p-ISSN : 2406-7733
e-ISSN : 2597-9922

PEMANFAATAN TEKNIK *MACHINE LEARNING* DALAM MEMPREDIKSI KEPADATAN LALU LINTAS GUNA EFISIENSI TRANSPORTASI

Rizal Muslim Sinaga[1], M. Rosyid Fauzan[2], Ega Pratama [3], M. Rizki Alfahri [4], Kana Saputra[5]
Ilmu Komputer, Matematika dan Ilmu Pengetahuan Alam,Universitas Negeri Medan
Jl. William Iskandar Ps. V, Kenangan Baru, Kec. Percut Sei Tuan, Kabupaten Deli Serdang, Sumatera Utara
E-mail: *sinaga.rizal456@gmail.com[1], rosyid.4233250009@mhs.unimed.ac.id[2], egap1840@gmail.com [3], mhdrizkialfahri@mhs.unimed.ac.id [4], kanasaputras@unimed.ac.id[5]

Abstrak - Kemacetan lalu lintas di kota-kota besar semakin meningkat akibat pertumbuhan jumlah kendaraan yang tidak sebanding dengan kapasitas jalan. Kondisi ini berdampak pada peningkatan waktu perjalanan, polusi udara, serta risiko kecelakaan. Penelitian ini bertujuan untuk memprediksi tingkat kepadatan lalu lintas di Kota Medan menggunakan metode pembelajaran mesin. Tiga algoritma yang digunakan dalam penelitian ini adalah regresi linear, *random forest*, dan *XGBoost*. Data yang digunakan mencakup informasi lalu lintas dan cuaca yang diperoleh dari sumber terbuka. Tahapan penelitian meliputi eksplorasi data, pra-pemrosesan, rekayasa fitur, pelatihan model, dan evaluasi menggunakan beberapa metrik pengukuran kesalahan. Hasil penelitian menunjukkan bahwa algoritma *Random Forest* dengan penyesuaian parameter memiliki performa terbaik dibandingkan dengan algoritma lainnya. Model ini dapat digunakan sebagai alternatif dalam pengelolaan lalu lintas guna meningkatkan efisiensi mobilitas di kawasan perkotaan.

Kata Kunci: efisiensi transportasi, kepadatan lalu lintas, *machine learning*, prediksi lalu lintas

I. PENDAHULUAN

Setiap tahunnya, kepadatan arus lalu lintas di kota-kota besar terus meningkat. Bertambahnya kepemilikan kendaraan bermotor, baik roda dua maupun roda empat, menjadi salah satu faktor utama yang menyebabkan padatnya lalu lintas. Hal ini diperparah dengan tidak adanya batasan yang ketat dalam kepemilikan kendaraan pribadi. Situasi lalu lintas yang semakin padat telah menjadi tantangan serius bagi pengelolaan transportasi di berbagai kota di seluruh dunia. Tingginya tingkat kemacetan tidak hanya memperpanjang waktu perjalanan, tetapi juga berdampak negatif pada lingkungan serta kesejahteraan masyarakat. Selain itu, padatnya lalu lintas juga meningkatkan risiko kecelakaan. Pertumbuhan populasi yang pesat turut mendorong peningkatan mobilitas, yang berakibat pada bertambahnya volume lalu lintas dan jumlah kendaraan (Huizen, 2024).

Penerapan teknik *machine learning* telah menunjukkan potensi besar dalam memprediksi kondisi lalu lintas. Berbagai algoritma, seperti *Random Forest* dan *XGBoost*, telah digunakan untuk membangun model prediksi dengan tingkat akurasi yang tinggi. Misalnya, penelitian oleh Sari et al. (2023) membandingkan algoritma *Random Forest* dan *XGBoost* dalam memprediksi kepadatan lalu lintas dan menemukan bahwa *Random Forest* memiliki keunggulan dalam akurasi dan efisiensi perhitungan. Selain itu, metode berbasis *deep learning* seperti *Long Short-Term Memory* (*LSTM*) juga telah diuji dalam prediksi lalu lintas multi-server dan menunjukkan performa yang baik dalam menangani data *time-series* (Sakir, 2023).

Kepadatan lalu lintas tentunya dipengaruhi oleh berbagai faktor, salah satunya adalah kondisi cuaca. Oleh karena itu, prediksi kepadatan lalu lintas menjadi sangat penting untuk membantu dalam menghindari dan mengatasi kemacetan. Prediksi yang akurat dapat memberikan manfaat yang signifikan bagi pengguna jalan, manajemen transportasi, serta perencanaan kota. Beberapa penelitian telah mengaplikasikan berbagai teknik *machine learning* seperti *K-Nearest Neighbors* (*KNN*), *Random Forest*, dan *XGBoost* untuk meningkatkan akurasi dalam memprediksi kepadatan lalu lintas (Febrianto et al., 2024). Selain itu, teknik pengolahan citra juga telah digunakan untuk mendeteksi kepadatan lalu lintas menggunakan metode seperti *Canny Edge Detection* dan analisis berbasis pemrosesan gambar (Kurniasari & Jalinas, 2020; Kurniawan, Sajati, & Dinaryanto, 2017).

Dalam penelitian ini, kami mengumpulkan data lalu lintas dan cuaca melalui API *OpenWeather* dan *TomTom*, dengan validasi lokasi menggunakan *OpenStreetMap*. Data yang diperoleh mencakup informasi seperti *timestamp*, lokasi, kecepatan lalu lintas, tingkat kemacetan, suhu, kelembaban, dan kondisi cuaca. Tahap *preprocessing* dilakukan dengan pembersihan data, normalisasi, serta rekayasa fitur untuk meningkatkan performa model. Selanjutnya, kami membandingkan algoritma *Regresi Linear*, *Random Forest*, dan *XGBoost* untuk menentukan model yang paling akurat dalam memprediksi kemacetan lalu lintas di Medan.

