

## Prediksi Situasi Lalu Lintas Menggunakan Machine Learning Dengan Algoritma K-Nearest Neighbors Classifier

Muhammad Fatur Febrianto<sup>1</sup>, Abimanyu Priyatno<sup>2</sup>, Hawna Adisty<sup>3</sup>, Anisa Fadilah Saputri<sup>4</sup>, Rizki Amanullah<sup>5</sup>, Thesa Pebrianti Krissella<sup>6</sup>, Nindy Irzavika<sup>7</sup>, Nur Hafifah Matondang<sup>8</sup>  
<sup>1,2,3,4,5,6,7,8</sup>Sistem Informasi, Fakultas Ilmu Komputer, Universitas Pembangunan Nasional Veteran Jakarta  
 Jl. RS. Fatmawati Raya, Pondok Labu, Kec. Cilandak, Kota Depok, Jawa Barat 12450  
<sup>1</sup>2110512002@mahasiswa.upnvj.ac.id, <sup>2</sup>2110512005@mahasiswa.upnvj.ac.id,  
<sup>3</sup>2110512008@mahasiswa.upnvj.ac.id, <sup>4</sup>2110512010@mahasiswa.upnvj.ac.id,  
<sup>5</sup>2110512025@mahasiswa.upnvj.ac.id, <sup>6</sup>2110512031@mahasiswa.upnvj.ac.id, <sup>7</sup>nindyrzavika@upnvj.ac.id,  
<sup>8</sup>nurhafifahmatondang@upnvj.ac.id

**Abstrak.** Setiap tahunnya kepadatan arus lalu lintas pada daerah kota-kota besar terus meningkat. Situasi lalu lintas yang semakin padat telah menjadi tantangan serius bagi pengelolaan transportasi di berbagai kota di seluruh dunia. Penelitian ini memberikan solusi untuk mengatasi kepadatan lalu lintas di kota-kota besar dengan menerapkan metode Machine Learning, khususnya Algoritma K-Nearest Neighbors (K-NN). Berfokus pada peningkatan kepemilikan kendaraan bermotor sebagai penyebab utama kepadatan lalu lintas, penelitian ini mengeksplorasi prediksi situasi lalu lintas guna meningkatkan efisiensi transportasi. Metodologi penelitian melibatkan enam tahap, termasuk pengumpulan data, pembagian dataset (80% data latih dan 20% data uji), dan pengujian K-NN Classifier dengan akurasi terbaik mencapai 85% melalui validasi silang. Kesimpulan menunjukkan bahwa K-NN Classifier dapat memprediksi situasi lalu lintas dengan tingkat akurasi tinggi, terutama pada data latih dengan nilai 1.0. Penerapan metode ini dapat meningkatkan pengelolaan lalu lintas perkotaan dan mendukung otoritas transportasi serta pemerintah kota dalam mengimplementasikan tindakan yang lebih tepat waktu. Meskipun demikian, penelitian ini memiliki keterbatasan, dan saran untuk penelitian selanjutnya mencakup eksplorasi aspek tertentu serta pengembangan metode, tool, dan pelatihan model untuk perbandingan dengan K-NN.

**Kata Kunci:** Machine Learning, K-Nearest Neighbors(K-NN), Validasi Silang, Prediksi Lalu Lintas.

### 1 Pendahuluan

Setiap tahunnya kepadatan arus lalu lintas pada daerah kota-kota besar terus meningkat, bertambahnya kepemilikan kendaraan bermotor baik roda dua maupun roda empat menjadi faktor membuat padatnya arus lalu lintas karena tidak ada batasan untuk memiliki kendaraan pribadi [1]. Situasi lalu lintas yang semakin padat telah menjadi tantangan serius bagi pengelolaan transportasi di berbagai kota di seluruh dunia [2]. Tingkat kemacetan yang tinggi tidak hanya dapat memengaruhi lama waktu perjalanan, akan tetapi dapat berdampak negatif pada lingkungan serta kesejahteraan masyarakat atau penduduknya [3]. Padatnya lalu lintas dapat meningkatkan resiko kecelakaan. Bertambahnya penduduk mendorong mobilitas pendukung meningkat yang berakibat bertambahnya volume lalu lintas dan jumlah kendaraan [4].

