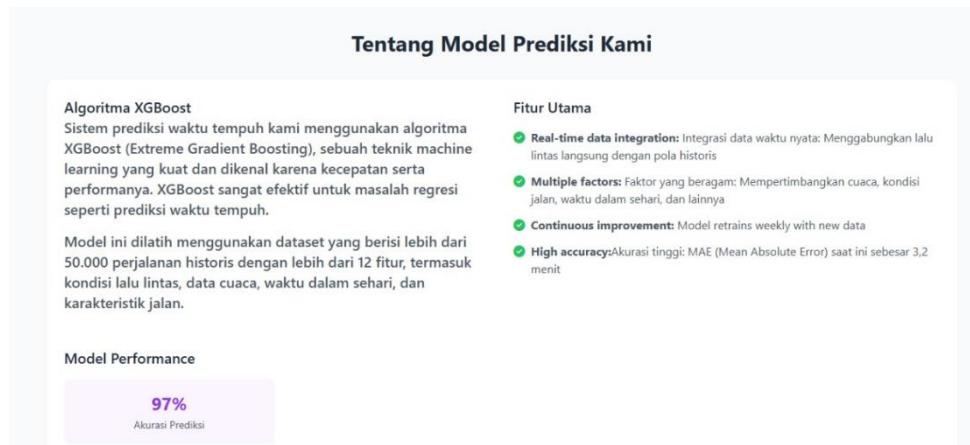
Setiap entri riwayat mencakup:

- a. Rute perjalanan (misalnya: Universitas Muhammadiyah Kota Sorong → Jalan Poncowati)
- b. Tanggal prediksi dilakukan

- c. Kondisi waktu, cuaca, dan kepadatan lalu lintas pada saat prediksi (ditampilkan dengan label berwarna seperti *Pagi*, *Cerah*, dan *Kepadatan Sedang*)
- d. Hasil prediksi waktu tempuh (misal: *15 min*), serta waktu aktual sebagai pembanding (*Sebenarnya: 15 min*)

Di bagian kanan atas halaman, tersedia fitur Filter dan Sortir yang memungkinkan pengguna menyaring hasil berdasarkan tanggal, lokasi, atau parameter kondisi lalu lintas. Ini memudahkan pengguna untuk meninjau kembali kondisi perjalanan yang mirip atau mengukur konsistensi sistem. Tombol “Lebih” di bagian bawah memungkinkan pengguna memuat lebih banyak data prediksi yang disimpan, menandakan bahwa sistem ini mendukung pagination atau lazy-loading agar tetap ringan saat digunakan.

6) Tampilan Tentang



Gambar 3.15 Tampilan Tentang

(Sumber: <https://www.figma.com/>)

Model prediksi waktu tempuh ini menggunakan *algoritma XGBoost (Extreme Gradient Boosting)*, sebuah teknik machine learning yang terkenal karena kecepatan dan keakuratannya dalam menangani permasalahan regresi. *Algoritma* ini sangat sesuai untuk memprediksi waktu tempuh perjalanan karena kemampuannya dalam memproses data besar dan mempertimbangkan banyak variabel secara bersamaan. Model dilatih menggunakan dataset yang berisi lebih dari 50.000 data perjalanan

historis dengan lebih dari 12 fitur penting, seperti kondisi lalu lintas, data cuaca, waktu dalam sehari, serta karakteristik jalan. Penggunaan fitur-fitur tersebut bertujuan untuk menghasilkan prediksi yang mendekati kondisi nyata di lapangan. Sistem ini memiliki beberapa keunggulan utama. Pertama, adanya integrasi data waktu nyata (*real-time data integration*), yang menggabungkan data lalu lintas langsung dengan pola historis untuk memberikan prediksi terkini. Kedua, model mempertimbangkan berbagai faktor (*multiple factors*) yang mempengaruhi waktu tempuh, seperti cuaca, kondisi jalan, dan waktu dalam sehari. Ketiga, model ini selalu mengalami perbaikan berkelanjutan (*continuous improvement*) dengan cara dilatih ulang setiap minggu menggunakan data terbaru, sehingga akurasi semakin meningkat seiring waktu. Keunggulan lain adalah tingkat akurasi tinggi (*high accuracy*), yang saat ini mencapai 97%, dengan rata-rata kesalahan prediksi (*Mean Absolute Error / MAE*) hanya sekitar 3,2 menit. Hal ini menunjukkan bahwa model mampu memberikan estimasi waktu tempuh yang sangat mendekati kenyataan dan dapat diandalkan untuk membantu pengguna dalam merencanakan perjalanan.

7) Tampilan Kontak Kami

The screenshot shows a contact form titled "Kontak Kami". On the left, there is a section labeled "Berikan Timbal Balik" containing fields for "nama" (name), "Email", "Subject" (set to "General Feedback"), and a large "Pesan" (Message) text area. Below these fields is a blue button labeled "Kirim Pesan" (Send Message). On the right, there is a section titled "Kelompok 1" featuring three circular icons with user icons. Below this are five small circular icons with communication-related symbols: a mail icon, a phone icon, a Twitter bird icon, and a red circle icon.

Gambar 3.16 Tampilan Halaman Kontak

(Sumber : <https://www.figma.com/>)

Gambar 3. 17 merupakan tampilan halaman Kontak Kami, dirancang sebagai sarana komunikasi dua arah antara pengguna dan tim pengembang sistem prediksi waktu tempuh. Di bagian kiri, terdapat formulir isian

bertajuk “Berikan Timbal Balik” yang memungkinkan pengguna mengirimkan masukan, kritik, saran, atau pertanyaan. Formulir ini terdiri dari input nama, alamat email, subjek pesan, serta kolom pesan utama. Pengguna dapat memilih subjek seperti *General Feedback*, *Bug Report*, atau *Request Fitur*, sehingga setiap pesan dapat diklasifikasikan dengan baik sesuai kebutuhan. Tombol “Kirim Pesan” ditempatkan di bawah formulir dengan desain mencolok menggunakan gradasi warna biru ke ungu, menandakan aksi utama pada halaman ini. Di sisi kanan halaman, terdapat informasi kontak dari tim pengembang yang tergabung dalam Kelompok 1, ditandai dengan ikon peran seperti pengguna, pengembang, dan pemetaan. Di bawahnya, tersedia berbagai ikon media komunikasi seperti email, telepon, *Twitter*, dan *GitHub* yang memberikan fleksibilitas bagi pengguna dalam memilih cara berinteraksi dengan tim. Antarmuka ini tidak hanya meningkatkan aksesibilitas dan keterbukaan, tetapi juga menciptakan kesan bahwa sistem ini dikembangkan secara profesional dan terbuka terhadap perbaikan berkelanjutan berdasarkan masukan pengguna.

f. Coding (Pengkodean)

Pada tahap ini, tim peneliti mulai membangun aplikasi berbasis web untuk prediksi waktu tempuh Sorong–Aimas menggunakan model *XGBoost*. Bahasa pemrograman utama yang digunakan adalah Python, dengan *framework Flask* untuk pengembangan *backend*. *Frontend* dikembangkan menggunakan *HTML*, *CSS*, dan *JavaScript* dengan tambahan *API Google Maps* untuk visualisasi rute perjalanan. Proses penulisan kode dilakukan di *Visual Studio Code*. Model *XGBoost* yang telah dilatih sebelumnya disimpan dalam format .json dan diintegrasikan ke dalam aplikasi untuk melakukan prediksi berdasarkan input pengguna seperti cuaca, waktu keberangkatan, kepadatan lalu lintas, dan jenis kendaraan.

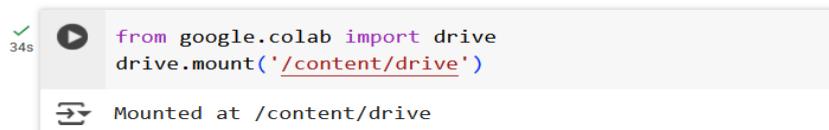
g. *Testing* (Pengujian)

Pada tahap ke empat penelitian ini melakukan pengujian. Pengujian perangkat lunak ditunjukkan untuk menguji semua elemen perangkat lunak yang dibuat.

3.4.5 Implementasi algoritma

Tahap kelima yang dilakukan peneliti dalam penelitian ini adalah implementasi dari web yang telah dibuat menggunakan metode *Xgboost*.

1. Mengakses *Google Drive* di *Google Colab*



```
34s  ⏴  from google.colab import drive
      drive.mount('/content/drive')

Mount at /content/drive
```

Gambar 3.17 Mengakses *Google Drive* di *Google Colab*

(Sumber: <https://colab.research.google.com/>)

Gambar 3.18 merupakan kode di atas ini digunakan untuk menghubungkan atau "mount" *Google Drive* ke lingkungan kerja *Google Colab*.

2. Instalasi dan Persiapan *Library* untuk Pemodelan Regresi dengan *XGBoost*



```
10s  ⏴  !pip install xgboost openpyxl

import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score
from xgboost import XGBRegressor

Requirement already satisfied: xgboost in /usr/local/lib/python3.11/dist-packages (2.1.4)
Requirement already satisfied: openpyxl in /usr/local/lib/python3.11/dist-packages (3.1.5)
Requirement already satisfied: numpy in /usr/local/lib/python3.11/dist-packages (from xgboost) (2.0.2)
Requirement already satisfied: nvidia-ncc1-cu12 in /usr/local/lib/python3.11/dist-packages (from xgboost) (2.21.5)
Requirement already satisfied: scipy in /usr/local/lib/python3.11/dist-packages (from xgboost) (1.15.3)
Requirement already satisfied: et-xmlfile in /usr/local/lib/python3.11/dist-packages (from openpyxl) (2.0.0)
```

Gambar 3.18 Instalasi dan Persiapan

(Sumber: <https://colab.research.google.com/>)

Gambar 3.19 di atas merupakan kode diatas ini digunakan untuk Menginstal *library* tambahan (*xgboost* dan *openpyxl*) yang belum tersedia secara default di *Google Colab*. Mengimpor *library* penting yang dibutuhkan untuk membaca data, memproses data, membagi data, melakukan pelatihan model regresi dengan *XGBoost*, dan mengevaluasi performa model.

3. Membaca File Excel dari Google Drive ke dalam DataFrame

```


file_path = '/content/drive/MyDrive/PrediksiWaktu/dataset_perjalanan.xlsx'
df = pd.read_excel(file_path)
df.head()


```

The screenshot shows a Jupyter Notebook cell with the code above. Below the code, a table displays the first 5 rows of the dataset. The columns are labeled: Cuaca, Kepadatan, Jam_Berangkat, Persimpangan, Ada_Perbaikan_Jalan, Hari, Tanjakan, Klasifikasi_Tanjakan, Tanjakan_Kecil, Tanjakan_Sedang, ..., Turunan_Besar, Tanjakan_Turunan, and Jarak_KM. The data includes various weather conditions like Berawan, Hujan, Cerah, and Sedang, along with other variables like traffic density, time of day, and road conditions.

Gambar 3.19 Membaca File Excel dari Google Drive

(Sumber: <https://colab.research.google.com/>)

Gambar 3.20 merupakan kode diatas ini digunakan untuk mengakses dan membaca *file Excel* yang tersimpan di *Google Drive* ke dalam bentuk *DataFrame* menggunakan *pandas*. *DataFrame* ini adalah struktur data yang mirip tabel dan umum digunakan untuk analisis data dan *machine learning*.

4. Menyiapkan Data untuk Pelatihan Model dengan *Encoding Label*

```


data = df.copy()

features = [
    'Cuaca', 'Kepadatan', 'Jam_Berangkat', 'Persimpangan',
    'Ada_Perbaikan_Jalan', 'Hari',
    'Tanjakan_Kecil', 'Tanjakan_Sedang', 'Tanjakan_Besar',
    'Turunan_Kecil', 'Turunan_Sedang', 'Turunan_Besar',
    'Tanjakan_Turunan', 'Jumlah_Berhenti', 'Jarak_KM'
]
target = 'Waktu_Tempuh'

categorical = ['Cuaca', 'Kepadatan', 'Ada_Perbaikan_Jalan', 'Hari']
le_dict = {}
for col in categorical:
    le = LabelEncoder()
    data[col] = le.fit_transform(data[col])
    le_dict[col] = le

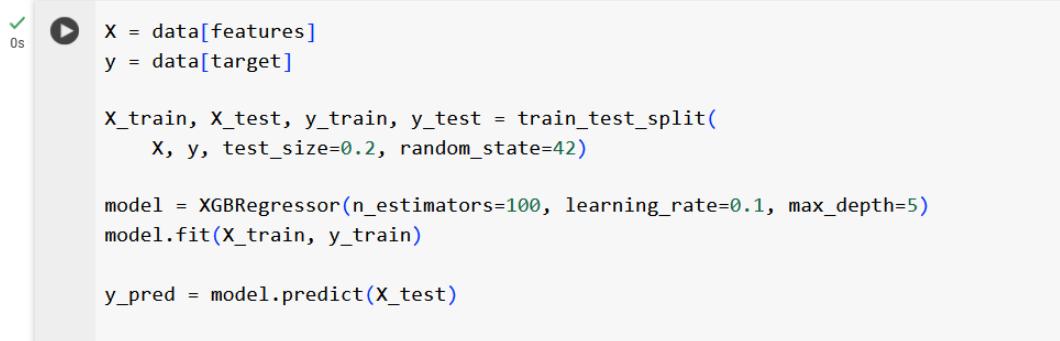

```

Gambar 3.20 Menyiapkan Data Untuk Pelatihan

(Sumber: <https://colab.research.google.com/>)

Gambar 3.21 merupakan kode diatas ini digunakan untuk Menyalin data asli agar tidak mengubah *DataFrame* aslinya. Menentukan fitur (*variabel input*) dan target (*variabel output*) untuk model. Mengubah data kategorikal (teks) menjadi angka dengan menggunakan *Label Encoding*, agar bisa digunakan dalam algoritma *machine learning* seperti *XGBoost*.

5. Melatih Model *XGBoost* untuk Prediksi Waktu Tempuh



```

✓ 0s
  X = data[features]
  y = data[target]

  X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.2, random_state=42)

  model = XGBRegressor(n_estimators=100, learning_rate=0.1, max_depth=5)
  model.fit(X_train, y_train)

  y_pred = model.predict(X_test)

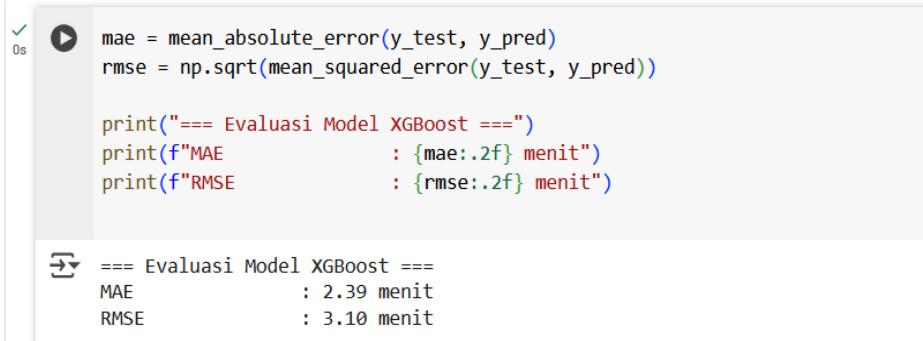
```

Gambar 3.21 Melatih Model *XGBoost*

(Sumber: <https://colab.research.google.com/>)

Gambar 3.22 Kode diatas ini melakukan pembagian data, pelatihan model *XGBoost*, dan prediksi terhadap data uji. Model ini digunakan untuk memprediksi nilai Waktu Tempuh berdasarkan fitur-fitur yang telah dipilih sebelumnya.