Dengan memanfaatkan teknik *machine learning* yang tepat, diharapkan prediksi kepadatan lalu lintas dapat dilakukan dengan lebih akurat. Hal

p-ISSN : 2406-7733
e-ISSN : 2597-9922

ini akan mendukung pengambilan keputusan yang lebih efektif dalam manajemen transportasi serta perencanaan kota, sehingga dapat mengurangi dampak negatif dari kemacetan dan meningkatkan efisiensi mobilitas di perkotaan.

II. TINJAUAN PUSTAKA

Teknologi Pengendalian Lalu Lintas Berbasis Citra

Salah satu pendekatan dalam pengendalian lalu lintas modern adalah pemanfaatan teknologi berbasis *citra* untuk mendeteksi dan mengelola pergerakan kendaraan secara otomatis. Dengan kemajuan teknik *pengolahan citra* (*image processing*), sistem ini mampu mengidentifikasi kendaraan secara *real-time* dan mengambil keputusan berbasis data visual. Efendi dan Hutabri (2024) mengembangkan sistem deteksi plat nomor kendaraan menggunakan kamera *webcam* dan *OpenCV*, di mana *citra* kendaraan diproses untuk membuka akses portal secara otomatis. Teknologi ini menawarkan keunggulan dalam meningkatkan efisiensi lalu lintas dengan cara mendeteksi keberadaan kendaraan secara lebih akurat tanpa bergantung pada sensor konvensional.

Metode Deteksi Kepadatan Lalu Lintas

Beberapa penelitian telah dilakukan untuk mendeteksi kepadatan lalu lintas menggunakan berbagai metode. Misalnya, Faradila, Fibriliyanti, dan Nasron (2017) menggunakan metode *wavelet* untuk mendeteksi objek kendaraan pada setiap jalur persimpangan dan menghitung lama waktu lampu lalu lintas berdasarkan tingkat kepadatan jalan. Kemudian, Abdurrafi, Alawiy, dan Basuki (2023) membangun sistem deteksi, klasifikasi, dan penghitungan kendaraan menggunakan algoritma *YOLOv3* dengan kamera *CCTV*, yang terbukti mampu mendeteksi kendaraan dengan akurasi tinggi, yaitu *Precision* sebesar 99%, *Recall* sebesar 90%, dan *F1 Score* sebesar 94%.

Penelitian lain oleh Kurniasari dan Jalinas (2020) menggunakan metode *Canny Edge Detection* untuk menentukan koordinat *Region of Interest* (ROI) pada jalan dan menghitung persentase kepadatan berdasarkan data video. Hasilnya menunjukkan bahwa metode *Canny Edge* berhasil mendeteksi kendaraan di jalan serta menentukan tingkat kepadatan lalu lintas.

Perbandingan Algoritma Prediksi Kepadatan Lalu Lintas

Sari et al. (2023) membandingkan performa algoritma *Random Forest* dan *XGBoost* dalam memprediksi kepadatan lalu lintas. Hasilnya menunjukkan bahwa kedua algoritma memiliki akurasi sekitar 95%, tetapi *XGBoost* memiliki keunggulan dalam kecepatan prediksi yang 532% lebih cepat dibandingkan dengan *Random Forest*.

Optimasi Jalur dengan Algoritma Koloni Semut

Fariza, Basofi, dan Hidayat (2020) menerapkan algoritma *Ant Colony Algorithm* pada peta untuk menemukan jalur optimal berdasarkan jarak tempuh dengan mempertimbangkan kepadatan lalu lintas. Hasil percobaan menunjukkan bahwa algoritma ini mampu menemukan jalur terbaik berdasarkan jumlah lajur jalan dan jenis kendaraan.

Prediksi Kepadatan Lalu Lintas Berbasis Data

Prediksi kepadatan lalu lintas dapat dilakukan menggunakan data sensor lalu lintas yang mencakup kecepatan kendaraan, volume kendaraan, serta tingkat kemacetan yang dikumpulkan secara real-time. Faktor lingkungan seperti suhu ekstrem dan hujan lebat juga dapat mempengaruhi tingkat kecelakaan dan kepadatan lalu lintas di daerah perkotaan.

Data dari *OpenStreetMap* (*OSM*) dapat digunakan untuk meningkatkan akurasi model prediksi lalu lintas karena menyediakan informasi rinci tentang jaringan jalan dan infrastruktur transportasi (Huizen, 2024). Beberapa metode yang umum digunakan untuk prediksi kepadatan lalu lintas meliputi regresi linear, *Random Forest*, dan *XGBoost*. Model berbasis data ini memberikan wawasan yang lebih baik dalam mencegah kemacetan, sehingga memungkinkan sistem manajemen lalu lintas merespons perubahan kondisi dengan lebih efektif (Febrianto et al., 2024).

Linear Regression

Menurut Rudi et al. (2023), analisis regresi sederhana digunakan untuk memahami pengaruh suatu variabel terhadap variabel lainnya. Dalam analisis ini, variabel yang mempengaruhi disebut variabel independen (*X*), sedangkan variabel yang dipengaruhi disebut variabel dependen (*Y*). Tujuan utama dari analisis regresi adalah membangun model matematis yang menggambarkan hubungan antara kedua variabel tersebut, mengukur tingkat perubahan variabel dependen akibat perubahan variabel independen, serta melakukan prediksi terhadap nilai *Y* berdasarkan nilai *X*.

