

MAKALAH

RISTEK DATATHON



Theofilus Arifin

Jonathan Ryan D.

Achmad Nashruddin R.

DAFTAR ISI

DAFTAR ISI	i
PENDAHULUAN.....	1
<i>DATA PREPROCESSING</i>	2
1. Ekstraksi Data	2
2. Penanganan Data <i>Null</i> dan Duplikat	2
3. Pengambilan Data Melalui <i>Overpass</i> API	2
4. Transformasi Data Jalan	2
5. Transformasi Data Titik.....	2
6. Transformasi Fitur Lit	3
7. Transformasi Fitur <i>Max speed</i>	3
8. Transformasi Fitur <i>Highway</i>	3
9. Transformasi Fitur Id Jalan	3
10. Transformasi Fitur Titik Mulai dan Titik Akhir.....	3
<i>FEATURE ENGINEERING</i>	4
1. Pemisahan <i>Date</i> dan <i>Time</i>	4
2. Transformasi Data <i>Date</i> menjadi Hari	4
3. Transformasi Data <i>Time</i> menjadi Jam	4
4. Kalkulasi Jarak dengan Menggunakan <i>Latitude</i> dan <i>Longitude</i>	4
5. Penghapusan Fitur	5
6. Penghapusan Data yang Tidak Lengkap	5
7. Penghapusan <i>Outlier</i>	5
8. <i>Scaling</i>	5
EXPLORATORY DATA ANALYSIS	6
1. Correlation Map	6
2. Analisis <i>Max speed</i>	6
3. Analisis Hari.....	7
4. Analisis Jam	8
<i>MODELING</i>	9
1. Random Forest	9
EVALUASI	10
ANALISIS HASIL PREDIKSI	11
KESIMPULAN	12

PENDAHULUAN

Makalah ini menjelaskan hasil penelitian analisis data dan pembuatan *time series forecasting* mengenai kecepatan rata-rata kendaraan di suatu jalan. Data yang dianalisis dalam makalah ini adalah kecepatan rata-rata kendaraan pada setiap jam dan pada titik yang berbeda dalam periode 1 Februari 2020 hingga 29 Februari 2020. Konteks analisis ini sangat relevan dalam rangka pemahaman dan perbaikan kondisi lalu lintas di suatu wilayah. Dengan mengidentifikasi pola fluktuasi kecepatan kendaraan dan faktor-faktor yang mempengaruhinya, kita dapat mengambil langkah-langkah yang lebih efektif untuk mengurangi kemacetan dan memastikan kelancaran perjalanan.

Isi makalah ini mencakup lima tahap yaitu *Data Preprocessing*, *Feature Engineering*, *Exploratory Data Analysis* (EDA), *Modeling*, dan Evaluasi Model. Tahap *Data Preprocessing* membahas langkah-langkah penting dalam mempersiapkan data untuk analisis, termasuk ekstraksi data, penanganan data yang tidak lengkap, dan transformasi fitur-fitur penting. Tahap *Feature Engineering* meliputi Transformasi suatu data dari satu bentuk ke bentuk lainnya dan penghapusan fitur dan data yang tidak diperlukan. Selanjutnya EDA dilakukan untuk mendapatkan wawasan awal mengenai pola-pola dalam data, melalui analisis korelasi, distribusi fitur, serta pemahaman tentang pengaruh variabel-variabel tertentu terhadap kecepatan kendaraan. Selanjutnya, tahap *Modeling* membahas proses pembuatan model prediksi kecepatan kendaraan. Dalam makalah ini, kami akan menggunakan metode *Random Forest* untuk tahap *Modeling*. Terakhir adalah tahap Evaluasi Model yang merupakan langkah penting dalam mengukur performa model yang telah dibuat. Dengan membandingkan prediksi dengan nilai aktual, kita dapat menilai sejauh mana model mampu meramalkan kecepatan kendaraan dengan akurat.

Dengan pemahaman yang mendalam terhadap data dan model yang diaplikasikan, makalah ini bertujuan untuk memberikan kontribusi dalam peningkatan kualitas analisis lalu lintas dan pengambilan keputusan yang lebih efektif dalam mengatasi permasalahan kemacetan jalan. Dengan demikian, makalah ini diharapkan dapat menjadi rujukan penting dalam pengembangan studi lebih lanjut dan implementasi solusi yang berkelanjutan.

DATA PREPROCESSING

Pada penelitian ini, kami menjelaskan langkah-langkah *Data Preprocessing* yang dilakukan dalam rangka mempersiapkan data untuk analisis lebih lanjut. Data yang digunakan berasal dari *file* CSV dan meliputi berbagai fitur seperti waktu_setempat, id_jalan, id_titik_mulai, id_titik_akhir, dan rerata_kecepatan. Berikut proses *Data Preprocessing* pada penelitian ini.

