

BAB II

KAJIAN PUSTAKA

2.1. Kajian Literatur

Integrasi antara *computer vision* dan *deep learning* telah membawa kemajuan signifikan dalam berbagai aplikasi, termasuk deteksi dan prediksi lalu lintas (Zheng et al., 2022). *Computer vision* memungkinkan sistem untuk mengenali dan menganalisis objek secara visual, sementara *deep learning* memberikan kemampuan pemrosesan data yang lebih kompleks untuk memahami pola lalu lintas secara lebih akurat. Dengan menggabungkan kedua teknologi ini, model deteksi dan prediksi kemacetan dapat dikembangkan dengan tingkat akurasi yang lebih tinggi, memungkinkan pengambilan keputusan yang lebih efektif dalam manajemen lalu lintas perkotaan.

YOLO (*You Only Look Once*) telah menjadi salah satu pendekatan utama dalam deteksi objek dan terus berkembang sejak model pertamanya. Model YOLO asli menekankan deteksi objek berkecepatan tinggi, sementara versi berikutnya, dari YOLOv2 hingga YOLOv5, berfokus pada peningkatan kecepatan dan akurasi untuk mencapai kinerja *real-time* tanpa mengorbankan kualitas deteksi (Terven dan Cordova-esparza, 2023).

YOLO telah diterapkan dalam berbagai aplikasi, seperti pengenalan objek yang dipegang oleh tangan serta deteksi jalur dan rintangan pada kendaraan otonom (Huu et al., 2022). Model ini terus dikembangkan dengan teknologi seperti *Neural Architecture Search* (NAS), *Reparameterized Generalized-FPN* (RepGFPN), penugasan label *AlignedOTA*, dan peningkatan distilasi, yang meningkatkan performanya dalam berbagai tugas deteksi objek (Xu et al., 2022). Salah satu penerapannya adalah dalam deteksi kebakaran, dengan hasil bahwa YOLOv8 menunjukkan akurasi lebih tinggi dibandingkan model sebelumnya (Talaat dan Eldin, 2023). Namun, peningkatan ini juga membawa kompleksitas komputasi yang dapat memengaruhi kecepatan inferensi dalam aplikasi *real-time* (Zhang et al., 2023).

Studi terbaru mengeksplorasi model deteksi objek berbasis YOLO, seperti YOLOv9, YOLOv10, dan YOLO11, untuk mendeteksi asap dan api di lingkungan hutan. Evaluasi dilakukan berdasarkan metrik kinerja utama, termasuk *Precision*, *Recall*, *F1-score*, dan *mean Average Precision (mAP)*, menggunakan dua *Dataset benchmark* yang mencakup berbagai variasi asap dan api. Model YOLO versi kecil, seperti YOLOv9t, YOLOv10n, dan YOLO11n, menunjukkan efektivitas tinggi dalam tugas deteksi kebakaran, dengan YOLO11n mencapai *Precision* sebesar 0,845, *Recall* 0,801, *mAP@50* 0,859, dan *mAP@50-95* 0,558. Selain itu, YOLO11x menunjukkan kinerja yang menjanjikan dibandingkan model YOLO lainnya dalam tugas deteksi asap dan api (Alkhamash, 2025).

Secara teoritis, YOLOv10 memperlihatkan penyempurnaan struktural atas YOLOv8 dalam menangani objek kecil dan cepat, misalnya klasifikasi sepeda ataupun truk mendapatkan hasil yang lebih baik dalam studi (Ding et al., 2024). Namun, implementasinya dalam konteks *real-time* lalu lintas belum optimal karena peningkatan kompleksitas arsitektur dapat memperlambat inferensi atau memerlukan sumber daya komputasi yang jauh lebih besar. Misalnya, studi lain menunjukkan bahwa meskipun versi lebih baru seperti v10 atau varian lain dapat meningkatkan akurasi (mAP) secara teoritis, parameter model dan FLOPS meningkat signifikan sehingga latensi inferensi bertambah (Sundaresan Geetha et al., 2024). Sebaliknya, YOLOv8 masih menawarkan kompromi terbaik antara kecepatan dan akurasi untuk sistem pemantauan lalu lintas berbasis CCTV atau *edge device* saat ini, karena arsitekturnya memang dioptimalkan untuk inferensi yang lebih ringan dan responsif (Rasyid & Sutopo, 2024). Sedangkan YOLOv10 lebih cocok untuk skenario dengan sumber daya komputasi besar atau sistem post-event analysis, di mana kecepatan *ultra-real-time* bukanlah kendala utama dan akurasi deteksi menjadi prioritas.