Persamaan regresi linier sederhana dinyatakan sebagai berikut:

$$Y = a + bX \quad \text{..... (1)}$$

dengan:

- a merupakan konstanta, yaitu nilai Y ketika X = 0, yang dihitung menggunakan persamaan:

$$a = \frac{(\Sigma Y)(\Sigma X^2) - (\sum X)(\sum XY)}{n\Sigma X^2 - (\Sigma X)^2} \quad \text{..... (2)}$$

p-ISSN : 2406-7733
e-ISSN : 2597-9922

- besarnya pengaruh X terhadap Y, yang diperoleh melalui persamaan:

$$b = \frac{n\sum XY - (\sum X)(\Sigma Y)}{n\Sigma X^2 - (\Sigma X)^2} \quad \text{..... (3)}$$

Dalam analisis regresi, hubungan antara X dan Y harus memiliki dasar teori atau bukti empiris dari penelitian sebelumnya agar hasil yang diperoleh memiliki validitas yang kuat.

XGBoost

XGBoost adalah pengembangan dari *gradient boosting*, yang diperkenalkan oleh Tianqi Chen pada 2014. Algoritma ini digunakan untuk regresi, klasifikasi, dan ranking, dengan prinsip pembaruan parameter secara berulang guna menurunkan *loss function* sebagai evaluasi model.

Keunggulan *XGBoost* terletak pada kemampuannya membangun *decision tree* yang lebih terstruktur, sehingga meningkatkan kinerja model sekaligus mengurangi risiko overfitting. Hasil prediksi diperoleh dengan menjumlahkan prediksi dari setiap pohon regresi yang dibangun. Algoritma ini juga cocok untuk data dengan fitur kategorikal dan tetap optimal meskipun terdapat ketidakseimbangan kelas dalam dataset.

Metode ini menggunakan fungsi objektif yang terdiri dari *loss function* dan regularisasi, seperti berikut:

$$Obj(\theta) = L(\theta) + \Omega(\theta) \quad \text{..... (4)}$$

dengan:

- L(θ) mengukur perbedaan antara nilai prediksi dan nilai aktual.
- Ω(θ) mengontrol kompleksitas model untuk mencegah *overfitting*.

Fungsi loss dapat dituliskan sebagai:

$$L(\theta) = \sum_{i=1}^{n} l(y_i,\ \hat{y}_i) \quad \text{..... (5)}$$

di mana y_i, adalah nilai aktual, $\hat{y}_i$nilai prediksi, dan n jumlah iterasi pelatihan. *XGBoost* menjadi metode unggulan dalam data science karena mampu menangani dataset besar secara efisien (Yulianti et al., 2023).

Random Forest

Amaliah et al., 2022 menyatakan bahwa *Random Forest* (RF) pertama kali diperkenalkan oleh Leo Breiman (2001) sebagai metode *ensemble* berbasis *decision tree* yang dapat meningkatkan akurasi klasifikasi dengan membangun banyak pohon keputusan secara acak. Tidak seperti metode *Classification and Regression Tree* (CART), RF tidak menerapkan pruning. Pemilihan fitur di setiap node internal dilakukan menggunakan *Indeks Gini*, yang dihitung sebagai:

$$Gini(S_i) = 1 - \sum_{i=0}^{C-1} p_i^2 \quad \text{..... (6)}$$

dengan p_i sebagai frekuensi relatif kelas C_i dalam dataset.

Proses pemisahan (*split*) menggunakan formula:

$$Gini_{split} = \sum_{i=0}^{k-1} \left(\frac{n_i}{n}\right) Gini(S_i) \quad \text{..... (7)}$$

Di mana n_i ini adalah jumlah sampel dalam subset S_i setelah pemisahan, dan n adalah jumlah total sampel pada node yang diberikan.

RF menggunakan pendekatan *majority voting* dari semua pohon yang dibangun. Fungsi margin dari RF diberikan oleh:

$$mr(X,Y) = P\Theta(h(X,\Theta) = Y) - \max\left(P\Theta(h(X,\Theta) = j)\right) untuk\ j \neq Y \quad \text{... (8)}$$

dengan $s = E_{\{X,Y\}[mr(X,Y)]}$ sebagai ukuran kekuatan himpunan pengklasifikasi. Dengan asumsi $s \geq 0$, batas atas kesalahan generalisasi RF dihitung sebagai:

$$P_E \leq \frac{\left(\bar{\rho}(1-s^2)\right)}{s^2} \quad \text{..... (9)}$$

Dengan $\bar{\rho}$ sebagai rata-rata korelasi antar pohon dalam RF, yang diperoleh dari:

$$\bar{\rho} = \frac{(E\Theta,\Theta'[\rho(\Theta,\Theta')\cdot sd(\Theta)\cdot sd(\Theta')])}{(E\Theta,\Theta'[sd(\Theta)\cdot sd(\Theta')])} \quad \text{..... (10)}$$

Metrik evaluasi

Dalam analisis regresi, tiga metrik evaluasi yang umum digunakan untuk menilai kinerja model adalah *Mean Squared Error* (MSE), *Root Mean Squared Error* (RMSE), dan *Mean Absolute Error* (MAE). MSE mengukur rata-rata dari kuadrat selisih antara nilai aktual dan prediksi, memberikan penekanan lebih besar pada kesalahan yang lebih besar karena efek pemangkatan. RMSE adalah akar kuadrat dari MSE, yang mengembalikan metrik ke skala yang sama dengan variabel target, sehingga memudahkan interpretasi ukuran rata-rata kesalahan. Sementara itu, MAE menghitung rata-rata dari nilai absolut selisih antara nilai aktual dan prediksi, memberikan gambaran langsung tentang besarnya kesalahan rata-rata tanpa mempertimbangkan arahnya. Pemilihan metrik yang tepat bergantung pada konteks dan tujuan analisis; misalnya, jika sensitivitas terhadap outlier diinginkan, MSE atau RMSE mungkin lebih sesuai, sedangkan MAE lebih robust terhadap outlier dan

p-ISSN : 2406-7733
e-ISSN : 2597-9922

memberikan interpretasi yang lebih intuitif tentang kesalahan rata-rata (Chicco et al., 2022).