6. Evaluasi Kinerja Model *XGBoost* untuk Prediksi Waktu Tempuh



```

✓ 0s
  mae = mean_absolute_error(y_test, y_pred)
  rmse = np.sqrt(mean_squared_error(y_test, y_pred))

  print("== Evaluasi Model XGBoost ==")
  print(f"MAE : {mae:.2f} menit")
  print(f"RMSE : {rmse:.2f} menit")

  ➔ == Evaluasi Model XGBoost ==
  MAE : 2.39 menit
  RMSE : 3.10 menit

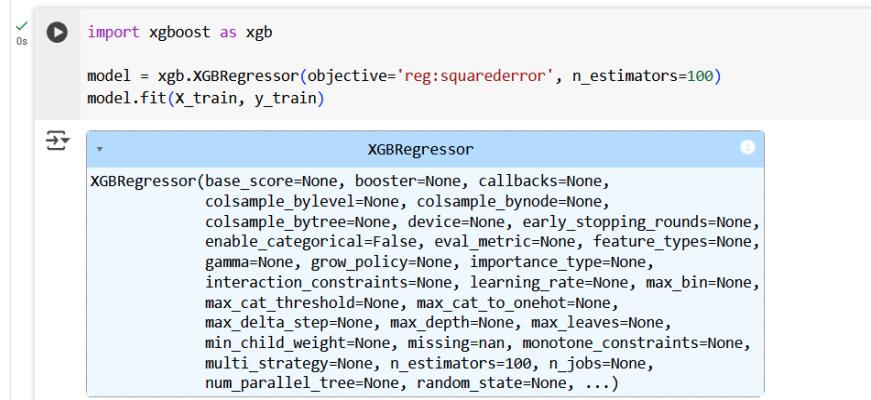
```

Gambar 3.22 Evaluasi Kinerja Model *XGBoost*

(Sumber: <https://colab.research.google.com/>)

Gambar 3.23 Kode diatas ini digunakan untuk mengukur seberapa baik performa model *XGBoost* dalam memprediksi Waktu Tempuh pada data uji.

7. Melatih Model *XGBoost Regressor* dengan Spesifikasi Tujuan Regresi



```

0s  ➔ import xgboost as xgb
      model = xgb.XGBRegressor(objective='reg:squarederror', n_estimators=100)
      model.fit(x_train, y_train)

      ➔ XGBRegressor
      XGBRegressor(base_score=None, booster=None, callbacks=None,
                  colsample_bylevel=None, colsample_bynode=None,
                  colsample_bytree=None, device=None, early_stopping_rounds=None,
                  enable_categorical=False, eval_metric=None, feature_types=None,
                  gamma=None, grow_policy=None, importance_type=None,
                  interaction_constraints=None, learning_rate=None, max_bin=None,
                  max_cat_threshold=None, max_cat_to_onehot=None,
                  max_delta_step=None, max_depth=None, max_leaves=None,
                  min_child_weight=None, missing='nan', monotone_constraints=None,
                  multi_strategy=None, n_estimators=100, n_jobs=None,
                  num_parallel_tree=None, random_state=None, ...)

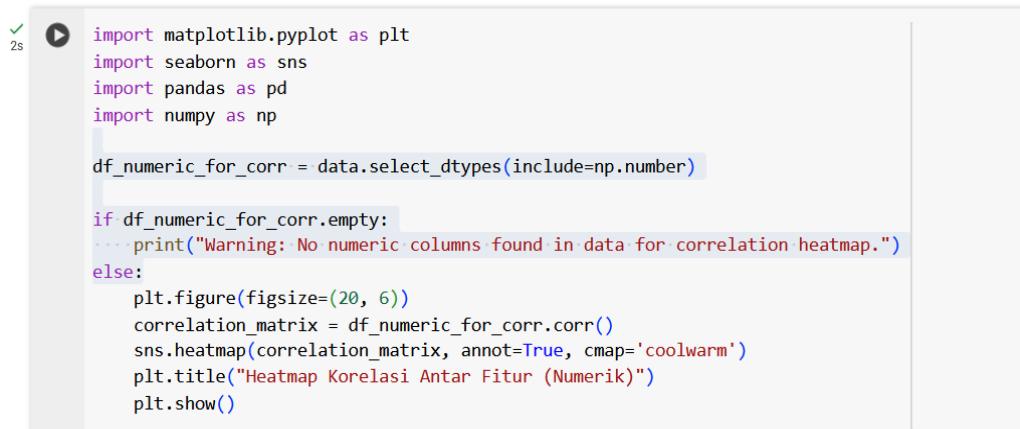
```

Gambar 3. 23 Melatih Model *XGBoost Regressor*

(Sumber: <https://colab.research.google.com/>)

Gambar 3.24 Kode diatas ini membuat dan melatih model regresi menggunakan *XGBoost* dengan tujuan untuk meminimalkan *squared error* (galat kuadrat). Model ini digunakan untuk menyelesaikan masalah regresi, seperti memprediksi Waktu Tempuh.

8. Hitmap Korelasi Antara Fitur Numerik Dengan Heatmap



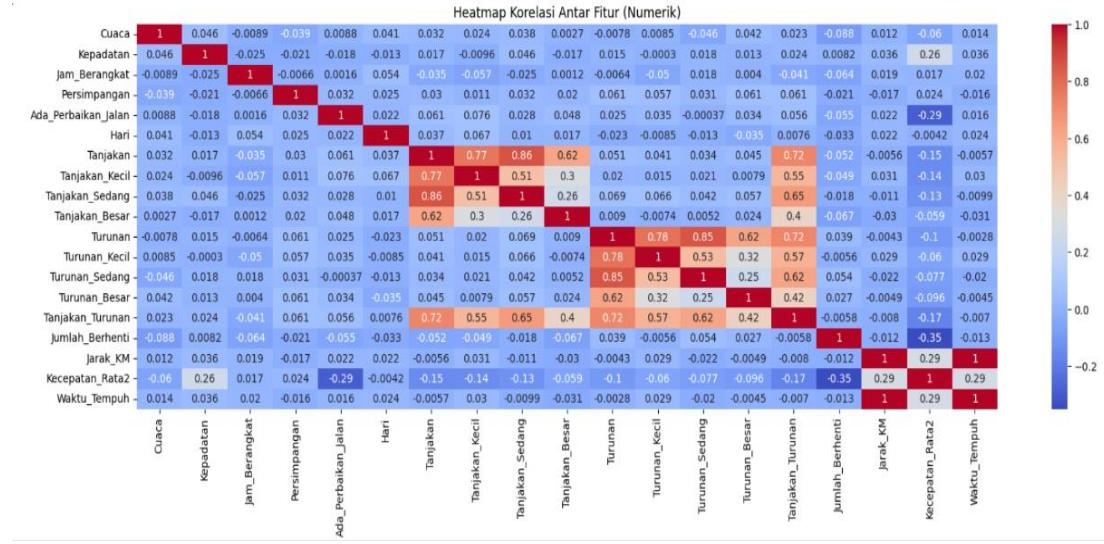
```

2s  ➔ import matplotlib.pyplot as plt
      import seaborn as sns
      import pandas as pd
      import numpy as np

      df_numeric_for_corr = data.select_dtypes(include=np.number)

      if df_numeric_for_corr.empty:
          ...print("Warning: No numeric columns found in data for correlation heatmap.")
      else:
          plt.figure(figsize=(20, 6))
          correlation_matrix = df_numeric_for_corr.corr()
          sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm')
          plt.title("Heatmap Korelasi Antar Fitur (Numerik)")
          plt.show()

```



Gambar 3.24 Hitmap Korelasi Antara Vitur Numerik

(Sumber: <https://colab.research.google.com/>)

Program ini bertujuan untuk menampilkan visualisasi hubungan antar fitur numerik (kolom angka) dalam sebuah dataset dalam bentuk heatmap korelasi. Heatmap ini membantu kita memahami seberapa kuat hubungan antara satu variabel dengan variabel lainnya.

9. Visualisasi Sebaran Waktu Tempuh Prediksi Vs Actual Per Rute

```

15s [37] import matplotlib.pyplot as plt
      import numpy as np

      # Buat prediksi simulasi (jika belum ada kolom prediksi asli)
      np.random.seed(42)
      df['Prediksi'] = df['Waktu_Tempuh'] + np.random.normal(0, 3, len(df))

      # Gabungkan asal dan tujuan jadi satu rute
      df['Rute'] = df['Lokasi_Asal'] + ' → ' + df['Lokasi_Tujuan']

      # Buat plot
      plt.figure(figsize=(10, 7))

      # Warna otomatis untuk tiap rute
      unique_routes = df['Rute'].unique()
      colors = plt.cm.tab20.colors # maksimal 20 warna berbeda
      for i, rute in enumerate(unique_routes):
          subset = df[df['Rute'] == rute]
          plt.scatter(
              subset['Waktu_Tempuh'],
              subset['Prediksi'],
              label=rute,
              alpha=0.7,
              s=30,
              color=colors[i % len(colors)])
      
```

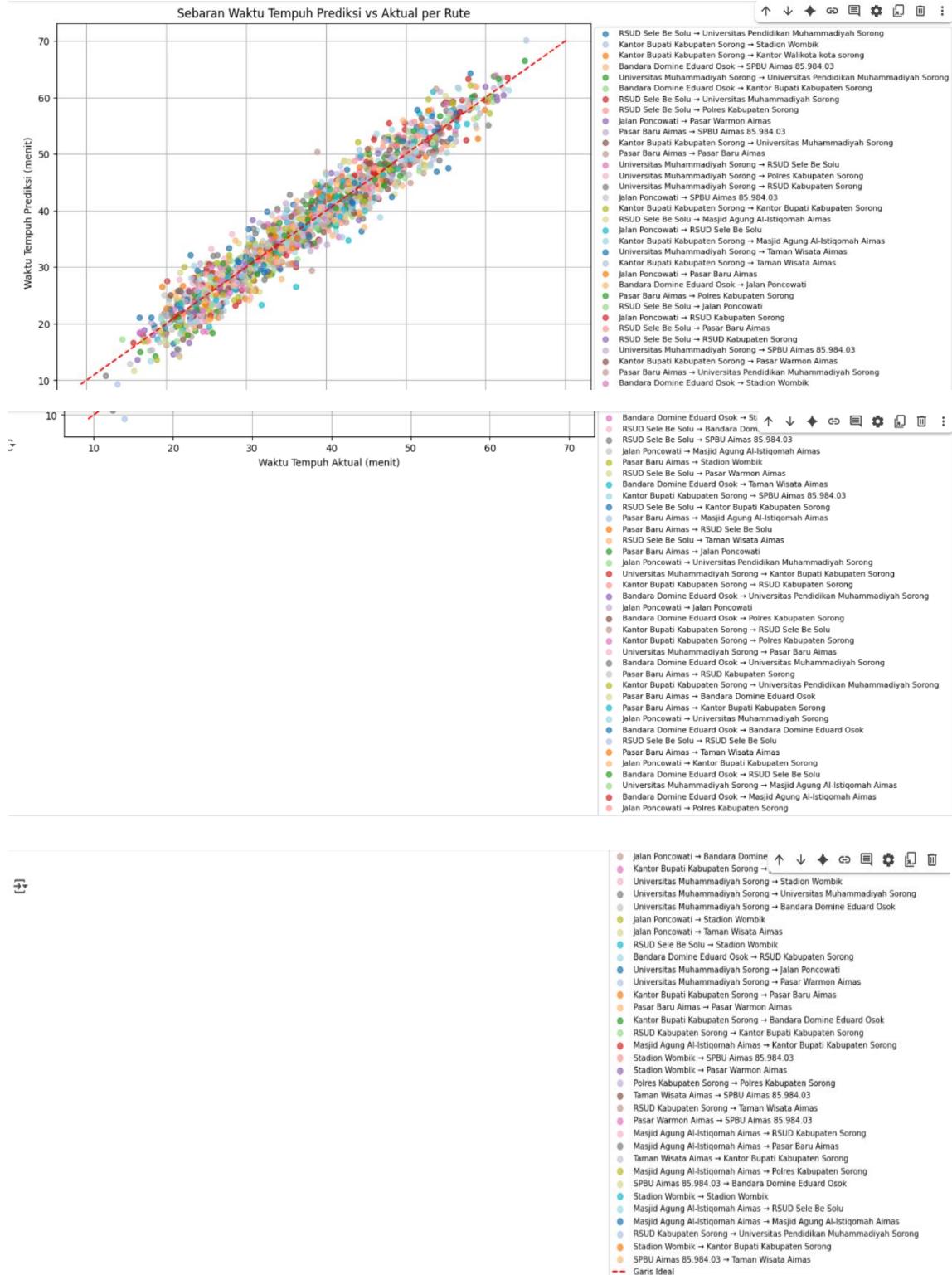


```

# Garis ideal
min_val = min(df['Waktu_Tempuh'].min(), df['Prediksi'].min())
max_val = max(df['Waktu_Tempuh'].max(), df['Prediksi'].max())
plt.plot([min_val, max_val], [min_val, max_val], 'r--', label='Garis Ideal')

# Label
plt.title('Sebaran Waktu Tempuh Prediksi vs Aktual per Rute')
plt.xlabel('Waktu Tempuh Aktual (menit)')
plt.ylabel('Waktu Tempuh Prediksi (menit)')
plt.legend(fontsize=8, loc='upper left', bbox_to_anchor=(1, 1))
plt.grid(True)
plt.tight_layout()
plt.show()

```



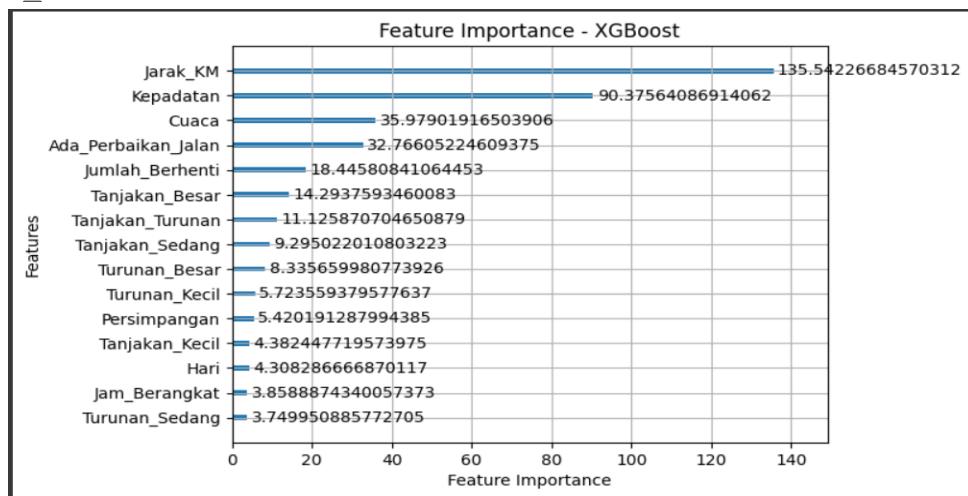
Gambar 3.25 Visualisasi Sebaran Waktu Tempuh

(Sumber: <https://colab.research.google.com/>)

Program ini digunakan untuk membuat grafik perbandingan antara waktu tempuh aktual dengan waktu tempuh hasil prediksi. Dalam grafik ini juga ditambahkan garis ideal sebagai acuan jika prediksi benar-benar sama dengan nilai aktual.

11. Visualisasi *Feature Importance* Berdasarkan *Gain* pada Model *XGBoost*

```
2s  ⏪ xgb.plot_importance(model, importance_type='gain', xlabel='Feature Importance')
    plt.title('Feature Importance - XGBoost')
    plt.show()
```



Gambar 3.26 Visualisasi *Feature Importance*

(Sumber: <https://colab.research.google.com/>)

Gambar 3.27 Kode diatas ini digunakan untuk menampilkan seberapa penting masing-masing fitur (variabel input) dalam membantu model *XGBoost* melakukan prediksi, berdasarkan *metrik gain*.

