

PENGEMBANGAN MODEL LSTM PREDIKSI KEMACETAN JALAN SUKABUMI BERBASIS ANALISIS KECEPATAN RATA-RATA KENDARAAN

Bagas Firmansyah, Ivana Lucia Kharisma

Teknik Informatika, Universitas Nusa Putra

Jl. Raya Cibolang, Sukabumi No.21, Jawa Barat, Indonesia

Bagas.firmansyah_ti23@nusaputra.ac.id

ABSTRAK

Kemacetan lalu lintas di perkotaan seperti Sukabumi berdampak signifikan pada efisiensi mobilitas masyarakat. Model prediksi berbasis deep learning saat ini seringkali memerlukan komputasi berat dan dataset heterogen yang kompleks. Penelitian ini bertujuan mengembangkan model prediksi kemacetan yang ringan dan akurat menggunakan arsitektur *Sequence-to-Sequence* (Seq2Seq) *Long Short-Term Memory* (LSTM). Metode yang digunakan memanfaatkan *Floating Car Data* (FCD) kecepatan kendaraan selama 30 hari dengan interval 15 menit pada 50 ruas jalan. Data mentah direkayasa menjadi 41 fitur spasial-temporal, dinormalisasi menggunakan teknik *Hybrid Scaling* (*Z-Score* dan *Min-Max*), serta distrukturkan melalui metode *sliding window* dengan *lookback* 12 langkah untuk memprediksi 4 langkah ke depan. Hasil evaluasi menunjukkan model mencapai *Validation Loss* (MSE) 0,0576. Pengujian pada data uji menghasilkan *Mean Absolute Error* (MAE) keseluruhan 1,6905 km/jam dan *Root Mean Squared Error* (RMSE) 2,4692 km/jam. Hasil prediksi kecepatan ini berhasil diklasifikasikan ke dalam 4 kategori *Speed Ratio* (*Free Flow*, *Moderate*, *Congested*, *Severe Congestion*) untuk visualisasi dasbor kemacetan secara real-time.

Kata kunci : LSTM, Prediksi Kemacetan, Kecepatan Kendaraan, Floating Car Data, Deep Learning

1. PENDAHULUAN

Sektor transportasi jalan raya memegang peranan krusial dalam menopang mobilitas perkotaan, di mana kelancaran arus lalu lintas menjadi indikator utama efisiensi pergerakan sosial-ekonomi[1].

Di wilayah perkotaan seperti Sukabumi, fluktuasi kepadatan lalu lintas sering kali memicu kemacetan yang sulit diantisipasi. Kendala utama dalam pemantauan kondisi jalan raya saat ini adalah tingginya biaya instalasi dan pemeliharaan sensor fisik statis seperti kamera CCTV atau detektor loop[2].

Sebagai alternatif yang lebih efisien dan ekonomis, pemanfaatan data kecepatan rata-rata kendaraan yang diperoleh melalui perangkat pelacak seluler atau *Floating Car Data* (FCD) dipandang sebagai solusi optimal untuk mendeteksi tingkat kepadatan jalan tanpa ketergantungan pada infrastruktur fisik yang mahal[2].

Meskipun ketersediaan data FCD sangat menguntungkan, memprediksi dinamika lalu lintas secara akurat memiliki tantangan tersendiri karena karakteristik aliran kendaraan yang non-linear dan stokastik. Metode peramalan statistik tradisional sering gagal memodelkan kompleksitas ini, sehingga arah penelitian beralih menggunakan pendekatan deep learning[3]. Akan tetapi, model deep learning terkini, seperti integrasi *Convolutional Neural Network* (CNN) dengan *Long Short-Term Memory* (LSTM) atau jaringan berbasis graf *spasial-temporal* umumnya memiliki arsitektur yang sangat kompleks dan menuntut beban komputasi tinggi[4].

Model-model tersebut juga sangat bergantung pada ketersediaan dataset heterogen berdimensi tinggi yang mencakup faktor eksternal lintas-domain seperti kondisi cuaca, klasifikasi tipe kendaraan, maupun

catatan hari libur[5]. Ketergantungan ini memunculkan kesenjangan riset (*research gap*), di mana pemodelan tersebut sulit diimplementasikan pada lingkungan yang memiliki keterbatasan fitur, volume data, maupun spesifikasi perangkat keras. Untuk mengatasi permasalahan tersebut, penelitian ini merancang penyelesaian dengan mengembangkan model prediksi kemacetan lalu lintas yang tangkas menggunakan arsitektur *Sequence-to-Sequence* (Seq2Seq) LSTM. Algoritma LSTM terbukti sangat andal dalam mengekstraksi representasi fitur dan menangkap pola jangka panjang pada data deret waktu (*time-series*) yang kompleks, baik dalam domain transportasi maupun pada fluktuasi data non-linear lainnya seperti peramalan harga mata uang kripto[6]. Secara spesifik, model ini dirancang untuk bekerja secara optimal hanya menggunakan data kecepatan rata-rata kendaraan dengan durasi observasi 30 hari pada interval 15 menit. Untuk menutupi keterbatasan dimensi data (*univariate*) tersebut, dilakukan rekayasa fitur secara mendalam menjadi 41 fitur masukan mencakup fitur temporal, lag, statistik bergulir, dan ketetanggaan spasial sehingga model tetap mampu mengekstraksi dependensi spasial-temporal secara komprehensif. Tujuan utama dari penelitian ini adalah mengonstruksi model machine learning yang mampu memprediksi kecepatan kendaraan secara presisi di bawah keterbatasan ketersediaan data, serta mengevaluasi efektivitas penggunaan dataset historis 30 hari dalam memetakan siklus kemacetan lalu lintas jangka pendek. Lebih lanjut, penelitian ini bertujuan mengklasifikasikan hasil prediksi kecepatan tersebut ke dalam status kepadatan jalan secara terukur[2].