Selain ketiga metrik tersebut, *Koefisien Determinasi* (R^2) juga sering digunakan untuk mengukur sejauh mana variabel independen dapat menjelaskan variabel dependen dalam model regresi. R^2 memiliki rentang nilai antara 0 hingga 1, di mana nilai mendekati 1 menunjukkan bahwa model mampu menjelaskan sebagian besar variasi dalam data, sedangkan nilai mendekati 0 menunjukkan bahwa model memiliki daya prediksi yang lemah (Chicco et al., 2022).

Berikut adalah rumus dari masing-masing metrik evaluasi:

-Mean Squared Error (MSE)

MSE menghitung rata-rata dari kuadrat selisih antara nilai aktual dan nilai prediksi:

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}i\,)^2 \text{..... } (11)$$

Keterangan:
- y_i = nilai aktual
- $\hat{y}$= nilai prediksi
- n= jumlah sampel

-Root Mean Squared Error (RMSE)

RMSE adalah akar dari MSE dan digunakan untuk memberikan gambaran tentang besarnya kesalahan dalam satuan yang sama dengan variabel target:

$$RMSE = \sqrt{\text{MSE}} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}i\,)^2} \text{.....} (12)$$

Penjelasan: RMSE lebih mudah diinterpretasikan dibandingkan MSE karena memiliki satuan yang sama dengan variabel target.

-Mean Absolute Error (MAE)

MAE menghitung rata-rata dari selisih absolut antara nilai aktual dan prediksi:

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}i\;i\,| \text{ ..... } (13)$$

Penjelasan: MAE tidak mengkuadratkan selisih nilai sehingga lebih tahan terhadap outlier dibandingkan MSE dan RMSE.

-R-squared Score (R^2 Score)

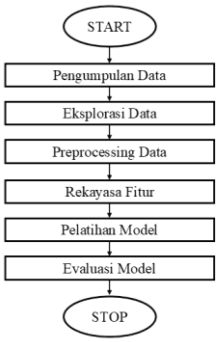
R^2 Score mengukur seberapa baik model menjelaskan variasi dalam data:

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}i\,)^2}{\sum_{i=1}^{n}(y_i - \hat{y}i\,)^2} \text{ ..... } (14)$$

Keterangan:
- R^2 = rata-rata nilai aktual
- Semakin tinggi nilai R^2 (mendekati 1), semakin baik model dalam menjelaskan data.

III. METODE PENELITIAN

Penelitian ini bertujuan untuk memprediksi kepadatan lalu lintas dengan menerapkan teknik *machine learning* menggunakan data lalu lintas dan cuaca yang dikumpulkan dari *OpenWeather API*, *TomTom API*, *OpenStreetMap* (OSM), *Nominatim*, dan *Overpass API*. Proses penelitian dilakukan melalui beberapa tahapan utama, yaitu eksplorasi data, preprocessing data, rekayasa fitur, pelatihan model, evaluasi model, dan prediksi kepadatan lalu lintas.



Gambar 1. Tahapan Penelitian

Pengumpulan Data

Data lalu lintas dan cuaca diperoleh dari berbagai sumber, termasuk *OpenWeather API* untuk data cuaca, *TomTom API* untuk informasi lalu lintas real-time, serta *OpenStreetMap* (OSM) dan *Overpass API* untuk data jaringan jalan dan geolokasi. *Nominatim* digunakan untuk proses geocoding. Dataset yang dikumpulkan mencakup 5000 data dengan atribut seperti nama jalan, koordinat, jenis jalan, jumlah lajur, suhu, kondisi cuaca, kecepatan kendaraan, tingkat kemacetan, serta waktu pengambilan data. Data dikumpulkan dalam radius 10 km di sekitar Medan selama lima hari (Jumat–Selasa), yang menjadi batasan dalam memprediksi pola lalu lintas di luar rentang tersebut.

Eksplorasi Data

Eksplorasi awal dilakukan untuk memahami struktur dan karakteristik dataset. Analisis mencakup distribusi nilai setiap fitur, seperti tren kecepatan rata-rata kendaraan dan tingkat kemacetan berdasarkan waktu. Selain itu, eksplorasi bertujuan mengidentifikasi nilai yang hilang, data tidak valid, serta mendeteksi anomali seperti outlier atau inkonsistensi yang dapat memengaruhi performa model. Hasil eksplorasi divisualisasikan untuk memberikan gambaran awal mengenai pola lalu lintas di kota Medan.

p-ISSN : 2406-7733
e-ISSN : 2597-9922

Preprocessing Data

Preprocessing bertujuan untuk meningkatkan kualitas data sebelum pelatihan model. Langkah-langkah yang dilakukan meliputi pembersihan data dengan menghapus nilai yang hilang, duplikat, dan data tidak relevan, serta ekstraksi fitur waktu dari kolom timestamp. Fitur kategori seperti *weather_condition*, *road_name*, dan *highway_type* dikonversi ke bentuk numerik menggunakan *one-hot encoding*, sementara fitur numerik dinormalisasi menggunakan *StandardScaler* agar memiliki skala yang seragam. Setelah *preprocessing*, eksplorasi ulang dilakukan untuk memastikan data siap digunakan dan menginterpretasikan pola lalu lintas di Medan berdasarkan dataset yang telah dibersihkan.