Dengan demikian, tentunya diperlukan pendekatan yang efektif dalam memprediksi dan mengelola situasi lalu lintas untuk mengetahui tingkat kemacetan sehingga dapat meningkatkan efisiensi transportasi. Untuk menyelesaikan permasalahan kemacetan perlu dilakukan prediksi kondisi lalu lintas [5]. Salah satu pendekatan yang dapat digunakan adalah penerapan metode Machine Learning, yang dapat memanfaatkan data historis lalu lintas untuk memprediksi tingkat situasi lintas. Menurut penelitian terdahulu dengan judul “Metode K-Nearest Neighbor Untuk Peramalan Kepadatan Arus Lalu Lintas” yang menunjukkan nilai eror yang cukup rendah dari penerapan K-NN sehingga kinerja algoritma K-NN mampu dalam memprediksi kondisi lalu lintas [6]. Pada penelitian lainnya, disebutkan bahwa penerapan K-NN memiliki potensi yang besar untuk memantau kondisi kemacetan lalu lintas dengan tingkat akurasi 96% [7], dan 89%[8], 90%[9]. K-NN memiliki keunggulan kompetitif dibandingkan pendekatan lain dalam memprediksi keadaan lalu lintas [10].

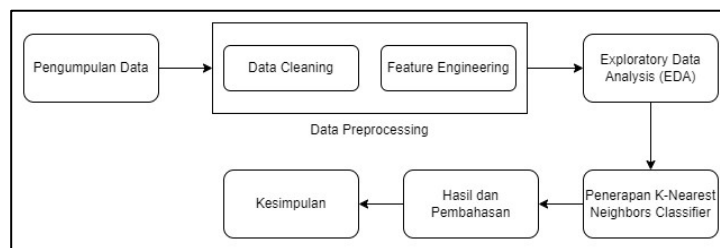
Pada penelitian dilakukan penerapan algoritma K-Nearest Neighbors Classifier untuk prediksi kondisi lalu lintas sebagai upaya untuk menghadirkan solusi yang inovatif dalam penanganan masalah yang terjadi dalam berlalu-lintas seperti kepadatan kendaraan hingga tingkat kemacetan lalu lintas. Berdasarkan permasalahan yang ditemukan perlu dibentuknya sistem yang dapat memprediksi kondisi situasi lalu lintas seperti tingkat kemacetan dengan akurasi tinggi, memungkinkan pihak terkait untuk mengambil tindakan preventif atau

korektif secara tepat waktu dengan penggunaan K-NN terhadap dataset atau kumpulan data mengenai kondisi lalu lintas. Kumpulan data ini berguna dalam perencanaan transportasi, manajemen kemacetan, dan analisis arus lalu lintas. Kumpulan data ini memungkinkan intervensi yang ditargetkan seperti optimalisasi sinyal dan penyesuaian jalur. Secara keseluruhan, kumpulan data ini memberdayakan pemangku kepentingan untuk membuat keputusan berdasarkan data, meningkatkan mobilitas perkotaan, dan menciptakan kota yang efisien dan berkelanjutan.

Dalam penelitian ini, algoritma K-Nearest Neighbors (K-NN) dipilih sebagai metode utama karena kemampuannya dalam mengklasifikasikan data berdasarkan kemiripan dengan data latih yang ada menunjukkan nilai akurasi yang lebih tinggi dibandingkan algoritma lainnya. Penerapan algoritma K-NN pada dataset kemacetan lalu lintas, diharapkan dapat membantu memahami permintaan kendaraan, mengidentifikasi area kemacetan, dan menginformasikan perbaikan infrastruktur. Hal ini memungkinkan peneliti mempelajari pola lalu lintas berdasarkan jam, hari, atau tanggal tertentu dan mengeksplorasi korelasinya dengan faktor eksternal. Ini mendukung penelitian transportasi tentang hubungan kendaraan dan perilaku lalu lintas yang diharapkan dapat membantu perencana kota dalam menilai dampak lalu lintas terhadap keputusan zonasi dan infrastruktur terkait dalam perencanaan dan pengelolaan lalu lintas perkotaan sehingga dapat diambil tindakan yang sesuai untuk mengelola lalu lintas dengan lebih efisien.