12. Optimasi *Hyperparameter XGBoost* Menggunakan *GridSearchCV*

```
0s  ⏪ from sklearn.linear_model import Ridge
    from sklearn.model_selection import train_test_split
    from sklearn.metrics import r2_score

    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

    ridge_model = Ridge(alpha=1.0)
    ridge_model.fit(X_train, y_train)

    y_pred = ridge_model.predict(X_test)
    r2 = r2_score(y_test, y_pred)
    print("R2 Ridge:", round(r2, 4))
```

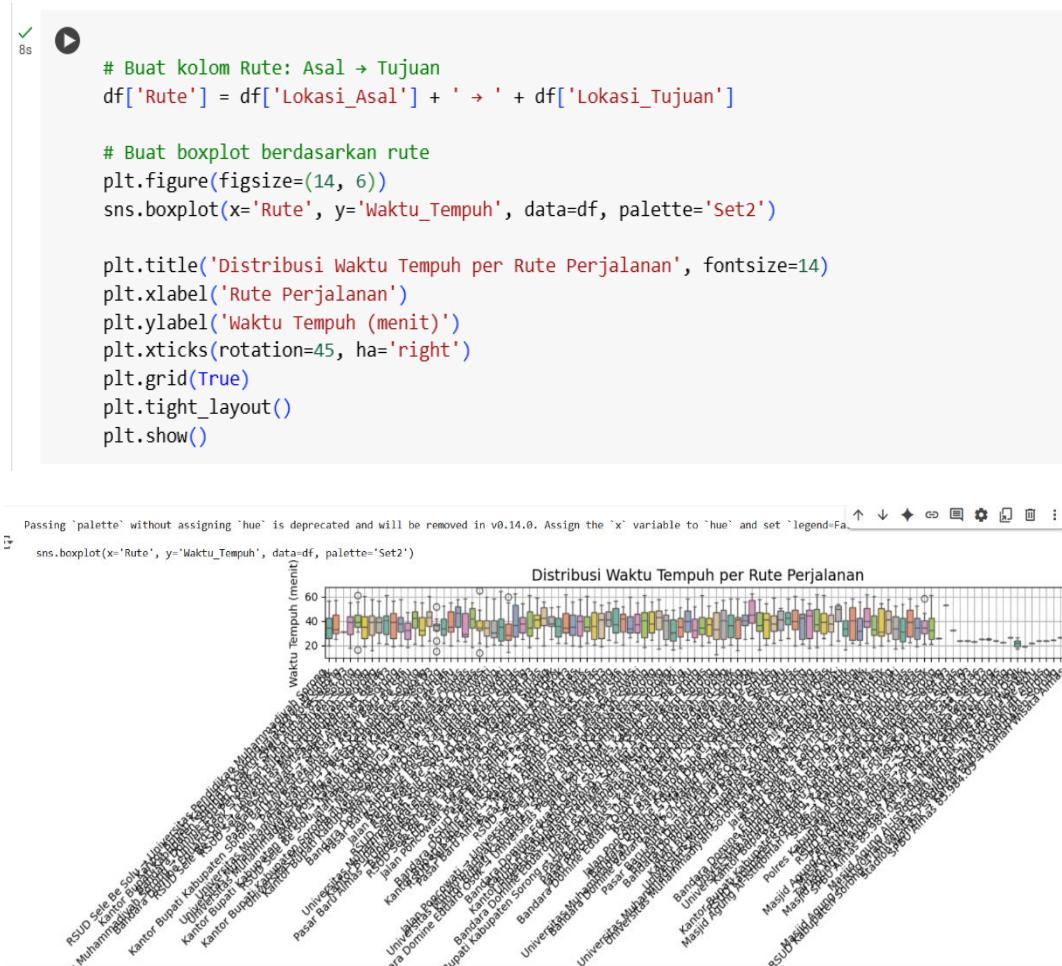
R² Ridge: 0.97

Gambar 3.27 Optimasi Hyperparameter XGBoost

(Sumber: <https://colab.research.google.com/>)

Kode ini digunakan untuk mencari kombinasi terbaik dari *hyperparameter* model *XGBoost* menggunakan teknik pencarian *grid* (*GridSearchCV*). Tujuannya adalah untuk meningkatkan performa model secara maksimal berdasarkan skor akurasi (*R² Score*).

13. Distribusi Waktu Tempuh Per Rute Perjalanan Dalam Bentuk Boxplot



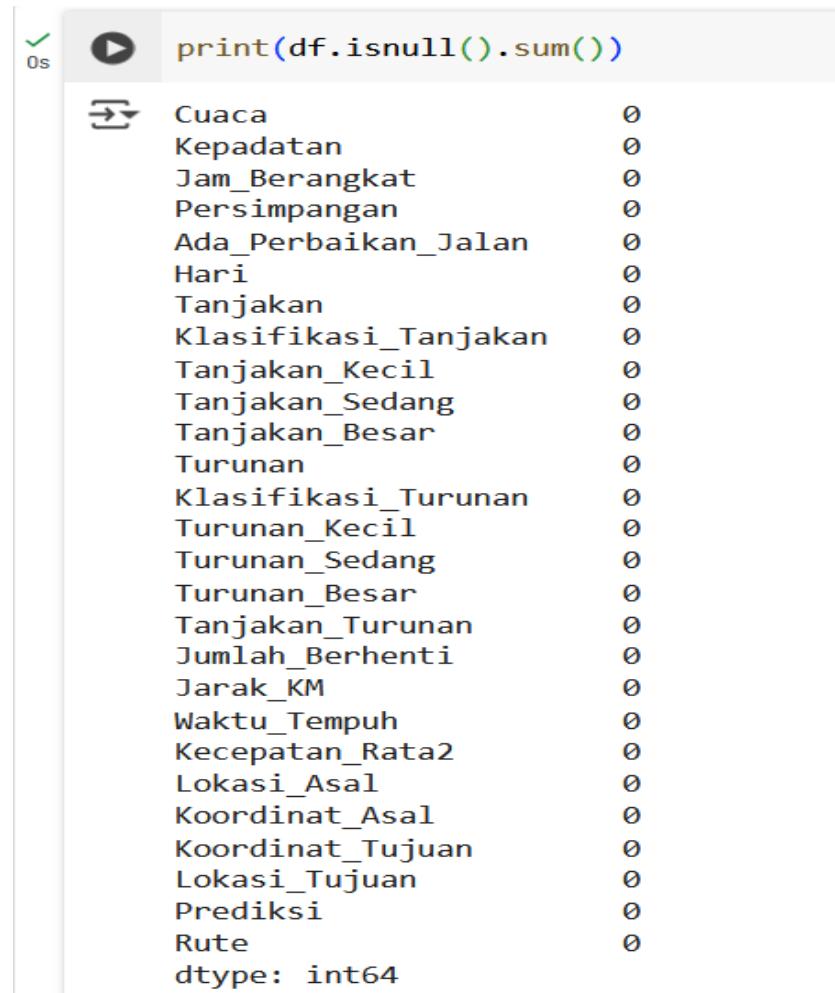
Gambar 3.28 Visualisasi Distribusi Waktu

(Sumber: <https://colab.research.google.com/>)

Program ini digunakan untuk menampilkan visualisasi distribusi waktu tempuh berdasarkan rute perjalanan dalam bentuk boxplot. Visualisasi ini

membantu kita melihat variasi durasi perjalanan antar rute tertentu, sekaligus mendeteksi data ekstrem (outlier)

14. Mengecek Jumlah Data Kosong (*Missing Values*) di Setiap Kolom



```
0s
print(df.isnull().sum())
Cuaca          0
Kepadatan       0
Jam_Berangkat   0
Persimpangan     0
Ada_Perbaikan_Jalan 0
Hari            0
Tanjakan         0
Klasifikasi_Tanjakan 0
Tanjakan_Kecil    0
Tanjakan_Sedang    0
Tanjakan_Besar    0
Turunan          0
Klasifikasi_Turunan 0
Turunan_Kecil      0
Turunan_Sedang      0
Turunan_Besar      0
Tanjakan_Turunan   0
Jumlah_Berhenti    0
Jarak_KM          0
Waktu_Tempuh        0
Kecepatan_Rata2    0
Lokasi_Asal        0
Koordinat_Asal      0
Koordinat_Tujuan    0
Lokasi_Tujuan        0
Prediksi           0
Rute              0
dtype: int64
```

Gambar 3.29 Mengecek Jumlah Data Kosong

(Sumber: <https://colab.research.google.com/>)

Program ini digunakan untuk mengecek apakah ada data yang kosong (missing values atau NaN) dalam dataset df. Pengecekan ini penting sebelum melakukan analisis atau pelatihan model, karena data kosong bisa menyebabkan error atau hasil analisis yang tidak akurat.

15. Mengecek Jumlah Data Duplikat di Dataset

```
✓ [17] print("Jumlah duplikat:", df.duplicated().sum())
→ Jumlah duplikat: 0
```

Gambar 3.30 Mengacak Data Duplikat

(Sumber: <https://colab.research.google.com/>)

Perintah ini digunakan untuk mengetahui berapa banyak baris yang duplikat (sama persis) di dalam *DataFrame* *df*. Jika ada dan ingin dihapus kamu hanya perlu memasukan *df = df.drop_duplicates()*.

16. Menghapus Data Duplikat dari DataFrame

```
✓ [18] df = df.drop_duplicates()
```

Gambar 3.31 Menghapus Data Duplikat

(Sumber: <https://colab.research.google.com/>)

Program ini digunakan untuk menghapus baris-baris data yang duplikat dari *DataFrame* *df*. Artinya, jika ada dua atau lebih baris yang isinya persis sama di semua kolom, maka baris-baris duplikat tersebut akan dihapus dan hanya satu yang disisakan.

17. Visualisasi Distribusi Waktu Tempuh Di Semua Rute Yang Telah

Ditentukan Dengan *Histogram* Dan *KDE*

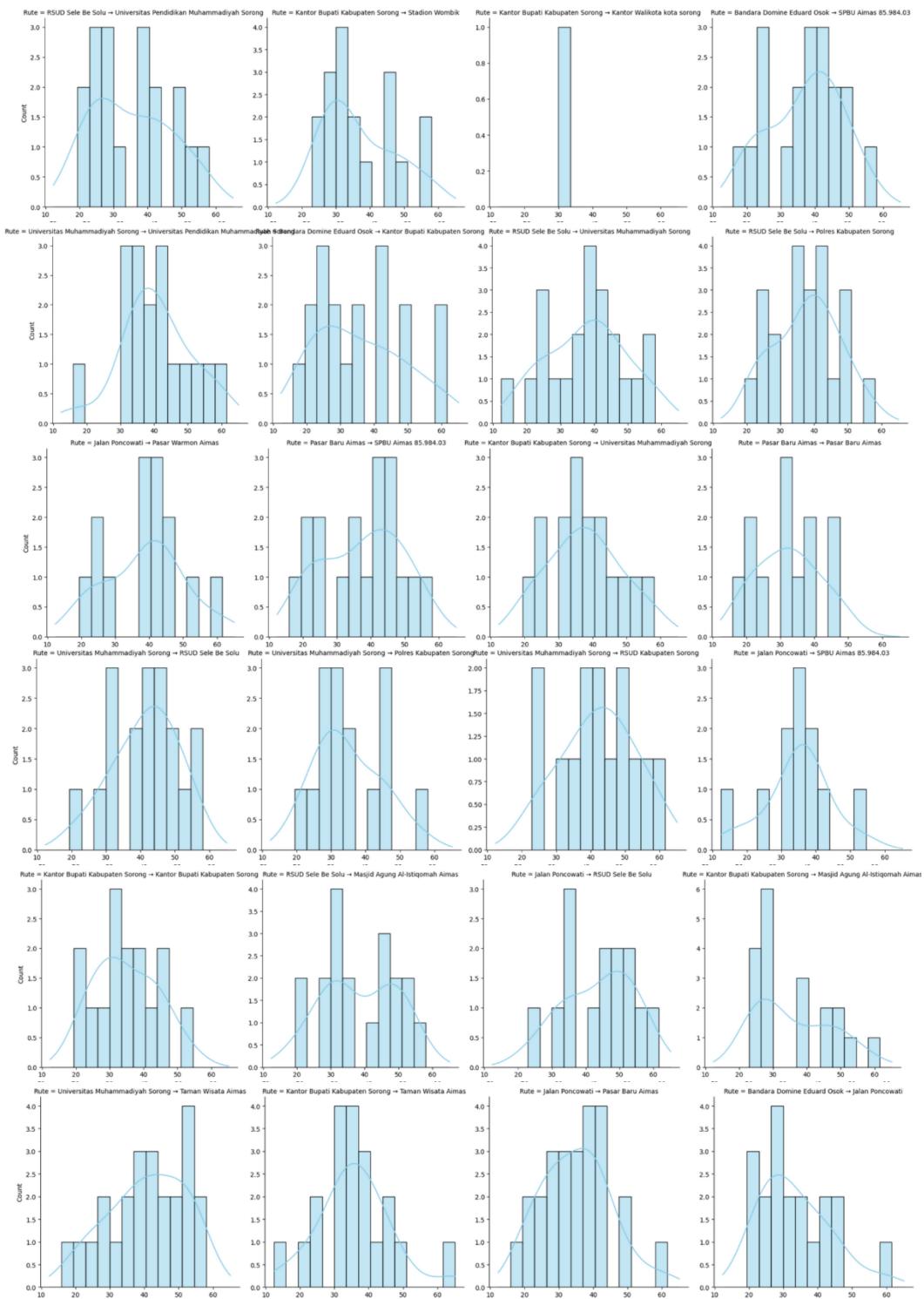
```
✓ [19] import seaborn as sns
import matplotlib.pyplot as plt

df['Rute'] = df['Lokasi_Asal'] + ' → ' + df['Lokasi_Tujuan']

g = sns.displot(
    data=df,
    x='Waktu_Tempuh',
    col='Rute',
    col_wrap=4,
    kde=True,
    bins=15,
    color='skyblue',
    facet_kws={'sharex': False, 'sharey': False}
)

g.fig.subplots_adjust(top=0.94)
g.fig.suptitle('Distribusi Waktu Tempuh per Rute (Semua Rute)', fontsize=16)
plt.show()
```

Distribusi Waktu Tempuh per Rute (Semua Rute)



Gambar 3.32 Visualisasi Distribusi Waktuh Tempuh

Sumber: <https://colab.research.google.com/>

Kode ini digunakan untuk menampilkan distribusi data Waktu_Tempuh dalam bentuk histogram, disertai dengan kurva *KDE (Kernel Density Estimation)*. Tujuannya adalah untuk memahami bentuk penyebaran data: apakah normal, miring (*skewed*), atau memiliki beberapa puncak.