Pelatihan Model

Pelatihan model membandingkan *Regresi Linear*, *Random Forest*, dan *XGBoost*. Data dibagi menjadi *training set* dan *testing set* untuk memastikan generalisasi yang baik. *Feature engineering* dilakukan dengan menambahkan indikator hari libur, kemacetan sebelumnya, perubahan kecepatan, serta interaksi cuaca dan hari libur. *Regresi Linear* digunakan sebagai baseline, sementara *Random Forest* dan *XGBoost* dilatih dengan *GridSearchCV* untuk mencari *hyperparameter* terbaik. Model yang telah dilatih disimpan menggunakan *Joblib* untuk penggunaan ulang tanpa pelatihan ulang.

Evaluasi dan Prediksi

Evaluasi model dilakukan menggunakan beberapa metrik utama, yaitu *Mean Squared Error* (MSE), *Root Mean Squared Error* (RMSE), *Mean Absolute Error* (MAE), dan R^2 *Score*. Metrik-metrik ini digunakan untuk mengukur seberapa baik model dapat memprediksi tingkat kemacetan dengan membandingkan hasil prediksi terhadap nilai aktual.

IV. HASIL DAN PEMBAHASAN

Sample Dataset

Sebelum melakukan eksplorasi data, berikut adalah dataset yang telah diambil secara realtime menggunakan *Overpass API*, *OpenWeather API*, *Nominatim*, dan *TomTom*:

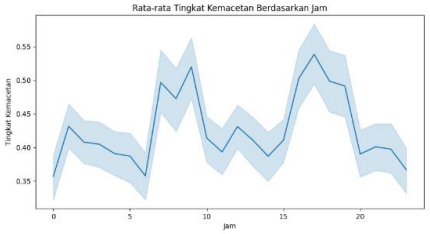
Tabel 1. Sample Dataset

road_name	latitude	longitude	highway_type	lane_count	temperature	weather_condition	current_speed	free_flow_speed	congestion_level	timestamp	hour
Jalan Pelangi	3,562748	98,692814	residential	2	29,6	Rain	23,2	29	0,2	23/03/2025 06.58	6
Gang Tentram	3,584919	98,624587	living_street	1	31,2	Rain	5	42	0,880952	23/03/2025 07.00	7
Jalan Listrik	3,58499	98,676612	secondary	2	27,4	Rain	6,596	26	0,7463	23/03/2025 07.01	7
Jalan Pertahanan	3,535404	98,71767	primary	3	31,9	Rain	25,79	35	0,2631	23/03/2025 07.02	7
Jalan Pelita	3,530036	98,719495	residential	2	29,7	Rain	5,4	27	0,8	23/03/2025 07.02	7
...	...	...	...	...	...	...	...	...	...	...	...

Dataset ini mencakup informasi lokasi, koordinat geografis, jenis jalan, jumlah lajur, suhu, kondisi cuaca, kecepatan kendaraan saat ini, kecepatan lalu lintas bebas, serta tingkat kemacetan pada berbagai waktu. Data ini kemudian dianalisis lebih lanjut untuk mengidentifikasi pola lalu lintas dan faktor-faktor yang berpengaruh terhadap kemacetan.

Eksplorasi Data Awal

Eksplorasi data awal dilakukan untuk memahami karakteristik dataset secara menyeluruh, bukan hanya untuk mengidentifikasi kekurangan yang perlu diperbaiki dalam tahap *preprocessing*, tetapi juga untuk menemukan pola yang dapat memberikan wawasan berharga. Dalam konteks analisis lalu lintas, eksplorasi data dapat mengungkapkan tren harian, pola jam sibuk, serta faktor-faktor eksternal yang berpengaruh terhadap kemacetan, seperti kondisi cuaca atau volume kendaraan. Misalnya, dari data yang dianalisis di Medan, dapat ditemukan bahwa kemacetan cenderung meningkat pada jam-jam tertentu, seperti pagi saat orang berangkat kerja dan sore saat pulang kerja. Selain itu, pola kemacetan mungkin juga dipengaruhi oleh faktor lain, seperti lokasi jalan, keberadaan persimpangan yang padat, atau bahkan kebijakan lalu lintas setempat. Dengan memahami pola ini, tidak hanya *preprocessing* data yang bisa dilakukan dengan lebih optimal, tetapi juga dapat menjadi dasar bagi kebijakan pengelolaan lalu lintas yang lebih efektif, seperti penyesuaian lampu lalu lintas atau penerapan sistem ganjil-genap pada jam-jam tertentu.



Gambar 2. Rata-rata tingkat kemacetan di medan

Preprocessing

Berdasarkan eksplorasi data, dilakukan *preprocessing* untuk meningkatkan kualitas dataset sebelum pelatihan model. *Missing values* diisi dengan median fitur numerik agar distribusi data tetap stabil. Informasi waktu dari *timestamp* diekstrak menjadi *hour* dan *day_of_week* untuk menangkap pola kemacetan berbasis waktu, lalu *timestamp* dihapus.

Fitur kategorikal seperti *weather_condition*, *road_name*, dan *highway_type* dikonversi menggunakan *one-hot encoding* agar dapat digunakan dalam model, sementara fitur numerik seperti *current_speed*, *free_flow_speed*, dan *temperature* dinormalisasi dengan *StandardScaler* agar memiliki skala yang seragam.

Untuk mengurangi dimensi data dan meningkatkan efisiensi model, seleksi fitur dilakukan menggunakan *SelectKBest* berbasis *f_regression*, memilih 50 fitur paling relevan terhadap tingkat kemacetan. Setelah itu, dataset

p-ISSN : 2406-7733
e-ISSN : 2597-9922

dibagi menjadi *train-test split* dengan rasio 80:20 untuk memastikan model dapat diuji dengan data yang belum terlihat.