## 2 Metodologi Penelitian

Tahapan pada metodologi penelitian ini dilakukan dalam 6 tahapan yang dimana setiap tahapan dari pengumpulan data sampai dengan kesimpulan dari hasil pengujian metode K-Nearest Neighbors Classifier. Pada tahap metode penelitian yang akan dilakukan adalah dengan pengumpulan data, Data Preprocessing, Exploratory Data Analysis, seperti pada *Gambar 1*.



**Gambar 1.** Alur Penelitian, gambar ini menunjukkan alur penelitian mulai dari pengumpulan data hingga kesimpulan

### 2.1 Pengumpulan Data

Pengumpulan data adalah langkah awal yang dilakukan dengan memperoleh informasi atau data mentah yang digunakan untuk melakukan analisis. Tahap ini melibatkan dari berbagai sumber baik itu sumber internal maupun sumber eksternal untuk membentuk dasar analisa lebih lanjut. Data lalu lintas yang digunakan pada penelitian ini adalah data sekunder, yaitu data yang telah ada berasal dari pihak ketiga. Pada tahap pengumpulan data dilakukan studi literatur untuk mendukung penelitian yang dilakukan.

### 2.2 Data Cleaning

Data *cleaning* atau pembersihan data adalah proses membersihkan dan mempersiapkan dataset agar sesuai untuk analisis atau penggunaan dalam model *machine learning*. Tujuan utama dari data *cleaning* adalah untuk mengatasi masalah-masalah yang mungkin ada dalam dataset, seperti *missing values*, *outliers*, atau inkonsistensi dalam format data. Proses ini sangat penting karena kualitas data yang baik adalah kunci keberhasilan analisis dan pembuatan model.

### 2.3 Feature Engineering

Feature Engineering adalah proses membuat atau memodifikasi variabel (fitur) dalam dataset untuk meningkatkan kinerja model *machine learning*. Tujuan utama dari feature engineering adalah untuk menyajikan informasi yang lebih relevan atau lebih bermakna kepada model, sehingga model dapat belajar pola yang lebih baik dan menghasilkan prediksi yang lebih kuat.

## 2.4. Exploratory Data Analysis

*Exploratory Data Analysis* (EDA) adalah suatu proses uji investigasi awal yang bertujuan untuk mengidentifikasi pola, menemukan anomali, menguji hipotesis dan memeriksa asumsi. EDA dilakukan untuk dapat mendeteksi kesalahan dari awal, dapat mengidentifikasi outlier, mengetahui hubungan antar data serta dapat menggali faktor-faktor penting dari data.

Berdasarkan hasil EDA yang dilakukan, diperoleh kesimpulan bahwa jumlah kendaraan mempunyai kontribusi paling besar terhadap kemacetan lalu lintas dan hari paling sibuk adalah hari Selasa, Rabu, dan Kamis. Puncak kemacetan berada antara pukul 08.00 – 10.00 dan 15.00 – 18.00, waktu tersebut merupakan waktu berangkat kerja dan waktu pulang kerja.

## 2.5. Penerapan K-Nearest Neighbors Classifier

Dalam algoritma K-Nearest Neighbors (KNN), perhitungan dapat dilakukan jika data bersifat numerik. Oleh karena itu, diperlukan pengolahan data kategorikal menjadi data numerik dengan memberikan bobot pada sub kategori. Setelah itu, dilakukan normalisasi data menggunakan rumus normalisasi Min-Max. yang ditunjukkan pada Persamaan (1). Rumus tersebut adalah sebagai berikut:

$$\text{Normalized Value} = \frac{(x - \min(X))}{(\max(X) - \min(X))} \quad (1)$$

Di mana  $x$  merupakan nilai sebelum dinormalisasi,  $\max(X)$  adalah nilai maksimum dari parameter,  $\min(X)$  adalah nilai minimum dari parameter, dan hasilnya adalah nilai yang telah dinormalisasi. Setelah data dinormalisasi, perhitungan dengan menggunakan algoritma KNN dapat dilakukan. K-Nearest Neighbors (KNN) merupakan salah satu algoritma klasifikasi dalam data mining yang digunakan untuk mencari pola baru dalam data. Algoritma ini menghubungkan pola data sebelumnya dengan data yang baru untuk mengklasifikasikan data ke dalam beberapa kelas berdasarkan atribut yang ada. KNN bekerja dengan menghitung jarak antara data untuk mengklasifikasikan berdasarkan data yang memiliki kemiripan yang paling besar. Perhitungan jarak pada KNN menggunakan rumus Euclidean Distance, yaitu:

$$\text{Euclidean Distance} = \sqrt{\sum_{i=1}^n (x_i - y_i)^2} \quad (2)$$

Dengan  $X_i$  adalah nilai pada data training,  $Y_i$  adalah nilai pada data uji, dan  $k$  adalah batas banyaknya data. Langkah-langkah dalam algoritma KNN meliputi menentukan nilai  $k$ , menghitung jarak Euclidean distance, melakukan pengurutan data, dan melakukan klasifikasi data berdasarkan kelas terbanyak dengan nilai  $k$  yang telah ditentukan. Untuk mengevaluasi nilai akurasi dari algoritma KNN, digunakan rumus Confusion Matrix, yang dapat dijabarkan dengan Persamaan (3).

$$\text{Accuracy} = \frac{\text{True Positive} + \text{True Negative}}{\text{Total Data}} \quad (3)$$

True Positive adalah jumlah data yang benar-benar terklasifikasi dengan benar sebagai positif, True Negative adalah jumlah data yang benar-benar terklasifikasi dengan benar sebagai negatif, dan Total Data adalah jumlah total data yang dievaluasi. Dengan langkah-langkah tersebut, implementasi K-Nearest Neighbors dapat memberikan prediksi yang akurat dan berguna dalam konteks prediksi tingkat kemacetan atau aplikasi lainnya.

## 3. Hasil dan Pembahasan

### 3.1 Implementasi

#### 3.1.1 Pengumpulan Data

Pada penelitian ini, dataset yang digunakan berasal dari repositori “Traffic Prediction Dataset” milik Hasibullah Aman di website Kaggle. Dataset ini memiliki 9 kolom yang berisi hingga 2976 baris data. Setiap kolom merupakan indikator situasi lalu lintas secara umum seperti Time, Date, Day of the week, CarCount,

BikeCount, BusCount, TruckCount, Total. Detail dataset “Traffic Prediction Dataset” ini dapat dilihat pada Gambar X. Adapun dataset pada penelitian ini dapat dilihat pada **Gambar 2** dan **Gambar 3**.

	A	B	C	D	E	F	G	H	I
1	Time	Date	Day of the week	CarCount	BikeCount	BusCount	TruckCount	Total	Traffic Situation
2	12:00:00 AM	10	Tuesday	31	0	4	4	39	low
3	12:15:00 AM	10	Tuesday	49	0	3	3	55	low
4	12:30:00 AM	10	Tuesday	46	0	3	6	55	low
5	12:45:00 AM	10	Tuesday	51	0	2	5	58	low
6	1:00:00 AM	10	Tuesday	57	6	15	16	94	normal
7	1:15:00 AM	10	Tuesday	44	0	5	4	53	low
8	1:30:00 AM	10	Tuesday	37	0	1	4	42	low
9	1:45:00 AM	10	Tuesday	42	4	4	5	55	low
10	2:00:00 AM	10	Tuesday	51	0	9	7	67	low

**Gambar 2.** 10 data teratas dari dataset “Traffic Prediction Dataset”

2968	9:30:00 PM	9	Thursday	106	18	13	27	164	high
2969	9:45:00 PM	9	Thursday	107	14	10	18	149	normal
2970	10:00:00 PM	9	Thursday	17	4	1	20	42	normal
2971	10:15:00 PM	9	Thursday	11	4	1	30	46	normal
2972	10:30:00 PM	9	Thursday	11	4	0	23	38	normal
2973	10:45:00 PM	9	Thursday	16	3	1	36	56	normal
2974	11:00:00 PM	9	Thursday	11	0	1	30	42	normal
2975	11:15:00 PM	9	Thursday	15	4	1	25	45	normal
2976	11:30:00 PM	9	Thursday	16	5	0	27	48	normal
2977	11:45:00 PM	9	Thursday	14	3	1	15	33	normal

**Gambar 3.** 10 data terbawah dari dataset “Traffic Prediction Dataset”

Bagian dari artikel yang berisi laporan hasil pengambilan data, hasil analisis data, dan interpretasi hasil. Pada bagian ini juga dapat dijelaskan bagaimana hubungan antara hasil penelitian dan ulasan teoretis dan studi relevan sebelumnya.