18. Visualisasi *Outlier* dan Distribusi Data Fitur Numerik dengan *Boxplot*

```
✓ 11m ➔ # =====
# ANALISIS BOX PLOT & DISTRIBUSI PER RUTE
# (MENAMPILKAN SEMUA RUTE)
# =====
from google.colab import drive
drive.mount('/content/drive')

import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

path = '/content/drive/My Drive/PrediksiWaktu/Dataset_Sorong_Aimas_Final.xlsx'
df = pd.read_excel(path)

df.columns = df.columns.str.strip()
df['Rute'] = df['Lokasi_Asal'].astype(str).str.strip() + " → " + \
    df['Lokasi_Tujuan'].astype(str).str.strip()

num_cols = df.select_dtypes(include=['int64', 'float64']).columns.tolist()
print(f"Kolom numerik ({len(num_cols)}):", num_cols)

show_histograms = True          # True = tampilkan hist+KDE per rute
wrap_cols       = 3             # berapa subplot per baris utk histogram

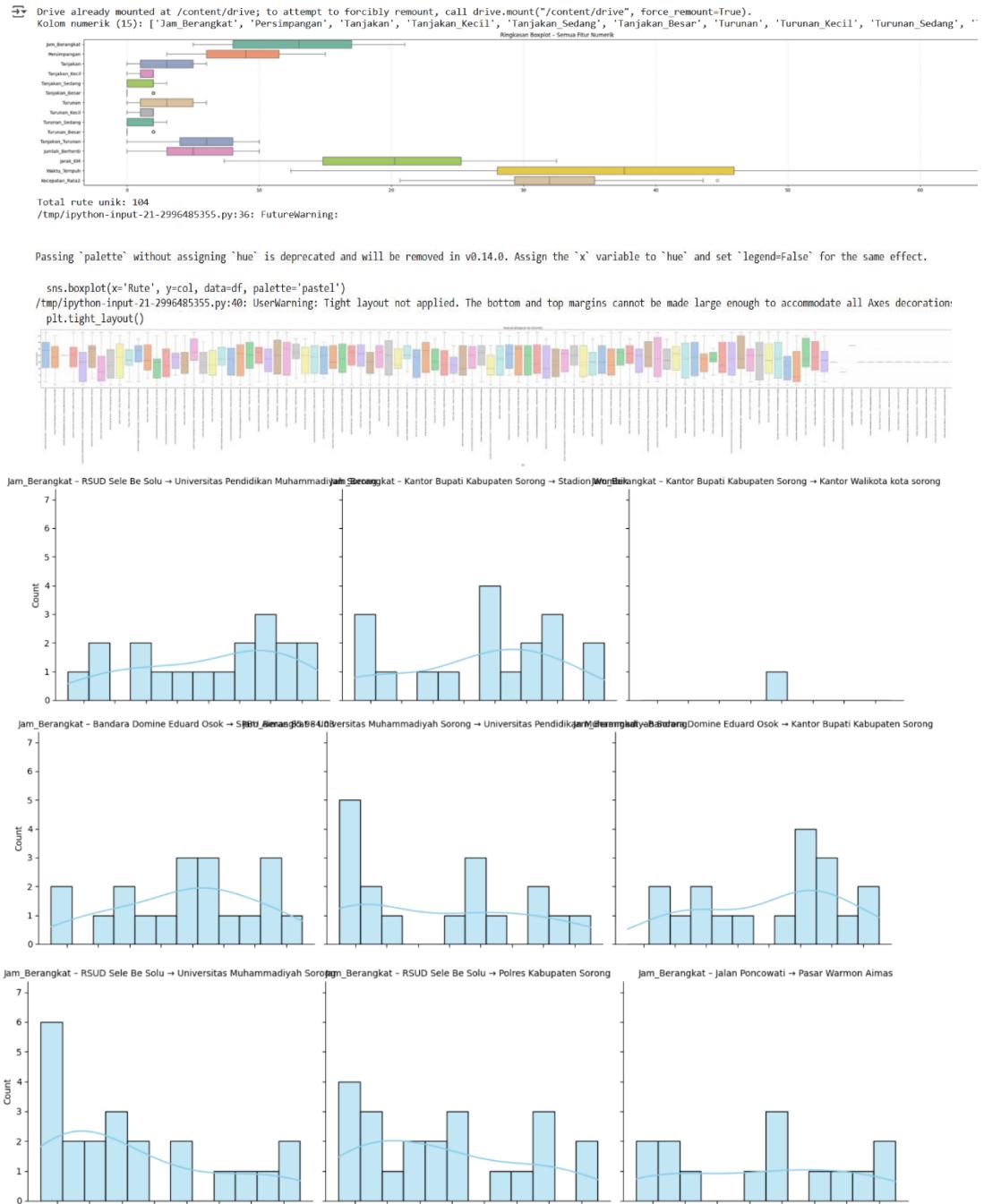
plt.figure(figsize=(2.5*len(num_cols), 6))
sns.boxplot(data=df[num_cols], orient='h', palette='Set2')
plt.title('Ringkasan Boxplot - Semua Fitur Numerik')
plt.grid(axis='x', linestyle='--', alpha=0.3)
plt.show()
```

```
✓ 11m ➔ all_routes = df['Rute'].unique()
print(f"Total rute unik: {len(all_routes)}")

for col in num_cols:
    plt.figure(figsize=(max(10, 1.2*len(all_routes)), 6))
    sns.boxplot(x='Rute', y=col, data=df, palette='pastel')
    plt.title(f'Boxplot {col} per Rute (Semua Rute)')
    plt.xticks(rotation=90)
    plt.grid(axis='y', linestyle='--', alpha=0.3)
    plt.tight_layout()
    plt.show()

    if show_histograms:
        g = sns.displot(
            data=df, x=col, col='Rute', col_wrap=wrap_cols,
            kde=True, color='skyblue', height=4, aspect=1.2
        )
        g.set_titles(col_template=f'{col} - {{col_name}}')
        g.set_axis_labels(col, 'Count')
        plt.tight_layout()
        plt.show()

print(" Selesai! Visualisasi dibuat untuk semua rute dan", len(num_cols), "fitur numerik.")
```



Gambar 3.33 Visualisasi *Outlier* dan Distribusi

(Sumber: <https://colab.research.google.com/>)

Kode ini digunakan untuk menampilkan *boxplot* (diagram kotak) untuk setiap fitur numerik yang terdaftar dalam num_cols. Tujuannya adalah untuk:

- Melihat distribusi data
- Mengetahui nilai median dan kuartil, serta

- c Mengidentifikasi outlier pada masing-masing fitur.
19. Pelatihan Ulang Model *XGBoost* untuk Prediksi Waktu Tempuh dengan Data Terstandarisasi

```
✓ 2s ➔ # REGRESI WAKTU_TEMPUH - XGBoost + Encoding & Scaling
from sklearn.preprocessing import StandardScaler
import xgboost as xgb
import matplotlib.pyplot as plt
import seaborn as sns
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score

df_model = df.copy()

categorical_cols = [
    'Cuaca', 'Kepadatan', 'Hari',
    'Klasifikasi_Tanjakan', 'Klasifikasi_Turunan',
    'Ada_Perbaikan_Jalan', 'Rute'
]

for col in categorical_cols:
    if col in df_model.columns:
        le = LabelEncoder()
        df_model[col] = le.fit_transform(df_model[col].astype(str))
    else:
        print(f"Kolom '{col}' tidak ditemukan & dilewati.")

drop_cols = ['Lokasi_Asal', 'Lokasi_Tujuan',
             'Koordinat_Asal', 'Koordinat_Tujuan']
df_model = df_model.drop(columns=[c for c in drop_cols if c in df_model.columns])
```

```
✓ 2s ➔ target_col = 'Waktu_Tempuh'
if target_col not in df_model.columns:
    raise KeyError(f"Kolom target '{target_col}' tidak ditemukan!")

X = df_model.drop(columns=[target_col])
y = df_model[target_col]

numeric_cols = X.select_dtypes(include=np.number).columns.tolist()
scaler = StandardScaler()
X[numeric_cols] = scaler.fit_transform(X[numeric_cols])

X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.2, random_state=42, shuffle=True
)

model = XGBRegressor(
    n_estimators=300,
    learning_rate=0.05,
    max_depth=6,
    subsample=0.8,
    colsample_bytree=0.8,
    random_state=42,
    objective='reg:squarederror'
)
model.fit(X_train, y_train)

y_pred = model.predict(X_test)

mae = mean_absolute_error(y_test, y_pred)
mse = mean_squared_error(y_test, y_pred)
rmse = np.sqrt(mse)
r2 = r2_score(y_test, y_pred)
```

```

✓ 2s  ➜ print(" Pelatihan selesai!")
print(f" MAE : {mae:.2f}")
print(f" MSE : {mse:.2f}")
print(f" RMSE : {rmse:.2f}")
print(f" R² : {r2:.4f}")

comparison_df = pd.DataFrame({
    "Nilai_Aslis": y_test.reset_index(drop=True),
    "Prediksi" : y_pred
})
print("\n")
print(comparison_df)
print(Generate)
print(View)
plt.figure()
plt.plot(comparison_df["Nilai_Aslis"].values, label="Nilai Asli", marker='o')
plt.plot(comparison_df["Prediksi"].values, label="Prediksi", marker='x', linestyle='--')
plt.title("Prediksi vs Nilai Asli")
plt.xlabel("Index")
plt.ylabel("Waktu Tempuh")
plt.legend()
plt.grid(True, linestyle='--', alpha=0.3)
plt.tight_layout()
plt.show()

➡️ Pelatihan selesai!
MAE : 0.93
MSE : 2.44
RMSE : 1.56
R² : 0.9716

```

➡️ **Contoh perbandingan:**

	Nilai_Aslis	Prediksi
0	17.876003	16.770905
1	35.575939	35.891571
2	24.536345	24.018579
3	41.689586	39.778023
4	21.080311	20.822083
5	36.069532	31.487959
6	28.056664	28.044310
7	34.271484	39.416561
8	42.931636	42.184643
9	28.919472	27.143606

Gambar 3.34 Pelatihan Ulang Model XGBoost

(Sumber: <https://colab.research.google.com/>)

Kode ini melakukan pelatihan ulang model *XGBoost Regressor* untuk memprediksi Waktu_Tempuh menggunakan data yang telah di-*encode* (untuk kolom kategorikal) dan di-*standarisasi* (untuk kolom numerik). Hasilnya kemudian dievaluasi menggunakan metrik *MAE*, *MSE*, *RMSE*, dan *R*², serta divisualisasikan perbandingan antara nilai asli dan prediksi.

20. Menampilkan Hasil Prediksi Waktu Tempuh dan Link *Google Maps* Berdasarkan Lokasi

```
# PREDIKSI WAKTU TEMPUH + GOOGLE MAPS LINK

import numpy as np
import pandas as pd
from sklearn.preprocessing import LabelEncoder, StandardScaler
from urllib.parse import quote_plus

df_pred = df.copy()

df_pred['Lokasi_Asal'] = df_pred['Lokasi_Asal'].replace({
    'Universitas Pendidikan Muhammadiyah Sorong': 'UNIMUDA Sorong'
})
df_pred['Lokasi_Tujuan'] = df_pred['Lokasi_Tujuan'].replace({
    'Universitas Pendidikan Muhammadiyah Sorong': 'UNIMUDA Sorong'
})
lokasi_asal = df_pred['Lokasi_Asal'].str.strip()
lokasi_tujuan = df_pred['Lokasi_Tujuan'].str.strip()

categorical_cols = [
    'Cuaca', 'Kepadatan', 'Hari',
    'Ada_Perbaikan_Jalan', 'Klasifikasi_Tanjakan', 'Klasifikasi_Turunan', 'Rute'
]

for col in categorical_cols:
    if col in df_pred.columns:
        le = LabelEncoder()
        df_pred[col] = le.fit_transform(df_pred[col].astype(str))

drop_cols = ['Lokasi_Asal', 'Lokasi_Tujuan', 'Koordinat_Asal', 'Koordinat_Tujuan', 'Waktu_Tempuh']
X_pred = df_pred.drop(columns=[c for c in drop_cols if c in df_pred.columns])

numeric_cols = X_pred.select_dtypes(include=np.number).columns
scaler = StandardScaler()
X_pred[numeric_cols] = scaler.fit_transform(X_pred[numeric_cols])

y_pred_all = model.predict(X_pred)

def make_maps_url(asal, tujuan):
    return f"https://www.google.com/maps/dir/?api=1&origin={quote_plus(asal)}&destination={quote_plus(tujuan)}"

result_df = pd.DataFrame({
    "Lokasi Asal": lokasi_asal,
    "Lokasi Tujuan": lokasi_tujuan,
    "Prediksi Waktu Tempuh (menit)": np.round(y_pred_all, 2)
})
result_df["Google Maps"] = result_df.apply(lambda r: make_maps_url(r["Lokasi Asal"], r["Lokasi Tujuan"]), axis=1)

pd.set_option("display.max_rows", None)
pd.set_option("display.max_colwidth", None)
print("Seluruh hasil prediksi:")
print(result_df.to_string(index=False))

excel_path = "/content/drive/My Drive/PrediksiWaktu/hasil_prediksi_lengkap.xlsx"
csv_path = "/content/drive/My Drive/PrediksiWaktu/hasil_prediksi_lengkap.csv"
result_df.to_excel(excel_path, index=False)
result_df.to_csv(csv_path, index=False)
```

Lokasi Asal	Lokasi Tujuan	Prediksi Waktu Tempuh (menit)	
RSUD Sele Be Solu	UNIMUDA Sorong	14.700000	https://www.google.com/maps/dir/?api=1&origin=RSUD+Sele+Be+Solu&destination=UNIMUDA+Sorong
Kantor Bupati Kabupaten Sorong	Stadion Wombik	17.010000	https://www.google.com/maps/dir/?api=1&origin=Kantor+Bupati+Kabupaten+Sorong&destination=Stadion+Wombik
Kantor Bupati Kabupaten Sorong	Kantor Walikota kota sorong	17.770000	https://www.google.com/maps/dir/?api=1&origin=Kantor+Bupati+Kabupaten+Sorong&destination=Kantor+Walikota+kota+sorong
Bandara Domine Eduard Osok	SPBU Almas 85, 984, 03	20.490000	https://www.google.com/maps/dir/?api=1&origin=Bandara+Domine+Eduard+Osok&destination=SPBU+Almas+85,+984,+03
Universitas Muhammadiyah Sorong	UNIMASA Sorong	35.340000	https://www.google.com/maps/dir/?api=1&origin=Universitas+Muhammadiyah+Sorong&destination=UNIMASA+Sorong
Bandara Domine Eduard Osok	Kantor Bupati Kabupaten Sorong	15.730000	https://www.google.com/maps/dir/?api=1&origin=Bandara+Domine+Eduard+Osok&destination=Kantor+Bupati+Kabupaten+Sorong
RSUD Sele Be Solu	Universitas Muhammadiyah Sorong	12.160000	https://www.google.com/maps/dir/?api=1&origin=RSUD+Sele+Be+Solu&destination=Universitas+Muhammadiyah+Sorong
RSUD Sele Be Solu	Polres Kabupaten Sorong	19.420000	https://www.google.com/maps/dir/?api=1&origin=RSUD+Sele+Be+Solu&destination=Polres+Kabupaten+Sorong
Jalan Poncowati	Pasar Warmon Alimas	46.169998	https://www.google.com/maps/dir/?api=1&origin=Jalan+Poncowati&destination=Pasar+Warmon+Alimas
Pasar Baru Alimas	SPBU Almas 85, 984, 03	29.809999	https://www.google.com/maps/dir/?api=1&origin=Pasar+Baru+Alimas&destination=SPBU+Almas+85,+984,+03
Kantor Bupati Kabupaten Sorong	Universitas Muhammadiyah Sorong	42.099999	https://www.google.com/maps/dir/?api=1&origin=Kantor+Bupati+Kabupaten+Sorong&destination=Universitas+Muhammadiyah+Sorong
Pasar Baru Alimas	Pasar Baru Alimas	26.639999	https://www.google.com/maps/dir/?api=1&origin=Pasar+Baru+Alimas&destination=Pasar+Baru+Alimas
Universitas Muhammadiyah Sorong	RSUD Sele Be Solu	38.810001	https://www.google.com/maps/dir/?api=1&origin=Universitas+Muhammadiyah+Sorong&destination=RSUD+Sele+Be+Solu
Universitas Muhammadiyah Sorong	Polres Kabupaten Sorong	27.009999	https://www.google.com/maps/dir/?api=1&origin=Universitas+Muhammadiyah+Sorong&destination=Polres+Kabupaten+Sorong
Universitas Muhammadiyah Sorong	RSUD Kabupaten Sorong	44.200001	https://www.google.com/maps/dir/?api=1&origin=Universitas+Muhammadiyah+Sorong&destination=RSUD+Kabupaten+Sorong
Jalan Poncowati	Pasar Baru Alimas	22.780001	https://www.google.com/maps/dir/?api=1&origin=Jalan+Poncowati&destination=Pasar+Baru+Alimas
Kantor Bupati Kabupaten Sorong	Kantor Bupati Kabupaten Sorong	24.760000	https://www.google.com/maps/dir/?api=1&origin=Kantor+Bupati+Kabupaten+Sorong&destination=Kantor+Bupati+Kabupaten+Sorong
RSUD Sele Be Solu	RSUD Sele Be Solu	42.090000	https://www.google.com/maps/dir/?api=1&origin=RSUD+Sele+Be+Solu&destination=RSUD+Sele+Be+Solu
Jalan Poncowati	RSUD Sele Be Solu	38.810001	https://www.google.com/maps/dir/?api=1&origin=Jalan+Poncowati&destination=RSUD+Sele+Be+Solu
Kantor Bupati Kabupaten Sorong	Masjid Agung Al-Istiqomah Almas	34.299999	https://www.google.com/maps/dir/?api=1&origin=Kantor+Bupati+Kabupaten+Sorong&destination=Masjid+Agung+Al-Istiqomah+Almas
Universitas Muhammadiyah Sorong	Taman Wisata Alimas	22.440001	https://www.google.com/maps/dir/?api=1&origin=Universitas+Muhammadiyah+Sorong&destination=Taman+Wisata+Alimas
Kantor Bupati Kabupaten Sorong	Pasar Baru Alimas	22.060001	https://www.google.com/maps/dir/?api=1&origin=Kantor+Bupati+Kabupaten+Sorong&destination=Pasar+Baru+Alimas
Jalan Poncowati	Polres Kabupaten Sorong	17.090000	https://www.google.com/maps/dir/?api=1&origin=Jalan+Poncowati&destination=Polres+Kabupaten+Sorong
bandara domine eduard osok	Polres Kabupaten Sorong	46.630001	https://www.google.com/maps/dir/?api=1&origin=bandara+domine+eduard+osok&destination=Polres+Kabupaten+Sorong
Pasar Baru Alimas	Jalan Poncowati	22.730000	https://www.google.com/maps/dir/?api=1&origin=Pasar+Baru+Alimas&destination=Jalan+Poncowati
RSUD Sele Be Solu	RSUD Kabupaten Sorong	41.180000	https://www.google.com/maps/dir/?api=1&origin=RSUD+Sele+Be+Solu&destination=RSUD+Kabupaten+Sorong
Jalan Poncowati	Pasar Baru Alimas	17.350000	https://www.google.com/maps/dir/?api=1&origin=Jalan+Poncowati&destination=Pasar+Baru+Alimas
Jalan Poncowati	RSUD Kabupaten Sorong	34.029999	https://www.google.com/maps/dir/?api=1&origin=Jalan+Poncowati&destination=RSUD+Kabupaten+Sorong
Universitas Muhammadiyah Sorong	Pasar Warmon Alimas	35.900002	https://www.google.com/maps/dir/?api=1&origin=Universitas+Muhammadiyah+Sorong&destination=Pasar+Warmon+Alimas
RSUD Sele Be Solu	RSUD Kabupaten Sorong	44.799999	https://www.google.com/maps/dir/?api=1&origin=RSUD+Sele+Be+Solu&destination=RSUD+Kabupaten+Sorong
Universitas Muhammadiyah Sorong	SPBU Almas 982, 03	43.220001	https://www.google.com/maps/dir/?api=1&origin=Universitas+Muhammadiyah+Sorong&destination=SPBU+Almas+982,+03
RSUD Sele Be Solu	SPBU Almas 982, 03	36.150002	https://www.google.com/maps/dir/?api=1&origin=RSUD+Sele+Be+Solu&destination=SPBU+Almas+982,+03