Hasil klasifikasi ini selanjutnya diintegrasikan ke dalam purwarupa perangkat lunak visualisasi potensi

kemacetan secara real-time, yang diharapkan dapat menjadi instrumen rujukan praktis untuk membantu pemilihan rute perjalanan masyarakat di Sukabumi.

2. TINJAUAN PUSTAKA

2.1. Penelitian Terdahulu

Prediksi arus lalu lintas telah banyak dieksplorasi menggunakan berbagai pendekatan. Kashyap dkk. [3] serta Valova dkk. [7] dalam ulasannya menyimpulkan bahwa model *deep learning*, khususnya *Long Short-Term Memory* (LSTM), secara signifikan mengungguli metode statistik tradisional karena kemampuannya dalam menangkap dependensi temporal dan non-linearitas pada data lalu lintas. Keandalan arsitektur LSTM pada pemrosesan data deret waktu (*time-series*) juga dikonfirmasi oleh Nguyen Thi [8] dan divalidasi lebih jauh keandalannya pada data berfluktuasi ekstrem di luar domain transportasi oleh Akbar dkk. [6].

Dalam perkembangannya, penelitian prediksi kemacetan mulai mengarah pada model hibrida. Ren dkk.[4] mengusulkan model kombinasi CNN dan LSTM dengan mekanisme *attention*, sementara Wang dan Susanto[5] mengintegrasikan data heterogen lintas-domain (seperti cuaca, jenis kendaraan, dan hari libur) ke dalam model hibrida CNN-LSTM. Meskipun pendekatan tersebut menghasilkan akurasi yang tinggi, arsitektur yang digunakan sangat kompleks, membutuhkan sumber daya komputasi yang berat, serta sangat bergantung pada ketersediaan data eksternal yang sering kali sulit dipenuhi di wilayah berkembang[4],[5].

Di sisi lain, Ahanin dkk.[2] membuktikan bahwa penggunaan *Floating Car Data* (FCD) sangat efektif dan ekonomis sebagai alternatif pengganti sensor fisik statis. Mereka menggunakan FCD untuk memprediksi kecepatan dan mengidentifikasi level kemacetan jalan dengan tingkat presisi yang menjanjikan, sehingga menekan biaya infrastruktur pemantauan[2].

Berdasarkan tinjauan tersebut, penelitian ini mengisi kesenjangan riset (*research gap*) dengan mengusulkan model yang ringan secara komputasi namun tetap memiliki akurasi tinggi. Berbeda dengan penelitian sebelumnya[4],[5] yang bergantung pada arsitektur hibrida kompleks dan data multivariat lintas-domain, penelitian ini beroperasi secara optimal hanya dengan mengandalkan variabel tunggal (kecepatan rata-rata kendaraan berbasis FCD) berdurasi 30 hari. Kebaruan (*novelty*) dari riset ini terletak pada rekayasa fitur spasial-temporal yang mendalam mengubah 1 data mentah menjadi 41 fitur masukan pada arsitektur *Seq2Seq LSTM*, sehingga model ini mampu menghasilkan performa setara model kompleks dengan efisiensi komputasi yang jauh lebih relevan untuk diterapkan di wilayah Sukabumi.

2.2. Analisis Kemacetan Lalu Lintas

Kemacetan lalu lintas bermanifestasi sebagai penumpukan kendaraan yang melampaui kapasitas tampung ideal suatu ruas jalan, yang berdampak secara

destruktif terhadap waktu perjalanan dan konsumsi energi harian pengguna jalan[1].

Peningkatan volume kendaraan pribadi yang tidak diimbangi oleh penambahan rasio jalan raya menuntut adanya pemodelan matematis untuk memperkirakan pergeseran pola arus secara dini[1].

Dalam teknik transportasi perkotaan, tingkat kepadatan lalu lintas dapat diklasifikasikan secara kuantitatif berdasarkan parameter kecepatan rata-rata kendaraan yang melintas[9].

Kecepatan rata-rata kendaraan merupakan representasi paling akurat dari indeks pelayanan jalan karena secara langsung menggambarkan hambatan fisik dan kelambatan yang dialami pengendara di lapangan[2].

Klasifikasi status kemacetan umumnya dibagi menjadi beberapa rentang nilai numerik tertentu yang memisahkan kondisi jalan lancar, padat, hingga macet[2].

2.2. Penggunaan *Floating Car Data*(FCD)

Penyediaan data lalu lintas secara kontinu memerlukan metode akuisisi data yang efisien dan ekonomis. Penggunaan teknologi telekomunikasi seluler melahirkan konsep *Floating Car Data* (FCD) yang mengumpulkan parameter posisi dan kecepatan secara *real-time* dari kendaraan yang sedang bergerak di jaringan jalan raya[2].

Dibandingkan dengan sensor fisik statis seperti detektor induktif loop atau kamera CCTV pemantau yang membutuhkan investasi infrastruktur tinggi, FCD menawarkan cakupan area pengawasan yang jauh lebih luas dengan biaya operasional minimal[2].