### 3.1.2 Data Cleaning

Pada penelitian ini menggunakan Google Colab berbasis web untuk memudahkan dalam melakukan proses penelitian prediksi situasi lalu lintas.

```
def handle_outliers(df, x):
    q1 = df[x].quantile(0.25)
    q3 = df[x].quantile(0.75)
    IQR = q3 - q1
    upper_limit = q3 + (1.5 * IQR)
    lower_limit = q1 - (1.5 * IQR)

    df_filtered = df[(df[x] >= lower_limit) & (df[x] <= upper_limit)]

    return df_filtered

df = handle_outliers(df, 'CarCount')
df = handle_outliers(df, 'BikeCount')
df = handle_outliers(df, 'BusCount')
df = handle_outliers(df, 'TruckCount')
df = handle_outliers(df, 'Total')
```

**Gambar 4.** Proses pembersihan data

Pada **Gambar 4** dilakukannya proses pembersihan data yang dimana untuk mengatasi permasalahan pada dataset yaitu adanya *outliers*. Proses data cleaning dalam penelitian ini juga berguna untuk menghasilkan kualitas prediksi data yang baik. Fungsi *handle\_outliers* digunakan untuk memastikan bahwa data yang akan dianalisis konsisten, akurat dan tidak ada data pencilon (*outliers*), sehingga model prediksi menjadi lebih akurat dan optimal.

### 3.1.3 Feature Engineering

Pada penelitian ini diterapkan feature engineering dengan menerapkan teknik ekstraksi fitur dengan mengonversi data waktu dalam format string menjadi representasi waktu dalam bentuk numerik yaitu float untuk memberikan info yang lebih informatif dari dataset yang telah diolah. Dari proses feature engineering, informasi "Time" akan ditampilkan menjadi lebih ringkas dan jelas dalam tipe data float sehingga hanya menampilkan format jam dan menit menggunakan fungsi `convert_time_to_float` seperti pada **Gambar 5**.

```
def convert_time_to_float(time_str):
    dt = datetime.strptime(time_str, '%I:%M:%S %p')
    hours = dt.hour
    minutes = dt.minute

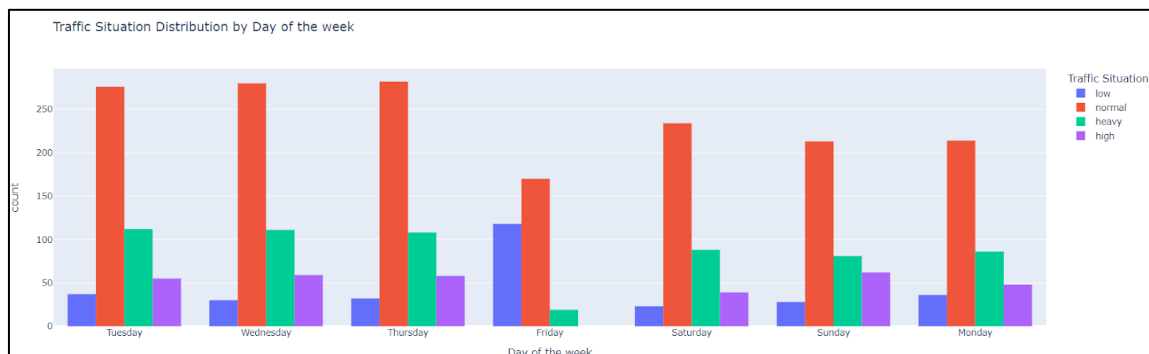
    time_float = float(f"{hours}.{minutes}")
    return time_float

df['Time'] = df['Time'].apply(convert_time_to_float)
```

**Gambar 5.** Menerapkan teknik ekstraksi fitur

### 3.1.4 Exploratory Data Analysis

Pada **Gambar 6** merupakan Traffic Situation yang di mana hasilnya menunjukkan bahwa dominan terhadap situasi normal. Pada dataset tersebut tingkat situasi “normal” dan “heavy” paling tinggi ialah pada hari Kamis (Thursday) dengan jumlah berbagai jenis kendaraan sebanyak 282, diikuti dengan hari Rabu, dan Selasa.