Penggunaan metode *Long Short-Term Memory (LSTM)* dan *Bidirectional LSTM (Bi-LSTM)* dalam prediksi kemacetan lalu lintas banyak dilakukan. *Transport for Greater Manchester (TfGM)* membangun model LSTM yang akurat dalam menangkap pola temporal dan mengungguli model

*Machine Learning*terkini (Essien et al., 2021). Metode ini juga diadopsi oleh *CityPulse EU FP7 project*, yang menggabungkan SARIMA dan Bi-LSTM, memberikan hasil terbaik dalam evaluasi MAE, MSE, RMSE, dan MAPE (Chahal et al., 2023). Selain itu, penelitian lain menunjukkan bahwa secara umum, Bi-LSTM unggul dalam prediksi aliran lalu lintas dibandingkan dengan LSTM (Zhai et al., 2022).

Metode statistik tradisional seperti *Autoregressive Integrated Moving Average* (ARIMA), meskipun banyak digunakan, mengalami kesulitan dalam menangkap pola lalu lintas yang nonlinier dan ketergantungan temporal (Sattarzadeh et al., 2023). Demikian pula, model *deep learning* klasik seperti LSTM memiliki kemampuan pembelajaran sekuensial yang kuat, tetapi memerlukan penyetelan *Hyperparameter* yang cermat untuk mengoptimalkan kinerjanya (Dhake et al., 2023). Bi-LSTM meningkatkan pembelajaran sekuensial dengan mempertimbangkan ketergantungan dua arah; namun, model ini tetap rentan terhadap masalah *overfitting* (C. Ma et al., 2025).

Beberapa penelitian terbaru telah mengimplementasikan arsitektur *deep learning* hibrida yang menggabungkan *Convolutional Neural Networks* (CNNs) dan LSTM, yang menunjukkan peningkatan dalam ekstraksi fitur *spatiotemporal* (Farhangmehr et al., 2025). Namun, model ini sering kali memerlukan *Dataset* dalam skala besar dan sumber daya komputasi yang tinggi, sehingga kurang praktis untuk aplikasi lalu lintas *real-time*. Di sisi lain, metodologi prediksi lalu lintas terkini mulai beralih pada arsitektur yang lebih canggih seperti *Temporal Convolutional Network* (TCN), *Graph Neural Network* (GNN), dan model berbasis Transformer. Sebagian besar pendekatan modern memanfaatkan *Spatial Temporal Graph Neural Networks* (STGNNs) untuk menganalisis hubungan spasial-temporal yang kompleks dari data lalu lintas yang dikumpulkan melalui ratusan sensor. STGNN menggabungkan GNN dan model sekuensial untuk membentuk struktur hibrida yang memungkinkan kedua jaringan tersebut berkolaborasi dalam menangkap dinamika ruang dan waktu. Namun, kolaborasi ini menyebabkan peningkatan kompleksitas arsitektur dan beban komputasi yang signifikan (Ju Ma et al., 2024).

Model hibrida yang mengintegrasikan algoritma optimasi dengan *deep learning* semakin mendapat perhatian. Misalnya, *Particle Swarm Optimization* (PSO) dan *Artificial Algorithm* (GA) telah digunakan untuk menyempurnakan *Hyperparameter* dalam jaringan LSTM, sehingga meningkatkan akurasi prediksi (Bharti et al., 2023)(X. Li et al., 2024). Namun, teknik optimasi ini sering menghadapi masalah konvergensi lambat dan jebakan pada solusi lokal. Algoritma Komodo Mlipir (KMA), sebuah algoritma metaheuristik yang dikembangkan berdasarkan strategi berburu Komodo, telah menunjukkan hasil yang menjanjikan dalam mengoptimalkan model *deep learning* dengan menyeimbangkan eksplorasi dan eksploitasi (Yeh, 2024). Dengan mengintegrasikan KMA dengan Bi-LSTM, pendekatan ini bertujuan untuk meningkatkan kinerja prediksi sambil mengurangi *overfitting*.