Hasil *preprocessing* disimpan dalam folder *data/preprocessed/*, termasuk dataset yang telah diproses, *StandardScaler*, dan daftar fitur terpilih untuk memastikan konsistensi saat inferensi. Dengan langkah ini, dataset menjadi lebih bersih, terstruktur, dan siap digunakan dalam analisis kemacetan.

Pelatihan Model

Model dilatih menggunakan tiga algoritma utama: *Regresi Linear*, *Random Forest* (Tuned), dan *XGBoost* (Tuned). Sebelum pelatihan, dilakukan rekayasa fitur dengan menambahkan beberapa fitur tambahan seperti indikator hari libur, kemacetan sebelumnya, perubahan kecepatan sebelumnya, serta interaksi antara cuaca dan waktu.

Regresi Linear digunakan sebagai model dasar untuk memahami hubungan linear antar variabel. *Random Forest* kemudian dilatih dengan optimasi *hyperparameter* menggunakan *GridSearchCV*, menyesuaikan jumlah pohon keputusan, kedalaman maksimum, serta parameter lainnya agar mendapatkan hasil yang optimal. Selanjutnya, *XGBoost* diterapkan dengan teknik optimasi serupa, termasuk *tuning* jumlah *estimator*, *learning rate*, dan parameter lainnya untuk meningkatkan performa model.

Setelah model selesai dilatih, model tersebut disimpan dalam bentuk file dan digunakan kembali untuk melakukan prediksi pada data uji. Data uji juga mengalami proses rekayasa fitur yang sama seperti pada saat pelatihan untuk memastikan konsistensi data sebelum digunakan dalam prediksi. Hasil prediksi kemudian disimpan untuk dianalisis lebih lanjut dan dibandingkan antar model.

Evaluasi Model

Setelah pelatihan, hasil evaluasi dari ketiga model dibandingkan untuk menentukan metode yang paling sesuai dalam memprediksi tingkat kemacetan.

Tabel 2. Tabel Evaluasi Model dan Waktu *Training*

No	Model	MSE	RMSE	MAE	R^2 Score	Training Time
1	Linear Regression	0.0311	0.1764	0.1326	0.5771	0.04 seconds
2	Random Forest (Tuned)	0.0106	0.1028	0.0585	0.8563	140.29 seconds
3	XGBoost (Tuned)	0.0122	0.1105	0.0676	0.8341	367.15 seconds

Hasil evaluasi menunjukkan bahwa *Linear Regression* memiliki kinerja yang paling rendah dibandingkan dua model lainnya, dengan *MSE* sebesar 0.0311, *RMSE* 0.1764, dan *MAE* 0.1326, serta R^2 *Score* hanya 0.5771. Nilai R^2 yang rendah menunjukkan bahwa model ini kurang mampu menjelaskan variasi kepadatan lalu lintas berdasarkan fitur yang digunakan. Meski demikian, model ini memiliki keunggulan dari sisi efisiensi, dengan waktu pelatihan hanya 0.04 detik, menjadikannya pilihan jika kecepatan pemrosesan lebih diutamakan dibandingkan akurasi.

Random Forest (Tuned) menunjukkan peningkatan yang signifikan dalam akurasi prediksi, dengan *MSE* 0.0106, *RMSE* 0.1028, dan *MAE* 0.0585, serta R^2 *Score* mencapai 0.8563. Model ini mampu menangkap hubungan non-linear dengan lebih baik dibandingkan *Linear Regression*, yang terlihat dari penurunan nilai *error* dan peningkatan skor R^2. Namun, proses *training* model ini jauh lebih lama, yaitu 140.29 detik, yang disebabkan oleh pencarian parameter optimal melalui *GridSearchCV*.

XGBoost (Tuned) memiliki performa yang sedikit lebih rendah dibandingkan *Random Forest* dalam hal akurasi, dengan *MSE* 0.0122, *RMSE* 0.1105, *MAE* 0.0676, dan R^2 *Score* 0.8341. Meskipun masih jauh lebih baik dibandingkan *Linear Regression*, hasil ini menunjukkan bahwa dalam kasus ini, *Random Forest* mampu menangkap pola lalu lintas dengan lebih baik. Selain itu, *XGBoost* memerlukan waktu pelatihan yang jauh lebih lama, yaitu 367.15 detik, menjadikannya model dengan waktu *training* terpanjang di antara ketiganya.

Secara keseluruhan, *Random Forest* (Tuned) menjadi pilihan terbaik dalam penelitian ini, karena memberikan keseimbangan terbaik antara akurasi dan waktu pelatihan. *XGBoost*, meskipun cukup akurat, membutuhkan waktu pelatihan yang lebih lama tanpa peningkatan performa yang signifikan dibandingkan *Random Forest*. Sementara itu, *Linear Regression* tetap menjadi model yang sangat cepat, tetapi kurang akurat dalam menangkap kompleksitas lalu lintas.

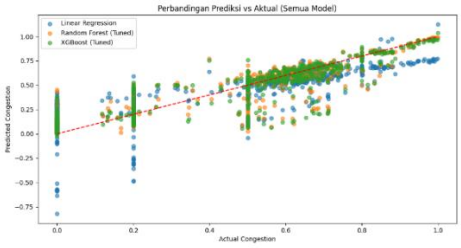
Penambahan fitur dalam eksperimen ini membantu meningkatkan kemampuan model dalam memahami pola lalu lintas yang kompleks. Namun, hasil evaluasi menunjukkan bahwa meskipun *Linear Regression* memiliki waktu pelatihan yang sangat cepat, model ini tidak mampu menangkap hubungan non-linear dengan baik. Di sisi lain, *Random Forest* dan *XGBoost* memberikan hasil prediksi yang lebih akurat setelah dilakukan *tuning parameter*, meskipun membutuhkan waktu

p-ISSN : 2406-7733
e-ISSN : 2597-9922

pelatihan yang lebih lama. Oleh karena itu, pendekatan yang digunakan dalam penelitian ini memastikan bahwa setiap model diberikan kesempatan yang sama untuk menunjukkan performanya dalam memprediksi kepadatan lalu lintas secara efisien.