**Gambar 6.** Traffic situation

### 3.1.5 Penerapan K-Nearest Neighbors Classifier

Pada penerapan K-NN classifier pada penelitian ini maka perlu dilakukan beberapa tahapan seperti feature selection dimana pada penelitian ini menghilangkan kolom 'Traffic Situation' untuk dijadikan sebagai variabel dependen dalam pemodelan. Dan selanjutnya membagi dataset menjadi 80% data latih dan 20% data uji. **Gambar 7** menunjukkan proses pembagian dataset dan penghapusan kolom “Traffic Situation”.

```
X = df.drop(columns='Traffic Situation')
y = df['Traffic Situation']

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

**Gambar 7.** Pembagian dataset

Selanjutnya dilakukan pencarian grid hyperparameter terbaik untuk melatih model menggunakan data latih untuk menampilkan nilai akurasi pada data uji setelah melatih model untuk memberikan perkiraan kinerja model pada data yang belum pernah ditemukan. Proses pencarian grid hyperparameter dapat dilihat pada **Gambar 8**.

```

preprocessor = ColumnTransformer([
    ('numeric', num_pipe(scaling='minmax'), ['Time', 'Date', 'CarCount', 'BikeCount', 'BusCount', 'TruckCount', 'Total']),
    ('categorical', cat_pipe(encoder='onehot'), ['Day of the week'])
])

pipeline = Pipeline([
    ('prep', preprocessor),
    ('algo', KNeighborsClassifier())
])

model = GridSearchCV(pipeline, gsp.knn_params, cv=3, n_jobs=-1, verbose=1)
model.fit(X_train, y_train)

print(model.best_params_)
print(model.score(X_train, y_train), model.best_score_, model.score(X_test, y_test))

```

Gambar 8. Pencarian grid hyperparameter

## 3.2 Pengujian

### 3.2.1 Model Accuracy

Pada penelitian ini, dataset yang digunakan berasal dari repositori “Traffic Prediction Dataset” milik Hasibullah Aman di website Kaggle. Dataset ini memiliki 9 kolom yang berisi hingga 2976 baris data. Setiap kolom merupakan indikator situasi lalu lintas secara umum seperti Time, Date, Day of the week, CarCount, BikeCount, BusCount, TruckCount, Total. Detail dataset “Traffic Prediction Dataset” ini dapat dilihat pada Gambar X. Adapun dataset pada penelitian ini dapat dilihat pada **Gambar 9**.