gambar 3.35 Menampilkan Hasil Waktuh Tempuh

(Sumber: <https://colab.research.google.com/>)

Program ini digunakan untuk memprediksi waktu tempuh perjalanan dari satu tempat ke tempat lain berdasarkan kondisi-kondisi seperti cuaca, kepadatan jalan, dan lain-lain. Selain itu, program ini juga menyediakan link langsung ke Google Maps agar pengguna bisa melihat rute perjalanannya.

3.4.6 Pengujian Sistem

Tahap keenam dalam penelitian ini adalah pengujian sistem untuk menilai apakah aplikasi prediksi waktu tempuh perjalanan Sorong–Aimas berbasis web telah berfungsi sesuai dengan tujuan yang ditetapkan. Pengujian ini mencakup evaluasi kesesuaian sistem dengan kebutuhan pengguna serta identifikasi dan perbaikan kesalahan yang mungkin terjadi. Dua metode pengujian yang digunakan adalah *Blackbox Testing* dan *Usability Testing*, dengan langkah-langkah sebagai berikut:

- Blackbox Testing*: Pengujian ini dilakukan dengan memeriksa fungsi-fungsi utama sistem tanpa melihat kode internalnya. Model yang telah dilatih menggunakan algoritma *XGBoost* diharapkan mencapai akurasi yang tinggi dalam memprediksi waktu tempuh. Pengguna memulai dengan membuka aplikasi web, memasukkan parameter seperti titik keberangkatan, tujuan, waktu keberangkatan, dan kondisi cuaca, lalu sistem akan memberikan prediksi waktu tempuh.

Pengujian ini bertujuan untuk memastikan bahwa sistem berfungsi tanpa kendala dan menghasilkan prediksi yang akurat.

No	Fitur yang diuji	input	Ekspetasi output	Hasil aktual	status
1.	Input Lokasi Awal dan Tujuan	Lokasi Awal: Kota Sorong Tujuan: Kabupaten Aimas	Sistem menampilkan estimasi waktu tempuh	Estimasi muncul	Berhasil
2.	Input Jam Keberangkatan	08:00 WIT	Sistem menyesuaikan prediksi berdasarkan jam sibuk	Prediksi disesuaikan	Berhasil
3.	Input Cuaca (hujan ringan)	(hujan ringan) Kondisi cuaca: Hujan ringan	Sistem mempengaruhi estimasi waktu (waktu tempuh lebih lama)	Waktu tempuh bertambah	Berhasil
4.	Input Lokasi GPS (Real-time)	Koordinat: -0.8792, 131.2612	Sistem menerima dan memproses lokasi untuk prediksi	Lokasi terdeteksi	Berhasil

5.	Tombol “Hitung Waktu Tempuh” ditekan	Semua input telah diisi	Sistem melakukan perhitungan dan menampilkan hasil prediksi	Hasil ditampilkan	Berhasil
6.	Input kosong atau tidak lengkap	Lokasi Awal dikosongkan	Sistem memberikan peringatan dan tidak memproses prediksi	Peringatan muncul	Berhasil
7.	Akses Website dari Perangkat Mobile	Browser Android	Antarmuka tampil responsif dan berfungsi normal	Tampilan responsif	Berhasil

b) *Usability Testing*: metode evaluasi yang digunakan untuk menilai tingkat kemudahan penggunaan suatu sistem, baik aplikasi website maupun mobile, melalui interaksi langsung pengguna. Pengujian ini bertujuan untuk memastikan bahwa sistem dapat digunakan secara efektif, efisien, dan memberikan kepuasan kepada pengguna. Evaluasi usability tidak hanya berfokus pada fungsi teknis, tetapi juga pada kenyamanan antarmuka, kemudahan memahami alur penggunaan, dan kejelasan informasi yang ditampilkan. Dengan demikian, Usability Testing berperan penting dalam mengidentifikasi permasalahan antarmuka yang mungkin membingungkan pengguna serta memberikan dasar bagi perbaikan lebih lanjut agar sistem lebih ramah pengguna.

1. Perencanaan pengujian

a. Tujuan

- 1) Menilai kemudahan penggunaan aplikasi
- 2) Mengukur kejelasan antarmuka dan alur navigasi
- 3) Mengetahui tingkat kepuasan pengguna terhadap sistem

b Responden

- 1) 5 orang pengguna awam yang belum pernah mencoba aplikasi

c Metode Pengujian

- 1) Observasi langsung, untuk mencatat waktu penyelesaian tugas, kesalahan, dan kebingungan pengguna

- 2) Kuesioner skala likert (1-5) dengan aspek penilaian, yaitu 1 = sangat buruk, 2 = buruk, 3 = cukup, 4 = baik, 5 = sangat baik.
- 3) Aspek yang dinilai, yaitu kemudahan penggunaan (*ease of use*), kejelasan antarmuka (*interface clarity*), kepuasan pengguna (*user satisfaction*)

2. Skenario Pengujian

Langkah-langkah yang dilakukan oleh responden selama pengujian adalah:

- a) Membuka aplikasi dan login untuk menilai kemudahan memulai penggunaan.
- b) Melakukan prediksi waktu tempuh perjalanan dengan memasukkan lokasi asal dan tujuan.
- c) Melihat visualisasi rute pada peta untuk mengecek kejelasan tampilan dan informasi sebelumnya.
- d) Melihat riwayat prediksi perjalanan untuk menilai kemudahan akses informasi sebelumnya.
- e) Logout dari aplikasi untuk menguji kemudahan dalam menutup sesi penggunaan.

3. Pengolahan dan Analisis Hasil

- a) Hasil pengujian dikumpulkan melalui observasi dan kuesioner.
- b) Sebagian besar responden berhasil menyelesaikan semua tugas dengan cepat dan tanpa bantuan.

- c) Beberapa pengguna mengalami kebingungan kecil saat pertama kali mengakses riwayat prediksi, namun tidak menghambat keseluruhan proses.
- d) Analisis menunjukkan:
 - 1) Kemudahan penggunaan: pengguna merasa alur sistem intuitif dan mudah diikuti.
 - 2) Kejelasan antarmuka: desain tampilan cukup jelas, tetapi disarankan untuk memperjelas ikon dan label.
 - 3) Kepuasan pengguna: mayoritas responden menyatakan puas dengan fungsi dan kecepatan sistem.

4. Rata-Rata Skor Kepuasan

Hasil penilaian kuesioner dari 5 responden menunjukkan nilai rata-rata sebagai berikut:

- a. Kemudahan Penggunaan: 4,4 (Sangat Baik)
- b. Kejelasan Antarmuka: 4,4 (Sangat Baik)
- c. Kepuasan Pengguna: 4,6 (Sangat Baik)

Secara keseluruhan, nilai rata-rata usability testing menunjukkan bahwa aplikasi sudah layak digunakan oleh pengguna. Perbaikan kecil seperti penambahan panduan singkat dan penyempurnaan desain tombol disarankan untuk meningkatkan pengalaman pengguna.

BAB IV

PENUTUP

4.1 KESIMPULAN

Berdasarkan hasil penelitian Implementasi Prediksi Waktu Tempuh Perjalanan Sorong–Aimas Berbasis Website Menggunakan Model XGBoost dan Fitur Lokasi Dinamis, sistem yang dikembangkan berhasil memberikan prediksi waktu tempuh dengan tingkat akurasi sebesar 0,97 (97%), yang menunjukkan kinerja model sangat baik dalam memprediksi waktu perjalanan. Model XGBoost mampu memproses data dengan efektif dan menangkap pola hubungan antara parameter seperti jarak, waktu keberangkatan, kepadatan lalu lintas, dan kondisi cuaca. Implementasi berbasis website juga berhasil menampilkan hasil prediksi dalam bentuk peta interaktif yang memudahkan pengguna memahami estimasi waktu tempuh secara real-time. Hasil pengujian sistem juga mendukung kelayakan aplikasi ini untuk digunakan oleh pengguna umum. Blackbox testing menunjukkan bahwa seluruh fungsi utama berjalan sesuai dengan yang diharapkan tanpa ada kesalahan fatal. Sementara itu, usability testing menghasilkan tingkat kepuasan pengguna yang tinggi dengan rata-rata skor kemudahan penggunaan 4,4, kejelasan antarmuka 4,4, dan kepuasan pengguna 4,6 yang masuk dalam kategori sangat baik. Meskipun demikian, pengguna memberikan masukan agar ditambahkan panduan penggunaan dan perbaikan tampilan antarmuka agar lebih menarik serta lebih responsif di perangkat mobile. Secara keseluruhan, penelitian ini berhasil menunjukkan bahwa penerapan XGBoost dengan fitur lokasi dinamis sangat potensial untuk meningkatkan efisiensi transportasi lokal, khususnya untuk rute perjalanan Sorong–Aimas.

4.2 SARAN

Berdasarkan hasil penelitian dan pengujian yang telah dilakukan, terdapat beberapa saran yang dapat dijadikan acuan untuk pengembangan sistem di masa mendatang. Pertama, antarmuka aplikasi disarankan untuk diperbaiki agar lebih responsif dan menarik, khususnya untuk tampilan di perangkat mobile. Penambahan panduan penggunaan singkat atau tutorial interaktif juga diharapkan dapat membantu pengguna baru memahami alur penggunaan aplikasi dengan lebih

mudah. Kedua, pengembangan sistem dapat diperluas dengan memanfaatkan data real-time seperti kondisi lalu lintas terkini dan pembaruan cuaca secara langsung untuk meningkatkan keakuratan prediksi waktu tempuh. Selain itu, model XGBoost dapat dioptimalkan lebih lanjut dengan hyperparameter tuning lanjutan atau dikombinasikan dengan metode machine learning lain untuk memperbaiki hasil prediksi. Terakhir, diperlukan pengujian lebih luas dengan jumlah responden yang lebih banyak agar evaluasi usability semakin representatif dan hasil prediksi dapat diuji keandalannya dalam berbagai kondisi perjalanan.

DAFTAR PUSTAKA

- Akurasi, P., Xgboost, A., & City, W. J. (2025). Perbandingan akurasi algoritma xgboost dan svr dalam prediksi harga cryptocurrency. *Jurnal Ilmu Komputer Dan Sistem Informasi PERBANDINGAN*, 13(11440), 1.
- Alkaff, M., & Rizky Baskara, A. (2023). Penerapan Metode XGBoost Untuk Memprediksi Jumlah Kejadian Kecelakaan Lalu Lintas di Kota Banjarmasin. In *Generation Journal* (Vol. 7, Issue 1).
- Asari, F. L., Meimaharini, R. S., & Khotimah, T. (2025). Implementasi Sistem Perpustakaan Berbasis Web untuk Meningkatkan Efisiensi Layanan Peminjaman dan Pengguna. *Bit-Tech(Binary Digital -Technology)*, 7(3), 1–5. <https://doi.org/10.32877/bt.v7i3.2185>
- Asnawi, M. F., Bisono, H. H., & Megantara, M. A. (2024). Aplikasi Prediksi Banjir Menggunakan Algoritma XGBoost Berbasis Website. *Journal of Economic, Management, Accounting and Technology (JEMATech)*, 7(2), 379–389.
- Astuti, W. P. (2024). Penggunaan Aplikasi Machine Learning (MI) dalam Kurikulum Perubahan Iklim. *(1,2) Universitas Pendidikan, Indonesia*, 5(MI), 5620–5631.
- Azzahra, A. F. (2025). Konsep Perancangan Komponen Desain Furnitur Limbah Kulit Kopi (Studi Kasus Kelompok Tani Kopi Manglayang Bandung). *Jurnal Desain Indonesia*, 07(1), 1–14.
- Cao, Y., Forssten, M. P., Sarani, B., Montgomery, S., & Mohseni, S. (2023). Development and Validation of an XGBoost-Algorithm-Powered Survival Model for Predicting In-Hospital Mortality Based on 545,388 Isolated Severe Traumatic Brain Injury Patients from the TQIP Database. *Journal of Personalized Medicine*, 13(9), 10.3390. <https://doi.org/10.3390/jpm13091401>
- Fajrul Haqqi, M., Saroji, S., & Prakoso, S. (2023). An implementation of XGBoost algorithm to estimate effective porosity on well log data. *Journal of Physics: Conference Series*, 2498(1), 1742–6596. <https://doi.org/10.1088/1742-6596/2498/1/012011>
- ForouzeshNejad, A. A., Arabikhan, F., & Aheleroff, S. (2024). Optimizing Project Time and Cost Prediction Using a Hybrid XGBoost and Simulated Annealing