Selain itu, beberapa model *deep learning* terbaru telah dikembangkan untuk meningkatkan akurasi prediksi arus lalu lintas. MFO-Bi-LSTM dirancang untuk prediksi arus lalu lintas jangka pendek dengan tingkat akurasi tinggi. Algoritma *Modified Firefly Optimization* (MFOA) digunakan untuk mengoptimalkan *Hyperparameter* secara otomatis, memungkinkan model Bi-LSTM menangkap pola data yang kompleks secara efisien sekaligus meningkatkan kemampuan pencarian global (Naheliya et al., 2024). Pendekatan ini menghindari jebakan pada solusi lokal dan menghasilkan prediksi yang lebih stabil. Selanjutnya, CAM-LSTM mengintegrasikan *Confined Attention Mechanism* (CAM) untuk meningkatkan akurasi prediksi arus lalu lintas dengan lebih fokus pada dinamika lalu lintas yang relevan. Mekanisme perhatian ini memungkinkan model menangkap fitur penting secara lebih efektif dibandingkan dengan model berbasis *Recurrent Neural Network* (RNN) konvensional (Chauhan dan Kumar, 2024a).

Model lain, FD-*Markov*-LSTM, menggabungkan Fundamental Diagram (FD), *Markov Chain*, dan LSTM untuk meningkatkan estimasi dan prediksi kondisi lalu lintas. Dengan mengintegrasikan status lalu lintas hierarkis, memanfaatkan transisi probabilistik, serta mempertimbangkan pola waktu residual, model ini mencapai akurasi yang lebih tinggi dibandingkan model

benchmark lainnya (Pan et al., 2024). Selain itu, *Hyper-Flophet* dikembangkan untuk prediksi arus lalu lintas dalam *Vehicular Ad Hoc Networks (VANETs)* dengan menggabungkan *Prophet* dan LSTM. Model ini meningkatkan penyesuaian parameter utama, memperkenalkan komponen regresor baru, serta mempertimbangkan faktor kejadian dan hari libur dalam prediksi, sehingga lebih unggul dibandingkan metode konvensional lainnya dalam lingkungan lalu lintas yang dinamis (Zaraket et al., 2024).

Berdasarkan tinjauan literatur ini, jelas bahwa pendekatan berbasis *deep learning* yang dikombinasikan dengan algoritma optimasi dan mekanisme perhatian adaptif dapat secara signifikan meningkatkan akurasi prediksi arus lalu lintas. Integrasi teknik seperti optimasi metaheuristik, diagram fundamental lalu lintas, dan mekanisme perhatian adaptif memperkaya kemampuan model dalam menangani kompleksitas data lalu lintas.

Dalam konteks ini, dapat disimpulkan bahwa metode Bi-LSTM telah menjadi pilihan yang efektif dalam memodelkan dan memprediksi kemacetan lalu lintas, mengintegrasikan berbagai sumber data dan memperoleh hasil yang superior dibandingkan dengan model tradisional. Adapun rincian lebih lanjut mengenai studi terkait penelitian terbaru dapat dilihat pada Tabel 2.1

Tabel 2.1 Detail Studi Terkait Penelitian Terbaru.

No	Penulis	Judul	Tujuan	Metodologi	Hasil	Catatan	Kelebihan	Kekurangan	Kontribusi
1.	Cheng-Jian Lin, Shiou-Yun Jeng, dan Hong-Wei Lioa (Lin et al., 2021) (Hindawi, Q2, November 2021)	<i>A Real-time Vehicle Counting, Speed Estimation, and Classification System Based on Virtual Detection Zone and YOLO</i> (Hindawi, Q2, November 2021)	Melakukan deteksi kendaraan secara <i>real-time</i> dan akurat. Klasifikasi dan Perhitungan Volume Kendaraan.	Dataset: Dataset ini terdiri dari 12.761 gambar <i>training</i> dan 3.190 gambar <i>testing</i> yang berasal dari berbagai sumber, termasuk gambar video lalu lintas dari kamera online di berbagai jalan di Taiwan, Montevideo Audio dan Video Dataset (MAVD), serta set data Pemantauan Lalu Lintas Jalan	Akurasi yang dihasilkan dari metodologi yang digunakan mencapai 99,1% pada siang hari, 98,6% pada malam hari, dan 98% pada hari hujan.	Dengan menggunakan kombinasi YOLOv4 untuk deteksi dan klasifikasi kendaraan serta GMM untuk perhitungan volume, proyek ini berhasil mencapai tingkat akurasi yang sangat tinggi dalam berbagai kondisi cuaca dan waktu.	<i>Dataset</i> dan skenario <i>testing</i> dalam penelitian ini cukup beragam sehingga memungkinkan analisis yang komprehensif dan mendalam terhadap berbagai aspek yang diteliti	Meskipun <i>Dataset</i> yang digunakan cukup beragam, namun sampel data yang diambil kurang kompleks karena terbatas pada situasi tertentu. Data hanya dikumpulkan pada pukul 10 pagi dan 8 malam, serta saat kondisi hujan saja sesuai	Pengembangan model deteksi kendaraan menggunakan kombinasi <i>Computer vision</i> dan <i>Machine Learning</i>