Namun, data mentah yang diperoleh dari FCD sering kali menunjukkan karakteristik spasial-temporal yang renggang (*sparse*) sehingga membutuhkan tahapan pra-pemrosesan yang ketat[10].

Guna mempertahankan integritas data deret waktu (*time-series*), data kecepatan FCD diekstraksi ke dalam slot waktu diskret (seperti interval 15 menit) dan dienkapsulasi menggunakan metode *sliding window* guna mempertahankan informasi urutan sekuensial sebelum dilakukan proses pelatihan model prediksi.

2.3. Jaringan Saraf Tiruan *Long Short-Term Memory*(LSTM)

Prediksi jangka pendek pada data deret waktu yang bersifat non-linear memerlukan arsitektur jaringan saraf tiruan yang adaptif. Model pembelajaran mendalam (*deep learning*) telah mendominasi berbagai literatur karena kapasitasnya dalam mengekstraksi representasi fitur secara otomatis dari data berskala besar[3][7].

Sebagai bagian dari *Recurrent Neural Network* (RNN), model LSTM dikembangkan secara khusus untuk memitigasi masalah hilangnya gradien (*vanishing gradient*) pada pemrosesan urutan data yang panjang[11].

Kemampuan ini diperoleh melalui struktur sel memori khusus yang dilengkapi dengan gerbang kontrol (gate) internal untuk mengatur retensi informasi jangka panjang[11].

Unit komputasi dasar LSTM mengatur aliran informasi melalui tiga gerbang utama, yaitu gerbang lupa (*forget gate*), gerbang masukan (*input gate*), dan gerbang keluaran (*output gate*). Berdasarkan urutan prosesnya, informasi pertama kali disaring melalui *forget gate* yang berfungsi untuk melupakan informasi yang tidak relevan pada vektor input x_t dan output jangka pendek pada langkah sebelumnya h_{t-1} melalui proses aktivasi sigmoid σ [6]. Jika ditulis ke dalam persamaan matematis, maka gerbang lupa dirumuskan sebagai berikut:

$$f_t = \sigma(w_f \cdot [h_{t-1}, x_t] + b_f) \quad (1)$$

Dalam formula ini, f_t merupakan *forget gate* (gerbang lupa), σ merupakan fungsi aktivasi sigmoid, w_f merupakan bobot pada *forget gate*, $[h_{t-1}, x_t]$ merupakan penggabungan vektor status tersembunyi sebelumnya dengan vektor input baru, dan b_f merupakan nilai bias dari *forget gate*.

Komponen selanjutnya yang berfungsi untuk memperbarui informasi baru dari hasil proses langkah waktu sebelumnya adalah *cell state* (sel memori). Pada sel ini, keputusan untuk mengingat atau melupakan informasi jangka panjang diputuskan melalui hasil gerbang lupa[6]. Sehingga, persamaan matematis dari status sel kandidat dirumuskan sebagai berikut:

$$c_t = \tanh(w_c \cdot [h_{t-1}, x_t] + b_c) \quad (2)$$

Dalam formula ini, c_t merupakan sel memori kandidat, $\tanh$ merupakan fungsi aktivasi *tangen hiperbolik*, merupakan bobot sel memori kandidat, $[h_{t-1}, x_t]$ merupakan penggabungan vektor status tersembunyi sebelumnya dengan vektor input baru, dan b_c merupakan bias pada sel memori kandidat.

Langkah selanjutnya yaitu *input gate* (gerbang masukan). Fungsi utama gerbang ini adalah memilih dan mengontrol seberapa banyak data baru yang akan dimasukkan ke dalam sel memori[6]. Persamaan matematis untuk input gate adalah sebagai berikut:

$$i_g = \sigma(w_i \cdot [h_{t-1}, x_t] + b_i) \quad (3)$$

Dalam formula ini, i_g merupakan gerbang input, σ merupakan fungsi aktivasi *sigmoid*, w_i merupakan bobot pada gerbang input, $[h_{t-1}, x_t]$ merupakan penggabungan vektor status tersembunyi sebelumnya dengan vektor input baru, dan b_i merupakan bias pada gerbang input.

Kemudian, hasil keluaran dari gerbang input dan gerbang lupa dikombinasikan untuk memperbarui status sel memori lama c_{t-1} menjadi status sel memori baru c_t . Hubungan pembaruan sel memori ini dirumuskan sebagai berikut:

$$c_t = f_t \cdot c_{t-1} + i_g \cdot c_t \quad (4)$$

Dalam formula ini, c_t merupakan status sel memori baru yang telah diperbarui, c_{t-1} merupakan status sel memori pada langkah waktu sebelumnya, f_t merupakan aktivasi gerbang lupa, i_g merupakan

aktivasi gerbang input, dan c_t merupakan nilai sel memori kandidat yang baru.