1. Ekstraksi Data

Data awal diekstraksi dari *file* CSV yang mencakup berbagai fitur penting seperti waktu_setempat, id_jalan, id_titik_mulai, id_titik_akhir, dan rerata_kecepatan. Data *train* terdiri dari sebanyak 398.648 data.

	waktu_setempat o	id_jalan int64	id_titik_mulai int...	id_titik_akhir int64	rerata_kecepatan f
0	2020-02-01 01:0...	691007296	21390008	1425033102	29.126
1	2020-02-01 01:0...	47010584	1677092762	579493410	46.576
2	2020-02-01 01:0...	22932408	26486694	1930267566	36.587
3	2020-02-01 01:0...	142479648	1111592522	3775231113	34.063
4	2020-02-01 01:0...	8504977	5940503398	5940503394	38.336

Gambar 1 Data Awal

2. Penanganan Data Null dan Duplikat

Data kemudian diperiksa untuk memastikan tidak adanya data *null* atau duplikat. Setelah pengecekan dilakukan, ditemukan bahwa tidak ada data yang *null* atau duplikat.

<pre>print(train.isnull().sum())</pre> <pre>waktu_setempat 0 id_jalan 0 id_titik_mulai 0 id_titik_akhir 0 rerata_kecepatan 0 dtype: int64</pre>	<pre>duplicates = train.duplicated() print(len(train[duplicates]))</pre> <pre>0</pre>
--	---

Gambar 2 Hasil Penanganan Data Null dan Duplikat

3. Pengambilan Data Melalui Overpass API

Data dari *OpenStreetMap* (OSM) diambil menggunakan *Overpass* API. *Overpass* API adalah sebuah layanan yang digunakan untuk melakukan *query* terhadap data *OpenStreetMap* (OSM). Proses ini memungkinkan kami untuk mengambil data OSM dengan lebih spesifik dan mendalam.

4. Transformasi Data Jalan

Kami mengambil data jalan melalui *Overpass* API dengan menggunakan parameter *id_jalan* dalam *query* API. Fitur-fitur yang diambil dari data jalan adalah *highway*, *lit*, dan *maxspeed*. Hanya fitur-fitur ini yang diambil karena hanya tiga data ini yang tersedia untuk setiap *id_jalan*.

5. Transformasi Data Titik

Data titik juga diambil melalui *Overpass* API dengan menggunakan parameter *id_titik*

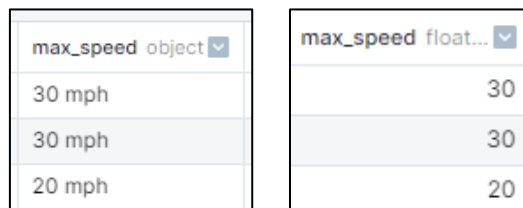
dalam *query* API. Data titik diambil dua kali untuk setiap data, mewakili titik awal dan titik akhir. Fitur yang diambil dari data titik adalah *latitude* dan *longitude*. Pada tahap ini, beberapa titik mungkin tidak memiliki data *latitude* dan *longitude*.

6. Transformasi Fitur Lit

Fitur lit yang mengindikasikan adanya lampu pencahayaan pada jalan diambil dari data jalan. Fitur ini bernilai boolean dengan nilai "yes" dan "no". Dalam data *train*, seluruh nilai fitur lit adalah "yes", sehingga fitur ini tidak akan berpengaruh pada pelatihan model dan akan dihapus.

7. Transformasi Fitur Max speed

Fitur *max speed*, yang mewakili kecepatan maksimum legal pada jalan, diambil dari data jalan. Nilai *max speed* awalnya dalam bentuk *string* dengan satuan mph. Kami menghilangkan *string* "mph" sehingga hanya angka yang tersisa.

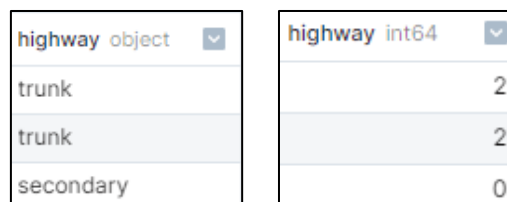


max_speed object	max_speed float...
30 mph	30
30 mph	30
20 mph	20

Gambar 3 Data Fitur Max speed Sebelum dan Sesudah Pemrosesan

8. Transformasi Fitur Highway

Fitur *highway*, yang mengkategorikan jenis-jenis jalan dalam peta, diambil dari data jalan. Nilai-nilai *highway* terdiri dari tiga jenis: *trunk*, *primary*, dan *secondary*. Kami melakukan *labeling* dengan mengubah nilai "secondary" menjadi 0, "primary" menjadi 1, dan "trunk" menjadi 2.



highway object	highway int64
trunk	2
trunk	2
secondary	0

Gambar 4 Data Fitur Highway Sebelum dan Sesudah Pemrosesan

9. Transformasi Fitur Id Jalan

Dari 398.648 data, terdapat hanya 20 id jalan yang berbeda. Kami melakukan *labeling* untuk mengubah nilai id jalan menjadi rentang antara 0 hingga 19, mengingat skala yang besar pada data id jalan yang semula berupa id acak.