- Algorithm. *Machines*, 12(12), 10.3390.
<https://doi.org/10.3390/machines12120867>
- Garma, M., Rianto, A., Prasasti, A. L., & Novianty, A. (2024). Implementasi Model XGBoost untuk Prediksi Jumlah Transaksi dan Total Pendapatan di Jaringan Restoran CV Balibul. *Jurnal Nasional SAINS Dan TEKNIK (JNST)*, 2(2), 43–46.
- Gelar Guntara, R. (2023). Pemanfaatan Google Colab Untuk Aplikasi Pendekripsi Masker Wajah Menggunakan Algoritma Deep Learning YOLOv7. *Jurnal Teknologi Dan Sistem Informasi Bisnis*, 5(1), 55–60.
<https://doi.org/10.47233/jteksis.v5i1.750>
- Jaka Naufal Semendawai, Indah Febiola, Bima Pamungkas, & Muhammad Deka Ruliansyah. (2021). Perancangan Aplikasi Otomatisasi Menggunakan Bahasa Pemrograman Python Pada Aktivitas Monitoring Pemakaian Data Harian Kartu Internet Of Things. *Jurnal Rekayasa Elektro Sriwijaya*, 3(1), 193–198.
<https://doi.org/10.36706/jres.v3i1.42>
- Jalan, D. M. T., Transportasi, P., Bali, D., Samsam, D., & Tabanan, K. (2024). METODE ESTIMASI INSTANTANEOUS MODEL PADA RUAS. *Jurnal Teknik Transportasi Logistik Dan Otomotif (JUTAGO)*, 1(1), 24–31.
- Jan Melvin Ayu Soraya Dachi, & Pardomuan Sitompul. (2023). Analisis Perbandingan Algoritma XGBoost dan Algoritma Random Forest Ensemble Learning pada Klasifikasi Keputusan Kredit. *Jurnal Riset Rumpun Matematika Dan Ilmu Pengetahuan Alam*, 2(2), 87–103.
<https://doi.org/10.55606/jurrimipa.v2i2.1470>
- Jodeiri Shokri, B., Mirzaghorbanali, A., McDougall, K., Karunasena, W., Nourizadeh, H., Entezam, S., Hosseini, S., & Aziz, N. (2024). Data-Driven Optimised XGBoost for Predicting the Performance of Axial Load Bearing Capacity of Fully Cementitious Grouted Rock Bolting Systems. *Applied Sciences (Switzerland)*, 14(21), 9925. <https://doi.org/10.3390/app14219925>
- Maharani, N. F., Aldiansyah, M., Nugraha, M. A., & Pibriana, D. (2024). Analisis Metode dan Bidang Pengembangan Sistem Informasi Menggunakan Systematic Literature Review. *Jurnal Teknologi Sistem Informasi*, 5(1), 45–56. <https://doi.org/10.35957/jtsi.v5i1.6844>

- Melinda, M., Ramadhan Na, S. R., Nurdin, Y., & Yunidar, Y. (2023). Implementation of System Development Life Cycle (SDLC) on IoT-Based Lending Locker Application. *Jurnal RESTI (Rekayasa Sistem Dan Teknologi Informasi)*, 7(4), 982–987. <https://doi.org/10.29207/resti.v7i4.5047>
- Momeni, E., He, B., Abdi, Y., & Armaghani, D. J. (2023). Novel Hybrid XGBoost Model to Forecast Soil Shear Strength Based on Some Soil Index Tests. *CMES - Computer Modeling in Engineering and Sciences*, 136(3), 2527–2550. <https://doi.org/10.32604/cmes.2023.026531>
- Pateman, D., & Pramudia, G. (2024). Analisis Bibliometrik pada Efektivitas UI/UX pada Penerapan Web Development. *Media Jurnal Informatika*, 16(1), 48. <https://doi.org/10.35194/mji.v16i1.3879>
- Penelitian, J., & Pendidikan, I. (2025). Pembelajaran Coding di Sekolah Dasar. *Jurnal Penelitian Ilmu Pendidikan Indonesia*, 4(1), 62–68.
- Putra, B. E. (2023). Aplikasi Peminjaman Buku Pada Perpustakaan Sma Nur Hasanah Kota Medan Berbasis Website. *Jurnal Nasional Teknologi Komputer*, 3(4), 285–296. <https://doi.org/10.61306/jnastek.v3i4.112>
- Rahmi, E. R., Yumami, E., & Hidayasari, N. (2023). Analisis Metode Pengembangan Sistem Informasi Berbasis Website: Systematic Literature Review. *Remik*, 7(1), 821–834. <https://doi.org/10.33395/remik.v7i1.12177>
- Rayadin, M. A., Musaruddin, M., Saputra, R. A., & Isnawaty, I. (2024). Implementasi Ensemble Learning Metode XGBoost dan Random Forest untuk Prediksi Waktu Penggantian Baterai Aki. *BIOS: Jurnal Teknologi Informasi Dan Rekayasa Komputer*, 5(2), 111–119.
- Sari, I. P., Azzahrah, A., Qathrunada, I. F., Lubis, N., & Anggraini, T. (2022). Perancangan Sistem Absensi Pegawai Kantoran Secara Online pada Website Berbasis HTML dan CSS. *Blend Sains Jurnal Teknik*, 1(1), 8–15. <https://doi.org/10.56211/blendsains.v1i1.66>
- Sayyendra, A. P., Andriani, N., Ritonga, A. F., & Ariska, M. (2024). Analisis Curah Hujan Di Papua Barat Menggunakan Metode Empirical Orthogonal Funtion (EOF). *Journal Online of Physics*, 9(3), 26–31.
- Senubekti, M. A., Dajoreyta, G. L., & Anggraini, N. (2024). Pembuatan Desain UI/UX dengan Metode Prototyping pada Aplikasi Layanan Pengadilan Negeri

- Bale Bandung menggunakan Figma. *Jurnal Informatika Terpadu*, 10(1), 1–10. <https://doi.org/10.54914/jit.v10i1.1001>
- Setiyani, L. (2021). Desain Sistem : Use Case Diagram Pendahuluan. *Prosiding Seminar Nasional : Inovasi & Adopsi Teknologi 2021*, 246(September), 246–260.
- Uminingsih, Nur Ichsanudin, M., Yusuf, M., & Suraya, S. (2022). Pengujian Fungsional Perangkat Lunak Sistem Informasi Perpustakaan Dengan Metode Black Box Testing Bagi Pemula. *STORAGE: Jurnal Ilmiah Teknik Dan Ilmu Komputer*, 1(2), 1–8. <https://doi.org/10.55123/storage.v1i2.270>
- Usroh, L., Laily, N., & Munir, F. (2022). Manajemen Waktu dan Self Regulated Learning pada Siswa. *Jurnal Psikologi : Jurnal Ilmiah Fakultas Psikologi Universitas Yudharta Pasuruan*, 9(1), 47–63. <https://doi.org/10.35891/jip.v9i1.2762>
- Wijaya, H., Hostiadi, D. P., & Triandini, E. (2024). OPTIMIZATION OF XGBOOST ALGORITHM USING PARAMETER TUNNING IN RETAIL SALES PREDICTION Jurnal Nasional Pendidikan Teknik Informatika : JANAPATI | 770. *Jurnal Teknologi Dan Sistem Komputer (JTSiskom)*, 13(3), 769–786. <https://doi.org/https://doi.org/10.23887/janapati.v13i3.82214>
- Wilyanto, N., Firnando, J., Franko, B., Tanzil, S. P., Tan, H. C., & Hartati, E. (2023). Pembuatan Website Menggunakan Visual Studio Code di SMA Xaverius 3 Palembang. *Fordicate*, 3(1), 1–8.
- Wisnu Sukma Maulana. (2022). Implementasi Arsitektur Pemrograman Microservices Untuk Migrasi Sistem Informasi Pt. Dwidaya World Wide (Dwidaya Tour and Travel) Dengan Metodologi Sdlc Rapid Application Development. *Jurnal Ilmiah Teknik*, 1(1), 09–19. <https://doi.org/10.56127/juit.v1i1.565>
- Yenny Dwi Suharyani, D. (2023). Perencanaan Strategis Berkelanjutan. *Jurnal Ilmiah Global Education*, 2(2), 767–778.
- Yulianti, S. (2024). Analisis Usability Testing Pada Website Ukm Ldk Syahid Uin Syarif Hidayatullah Jakarta Menggunakan Metode System Usability Scale (Sus). *JATI (Jurnal Mahasiswa Teknik Informatika)*, 8(4), 8190–8196. <https://doi.org/10.36040/jati.v8i4.10715>

- Zahra, A., Riani, L., Kurniawan, F., Darmansah, F. Z., Valenza, I. L., & Nasir, M. (2024). Pengujian Black Box pada Website Jasa Angkutan dan Ekspedisi Menggunakan Teknik Boundary Value Analysis. *Media Jurnal Informatika*, 16(2), 1–7.
- Akuras, P., Xgboost, A., & City, W. J. (2025). Perbandingan akurasi algoritma xgboost dan svr dalam prediksi harga cryptocurrency. *Jurnal Ilmu Komputer Dan Sistem Informasi PERBANDINGAN*, 13(11440), 1.
- Alkaff, M., & Rizky Baskara, A. (2023). Penerapan Metode XGBoost Untuk Memprediksi Jumlah Kejadian Kecelakaan Lalu Lintas di Kota Banjarmasin. In *Generation Journal* (Vol. 7, Issue 1).
- Asari, F. L., Meimaharini, R. S., & Khotimah, T. (2025). Implementasi Sistem Perpustakaan Berbasis Web untuk Meningkatkan Efisiensi Layanan Peminjaman dan Pengguna. *Bit-Tech(Binary Digital -Technology)*, 7(3), 1–5. <https://doi.org/10.32877/bt.v7i3.2185>
- Asnawi, M. F., Bisono, H. H., & Megantara, M. A. (2024). Aplikasi Prediksi Banjir Menggunakan Algoritma XGBoost Berbasis Website. *Journal of Economic, Management, Accounting and Technology (JEMATech)*, 7(2), 379–389.
- Astuti, W. P. (2024). Penggunaan Aplikasi Machine Learning (MI) dalam Kurikulum Perubahan Iklim. *(1,2) Universitas Pendidikan, Indonesia*, 5(MI), 5620–5631.
- Azzahra, A. F. (2025). Konsep Perancangan Komponen Desain Furnitur Limbah Kulit Kopi (Studi Kasus Kelompok Tani Kopi Manglayang Bandung). *Jurnal Desain Indonesia*, 07(1), 1–14.
- Cao, Y., Forssten, M. P., Sarani, B., Montgomery, S., & Mohseni, S. (2023). Development and Validation of an XGBoost-Algorithm-Powered Survival Model for Predicting In-Hospital Mortality Based on 545,388 Isolated Severe Traumatic Brain Injury Patients from the TQIP Database. *Journal of Personalized Medicine*, 13(9), 10.3390. <https://doi.org/10.3390/jpm13091401>
- Fajrul Haqqi, M., Saroji, S., & Prakoso, S. (2023). An implementation of XGBoost algorithm to estimate effective porosity on well log data. *Journal of Physics: Conference Series*, 2498(1), 1742–6596. <https://doi.org/10.1088/1742-6596/2498/1/012011>

- ForouzeshNejad, A. A., Arabikhan, F., & Aheleroff, S. (2024). Optimizing Project Time and Cost Prediction Using a Hybrid XGBoost and Simulated Annealing Algorithm. *Machines*, 12(12), 10.3390. <https://doi.org/10.3390/machines12120867>
- Garma, M., Rianto, A., Prasasti, A. L., & Novianty, A. (2024). Implementasi Model XGBoost untuk Prediksi Jumlah Transaksi dan Total Pendapatan di Jaringan Restoran CV Balibul. *Jurnal Nasional SAINS Dan TEKNIK (JNST)*, 2(2), 43–46.
- Gelar Guntara, R. (2023). Pemanfaatan Google Colab Untuk Aplikasi Pendekripsi Masker Wajah Menggunakan Algoritma Deep Learning YOLOv7. *Jurnal Teknologi Dan Sistem Informasi Bisnis*, 5(1), 55–60. <https://doi.org/10.47233/jteksis.v5i1.750>
- Jaka Naufal Semendawai, Indah Febiola, Bima Pamungkas, & Muhammad Deka Ruliansyah. (2021). Perancangan Aplikasi Otomatisasi Menggunakan Bahasa Pemrograman Python Pada Aktivitas Monitoring Pemakaian Data Harian Kartu Internet Of Things. *Jurnal Rekayasa Elektro Sriwijaya*, 3(1), 193–198. <https://doi.org/10.36706/jres.v3i1.42>
- Jalan, D. M. T., Transportasi, P., Bali, D., Samsam, D., & Tabanan, K. (2024). METODE ESTIMASI INSTANTANEOUS MODEL PADA RUAS. *Jurnal Teknik Transportasi Logistik Dan Otomotif(JUTAGO)*, 1(1), 24–31.
- Jan Melvin Ayu Soraya Dachi, & Pardomuan Sitompul. (2023). Analisis Perbandingan Algoritma XGBoost dan Algoritma Random Forest Ensemble Learning pada Klasifikasi Keputusan Kredit. *Jurnal Riset Rumpun Matematika Dan Ilmu Pengetahuan Alam*, 2(2), 87–103. <https://doi.org/10.55606/jurrimipa.v2i2.1470>
- Jodeiri Shokri, B., Mirzaghorbanali, A., McDougall, K., Karunasena, W., Nourizadeh, H., Entezam, S., Hosseini, S., & Aziz, N. (2024). Data-Driven Optimised XGBoost for Predicting the Performance of Axial Load Bearing Capacity of Fully Cementitious Grouted Rock Bolting Systems. *Applied Sciences (Switzerland)*, 14(21), 9925. <https://doi.org/10.3390/app14219925>
- Maharani, N. F., Aldiansyah, M., Nugraha, M. A., & Pibriana, D. (2024). Analisis Metode dan Bidang Pengembangan Sistem Informasi Menggunakan

- Systematic Literature Review. *Jurnal Teknologi Sistem Informasi*, 5(1), 45–56. <https://doi.org/10.35957/jtsi.v5i1.6844>
- Melinda, M., Ramadhan Na, S. R., Nurdin, Y., & Yunidar, Y. (2023). Implementation of System Development Life Cycle (SDLC) on IoT-Based Lending Locker Application. *Jurnal RESTI (Rekayasa Sistem Dan Teknologi Informasi)*, 7(4), 982–987. <https://doi.org/10.29207/resti.v7i4.5047>
- Momeni, E., He, B., Abdi, Y., & Armaghani, D. J. (2023). Novel Hybrid XGBoost Model to Forecast Soil Shear Strength Based on Some Soil Index Tests. *CMES - Computer Modeling in Engineering and Sciences*, 136(3), 2527–2550. <https://doi.org/10.32604/cmes.2023.026531>
- Pateman, D., & Pramudia, G. (2024). Analisis Bibliometrik pada Efektivitas UI/UX pada Penerapan Web Development. *Media Jurnal Informatika*, 16(1), 48. <https://doi.org/10.35194/mji.v16i1.3879>
- Penelitian, J., & Pendidikan, I. (2025). Pembelajaran Coding di Sekolah Dasar. *Jurnal Penelitian Ilmu Pendidikan Indonesia*, 4(1), 62–68.
- Putra, B. E. (2023). Aplikasi Peminjaman Buku Pada Perpustakaan Sma Nur Hasanah Kota Medan Berbasis Website. *Jurnal Nasional Teknologi Komputer*, 3(4), 285–296. <https://doi.org/10.61306/jnastek.v3i4.112>
- Rahmi, E. R., Yumami, E., & Hidayasari, N. (2023). Analisis Metode Pengembangan Sistem Informasi Berbasis Website: Systematic Literature Review. *Remik*, 7(1), 821–834. <https://doi.org/10.33395/remik.v7i1.12177>
- Rayadin, M. A., Musaruddin, M., Saputra, R. A., & Isnawaty, I. (2024). Implementasi Ensemble Learning Metode XGBoost dan Random Forest untuk Prediksi Waktu Penggantian Baterai Aki. *BIOS: Jurnal Teknologi Informasi Dan Rekayasa Komputer*, 5(2), 111–119.
- Sari, I. P., Azzahrah, A., Qathrunada, I. F., Lubis, N., & Anggraini, T. (2022). Perancangan Sistem Absensi Pegawai Kantoran Secara Online pada Website Berbasis HTML dan CSS. *Blend Sains Jurnal Teknik*, 1(1), 8–15. <https://doi.org/10.56211/blendsains.v1i1.66>
- Sayyendra, A. P., Andriani, N., Ritonga, A. F., & Ariska, M. (2024). Analisis Curah Hujan Di Papua Barat Menggunakan Metode Empirical Orthogonal Funtion (EOF). *Journal Online of Physiscs*, 9(3), 26–31.