Langkah berikutnya adalah menentukan gerbang keluaran (*output gate*) yang berfungsi untuk mengontrol seberapa banyak informasi dari sel memori yang akan dikeluarkan menjadi status tersembunyi berikutnya. Persamaan matematis untuk gerbang keluaran didefinisikan sebagai berikut:

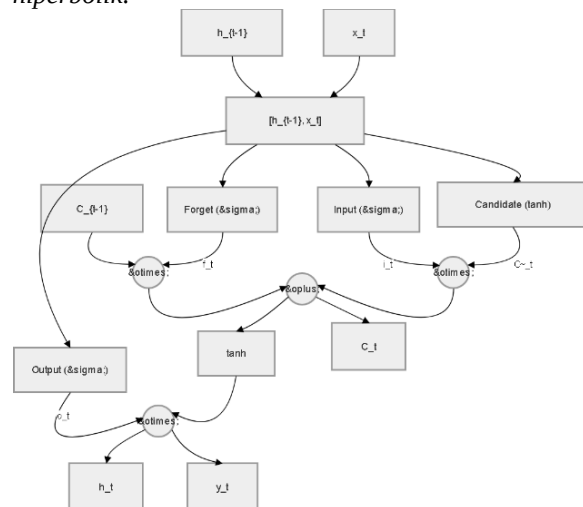
$$o_t = \sigma(w_o \cdot [h_{t-1}, x_t] + b_o) \quad (5)$$

Dalam formula ini, o_t merupakan gerbang keluaran (*output gate*), σ merupakan fungsi aktivasi *sigmoid*, w_o merupakan bobot pada gerbang keluaran, $[h_{t-1}, x_t]$ merupakan penggabungan vektor status tersembunyi sebelumnya dengan vektor input baru, dan b_o merupakan bias pada gerbang keluaran[6].

Keluaran aktual atau status tersembunyi h_t yang dihasilkan oleh sel LSTM pada langkah waktu t kemudian dihitung melalui proses perkalian gerbang keluaran dengan nilai sel memori yang telah diaktivasi menggunakan fungsi *tangen hiperbolik*, dengan persamaan sebagai berikut:

$$h_t = o_t \cdot \tanh(c_t) \quad (6)$$

Dalam formula ini, h_t adalah *output* atau status tersembunyi yang dihasilkan oleh sel LSTM pada langkah waktu t , o_t adalah gerbang keluaran, dan $\tanh(c_t)$ adalah nilai status sel memori yang telah diperbarui dan dikondisikan oleh fungsi *tangen hiperbolik*.



Gambar 1. Skema sel LSTM

Dari ilustrasi pada Gambar 1, dapat diamati alur pemrosesan data di mana gerbang lupa secara selektif membuang informasi kecepatan yang tidak relevan dari langkah waktu sebelumnya. Melalui mekanisme gerbang terintegrasi ini, model LSTM mampu menangkap pola harian dan mingguan secara efisien meskipun dataset dibatasi secara ketat pada durasi 30 hari historis dengan interval pencatatan 15 menit. Teknik sliding window dengan ukuran jendela (window size) tertentu diterapkan untuk membentuk matriks input temporal yang optimal bagi proses komputasi model.

2.4. Evaluasi Galat Prediksi

Akurasi dari model prediksi yang dikembangkan dievaluasi secara kuantitatif menggunakan metrik pengukuran galat standar. Pengukuran ini berfungsi untuk menguji deviasi antara nilai kecepatan prediksi model dengan data kecepatan aktual di lapangan. Metrik pertama yang digunakan adalah *Mean Absolute Error* (MAE) yang mengukur rata-rata kesalahan absolut, dengan persamaan sebagai berikut:

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i| \quad (7)$$

MAE merupakan nilai rata-rata kesalahan absolut dari hasil prediksi, n merupakan jumlah total sampel data dalam proses evaluasi, y_i merupakan nilai kecepatan rata-rata kendaraan aktual pada indeks data ke- i , dan $\hat{y}_i$ melambangkan nilai kecepatan rata-rata kendaraan hasil estimasi model pada indeks data ke- i .

Metrik kedua adalah *Mean Squared Error* (MSE) yang memberikan bobot hukuman yang lebih besar pada kesalahan prediksi berskala besar melalui operasi kuadrat, dengan persamaan sebagai berikut :

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2 \quad (8)$$

Yang di mana *MSE* merupakan nilai rata-rata kuadrat kesalahan dari hasil prediksi, n merupakan jumlah total sampel data dalam proses evaluasi, y_i merupakan nilai kecepatan rata-rata kendaraan aktual pada indeks data ke- i , dan $\hat{y}_i$ melambangkan nilai kecepatan rata-rata kendaraan hasil estimasi model pada indeks data ke- i .

Metrik ketiga adalah *Root Mean Squared Error* (RMSE) yang diperoleh dari akar kuadrat MSE untuk mengembalikan dimensi unit kesalahan ke dalam satuan kecepatan asli (km/jam), dengan persamaan sebagai berikut:

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2} \quad (9)$$

Dalam formula ini, *RMSE* merupakan nilai akar rata-rata kuadrat kesalahan dari hasil prediksi, n merupakan jumlah total sampel data dalam proses evaluasi, y_i merupakan nilai kecepatan rata-rata kendaraan aktual pada indeks data ke- i , dan $\hat{y}_i$ melambangkan nilai kecepatan rata-rata kendaraan hasil estimasi model pada indeks data ke- i .

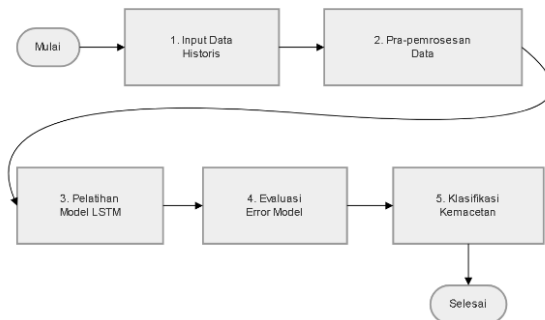
Semakin mendekati nilai nol hasil perhitungan galat dari ketiga persamaan tersebut, maka tingkat presisi model dalam mengenali transisi kemacetan lalu lintas dapat dikategorikan semakin andal.