10. Transformasi Fitur Titik Mulai dan Titik Akhir

Meskipun kami telah mengambil *latitude* dan *longitude* dari titik mulai dan titik akhir, kedua fitur ini tetap diperlukan untuk pelatihan model karena dua titik berbeda pada satu jalan yang sama dapat memiliki tingkat kemacetan yang berbeda. Dari 398.648 data, terdapat 488 id titik yang berbeda. Kami melakukan *labeling* untuk mengubah nilai id titik mulai dan titik akhir menjadi rentang antara 0 hingga 487.

FEATURE ENGINEERING

Dalam analisis data yang lebih mendalam, *Feature Engineering* adalah langkah penting untuk mengubah data mentah menjadi representasi yang lebih informatif dan sesuai dengan tujuan analisis. Dalam makalah ini, kami akan membahas berbagai teknik pengolahan fitur yang diterapkan pada data, dengan fokus pada pemisahan data waktu, transformasi data geografis, dan penanganan *outlier*.

1. Pemisahan *Date* dan *Time*

Fitur "waktu setempat" merupakan data *time series* yang mencakup informasi tanggal dan waktu. Dalam penelitian ini, kami akan memisahkan fitur waktu menjadi dua bagian, yaitu "*date*" yang berisi tahun, bulan, dan tanggal; serta "*time*" yang berisi jam, menit, dan detik. Hal ini bertujuan untuk memungkinkan analisis yang lebih terperinci terkait hubungan antara waktu dan fitur lainnya.

waktu_setempat object	date object	time object
2020-02-01 01:00:00+00:00	2020-02-01	01:00:00
2020-02-01 01:00:00+00:00	2020-02-01	01:00:00
2020-02-01 01:00:00+00:00	2020-02-01	01:00:00

Gambar 5 Fitur Waktu Setempat Dipisah menjadi Date dan Time

2. Transformasi Data *Date* menjadi Hari

Pengolahan lebih lanjut dilakukan dengan mengubah fitur *date* menjadi bentuk hari dalam seminggu. Proses ini menghasilkan hari dalam bentuk bilangan bulat, yang akan digunakan sebagai fitur dalam analisis. Tabel berikut menggambarkan *mapping* antara label dan hari

Label	Hari
0	Senin
1	Selasa
2	Rabu
3	Kamis
4	Jumat
5	Sabtu
6	Minggu

Tabel 1 Labeling Hari Pada Fitur Date

3. Transformasi Data *Time* menjadi Jam

Dalam data *time*, nilai menit dan detik pada setiap data selalu sama, yaitu 00. Oleh karena itu, data yang akan digunakan hanya informasi jam saja. Fitur *time* akan menggambarkan jam pada waktu tertentu dan disimpan dalam format bilangan bulat.

4. Kalkulasi Jarak dengan Menggunakan *Latitude* dan *Longitude*

Tahap selanjutnya adalah melakukan kalkulasi pada *latitude* dan *longitude* dari titik awal dan titik akhir untuk mengetahui jarak dari dua titik tersebut. Proses kalkulasi jarak

akan menggunakan persamaan *haversine*. Persamaan *haversine distance* digunakan untuk menghitung jarak antara dua titik koordinat (*latitude* dan *longitude*) pada permukaan bola. Pada penelitian ini satuan jarak yang digunakan adalah kilometer karena menyesuaikan dengan rerata kecepatan yang memiliki satuan kilometer per jam. *Haversine distance* akan dihitung menggunakan persamaan 1, 2, dan 3.

$$distance = R \cdot c \quad (1)$$

$$c = 2 \cdot a \cdot \tan 2(\sqrt{a}, \sqrt{1-a}) \quad (2)$$

$$a = \sin^2\left(\frac{\Delta lat}{2}\right) + \cos\left(\frac{\pi \cdot lat_1}{180}\right) \cdot \cos\left(\frac{\pi \cdot lat_2}{180}\right) \cdot \sin^2\left(\frac{\Delta lon}{2}\right) \quad (3)$$

Keterangan:

- lat_1 dan lat_2 adalah *latitude* dari titik awal dan titik akhir, dalam radian.
- Δlat adalah selisih *latitude* antara titik akhir dan titik awal, dalam radian.
- Δlon adalah selisih *longitude* antara titik akhir dan titik awal, dalam radian.
- R adalah radius bumi (6371 km)

5. Penghapusan Fitur

Dalam konteks analisis ini, fitur *latitude* dan *longitude* dari titik mulai dan titik akhir menjadi kurang relevan setelah perhitungan jarak dilakukan. Oleh karena itu, fitur-fitur ini akan dihapus dari data untuk menghindari redundansi dalam representasi data

6. Penghapusan Data yang Tidak Lengkap

Proses pengambilan data dari *Overpass API* dapat menghasilkan data yang tidak lengkap, terutama dalam hal data *latitude* dan *longitude*. Karena perhitungan jarak memerlukan kedua informasi ini, data yang tidak lengkap akan dihapus dari *dataset* untuk menjaga integritas analisis.