```

Fitting 3 folds for each of 90 candidates, totalling 270 fits
{'algo__n_neighbors': 7, 'algo__p': 1, 'algo__weights': 'distance'}
1.0 0.8503665373005607 0.8896551724137931

```

Gambar 9. Score dari accuracy model

Pengujian yang telah dilakukan didapatkan Metode K-Nearest Neighbour mendapat score accuracy 0.8896 (89%) setelah melatih model. Ini memberikan perkiraan kinerja model pada data yang belum pernah dilihat sebelumnya, selain itu model juga berhasil memprediksi dengan sempurna pada data latih.

## 4. Kesimpulan dan Saran

### 4.1 Kesimpulan

Dari penelitian yang sudah dilakukan dapat disimpulkan bahwa proses klasifikasi situasi kondisi lalu lintas dengan metode yang diterapkan yaitu K-Nearest Neighbours didapatkan metode ini memiliki nilai accuracy yang didapat yaitu 0.8896551724137931 (89%) yang didapatkan setelah melakukan pengolahan data seperti pengumpulan data yang menghasilkan pembagian dataset menjadi 80% data latih dan 20% data uji melalui proses feature selection dan pencarian grid hypermeter.

Hasil pengujian akurasi dengan menggunakan tools google collab dengan metode K-Nearest Neighbours menghasilkan nilai akurasi terbaik yang ditemukan selama pencarian grid menggunakan validasi silang sebesar 0.8503665373005607 (85%) dan akurasi pada data latih setelah pelatihan model memiliki nilai 1.0, yang berarti model berhasil memprediksi dengan sempurna pada data latih, sehingga metode K-Nearest Neighbours sudah mendekati nilai akurasi sempurna.

Dengan hasil pengujian tersebut, tentunya dapat membantu instansi terkait dalam melakukan prediksi kondisi situasi lalu lintas seperti tingkat kemacetan dengan akurasi tinggi dan memungkinkan instansi tersebut untuk mengambil tindakan preventif atau korektif secara tepat.

### 4.2 Saran

Dari penelitian yang sudah dilakukan maka peneliti memberikan usulan maupun saran untuk penelitian selanjutnya dapat memanfaatkan metode yang berbeda, tool yang berbeda serta pelatihan model yang sekiranya dapat dijadikan sebagai pembandingan dengan metode K-Nearest neighbour sehingga pemahaman akan situasi lalu lintas lebih rinci dan akurat.



## Referensi

- [1] M. F. Abdi, S. Y. Qodarbaskoro, A. Alfani, K. Kusri, and D. Maulina, 'KLASIFIKASI PEMBAGIAN ARUS LALU LINTAS MENGGUNAKAN ALOGARITMA NAÏVE BAYES DAN MODEL LINEAR', *Technologia: Jurnal Ilmiah*, vol. 12, no. 4, p. 203, Oct. 2021, doi: 10.31602/tji.v12i4.5626.
- [2] M. Ashifuddin Mondal and Z. Rehena, 'Intelligent Traffic Congestion Classification System using Artificial Neural Network', in *Companion Proceedings of The 2019 World Wide Web Conference*, New York, NY, USA: ACM, May 2019, pp. 110–116. doi: 10.1145/3308560.3317053.
- [3] M. Ilham, 'SUBSISTEM KONTROL LAMPU LALU LINTAS: KLASIFIKASI BESAR ARUS KENDARAAN BERMOTOR MENGGUNAKAN ALGORITMA C4. 5 DAN PREDIKSI TAMBAHAN DURASI LAMPU HIJAU', Doctoral Dissertation, Universitas Gadjah Mada, Yogyakarta, 2021.
- [4] D. W. Sharfina and A. Saputra, 'Analisis Tingkat Kemacetan Lalu Lintas di Ruas Jalan Arteri dan Kolektor Kota Magelang Menggunakan Sistem Informasi Geografis', Universitas Muhammadiyah Surakarta, 2022.
- [5] N. Irzavika and G. A. P. Saptawati, 'Analysis Performance of Support Vector Regression (SVR) for Congestion Prediction', in *2022 International Conference on Informatics, Multimedia, Cyber and Information System (ICIMCIS)*, IEEE, Nov. 2022, pp. 1–5. doi: 10.1109/ICIMCIS56303.2022.10017641.
- [6] M. Sho'imah, R. Wijanarko, and N. E. Budiyanto, 'METODE K-NEAREST NEIGHBOR UNTUK PERAMALAN KEPADATAN ARUS LALU LINTAS DI GERBANG TOL MANYARAN SEMARANG', *JURNAL ILMIAH MOMENTUM*, vol. 14, no. 2, Nov. 2018, doi: 10.36499/jim.v14i2.2517.
- [7] K. E. N. T. Rabe, E. R. Arboleda, and R. M. Dellosa, 'Fuzzy logic based vehicular congestion estimation monitoring system using image processing and kNN classifier', *Int. J. Sci. Tehnol. Res*, vol. 8, no. 08, pp. 1377–1380, 2019.
- [8] S. Rahayu, B. R. Asmoro, and E. Rinal, 'Classification of Congestion in Jakarta Using KNN, Naïve Bayes and Decision Tree Method', *Jurnal Syntax Admiration*, vol. 4, no. 7, pp. 928–952, Jul. 2023, doi: 10.46799/jsa.v4i7.654.
- [9] L. Zhang, Q. Liu, W. Yang, N. Wei, and D. Dong, 'An Improved K-nearest Neighbor Model for Short-term Traffic Flow Prediction', *Procedia Soc Behav Sci*, vol. 96, pp. 653–662, Nov. 2013, doi: 10.1016/j.sbspro.2013.08.076.
- [10] H. Yu, N. Ji, Y. Ren, and C. Yang, 'A Special Event-Based K-Nearest Neighbor Model for Short-Term Traffic State Prediction', *IEEE Access*, vol. 7, pp. 81717–81729, 2019, doi: 10.1109/ACCESS.2019.2923663.