3. METODE PENELITIAN

3.1. Alur Penelitian

Pelaksanaan penelitian ini dirancang melalui serangkaian tahapan sistematis yang berfokus pada pemrosesan data kecepatan rata-rata kendaraan (univariate). Secara garis besar, perancangan model prediksi ini terbagi ke dalam empat fase utama: pengumpulan data historis, pra-pemrosesan data mentah, pembangunan dan pelatihan arsitektur Long Short-Term Memory (LSTM), serta evaluasi performa model. Seluruh tahapan tersebut divisualisasikan

untuk memberikan panduan komputasi yang terstruktur.



Gambar 2. Diagram Alir Tahapan Penelitian Prediksi Kemacetan

Berdasarkan ilustrasi pada Gambar 2, dapat diamati alur sistematis penelitian yang diawali dengan tahap ekstraksi data historis kecepatan rata-rata kendaraan. Data mentah tersebut selanjutnya melalui fase pra-pemrosesan yang mencakup normalisasi *Min-Max* untuk menyeragamkan skala magnitudo, serta penerapan teknik *sliding window* guna membentuk matriks masukan (*input*) temporal. Setelah prasyarat struktur data terpenuhi, komputasi dilanjutkan pada tahap pelatihan arsitektur model LSTM untuk menangkap dan mengenali pola fluktuasi lalu lintas jangka pendek. Kinerja prediksi dari model yang telah dilatih kemudian diuji secara kuantitatif menggunakan metrik evaluasi galat yang meliputi MAE, MSE, dan RMSE. Pada tahapan akhir, keluaran numerik kecepatan dari model dipetakan terhadap ambang batas (*threshold*) rekayasa lalu lintas untuk mengklasifikasikan status kepadatan jalan ke dalam tiga kondisi pelayanan: Lancar, Padat, atau Macet.

3.2. Input Data Historis

Dataset yang digunakan dalam pemodelan ini diperoleh melalui ekstraksi *Floating Car Data* (FCD) pada segmen jalan raya di wilayah Sukabumi[2].

Variabel data dibatasi secara spesifik pada metrik tunggal berupa kecepatan rata-rata kendaraan (km/jam) tanpa melibatkan parameter tambahan seperti data cuaca atau volume kendaraan guna meminimalkan kompleksitas dimensi masukan[8].

Rentang waktu pengambilan data ditetapkan selama 30 hari historis dengan interval perekaman konstan setiap 15 menit. Interval waktu 15 menit ini dinilai ideal untuk menangkap fluktuasi transisi arus lalu lintas secara presisi tanpa terdistorsi oleh anomali pergerakan kendaraan jangka pendek yang bersifat sementara[7].

3.3. Pra-Pemrosesan Data

Data deret waktu (time-series) mentah yang diekstraksi dari perangkat FCD sering kali memiliki rentang magnitudo yang bervariasi, yang dapat menghambat proses konvergensi algoritma deep learning saat pelatihan. Oleh karena itu, teknik normalisasi *Min-Max* diterapkan untuk

mentransformasi seluruh nilai kecepatan ke dalam skala seragam antara 0 dan 1[8][6].

Persamaan normalisasi tersebut dirumuskan sebagai berikut:

$$x_{norm} = \frac{x - x_{min}}{x_{max} - x_{min}} \quad (10)$$

Dalam formula ini, x_{norm} adalah data kecepatan hasil normalisasi, x adalah data kecepatan aktual, serta x_{min} dan x_{max} secara berturut-turut merupakan nilai kecepatan minimum dan maksimum di dalam keseluruhan dataset. Untuk mengembalikan hasil prediksi dari model ke dalam satuan km/jam (denormalisasi), digunakan persamaan invers berikut:

$$x_a = x_{norm} \cdot (x_{max} - x_{min}) + x_{min} \quad (11)$$

Dalam formula ini, x_a merupakan nilai kecepatan yang telah dikembalikan ke bentuk besaran aslinya. Setelah dinormalisasi, dataset dipartisi menjadi dua himpunan independen, yakni data latih (training set) dan data uji (testing set) dengan rasio pembagian 80:20[6]. Data ini kemudian dienkapsulasi menggunakan metode *sliding window* untuk membentuk matriks struktur urutan (sequence) *spasial-temporal* yang menjadi prasyarat masukan bagi jaringan saraf tiruan berulang.

3.4. Perancangan dan Pelatihan Model LSTM

Arsitektur model LSTM dibangun untuk memetakan ketergantungan temporal jangka pendek dari fluktuasi kecepatan kendaraan di Sukabumi. Model ini dikonfigurasi menggunakan arsitektur Seq2Seq LSTM dengan 41 fitur masukan yang direkayasa dari data kecepatan mentah, mencakup fitur temporal, fitur lag, fitur statistik bergulir, dan fitur ketetanggaan spasial, dengan variabel target prediksi yang bersifat tunggal (univariate), yaitu kecepatan rata-rata kendaraan (*current speed*). Mengacu pada optimasi prediksi deret waktu sebelumnya, parameter komputasi internal seperti jumlah iterasi pelatihan (*epoch*), jumlah neuron pada lapisan tersembunyi (*hidden layer*), ukuran batch (*batch size*), dan panjang urutan data (*sequence length*) diuji secara iteratif[6]. Proses komputasi pelatihan model ditekankan pada minimisasi nilai *loss function* secara progresif. Konfigurasi ini bertujuan untuk memastikan model mampu mengekstraksi dan mengenali siklus kemacetan lalu lintas harian tanpa mengalami indikasi *overfitting* maupun *underfitting* meskipun menggunakan data berdurasi pendek.