7. Penghapusan Outlier

Penghapusan *outlier* adalah langkah penting dalam pengolahan data. Dalam penelitian ini, kami akan menerapkan penghapusan *outlier* berdasarkan kelompok data jalan masing-masing. Setiap kelompok data jalan (dari 0 hingga 19) akan dilakukan *filtering* dan penghapusan *outlier* secara terpisah. Hal ini dilakukan untuk mencegah penghapusan data yang valid dan relevan dari kelompok lain. Data yang dianggap *outlier* pada tiap jalan adalah data yang memiliki sebuah nilai fitur yang memiliki selisih sebesar tiga kali dari standar deviasi fitur tersebut.

8. Scaling

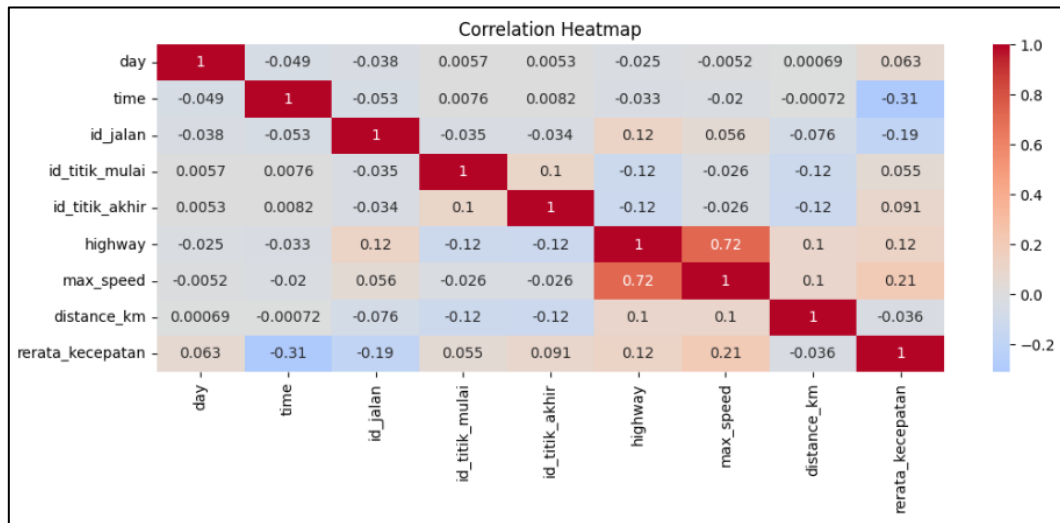
Scaling dilakukan pada semua fitur yang ada pada data. Metode *Scaling* yang digunakan adalah *Standart Scaler* dari library *sklearn*. *Scaling* dilakukan agar semua fitur memiliki nilai yang setara antara satu fitur dengan fitur yang lainnya

EXPLORATORY DATA ANALYSIS

Exploratory Data Analysis atau EDA adalah tahap analisis data yang bertujuan untuk memahami karakteristik dan pola data yang ada. Pada bagian ini, kami akan melakukan eksplorasi data yang relevan dengan fokus pada pemahaman korelasi antara fitur.

1. Correlation Map

Salah satu aspek penting dalam EDA adalah pemahaman tentang korelasi antara variabel-variabel yang ada dalam *dataset*. Korelasi map merupakan alat yang digunakan untuk memvisualisasikan hubungan korelasi antara pasangan variabel.