- Senubekti, M. A., Dajoreyta, G. L., & Anggraini, N. (2024). Pembuatan Desain UI/UX dengan Metode Prototyping pada Aplikasi Layanan Pengadilan Negeri Bale Bandung menggunakan Figma. *Jurnal Informatika Terpadu*, 10(1), 1–10. <https://doi.org/10.54914/jit.v10i1.1001>
- Setiyani, L. (2021). Desain Sistem: Use Case Diagram Pendahuluan. *Prosiding Seminar Nasional : Inovasi & Adopsi Teknologi 2021*, 246(September), 246–260.
- Uminingsih, Nur Ichsanudin, M., Yusuf, M., & Suraya, S. (2022). Pengujian Fungsional Perangkat Lunak Sistem Informasi Perpustakaan Dengan Metode Black Box Testing Bagi Pemula. *STORAGE: Jurnal Ilmiah Teknik Dan Ilmu Komputer*, 1(2), 1–8. <https://doi.org/10.55123/storage.v1i2.270>
- Usroh, L., Laily, N., & Munir, F. (2022). Manajemen Waktu dan Self Regulated Learning pada Siswa. *Jurnal Psikologi : Jurnal Ilmiah Fakultas Psikologi Universitas Yudharta Pasuruan*, 9(1), 47–63. <https://doi.org/10.35891/jip.v9i1.2762>
- Wijaya, H., Hostiadi, D. P., & Triandini, E. (2024). OPTIMIZATION OF XGBOOST ALGORITHM USING PARAMETER TUNNING IN RETAIL SALES PREDICTION Jurnal Nasional Pendidikan Teknik Informatika : JANAPATI | 770. *Jurnal Teknologi Dan Sistem Komputer (JTSiskom)*, 13(3), 769–786. <https://doi.org/https://doi.org/10.23887/janapati.v13i3.82214>
- Wilyanto, N., Firnando, J., Franko, B., Tanzil, S. P., Tan, H. C., & Hartati, E. (2023). Pembuatan Website Menggunakan Visual Studio Code di SMA Xaverius 3 Palembang. *Fordicate*, 3(1), 1–8.
- Wisnu Sukma Maulana. (2022). Implementasi Arsitektur Pemrograman Microservices Untuk Migrasi Sistem Informasi Pt. Dwidaya World Wide (Dwidaya Tour and Travel) Dengan Metodologi Sdlc Rapid Application Development. *Jurnal Ilmiah Teknik*, 1(1), 09–19. <https://doi.org/10.56127/juit.v1i1.565>
- Yenny Dwi Suharyani, D. (2023). Perencanaan Strategis Berkelanjutan. *Jurnal Ilmiah Global Education*, 2(2), 767–778.
- Yulianti, S. (2024). Analisis Usability Testing Pada Website Ukm Ldk Syahid Uin Syarif Hidayatullah Jakarta Menggunakan Metode System Usability Scale

- (Sus). *JATI (Jurnal Mahasiswa Teknik Informatika)*, 8(4), 8190–8196. <https://doi.org/10.36040/jati.v8i4.10715>
- Zahra, A., Riani, L., Kurniawan, F., Darmansah, F. Z., Valenza, I. L., & Nasir, M. (2024). Pengujian Black Box pada Website Jasa Angkutan dan Ekspedisi Menggunakan Teknik Boundary Value Analysis. *Media Jurnal Informatika*, 16(2), 1–7.
- Akurasi, P., Xgboost, A., & City, W. J. (2025). Perbandingan akurasi algoritma xgboost dan svr dalam prediksi harga cryptocurrency. *Jurnal Ilmu Komputer Dan Sistem Informasi PERBANDINGAN*, 13(11440), 1.
- Alkaff, M., & Rizky Baskara, A. (2023). Penerapan Metode XGBoost Untuk Memprediksi Jumlah Kejadian Kecelakaan Lalu Lintas di Kota Banjarmasin. In *Generation Journal* (Vol. 7, Issue 1).
- Asari, F. L., Meimaharini, R. S., & Khotimah, T. (2025). Implementasi Sistem Perpustakaan Berbasis Web untuk Meningkatkan Efisiensi Layanan Peminjaman dan Pengguna. *Bit-Tech(Binary Digital -Technology)*, 7(3), 1–5. <https://doi.org/10.32877/bt.v7i3.2185>
- Asnawi, M. F., Bisono, H. H., & Megantara, M. A. (2024). Aplikasi Prediksi Banjir Menggunakan Algoritma XGBoost Berbasis Website. *Journal of Economic, Management, Accounting and Technology (JEMATech)*, 7(2), 379–389.
- Astuti, W. P. (2024). Penggunaan Aplikasi Machine Learning (Ml) dalam Kurikulum Perubahan Iklim. (1,2) *Universitas Pendidikan, Indonesia*, 5(Ml), 5620–5631.
- Azzahra, A. F. (2025). Konsep Perancangan Komponen Desain Furnitur Limbah Kulit Kopi (Studi Kasus Kelompok Tani Kopi Manglayang Bandung). *Jurnal Desain Indonesia*, 07(1), 1–14.
- Cao, Y., Forssten, M. P., Sarani, B., Montgomery, S., & Mohseni, S. (2023). Development and Validation of an XGBoost-Algorithm-Powered Survival Model for Predicting In-Hospital Mortality Based on 545,388 Isolated Severe Traumatic Brain Injury Patients from the TQIP Database. *Journal of Personalized Medicine*, 13(9), 10.3390. <https://doi.org/10.3390/jpm13091401>
- Fajrul Haqqi, M., Saroji, S., & Prakoso, S. (2023). An implementation of XGBoost algorithm to estimate effective porosity on well log data. *Journal of Physics:*

- Conference Series*, 2498(1), 1742–6596. <https://doi.org/10.1088/1742-6596/2498/1/012011>
- ForouzeshNejad, A. A., Arabikhan, F., & Aheleroff, S. (2024). Optimizing Project Time and Cost Prediction Using a Hybrid XGBoost and Simulated Annealing Algorithm. *Machines*, 12(12), 10.3390. <https://doi.org/10.3390/machines12120867>
- Garma, M., Rianto, A., Prasasti, A. L., & Novianty, A. (2024). Implementasi Model XGBoost untuk Prediksi Jumlah Transaksi dan Total Pendapatan di Jaringan Restoran CV Balibul. *Jurnal Nasional SAINS Dan TEKNIK (JNST)*, 2(2), 43–46.
- Gelar Guntara, R. (2023). Pemanfaatan Google Colab Untuk Aplikasi Pendekripsi Masker Wajah Menggunakan Algoritma Deep Learning YOLOv7. *Jurnal Teknologi Dan Sistem Informasi Bisnis*, 5(1), 55–60. <https://doi.org/10.47233/jteksis.v5i1.750>
- Jaka Naufal Semendawai, Indah Febiola, Bima Pamungkas, & Muhammad Deka Ruliansyah. (2021). Perancangan Aplikasi Otomatisasi Menggunakan Bahasa Pemrograman Python Pada Aktivitas Monitoring Pemakaian Data Harian Kartu Internet Of Things. *Jurnal Rekayasa Elektro Sriwijaya*, 3(1), 193–198. <https://doi.org/10.36706/jres.v3i1.42>
- Jalan, D. M. T., Transportasi, P., Bali, D., Samsam, D., & Tabanan, K. (2024). METODE ESTIMASI INSTANTANEOUS MODEL PADA RUAS. *Jurnal Teknik Transportasi Logistik Dan Otomotif(JUTAGO)*, 1(1), 24–31.
- Jan Melvin Ayu Soraya Dachi, & Pardomuan Sitompul. (2023). Analisis Perbandingan Algoritma XGBoost dan Algoritma Random Forest Ensemble Learning pada Klasifikasi Keputusan Kredit. *Jurnal Riset Rumpun Matematika Dan Ilmu Pengetahuan Alam*, 2(2), 87–103. <https://doi.org/10.55606/jurrimipa.v2i2.1470>
- Jodeiri Shokri, B., Mirzaghorbanali, A., McDougall, K., Karunasena, W., Nourizadeh, H., Entezam, S., Hosseini, S., & Aziz, N. (2024). Data-Driven Optimised XGBoost for Predicting the Performance of Axial Load Bearing Capacity of Fully Cementitious Grouted Rock Bolting Systems. *Applied Sciences (Switzerland)*, 14(21), 9925. <https://doi.org/10.3390/app14219925>

- Maharani, N. F., Aldiansyah, M., Nugraha, M. A., & Pibriana, D. (2024). Analisis Metode dan Bidang Pengembangan Sistem Informasi Menggunakan Systematic Literature Review. *Jurnal Teknologi Sistem Informasi*, 5(1), 45–56. <https://doi.org/10.35957/jtsi.v5i1.6844>
- Melinda, M., Ramadhan Na, S. R., Nurdin, Y., & Yunidar, Y. (2023). Implementation of System Development Life Cycle (SDLC) on IoT-Based Lending Locker Application. *Jurnal RESTI (Rekayasa Sistem Dan Teknologi Informasi)*, 7(4), 982–987. <https://doi.org/10.29207/resti.v7i4.5047>
- Momeni, E., He, B., Abdi, Y., & Armaghani, D. J. (2023). Novel Hybrid XGBoost Model to Forecast Soil Shear Strength Based on Some Soil Index Tests. *CMES - Computer Modeling in Engineering and Sciences*, 136(3), 2527–2550. <https://doi.org/10.32604/cmes.2023.026531>
- Pateman, D., & Pramudia, G. (2024). Analisis Bibliometrik pada Efektivitas UI/UX pada Penerapan Web Development. *Media Jurnal Informatika*, 16(1), 48. <https://doi.org/10.35194/mji.v16i1.3879>
- Penelitian, J., & Pendidikan, I. (2025). Pembelajaran Coding di Sekolah Dasar. *Jurnal Penelitian Ilmu Pendidikan Indonesia*, 4(1), 62–68.
- Putra, B. E. (2023). Aplikasi Peminjaman Buku Pada Perpustakaan Sma Nur Hasanah Kota Medan Berbasis Website. *Jurnal Nasional Teknologi Komputer*, 3(4), 285–296. <https://doi.org/10.61306/jnastek.v3i4.112>
- Rahmi, E. R., Yumami, E., & Hidayasari, N. (2023). Analisis Metode Pengembangan Sistem Informasi Berbasis Website: Systematic Literature Review. *Remik*, 7(1), 821–834. <https://doi.org/10.33395/remik.v7i1.12177>
- Rayadin, M. A., Musaruddin, M., Saputra, R. A., & Isnawaty, I. (2024). Implementasi Ensemble Learning Metode XGBoost dan Random Forest untuk Prediksi Waktu Penggantian Baterai Aki. *BIOS: Jurnal Teknologi Informasi Dan Rekayasa Komputer*, 5(2), 111–119.
- Sari, I. P., Azzahrah, A., Qathrunada, I. F., Lubis, N., & Anggraini, T. (2022). Perancangan Sistem Absensi Pegawai Kantoran Secara Online pada Website Berbasis HTML dan CSS. *Blend Sains Jurnal Teknik*, 1(1), 8–15. <https://doi.org/10.56211/blendsains.v1i1.66>
- Sayyendra, A. P., Andriani, N., Ritonga, A. F., & Ariska, M. (2024). Analisis Curah

- Hujan Di Papua Barat Menggunakan Metode Empirical Orthogonal Funtion (EOF). *Journal Online of Physiscs*, 9(3), 26–31.
- Senubekti, M. A., Dajoreyta, G. L., & Anggraini, N. (2024). Pembuatan Desain UI/UX dengan Metode Prototyping pada Aplikasi Layanan Pengadilan Negeri Bale Bandung menggunakan Figma. *Jurnal Informatika Terpadu*, 10(1), 1–10. <https://doi.org/10.54914/jit.v10i1.1001>
- Setiyani, L. (2021). Desain Sistem : Use Case Diagram Pendahuluan. *Prosiding Seminar Nasional : Inovasi & Adopsi Teknologi 2021*, 246(September), 246–260.
- Uminingsih, Nur Ichsanudin, M., Yusuf, M., & Suraya, S. (2022). Pengujian Fungsional Perangkat Lunak Sistem Informasi Perpustakaan Dengan Metode Black Box Testing Bagi Pemula. *STORAGE: Jurnal Ilmiah Teknik Dan Ilmu Komputer*, 1(2), 1–8. <https://doi.org/10.55123/storage.v1i2.270>
- Usroh, L., Laily, N., & Munir, F. (2022). Manajemen Waktu dan Self Regulated Learning pada Siswa. *Jurnal Psikologi : Jurnal Ilmiah Fakultas Psikologi Universitas Yudharta Pasuruan*, 9(1), 47–63. <https://doi.org/10.35891/jip.v9i1.2762>
- Wijaya, H., Hostiadi, D. P., & Triandini, E. (2024). OPTIMIZATION OF XGBOOST ALGORITHM USING PARAMETER TUNNING IN RETAIL SALES PREDICTION Jurnal Nasional Pendidikan Teknik Informatika : JANAPATI | 770. *Jurnal Teknologi Dan Sistem Komputer (JTSiskom)*, 13(3), 769–786. <https://doi.org/https://doi.org/10.23887/janapati.v13i3.82214>
- Wilyanto, N., Firnando, J., Franko, B., Tanzil, S. P., Tan, H. C., & Hartati, E. (2023). Pembuatan Website Menggunakan Visual Studio Code di SMA Xaverius 3 Palembang. *Fordicate*, 3(1), 1–8.
- Wisnu Sukma Maulana. (2022). Implementasi Arsitektur Pemrograman Microservices Untuk Migrasi Sistem Informasi Pt. Dwidaya World Wide (Dwidaya Tour and Travel) Dengan Metodologi Sdlc Rapid Application Development. *Jurnal Ilmiah Teknik*, 1(1), 09–19. <https://doi.org/10.56127/juit.v1i1.565>
- Yenny Dwi Suharyani, D. (2023). Perencanaan Strategis Berkelanjutan. *Jurnal Ilmiah Global Education*, 2(2), 767–778.

- Yulianti, S. (2024). Analisis Usability Testing Pada Website Ukm Ldk Syahid Uin Syarif Hidayatullah Jakarta Menggunakan Metode System Usability Scale (Sus). *JATI (Jurnal Mahasiswa Teknik Informatika)*, 8(4), 8190–8196. <https://doi.org/10.36040/jati.v8i4.10715>
- Zahra, A., Riani, L., Kurniawan, F., Darmansah, F. Z., Valenza, I. L., & Nasir, M. (2024). Pengujian Black Box pada Website Jasa Angkutan dan Ekspedisi Menggunakan Teknik Boundary Value Analysis. *Media Jurnal Informatika*, 16(2), 1–7.