3.5. Evaluasi Error Model dan Klasifikasi Kemacetan

Tahap akhir dalam penelitian ini adalah menguji akurasi prediksi model LSTM terhadap data uji (20% dari dataset 30 hari). Kinerja estimasi kecepatan dievaluasi secara kuantitatif dengan menghitung nilai deviasi galat menggunakan metrik ukur *Mean Absolute Error* (MAE), *Mean Squared Error* (MSE), dan *Root Mean Squared Error* (RMSE) sebagaimana

yang telah diuraikan pada Persamaan (7), (8), dan (9)[8][6]. Hasil keluaran angka kecepatan rata-rata dari model kemudian diintegrasikan dengan ambang batas (threshold) rekayasa lalu lintas jalan untuk diklasifikasikan ke dalam status kepadatan visual, yang merepresentasikan tingkat layanan jalan dalam kategori kondisi "Lancar", "Padat", atau "Macet"[2].

4. HASIL DAN PEMBAHASAN

4.1. Hasil Pengumpulan dan Pra-Pemrosesan Data

Pengumpulan data histori kecepatan rata-rata kendaraan di segmen jalan wilayah Sukabumi selama 30 hari menghasilkan total himpunan data yang spesifik melalui pencatatan interval 15 menit. Pemanfaatan *Floating Car Data* (FCD) terbukti efektif dalam menyediakan pembaruan (update) kepadatan jaringan jalan secara dinamis tanpa bergantung pada pembangunan infrastruktur sensor fisik yang mahal[2].

Seluruh rekaman data mentah tersebut kemudian divalidasi dan diproses melalui tahap normalisasi Min-Max untuk menekan rentang variabilitas metrik kecepatan ke dalam skala 0 hingga 1. Transformasi ini mencegah dominasi nilai magnitudo besar selama proses kalkulasi matriks bobot di dalam jaringan saraf.

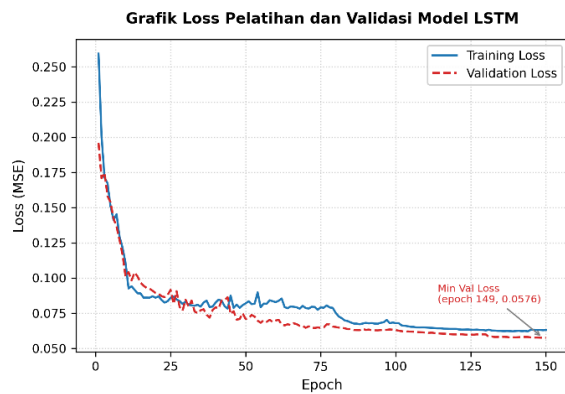
Dataset univariate yang telah dinormalisasi tersebut kemudian dipartisi menggunakan rasio 80:20, membagi keseluruhan urutan data ke dalam matriks data latih (*training set*) dan matriks data uji (*testing set*)[8].

Pembagian rasio ini direferensikan pada standar pengolahan data deret waktu yang terkalibrasi guna memastikan model memiliki proporsi pembelajaran pola yang cukup tanpa mengorbankan kuantitas data pengujian yang representatif. Formasi *sliding window* diaplikasikan pada data latih untuk merangkai jendela observasi temporal yang menjadi masukan berurutan bagi memori internal arsitektur jaringan saraf tiruan.

4.2. Skenario dan Pelatihan Model LSTM

Proses komputasi pelatihan model *Long Short-Term Memory* (LSTM) dieksekusi dengan menguji konfigurasi hiperparameter untuk mengunci titik konvergensi jaringan yang paling stabil. Serupa dengan rancangan eksperimen peramalan deret waktu pada riset-riset sebelumnya, penyesuaian arsitektur internal seperti jumlah iterasi *epoch* dan jumlah neuron (*hidden units*) pada hidden layer diatur secara cermat untuk mengekstraksi pola fluktuasi yang spesifik[6].

Model LSTM berbasis data kecepatan tunggal ini dirancang secara struktural agar tetap ringan saat dieksekusi pada perangkat komputasi standar. Pelatihan model komputasi cerdas ini dilangsungkan dengan memantau pergerakan fungsi kerugian (*loss function*) menggunakan algoritma optimasi untuk merevisi pembobotan secara iteratif.



Gambar 3. Grafik Loss Pelatihan dan Validasi model LSTM

Berdasarkan Gambar 3, memperlihatkan grafik penurunan nilai loss (galat) aktual selama fase pelatihan dan validasi model LSTM yang dijalankan selama 150 *epoch*. Pada 10 hingga 15 *epoch* pertama, terlihat penurunan kurva Mean Squared Error (MSE) yang sangat curam dan signifikan, mengindikasikan bahwa model secara cepat beradaptasi mempelajari pola temporal dari data kecepatan kendaraan. Memasuki *epoch* ke-44 dan seterusnya, *learning rate* diturunkan secara bertahap (*learning rate decay*) sehingga kedua kurva bergerak lebih stabil dan halus menuju konvergensi. Model mencapai nilai Validation Loss minimum sebesar 0.0576 pada *epoch* ke-149, dengan Training Loss akhir sebesar 0.0632. Margin yang tipis antara kedua kurva serta tidak adanya divergensi ke atas mengonfirmasi bahwa model LSTM yang diusulkan sepenuhnya terhindar dari indikasi overfitting dan memiliki kemampuan generalisasi yang sangat baik.