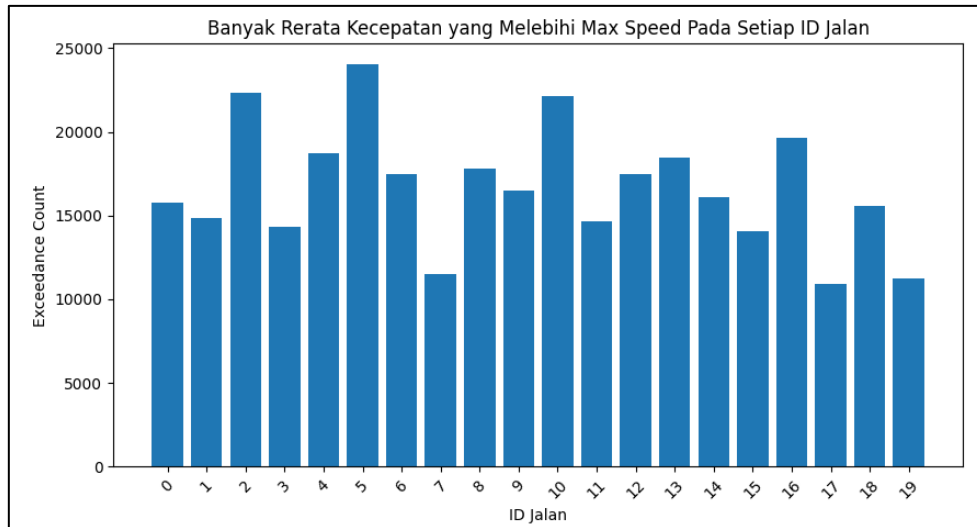


Gambar 6 Correlation Map Pada Dataset Setelah Pemrosesan

Dalam analisis ini, terlihat bahwa terdapat korelasi yang signifikan antara fitur waktu (*time*) dan rerata kecepatan. Hubungan tersebut bersifat negatif yang kuat, mengindikasikan bahwa semakin tinggi nilai variabel waktu, maka tingkat kemacetan yang terjadi akan semakin tinggi pula. Selain itu, terdapat juga hubungan korelasi yang tinggi antara *max speed* dan rerata kecepatan. Temuan ini menggambarkan bahwa setiap semakin tinggi *max speed* pada suatu jalan rata-rata kecepatan pada jalan tersebut juga akan semakin tinggi yang mengindikasikan kemacetan yang rendah.

2. Analisis Max speed

Pada analisis ini, kami membandingkan nilai *max speed* dengan rerata kecepatan. Hasil dari analisis ini memberikan wawasan bahwa sebanyak 87.19% dari data memiliki rerata kecepatan yang lebih tinggi daripada *max speed* yang ditetapkan. Fakta ini mengindikasikan bahwa ada ketidakpatuhan pengendara terhadap batas kecepatan yang diberlakukan di tiap jalan.

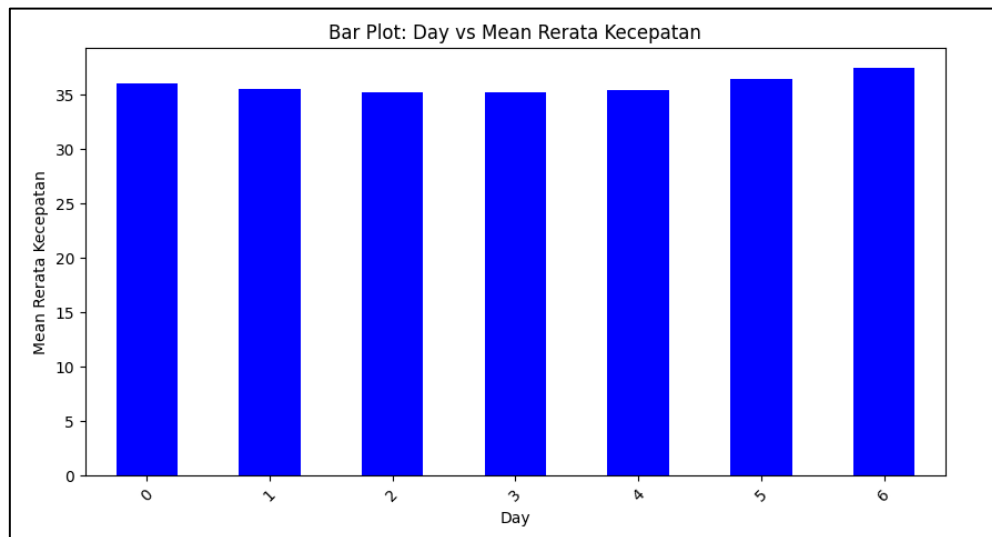


Gambar 7 Bar Plot Banyak Rerata Kecepatan yang melebihi Max speed Pada Tiap Jalan

Gambar yang dipresentasikan mengungkapkan bahwa pada setiap segmen jalan, terdapat pengendara yang melakukan pelanggaran dengan melebihi batas kecepatan yang telah ditetapkan secara legal. Analisis tersebut mengindikasikan bahwa pelanggaran terbanyak tercatat pada jalan yang diidentifikasi dengan label 5, sementara jalan yang ditandai dengan label 17 mencatat jumlah pelanggaran yang paling sedikit.

3. Analisis Hari

Pemahaman mengenai pola pergerakan lalu lintas dalam setiap hari dilakukan melalui analisis harian, di mana rata-rata kecepatan diukur pada tiap hari. Hari-hari dalam analisis direpresentasikan dengan nilai bilangan bulat sesuai dengan tabel 1. Visualisasi perbandingan hasil analisis ini diilustrasikan melalui bar plot berikut.