Analisis dan Perbaikan Model

Hasil pengujian divisualisasikan dalam *scatter plot* Perbandingan Prediksi vs Aktual untuk ketiga model yang digunakan dalam penelitian ini, yaitu *Regresi Linear*, *Random Forest* (Tuned), dan *XGBoost* (Tuned). Sumbu X pada grafik merepresentasikan nilai kepadatan lalu lintas aktual berdasarkan data uji, sedangkan sumbu Y menunjukkan hasil prediksi dari masing-masing model. Garis merah putus-putus pada grafik menunjukkan batas ideal (y = x), yang mengindikasikan prediksi sempurna di mana setiap titik data seharusnya sejajar dengan garis tersebut.



Gambar 3. Visualisasi Hasil Prediksi Kemacetan dengan Ketiga Model

Dari hasil visualisasi, terlihat bahwa *XGBoost* dan *Random Forest* menghasilkan prediksi yang lebih akurat dibandingkan dengan *Regresi Linear*. Hal ini dapat diamati dari bagaimana titik-titik data yang dihasilkan oleh model *Random Forest* dan *XGBoost* lebih tersebar di sekitar garis diagonal dibandingkan dengan *Regresi Linear*, yang menunjukkan penyimpangan lebih besar terutama pada nilai kepadatan lalu lintas yang sangat rendah atau sangat tinggi. Penyimpangan ini menunjukkan bahwa model berbasis pohon keputusan mampu menangkap pola hubungan yang lebih kompleks dalam data dibandingkan dengan *Regresi Linear* yang lebih sederhana dan berbasis hubungan linier.

Pola penyebaran data pada *scatter plot* juga menunjukkan bahwa semua model masih memiliki tingkat *error* tertentu, terutama dalam memprediksi kepadatan lalu lintas pada kondisi ekstrem, seperti saat kepadatan sangat tinggi atau sangat rendah. Hal ini dapat disebabkan oleh beberapa faktor, di antaranya keterbatasan jumlah data latih, kompleksitas sistem lalu lintas yang sulit ditangkap oleh model, serta faktor eksternal yang tidak termasuk dalam dataset, seperti kejadian tidak terduga (misalnya kecelakaan atau perubahan kebijakan lalu lintas mendadak).

Untuk meningkatkan akurasi prediksi model, beberapa langkah dapat diterapkan. Salah satunya adalah dengan menambahkan fitur-fitur yang lebih relevan dan berkontribusi signifikan terhadap prediksi kepadatan lalu lintas, misalnya informasi lebih detail mengenai jenis kendaraan, kepadatan pada ruas jalan sekitar, atau faktor waktu seperti durasi lampu lalu lintas. Selain itu, teknik *feature selection* dapat diterapkan untuk menghilangkan fitur yang kurang berkontribusi sehingga model dapat berfokus pada variabel yang lebih bermakna.

Selain optimasi fitur, penambahan jumlah data latih juga dapat membantu model dalam menangkap pola yang lebih beragam dalam kepadatan lalu lintas. Jika dataset yang lebih besar tersedia, model dapat dilatih dengan variasi kondisi yang lebih luas, sehingga kemampuannya dalam menangani berbagai skenario lalu lintas dapat meningkat.

Selain itu, eksplorasi terhadap model lain seperti *LightGBM* atau pendekatan berbasis *deep learning* juga dapat dipertimbangkan, terutama jika akurasi prediksi masih perlu ditingkatkan lebih lanjut. Model berbasis *deep learning* seperti *Recurrent Neural Networks* (*RNN*) atau *Long Short-Term Memory* (*LSTM*) dapat menangani pola data yang bersifat *time-series* dengan lebih baik, sehingga mampu memahami hubungan temporal dalam pergerakan lalu lintas. Dengan langkah-langkah perbaikan ini, diharapkan model dapat semakin akurat dalam memprediksi kepadatan lalu lintas serta memberikan wawasan yang lebih baik dalam analisis data lalu lintas untuk keperluan perencanaan dan pengambilan keputusan.

V. KESIMPULAN DAN SARAN

Kesimpulan

Penelitian ini bertujuan untuk mengembangkan model prediksi kepadatan lalu lintas berdasarkan data lalu lintas historis serta faktor eksternal seperti cuaca dan hari libur. Dari hasil analisis dan evaluasi, diperoleh beberapa temuan sebagai berikut:

1. Model *Random Forest* menunjukkan performa terbaik dengan nilai RMSE terendah, sehingga lebih akurat dibandingkan *Regresi Linear* dan *XGBoost* dalam memprediksi kepadatan lalu lintas.
2. *XGBoost* memiliki keunggulan dalam kecepatan prediksi, tetapi sedikit kalah akurat dibandingkan *Random Forest*.
3. Model *Regresi Linear* kurang mampu menangkap pola kemacetan yang kompleks, sehingga menghasilkan tingkat kesalahan yang lebih tinggi dibandingkan model lainnya.
4. Hasil penelitian ini dapat dimanfaatkan dalam manajemen transportasi untuk mendukung pengambilan keputusan yang lebih baik terkait pengaturan lalu lintas dan perencanaan infrastruktur.

p-ISSN : 2406-7733
e-ISSN : 2597-9922

5. Masih terdapat ruang untuk meningkatkan akurasi model dengan memperkaya data atau menggunakan metode yang lebih canggih.