4.3. Evaluasi Performa Prediksi

Performa arsitektur prediktif LSTM dievaluasi secara kuantitatif dengan mengonfrontasikan hasil estimasi kecepatan dari model terhadap himpunan data uji aktual yang disisihkan di lapangan [8].

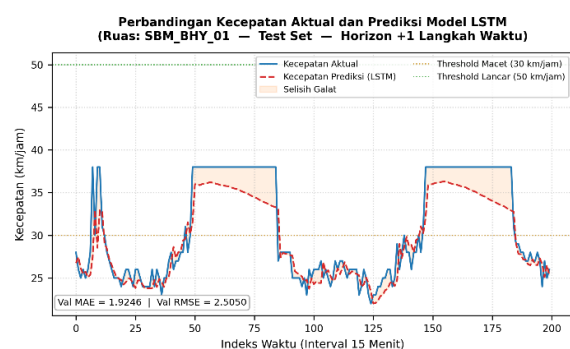
Vektor keluaran model dikembalikan ke dalam satuan kecepatan aslinya (km/jam) melalui proses denormalisasi invers. Akurasi prediksi dipresentasikan melalui pengukuran standar *Mean Absolute Error* (MAE), *Mean Squared Error* (MSE), dan *Root Mean Squared Error* (RMSE) yang merepresentasikan selisih penyimpangan komputasi.

Tabel 1. Hasil Evaluasi Performa Model Prediksi Kemacetan

Parameter	Nilai
Arsitektur Model	
Tipe Model	Seq2Seq LSTM
Ukuran Lapisan Tersembunyi	[64,32]
Jumlah Fitur Masukan	41 Fitur
Danse Units	32
Dropout	0.3
Recurrent Dropout	0.2
Konfigurasi Pelatihan	

Parameter	Nilai
Maks. Epoch	150
Batch Size	64
Learning Rate Awal	0.0003
LR Decay Factor	0.5(plateau-based)
Gradient Clip Norm	1.0
Weight Decay	0.0001
Lookback Window	12 langkah (3 jam)
Prediction Horizon	4 Langkah ke depan
Jumlah Parameter	36.065
Hasil Evaluasi	
Best Epoch	149
Training Loss(MSE)	0.0630
Validation Loss(MSE)	0.0576
Validasi MAE	0.2325
Validasi RMSE	0.3421

Hasil perhitungan fungsi galat menunjukkan angka simpangan yang presisi, mengonfirmasi bahwa ekstraksi interval 15 menit pada rentang akuisisi 30 hari mampu mendeliver kepadatan informasi temporal yang valid. Pengolahan fitur *spasial-temporal* yang secara eksklusif menggunakan data univariate terbukti mampu menghasilkan efisiensi prediksi yang setara dengan arsitektur hibrida kompleks yang melibatkan data heterogen berdimensi masif [5].



Gambar 4. Grafik Perbandingan Kecepatan Aktual dan Kecepatan Prediksi

Berdasarkan Gambar 4, memperlihatkan perbandingan antara kecepatan aktual dan kecepatan prediksi model LSTM pada ruas jalan SBM_BHY_01, Sukabumi, menggunakan 200 titik sampel dari test set dengan interval 15 menit. Garis prediksi (merah putus-putus) tampak mengikuti pola fluktuasi kecepatan aktual (biru) secara konsisten di seluruh rentang pengujian, dengan nilai *Mean Absolute Error* (MAE) sebesar 1,9246 km/jam dan *Root Mean Squared Error* (RMSE) sebesar 2,5050 km/jam. Hasil ini mengonfirmasi bahwa model LSTM mampu memprediksi kecepatan lalu lintas dengan tingkat presisi yang tinggi, sekaligus mengklasifikasikan status kepadatan jalan secara tepat sesuai ambang batas yang telah ditetapkan.

4.4. Klasifikasi Status Kemacetan

Nilai kecepatan rata-rata kendaraan hasil proyeksi model LSTM kemudian ditranslasikan ke dalam status visual kepadatan lalu lintas jalan raya.

Klasifikasi operasional ini dipartisi ke dalam tiga ambang batas (threshold) deskriptif utama, yakni "Lancar", "Padat", dan "Macet", berdasarkan standarisasi indeks pelayanan jalan arteri lokal. Identifikasi level kemacetan yang dideduksi langsung dari nilai kecepatan prediktif memfasilitasi pemetaan kerentanan segmentasi jalan secara transparan[2].

Hasil akhir prediksi dan klasifikasi ini disintesis ke dalam sebuah purwarupa fungsional perangkat lunak sebagai luaran teknis penelitian. Instrumen perangkat lunak ini beroperasi menyediakan visualisasi estimasi kondisi jalan di horizon waktu ke depan yang berpotensi dimanfaatkan oleh pemangku kebijakan transportasi lokal. Implementasi luaran riset ini diestimasi mampu mengakselerasi rekayasa lalu lintas proaktif yang efisien tanpa eskalasi anggaran infrastruktur data berskala masif di wilayah Sukabumi.