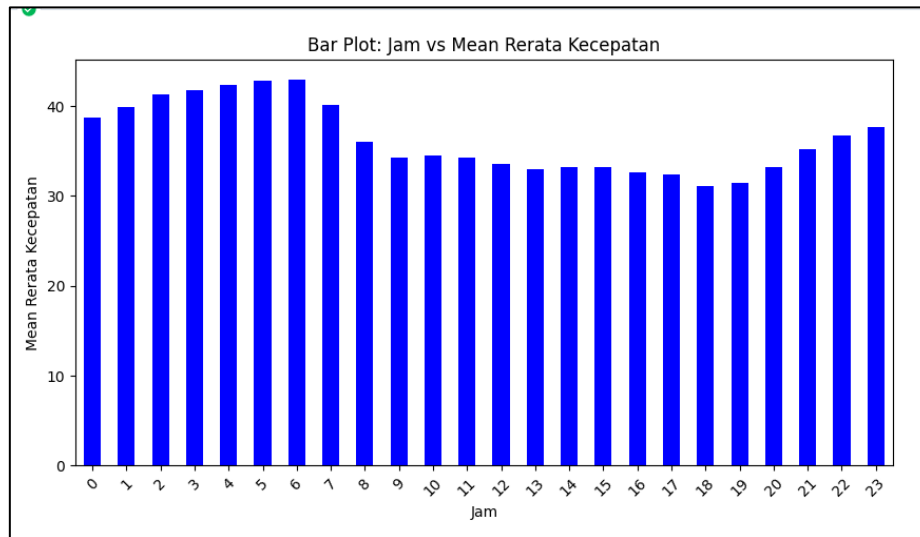
Gambar 8 Bar Plot Mean Rerata Kecepatan Per Hari

Pada gambaran bar plot yang disajikan, terlihat dengan jelas bahwa rata-rata kecepatan pada hari Sabtu dan Minggu cenderung lebih tinggi dibandingkan dengan hari-hari lainnya. Temuan ini mengindikasikan bahwa kondisi lalu lintas pada akhir pekan cenderung lebih lancar dibandingkan hari-hari kerja. Pada hari Senin, rata-rata kecepatan mulai menurun, yang mencerminkan transisi dari periode libur ke periode

aktivitas kerja. Sementara itu, pada hari kerja yaitu Selasa hingga Jumat, rata-rata kecepatan relatif konsisten dan lebih rendah dibandingkan dengan hari-hari lainnya. Dengan demikian, dapat diambil kesimpulan bahwa tingkat kemacetan cenderung lebih tinggi pada hari-hari kerja dibandingkan dengan akhir pekan.

4. Analisis Jam

Pendekatan analisis dilakukan melalui perbandingan rata-rata kecepatan pada setiap jam dalam satu hari. Visualisasi hasil analisis diwujudkan dalam bentuk diagram batang berikut.



Gambar 9 Bar Plot Mean Rerata Kecepatan Per Jam

Diagram batang yang disajikan menggambarkan pola fluktuasi rata-rata kecepatan sepanjang interval waktu dalam suatu hari. Penelitian ini mengungkapkan bahwa pada jam 0 hingga 6 pagi, tingkat kecepatan rata-rata cenderung tidak menunjukkan kemacetan yang signifikan. Namun, saat mencapai jam 7 pagi, tampak adanya peningkatan tajam dalam tingkat kemacetan yang berkelanjutan hingga jam 6 malam. Selanjutnya, kurva kemacetan menunjukkan penurunan pada interval waktu jam 7 malam hingga 11 malam. Temuan ini menggambarkan bahwa pada jam 0 hingga 6 pagi, kemacetan cenderung rendah karena mayoritas masyarakat sedang istirahat dan aktivitasnya minimal. Peningkatan kemacetan pada jam 7 pagi mengindikasikan dimulainya aktivitas harian seperti perjalanan menuju tempat kerja, yang berdampak pada lonjakan lalu lintas. Pada siang hari, kemacetan mencapai puncaknya karena banyaknya individu yang melakukan perjalanan dalam berbagai aktivitas. Jam 7 malam menandai penurunan kemacetan yang mengindikasikan banyaknya individu yang sudah kembali pulang ke rumah atau menghentikan aktivitas berkendara. Peningkatan aktivitas kembali terjadi pada pagi berikutnya, membentuk pola fluktuasi yang berulang. Temuan ini menggambarkan bahwa tingkat kemacetan mengalami fluktuasi berdasarkan interval waktu tertentu. Peningkatan kemacetan pada awal pagi dan menjelang malam berkaitan dengan aktivitas perjalanan masyarakat, seperti pergi dan pulang dari tempat kerja. Sementara itu, penurunan kemacetan selama malam hari mencerminkan sedikitnya aktivitas berkendara masyarakat pada jam tersebut.

MODELING

Tahap *Modeling* merupakan tahap model yang akan digunakan untuk memprediksi nilai rerata kecepatan dibentuk. Setelah model sudah dibentuk, model akan dilatih menggunakan data *training*. Pada kasus ini algoritma yang digunakan adalah *Random Forest*.