Saran

Berdasarkan temuan penelitian ini, beberapa pengembangan lebih lanjut yang dapat dilakukan meliputi:

1. Memperbesar dan memperkaya dataset dengan informasi tambahan, seperti jenis kendaraan, kepadatan pada ruas jalan sekitar, serta durasi lampu lalu lintas, guna meningkatkan akurasi prediksi.
2. Menerapkan teknik *feature selection* untuk mengidentifikasi dan menghapus fitur yang kurang berkontribusi, sehingga model dapat lebih fokus pada variabel yang signifikan.
3. Mengeksplorasi model *machine learning* lainnya, seperti *LightGBM* atau metode berbasis *deep learning* seperti *Recurrent Neural Networks* (*RNN*) dan *Long Short-Term Memory* (*LSTM*), yang lebih efektif dalam menangkap pola data *time-series*.
4. Mengembangkan sistem berbasis web atau mobile untuk menyajikan hasil prediksi secara real-time, sehingga dapat digunakan oleh masyarakat atau instansi terkait untuk perencanaan lalu lintas yang lebih baik.
5. Mengintegrasikan data dari sumber lain, seperti GPS kendaraan atau sensor *IoT*, untuk memberikan wawasan yang lebih kaya mengenai pola lalu lintas.

DAFTAR PUSTAKA

Abdurrafi, D. A., Alawiy, M. T., & Basuki, B. M. (2023). Deteksi klasifikasi dan menghitung kendaraan berbasis algoritma You Only Look Once (YOLO) menggunakan kamera CCTV. *Science Electro, 16*(3), 1–6.

Amaliah, S., Nusrang, M., & Aswi, A. (2022). Penerapan metode Random Forest untuk klasifikasi varian minuman kopi di Kedai Kopi Konijiwa Bantaeng. *VARIANSI: Journal of Statistics and Its Application on Teaching and Research, 4*(3), 121-127.

Chicco, D., Warrens, M. J., & Jurman, G. (2021). The coefficient of determination R-squared is more informative than SMAPE, MAE, MAPE, MSE, and RMSE in regression analysis evaluation. *PeerJ Computer Science, 7*, e623. https://doi.org/10.7717/peerj-cs.623

Efendi, I., & Hutabri, E. (2024). Perancangan Sistem Deteksi Plat Kendaraan Bermotor Menggunakan OpenCV Berbasis Web. *Computer and Science Industrial Engineering (COMASIE)*, *11*(4), 10-19. https://doi.org/10.33884/comasiejournal.v11i4.9127

Faradila, L. R., Fibriliyanti, Y., & Nasron. (2017). Deteksi kepadatan dan pembagian waktu pada simulasi lampu lalu lintas di persimpangan. In *Prosiding SNATIF ke-4* (pp. 335–339).

Fariza, A., Basofi, A., & Hidayat, M. R. (2020). Pencarian jalur berdasarkan kepadatan lalu lintas di Surabaya menggunakan algoritma koloni semut. *Journal of Applied Computer Science and Technology, 1*(2), 50–55. https://doi.org/10.52158/jacost.v1i2.10

Febrianto, M. F., Priyatno, A., Adisty, H., Saputri, A. F., Amanullah, R., Krissella, T. P., & Matondang, N. H. (2024). Prediksi situasi lalu lintas menggunakan machine learning dengan algoritma K-Nearest Neighbors Classifier. *Informatik: Jurnal Ilmu Komputer, 20*(1), 28–34. https://doi.org/10.52958/iftk.v20i1.7042

Huizen, R. R. (2024). Optimalisasi rekayasa lalu lintas melalui teknologi deteksi objek. *Jurnal Sistem dan Informatika (JSI), 18*(2), 111–117.

Kurniawan, F., Sajati, H., & Dinaryanto, O. (2017). Image processing technique for traffic density estimation. *International Journal of Engineering and Technology, 9*(2), 1496–1503. https://doi.org/10.21817/ijet/2017/v9i2/170902117

Kurniasari, A., & Jalinas. (2020). Pendeteksian tingkat kepadatan jalan menggunakan metode Canny Edge Detection. *Jurnal Ilmiah Teknologi dan Rekayasa, 25*(3), 239–248. https://doi.org/10.35760/tr.2020.v25i3.3419

Rudi, W. S., Pranoto, Y. A., & Ariwibisono, F. X. (2023). Penerapan metode regresi linier dalam peramalan penjualan kue di Toko Karya Bahari Samarinda berbasis website. *JATI (Jurnal Mahasiswa Teknik Informatika, 7*(4), 2451-2457. https://doi.org/10.36040/jati.v7i4.7547

Sakir, R. K. A. (2023). Pengujian Long-Short Term Memory (LSTM) pada prediksi trafik lalu lintas menggunakan multi server. *Jurnal Teknologi Elekterika, 20*(1), 14–19.

Sari, F. I., Gunawan, E. L., Adhigiadany, C. A., Lisanthoni, A., Data, S., & Timur, J. (2023). Model prediksi kepadatan lalu lintas: Perbandingan antara algoritma Random Forest dan XGBoost. In *Prosiding Seminar Nasional Sains Data* (Vol. 3, No. 1, pp. 296-303).

Yulianti, S. E. H., Soesanto, O., & Sukmawaty, Y. (2022). Penerapan metode Extreme Gradient Boosting (XGBoost) pada klasifikasi nasabah kartu kredit. *Journal of Mathematics: Theory and Applications, 4(1), 21-2.* *https://doi.org/10.31605/jomta.v4i1.1792*