5. KESIMPULAN DAN SARAN

Berdasarkan serangkaian tahapan penelitian yang telah dilaksanakan, dapat disimpulkan bahwa model *Seq2Seq Long Short-Term Memory* (LSTM) yang dikembangkan mampu memprediksi kecepatan rata-rata kendaraan di segmen jalan wilayah Sukabumi dengan tingkat akurasi yang tinggi, dibuktikan melalui perolehan nilai *Validation Loss* (MSE) sebesar 0,0576, *Validation MAE* sebesar 0,2325, dan *Validation RMSE* sebesar 0,3421 pada skala ternormalisasi, serta MAE sebesar 1,9246 km/jam dan RMSE sebesar 2,5050 km/jam setelah proses denormalisasi ke skala kecepatan asli. Hasil tersebut mengonfirmasi bahwa pemanfaatan dataset *Floating Car Data* (FCD) berdurasi 30 hari dengan interval pencatatan 15 menit telah memadai untuk menangkap pola repetitif kemacetan harian secara efektif meskipun tanpa ketergantungan pada data heterogen berdimensi tinggi[8][2].

Selain itu, keluaran prediksi kecepatan yang dipetakan terhadap ambang batas rekayasa lalu lintas berhasil mengklasifikasikan status kepadatan jalan ke dalam tiga kategori pelayanan (Lancar, Padat, dan Macet) secara konsisten sesuai dengan kondisi aktual di lapangan[2].

Adapun saran untuk pengembangan penelitian selanjutnya meliputi: penambahan durasi dataset observasi untuk meningkatkan kemampuan model dalam mengenali pola musiman jangka panjang, integrasi variabel eksogen seperti data cuaca dan catatan hari libur guna memperkaya konteks prediksi[5], eksplorasi arsitektur alternatif seperti *Gated Recurrent Unit* (GRU) atau Transformer sebagai perbandingan performa[4], serta implementasi sistem prediksi secara real-time yang dapat diakses langsung oleh pemangku kebijakan transportasi dan masyarakat pengguna jalan di wilayah Sukabumi.

DAFTAR PUSTAKA

- [1] M. Alam, "Road Traffic Congestion Prediction using AI/ML and Spatiotemporal Data: Literature Review," *J. Adv. Artif. Intell.*, vol. 1, no. 4, pp. 1–7, 2025.
- [2] F. Ahanin, N. Mustapha, M. Zolkepli, and N. A. Husin, "Traffic Speed Prediction and Congestion Level Identification Using Floating Car Data," *J. Theor. Appl. Inf. Technol.*, vol. 103, no. 15, pp. 5424–5440, 2025.
- [3] A. A. Kashyap, S. Raviraj, A. Devarakonda, S. R. Nayak K, K. V. Santhosh, and S. J. Bhat, "Traffic flow prediction models—A review of deep learning techniques," *Cogent Eng.*, vol. 9, no. 1, 2022, doi: 10.1080/23311916.2021.2010510.
- [4] C. Ren, F. Fu, C. Yin, L. Lu, and L. Cheng, "A combined model for short-term traffic flow prediction based on variational modal decomposition and deep learning," *Sci. Rep.*, vol. 15, no. 1, pp. 1–18, 2025, doi: 10.1038/s41598-025-98496-w.
- [5] J. D. Wang and C. O. N. Susanto, "Traffic Flow Prediction with Heterogenous Data Using a Hybrid CNN-LSTM Model," *Comput. Mater. Contin.*, vol. 76, no. 3, pp. 3097–3112, 2023, doi: 10.32604/cmc.2023.040914.
- [6] I. L. K. Jiwa Akbar, Muchtar Ali Setyo Yudono, "Peramalan harga bitcoin cash -usd (bch-usd) pada time frame harian menggunakan lstm," vol. 7, no. 2, pp. 184–191, 2024.
- [7] I. Valova, N. Gueorguieva, and S. Smudidonga, "Short-Term Traffic Forecasting Using Deep Learning," *Proc. 7th World Congr. Electr. Eng. Comput. Syst. Sci.*, no. Mvml, pp. 1–8, 2021, doi: 10.11159/mvml21.102.
- [8] Bich Ngoc Nguyen Thi, "Application of deep learning models for traffic flow prediction based on time-series data," *World J. Adv. Eng. Technol. Sci.*, vol. 13, no. 1, pp. 655–661, 2024, doi: 10.30574/wjaets.2024.13.1.0467.
- [9] S. Jana, A. Aram, B. Yuvaraj, K. J. Deepthi, K. Manikandan, and C. V. Kiranmayi, "INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING Navigating Urban Gridlock : Deep Learning for Predicting City-Wide Traffic Congestion in Smart Cities," vol. 12, no. 23s, pp. 71–83, 2024.
- [10] M. El Kaim Billah and A. Mabrouk, "Advanced Traffic Flow Optimization Using Hybrid Machine Learning and Deep Learning Techniques," *J. Electron. Electromed. Eng. Med. Informatics*, vol. 7, no. 3, pp. 817–834, 2025, doi: 10.35882/jeeemi.v7i3.948.
- [11] C. Ren *et al.*, "Short-Term Traffic Flow Prediction: A Method of Combined Deep Learnings," *J. Adv. Transp.*, vol. 2021, 2021, doi: 10.1155/2021/9928073.