1. Random Forest

Random Forest merupakan sebuah algoritma *machine learning* yang cukup fleksibel dan dapat digunakan pada berbagai data. Algoritma ini bekerja dengan menggabungkan hasil beberapa decision tree untuk menghasilkan sebuah nilai output. *Random Forest* dapat digunakan baik pada kasus klasifikasi maupun regresi. Pada penelitian ini, *Random Forest* digunakan dengan *hyperparameter* tuning sebagai berikut ini.

<i>Hyperparameter</i>	<i>Value</i>
<i>n_estimators</i>	100
<i>random_state</i>	42

Tabel 2 Hyperparameter Tuning Pada Model Random Forest

Model ini akan dilatih dengan menggunakan data *training* hasil *preprocessing* dan *Feature Engineering*. Hasil dari *training* model *Random Forest* pada penelitian ini dapat dilihat pada tabel di bawah ini.

Waktu <i>Training</i>	Akurasi
2 menit	89.52

Tabel 3 Hasil Dari Proses Training Model Random Forest

Dapat dilihat bahwa model *Random Forest* ini memiliki waktu *training* yang sangat singkat yaitu 2 menit namun memiliki nilai akurasi yang cukup tinggi yaitu 89.52%. Akurasi ini adalah akurasi seberapa baik model dalam melakukan regresi terhadap data *training*. Hasil akhir dari model *Random Forest* ini akan membantu dalam menganalisis dan memahami faktor-faktor yang mempengaruhi rata-rata kecepatan kendaraan di berbagai jalan berdasarkan data geografis dan fitur-fitur lainnya.

EVALUASI

Tahap evaluasi merupakan proses penting untuk mengukur sejauh mana kemampuan model dalam melakukan prediksi. Pada tahap ini, berbagai metrik digunakan untuk menganalisis performa model dalam konteks tugas regresi. Dalam kasus ini, kita akan membahas dua metrik evaluasi, yaitu *Mean Squared Error* (MSE), *Mean Absolute Error* (MAE), dan *Symmetric Mean Absolute Percentage Error* (sMAPE).

1. *Mean Squared Error* (MSE)

Metrik ini mengukur rata-rata dari kuadrat selisih antara nilai prediksi model dengan nilai aktual. Secara matematis, MSE dihitung dengan menjumlahkan kuadrat dari selisih antara setiap prediksi dan nilai aktual, kemudian diambil rata-ratanya. Nilai MSE yang lebih rendah menunjukkan bahwa model memiliki tingkat kesalahan yang lebih kecil dalam meramalkan nilai aktual.

2. *Mean Absolute Error* (MAE)

Metrik ini mengukur rata-rata dari selisih absolut antara nilai prediksi model dengan nilai aktual. MAE lebih tahan terhadap nilai ekstrem karena menggunakan nilai absolut, sehingga memberikan gambaran keseluruhan tentang seberapa besar kesalahan prediksi. Seperti MSE, nilai MAE yang lebih rendah menunjukkan performa model yang lebih baik dalam memprediksi data.

3. *Symmetric Mean Absolute Percentage Error* (sMAPE)

Metrik ini mengukur besarnya kesalahan prediksi dalam bentuk persentase dari nilai aktual, namun dengan pendekatan simetris, yang berarti baik *overestimation* (prediksi lebih tinggi dari aktual) maupun *underestimation* (prediksi lebih rendah dari aktual) dianggap memiliki dampak yang setara terhadap nilai kesalahan..

Hasil yang dihasilkan dalam melakukan evaluasi terhadap model dengan melakukan prediksi terhadap data *training* adalah sebagai berikut.

MSE = 6.223467012118513

MAE = 1.7503251177325638

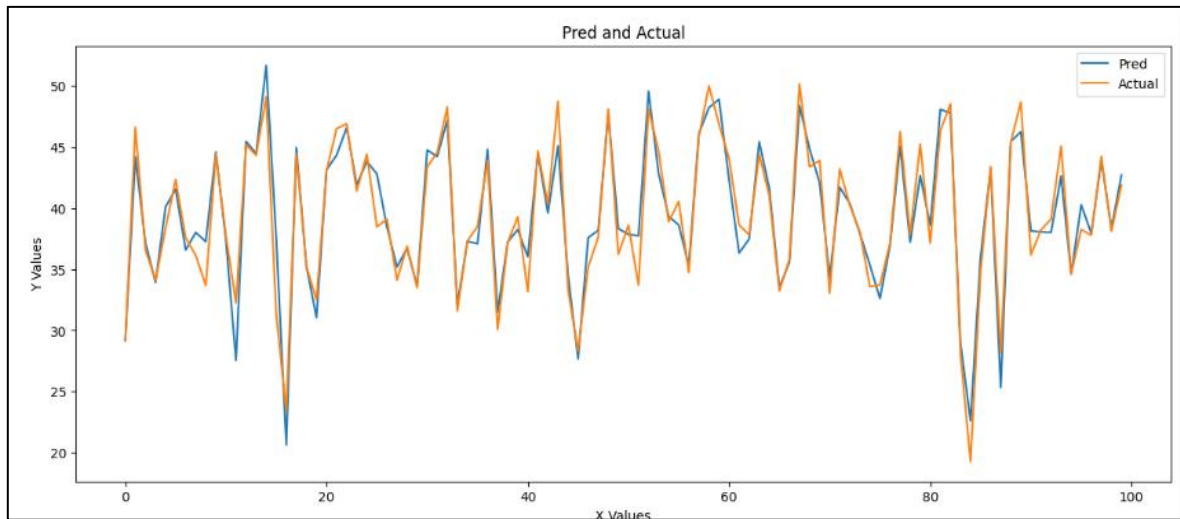
Hasil yang dihasilkan dalam melakukan evaluasi terhadap model dengan melakukan prediksi terhadap 30% data testing adalah sebagai berikut.