LAMPIRAN

Lampiran 1 Evaluasi Pengerjaan Kelompok 1

No	Nama	Bagian Pengerjaan Tugas Besar
1	Marsha Laura Putty	: Mengkordinasi kelompok, Menyusun Laporan, Cari Jurnal.
2	Diva Caesary Dzuhiati	: Menyusun Laporan, Membuat Tabel, Flowchart
3	Margareth V Berhitu	: Menyusun Laporan Dan Membuat Usecase Sistem
4	Yosia Tangdi	: Menyusun Laporan BAB 2 BAB 3 dan Membuat Google Colab
5	Fredrik A Tamumete	: Membuat desain UI Google Colab dan Sistem

Mengetahui Dosen pengampuh Mata Kuliah Algoritma dan Pemograman

Fajar R.B Putra,S.Kom.,M.Kom.
NIDN.1428099501



Lampiran 2 Lembar Asistensi

**UNIVERSITAS MUHAMMADIYAH SORONG
PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNIK**

Jl. Pendidikan no 27 km 8,kel. malaimsimsa, Telp (0951322382) Kota Sorong

Lembar Asistensi

Prediksi Nilai Akhir

Kelompok 1

- | | |
|--------------------------|----------------|
| 1. Marsha Laura Putty | (202455202040) |
| 2. Diva Caesary Dzuhiati | (202455202076) |
| 3. Margareth V Berhitu | (202455202060) |
| 4. Yosia Tangdi | (202455202068) |
| 5. Fredrik A Tamunete | (202455202091) |

No	Hari / tanggal	Variable Asistensi	Paraf Asdos	Paraf Dosen
1.		Judul		
2.	Minggu, 4 Mei 2025	Bab I Pendahuluan		
3.	Selasa, 20 Mei 2025	Bab II Tinjauan Pustaka		
4.	Sabtu, 28 Juni 2025	Bab III Metode Penelitian		
5.	Sabtu, 12 Juli 2025	Bab IV Penutup		
6.		Progres Proyek Prediksi nilai akhir		

Mengetahui

Asisten Dosen mata kuliah Algoritma dan Pemograman

Dosen Pengampu

FAJAR R.B PUTRA,S.KOM.,M.KOM.
NIDN. 1428099501

Asisten Dosen

ILHAM
Nim : 202255202071

Lampiran 3 Dokumentasi Kelompok

Lampiran 4 Github

<https://github.com/kelompok1-Alpro/Laporan-Alpro.git>

Lampiran 5 Implementasi Google Colab

Hubungkan Colab Dengan Drive

```
from google.colab import drive
drive.mount('/content/drive')
```

Install Library Xgboost

```
!pip install xgboost openpyxl
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score
from xgboost import XGBRegressor
```

Mengambil Atau Memuat Dataset

```
file_path = '/content/drive/MyDrive/PrediksiWaktu/dataset_perjalanan.xlsx'
df = pd.read_excel(file_path)
df.head()
```

Memisahkan Fitur Target Dan Kategori

```
data = df.copy()
features = [
    'Cuaca', 'Kepadatan', 'Jam_Berangkat', 'Persimpangan',
    'Ada_Perbaikan_Jalan', 'Hari',
    'Tanjakan_Kecil', 'Tanjakan_Sedang', 'Tanjakan_Besar',
    'Turunan_Kecil', 'Turunan_Sedang', 'Turunan_Besar',
    'Tanjakan_Turunan', 'Jumlah_Berhenti', 'Jarak_KM'
]
target = 'Waktu_Tempuh'
```

```
categorical = ['Cuaca', 'Kepadatan', 'Ada_Perbaikan_Jalan', 'Hari']
le_dict = {}
for col in categorical:
    le = LabelEncoder()
    data[col] = le.fit_transform(data[col])
    le_dict[col] = le
```

Melakukan Pelebelan Pada Features Dan Target

```
X = data[features]
y = data[target]
```

```
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.2, random_state=42)
```

```
model = XGBRegressor(n_estimators=100, learning_rate=0.1, max_depth=5)
model.fit(X_train, y_train)
```

```
y_pred = model.predict(X_test)
```

Melakukan Evaluasi Model Xgboost

```
mae = mean_absolute_error(y_test, y_pred)
rmse = np.sqrt(mean_squared_error(y_test, y_pred))
```

```
print("== Evaluasi Model XGBoost ==")
```

```
print(f"MAE : {mae:.2f} menit")
print(f"RMSE : {rmse:.2f} menit")
```

Melatih Model Xgboost

```
import xgboost as xgb
```

```
model = xgb.XGBRegressor(objective='reg:squarederror', n_estimators=100)
model.fit(X_train, y_train)
```

Menampilkan Heat Map Dan Korelasi Antar Fitur

```
import matplotlib.pyplot as plt
```

```
import seaborn as sns
```

```
import pandas as pd
```

```
import numpy as np
```

```
df_numeric_for_corr = data.select_dtypes(include=np.number)
```

```
if df_numeric_for_corr.empty:
```

```
    print("Warning: No numeric columns found in data for correlation heatmap.")
```

```
else:
```

```
    plt.figure(figsize=(20, 6))
```

```
    correlation_matrix = df_numeric_for_corr.corr()
```

```
    sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm')
```

```
    plt.title("Heatmap Korelasi Antar Fitur (Numerik)")
```

```
    plt.show()
```

Menampilkan Sebaran Waktu Prediksi Vs Aktual Per Rute

```
import matplotlib.pyplot as plt
```

```
import numpy as np
```

```
# Buat prediksi simulasi (jika belum ada kolom prediksi asli)
```

```
np.random.seed(42)
```

```
df['Prediksi'] = df['Waktu_Tempuh'] + np.random.normal(0, 3, len(df))
```

```
# Gabungkan asal dan tujuan jadi satu rute
```

```
df['Rute'] = df['Lokasi_Asal'] + ' → ' + df['Lokasi_Tujuan']
```

```
# Buat plot
```

```
plt.figure(figsize=(10, 7))
```

```
# Warna otomatis untuk tiap rute
```

```

unique_routes = df['Rute'].unique()
colors = plt.cm.tab20.colors # maksimal 20 warna berbeda
;
for i, rute in enumerate(unique_routes):
    subset = df[df['Rute'] == rute]
    plt.scatter(
        subset['Waktu_Tempuh'],
        subset['Prediksi'],
        label=rute,
        alpha=0.7,
        s=30,
        color=colors[i % len(colors)])
)

# Garis ideal
min_val = min(df['Waktu_Tempuh'].min(), df['Prediksi'].min())
max_val = max(df['Waktu_Tempuh'].max(), df['Prediksi'].max())
plt.plot([min_val, max_val], [min_val, max_val], 'r--', label='Garis Ideal')

# Label
plt.title('Sebaran Waktu Tempuh Prediksi vs Aktual per Rute')
plt.xlabel('Waktu Tempuh Aktual (menit)')
plt.ylabel('Waktu Tempuh Prediksi (menit)')
plt.legend(fontsize=8, loc='upper left', bbox_to_anchor=(1, 1))
plt.grid(True)
plt.tight_layout()
plt.show()

```

Menampilkan Fitur Penting Xgboost

```

xgb.plot_importance(model, importance_type='gain', xlabel='Feature Importance')
plt.title('Feature Importance - XGBoost')
plt.show()

```

Menampilkan Hasil Akurasi Model

```

from sklearn.linear_model import Ridge
from sklearn.model_selection import train_test_split
from sklearn.metrics import r2_score

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=42)

ridge_model = Ridge(alpha=1.0)
ridge_model.fit(X_train, y_train)

y_pred = ridge_model.predict(X_test)
r2 = r2_score(y_test, y_pred)
print("R2 Ridge:", round(r2, 4))

```

Box Plot Distribusi Waktu Per Rute Perjalanan

```
# Buat kolom Rute: Asal → Tujuan
df['Rute'] = df['Lokasi_Asal'] + ' → ' + df['Lokasi_Tujuan']

# Buat boxplot berdasarkan rute
plt.figure(figsize=(14, 6))
sns.boxplot(x='Rute', y='Waktu_Tempuh', data=df, palette='Set2')

plt.title('Distribusi Waktu Tempuh per Rute Perjalanan', fontsize=14)
plt.xlabel('Rute Perjalanan')
plt.ylabel('Waktu Tempuh (menit)')
plt.xticks(rotation=45, ha='right')
plt.grid(True)
plt.tight_layout()
plt.show()
```

Mengecek Kolom Yang Tidak Ter Panggil

```
print(df.isnull().sum())
```

Mengecek Jika Ada Duplikat

```
print("Jumlah duplikat:", df.duplicated().sum())
```

Menghapus Duplikat

```
df = df.drop_duplicates()
```

Distribusi Waktu Tempuh Per Rute (Semua Rute)

```
df['Rute'] = df['Lokasi_Asal'] + ' → ' + df['Lokasi_Tujuan']
```

```
g = sns.displot(
    data=df,
    x='Waktu_Tempuh',
    col='Rute',
    col_wrap=4,
    kde=True,
    bins=15,
    color='skyblue',
    facet_kws={'sharex': False, 'sharey': False}
)
```

```
g.fig.subplots_adjust(top=0.94)
g.fig.suptitle('Distribusi Waktu Tempuh per Rute (Semua Rute)', fontsize=16)
plt.show()
```

Analisis Box Plot & Distribusi Per Rute

(Menampilkan Semua Rute)

```
import matplotlib.pyplot as plt
import seaborn as sns
import pandas as pd
import numpy as np
```

```
df.columns = df.columns.str.strip()
df['Rute'] = df['Lokasi_Asal'].astype(str).str.strip() + " → " + \
    df['Lokasi_Tujuan'].astype(str).str.strip()
```

```
num_cols = df.select_dtypes(include=['int64', 'float64']).columns.tolist()
print(f'Kolom numerik ({len(num_cols)}):', num_cols)
```

```
show_histograms = True      # True = tampilkan hist+KDE per rute
wrap_cols     = 3          # berapa subplot per baris utk histogram
```

```
plt.figure(figsize=(2.5*len(num_cols), 6))
sns.boxplot(data=df[num_cols], orient='h', palette='Set2')
plt.title('Ringkasan Boxplot – Semua Fitur Numerik')
plt.grid(axis='x', linestyle='--', alpha=0.3)
plt.show()
```

```
all_routes = df['Rute'].unique()
print(f'Total rute unik: {len(all_routes)}")
```

```
for col in num_cols:
    plt.figure(figsize=(max(10, 1.2*len(all_routes)), 6))
    sns.boxplot(x='Rute', y=col, data=df, palette='pastel')
    plt.title(f'Boxplot {col} per Rute (Semua Rute)')
    plt.xticks(rotation=90)
    plt.grid(axis='y', linestyle='--', alpha=0.3)
    plt.tight_layout()
    plt.show()
```

```
if show_histograms:
    g = sns.displot(
        data=df, x=col, col='Rute', col_wrap=wrap_cols,
        kde=True, color='skyblue', height=4, aspect=1.2
    )
    g.set_titles(col_template=f'{col} - {{col_name}}')
    g.set_axis_labels(col, 'Count')
    plt.tight_layout()
    plt.show()
```

```
print(" Selesai! Visualisasi dibuat untuk semua rute dan", len(num_cols), "fitur numerik.")
```

Regresi Waktu_Tempuh – Xgboost + Encoding & Scaling

```

from sklearn.preprocessing import StandardScaler
import xgboost as xgb
import matplotlib.pyplot as plt
import seaborn as sns
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score

df_model = df.copy()

categorical_cols = [
    'Cuaca', 'Kepadatan', 'Hari',
    'Klasifikasi_Tanjakan', 'Klasifikasi_Turunan',
    'Ada_Perbaikan_Jalan', 'Rute'
]

for col in categorical_cols:
    if col in df_model.columns:
        le = LabelEncoder()
        df_model[col] = le.fit_transform(df_model[col].astype(str))
    else:
        print(f"Kolom '{col}' tidak ditemukan & dilewati.")

drop_cols = ['Lokasi_Asal', 'Lokasi_Tujuan',
             'Koordinat_Asal', 'Koordinat_Tujuan']
df_model = df_model.drop(columns=[c for c in drop_cols if c in
                                   df_model.columns])

target_col = 'Waktu_Tempuh'
if target_col not in df_model.columns:
    raise KeyError(f"Kolom target '{target_col}' tidak ditemukan!")

X = df_model.drop(columns=[target_col])
y = df_model[target_col]

numeric_cols = X.select_dtypes(include=np.number).columns.tolist()
scaler = StandardScaler()
X[numeric_cols] = scaler.fit_transform(X[numeric_cols])

X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.2, random_state=42, shuffle=True
)

model = XGBRegressor(
    n_estimators=300,
    learning_rate=0.05,
)

```

```

max_depth=6,
subsample=0.8,
colsample_bytree=0.8,
random_state=42,
objective='reg:squarederror'
)
model.fit(X_train, y_train)

y_pred = model.predict(X_test)

mae = mean_absolute_error(y_test, y_pred)
mse = mean_squared_error(y_test, y_pred)
rmse = np.sqrt(mse)
r2 = r2_score(y_test, y_pred)

print(" Pelatihan selesai!")
print(f" MAE : {mae:.2f}")
print(f" MSE : {mse:.2f}")
print(f" RMSE : {rmse:.2f}")
print(f" R2 : {r2:.4f}")

comparison_df = pd.DataFrame({
    "Nilai_Aslu": y_test.reset_index(drop=True),
    "Prediksi" : y_pred
})
print("\n📋 Contoh perbandingan:")
print(comparison_df.head(10))

plt.figure(figsize=(10,5))
plt.plot(comparison_df["Nilai_Aslu"].values, label="Nilai Asli", marker='o')
plt.plot(comparison_df["Prediksi"].values, label="Prediksi", marker='x',
linestyle='--')
plt.title("Prediksi vs Nilai Asli")
plt.xlabel("Index")
plt.ylabel("Waktu Tempuh")
plt.legend()
plt.grid(True, linestyle='--', alpha=0.3)
plt.tight_layout()
plt.show()

```

Prediksi Waktu Tempuh + Google Maps Link

```

import numpy as np
import pandas as pd
from sklearn.preprocessing import LabelEncoder, StandardScaler
from urllib.parse import quote_plus

df_pred = df.copy()

```

```

df_pred['Lokasi_Asal'] = df_pred['Lokasi_Asal'].replace({
    'Universitas Pendidikan Muhammadiyah Sorong': 'UNIMUDA Sorong'
})
df_pred['Lokasi_Tujuan'] = df_pred['Lokasi_Tujuan'].replace({
    'Universitas Pendidikan Muhammadiyah Sorong': 'UNIMUDA Sorong'
})
lokasi_asal = df_pred['Lokasi_Asal'].str.strip()
lokasi_tujuan = df_pred['Lokasi_Tujuan'].str.strip()

categorical_cols = [
    'Cuaca', 'Kepadatan', 'Hari',
    'Ada_Perbaikan_Jalan', 'Klasifikasi_Tanjakan', 'Klasifikasi_Turunan', 'Rute'
]

for col in categorical_cols:
    if col in df_pred.columns:
        le = LabelEncoder()
        df_pred[col] = le.fit_transform(df_pred[col].astype(str))

drop_cols = ['Lokasi_Asal', 'Lokasi_Tujuan', 'Koordinat_Asal',
'Koordinat_Tujuan', 'Waktu_Tempuh']
X_pred = df_pred.drop(columns=[c for c in drop_cols if c in df_pred.columns])

numeric_cols = X_pred.select_dtypes(include=np.number).columns
scaler = StandardScaler()
X_pred[numeric_cols] = scaler.fit_transform(X_pred[numeric_cols])

y_pred_all = model.predict(X_pred)

def make_maps_url(asal, tujuan):
    return
f"https://www.google.com/maps/dir/?api=1&origin={quote_plus(asal)}&destination={quote_plus(tujuan)}"

result_df = pd.DataFrame({
    "Lokasi Asal": lokasi_asal,
    "Lokasi Tujuan": lokasi_tujuan,
    "Prediksi Waktu Tempuh (menit)": np.round(y_pred_all, 2)
})
result_df["Google Maps"] = result_df.apply(lambda r: make_maps_url(r["Lokasi Asal"], r["Lokasi Tujuan"]), axis=1)

pd.set_option("display.max_rows", None)
pd.set_option("display.max_colwidth", None)
print("Seluruh hasil prediksi:")
print(result_df.to_string(index=False))

```

```
excel_path = "/content/drive/My  
Drive/PrediksiWaktu/hasil_prediksi_lengkap.xlsx"  
csv_path = "/content/drive/My Drive/PrediksiWaktu/hasil_prediksi_lengkap.csv"  
result_df.to_excel(excel_path, index=False)  
result_df.to_csv(csv_path, index=False)
```

Menyimpan Model

```
import joblib
```

```
joblib.dump(model, 'xgboost_model.pkl')  
joblib.dump(le_dict, 'le_dict.pkl')
```