sMAPE = 8.56306

Ketiga nilai MSE, MAE, dan sMAPE pada model menunjukkan ukuran yang relatif kecil, hal ini mengindikasikan kemampuan model dalam prediksi yang baik.

ANALISIS HASIL PREDIKSI

Proses analisis hasil prediksi dilaksanakan melalui langkah prediksi menggunakan model yang telah melalui tahap pelatihan dan data *training* untuk perbandingan antara prediksi dan nilai aktual. Rangkaian perbandingan ini dapat direpresentasikan secara visual melalui metode visualisasi plot pada pustaka Matplotlib. Berikutnya, kami menyajikan hasil perbandingan antara prediksi dan nilai aktual, khususnya untuk 100 data pertama.



Gambar 10 Perbandingan Prediction dan Actual pada Data Training

Dari gambar di atas, terlihat bahwa prediksi model ditunjukkan melalui kurva grafik berwarna biru, sementara nilai aktual direpresentasikan oleh kurva grafik berwarna oranye. Perbandingan visual antara prediksi dan nilai aktual tergambar dalam plot tersebut, dan tampak bahwa terdapat kesamaan pola antara keduanya dengan beberapa data yang cocok. Namun, perlu diperhatikan bahwa sejumlah data juga menunjukkan perbedaan dan deviasi dari kesamaan pola tersebut.

Meskipun terdapat variasi dalam beberapa data yang tidak sesuai dengan pola umum, perbedaan (*offset*) dari prediksi terhadap nilai aktual tidak memiliki jarak yang signifikan. Hal ini menunjukkan bahwa model yang digunakan memiliki tingkat performa yang layak dalam melakukan prediksi data. Meskipun beberapa data menunjukkan perbedaan, *offset* dari nilai prediksi terhadap nilai aktual tetap berada dalam batas yang masuk akal. Penjelasan ini menggarisbawahi bahwa model memiliki kemampuan yang memadai dalam melakukan prediksi data dengan tetap mempertahankan akurasi yang relatif tinggi, meskipun beberapa deviasi terjadi.

KESIMPULAN

Penelitian ini menghasilkan pemahaman mendalam mengenai analisis data dan prediksi *time series* terhadap kecepatan rata-rata kendaraan dalam suatu konteks lalu lintas urban. Penggunaan data berbasis jam dan lokasi pada periode yang relevan menghasilkan representasi temporal yang kaya dan signifikan. Melalui proses *Data Preprocessing*, *Feature Engineering*, *Exploratory Data Analysis* (EDA), *Modeling*, dan Evaluasi Model, kami berhasil mengembangkan model regresi yang kuat dengan menerapkan metode *Random Forest* yang dioptimalisasi dengan *hyperparameter tuning*.

Dengan akurasi mencapai 89.52%, model ini memiliki kapabilitas meramalkan kecepatan kendaraan dengan tingkat akurasi yang memadai. Penelitian ini mengungkapkan pola korelasi yang signifikan antara waktu, geolokasi, dan faktor eksternal seperti batas kecepatan maksimum dengan tingkat kemacetan. Analisis hasil prediksi memberikan gambaran menyeluruh tentang pola fluktuasi lalu lintas sepanjang waktu dan memberikan wawasan mengenai hubungan antara faktor-faktor tertentu dengan perubahan kecepatan kendaraan.

Dalam konteks pengembangan infrastruktur dan perencanaan lalu lintas, penelitian ini memiliki implikasi penting dalam menyusun solusi berkelanjutan guna mengurangi kemacetan dan meningkatkan kelancaran perjalanan di lingkungan perkotaan. Diharapkan temuan ini dapat memicu studi lebih mendalam serta mendorong implementasi strategi intervensi yang lebih efektif dalam upaya mengoptimalkan mobilitas dan mengatasi tantangan lalu lintas di masa mendatang.