No	Penulis	Judul	Tujuan	Metodologi	Hasil	Catatan	Kelebihan	Kekurangan	Kontribusi
				GARM (GRAM-RTM). Model: YOLOv4 untuk klasifikasi kendaraan dan <i>Gaussian mixture model</i> (GMM) untuk menghitung jumlah kendaraan.		Deteksi kendaraan secara <i>real-time</i> dan akurat menjadi kunci dalam meningkatkan efisiensi dan keamanan lalu lintas.		dengan skenario yang ditentukan.	
2.	Naif Al Mudawi, Asifa Mehmood Qureshi, Maha Abdelhaq, Abdullah Alshahran	<i>Vehicle Detection and Classification via YOLOv8 and Deep Belief Network over Aerial Image Sequences</i>	Melakukan deteksi dan klasifikasi kendaraan dalam urutan gambar udara menggunakan YOLOv8 dan	Dataset: Dataset yang digunakan adalah <i>Aerial Imagery</i> (VEDAI) dan <i>Vehicle Aerial Imagery from a Drone</i> (VAID) Model: Untuk mendeteksi	Model yang diusulkan mencapai akurasi sebesar 95,6% pada <i>Aerial Imagery</i>	Pendekatan yang menggabungkan YOLOv8 untuk deteksi dan <i>Deep Belief Network</i> untuk klasifikasi, bersama dengan penggunaan metode ekstraksi	Model yang diusulkan sudah menggunakan versi YOLO terbaru yaitu YOLOv8	Spesifikasi perangkat yang digunakan untuk menguji model ini terlalu rendah, sehingga harusnya perlu untuk dijelaskan bagaimana cara	Pengembangan model deteksi kendaraan menggunakan kombinasi <i>Computer vision</i> dan

No	Penulis	Judul	Tujuan	Metodologi	Hasil	Catatan	Kelebihan	Kekurangan	Kontribusi
	i, Abdulwahab Alazeb, Mohammed Alonazi, dan Asaad Algarni (Al Mudawi et al., 2023)	(MDPI, Q2, Oktober 2023)	<i>Deep Belief Network</i>	kendaraan dalam setiap <i>frame</i> yang diekstraksi, penelitian ini menggunakan YOLOv8, yang dapat mendeteksi objek kecil dengan efektif. Dan selanjutnya, semua kendaraan yang terdeteksi dikenakan ekstraksi fitur SIFT, ORB, dan KAZE untuk mengklasifikasikannya ke dalam	(VEDAI) dan 94,6% pada <i>Dataset Vehicle Aerial Imagery from a Drone</i> (VAID)	fitur, membuktikan dirinya sebagai solusi yang berhasil untuk tugas deteksi dan klasifikasi kendaraan dalam konteks gambar udara.		<i>testing</i> detail dengan menggunakan spesifikasi rendah seperti ini,	<i>Machine Learning</i>

No	Penulis	Judul	Tujuan	Metodologi	Hasil	Catatan	Kelebihan	Kekurangan	Kontribusi
				beberapa kelas kendaraan.					
3.	Xiaoxu Liu dan Wei Qi Yan (X. Liu dan Yan, 2023)	<i>Vehicle-Related Distance Estimation Using Customized YOLOv7</i> (<i>International Conference, 2022</i>)	Membangun model yang cepat dan tangguh untuk estimasi jarak kendaraan menggunakan YOLOv7	Dataset: Dataset yang digunakan adalah KITTI <i>Dataset</i> Model: YOLOv7 pertama-tama mengubah ukuran gambar input menjadi 640x640 dan memasukkannya ke dalam jaringan <i>backbone</i> , kemudian menghasilkan peta fitur dengan tiga lapisan berbeda	Model estimasi jarak dan mendapatkan nilai 4,253 pada RMSE.	Menggunakan YOLOv7 dengan <i>Dataset</i> KITTI telah membuktikan efektivitasnya dalam membangun model estimasi jarak kendaraan yang cepat dan tangguh. Hasil evaluasi dengan nilai RMSE sebesar 4,253 mengindikasikan	Penelitian ini memberikan gambaran baru terhadap permasalahan lain di lalu lintas	Informasi publikasi artikel kurang lengkap	Pengembangan studi kasus penggunaan teknologi YOLO untuk masalah lain pada kasus lalu lintas, pada penelitian ini adalah sistem deteksi objek untuk

No	Penulis	Judul	Tujuan	Metodologi	Hasil	Catatan	Kelebihan	Kekurangan	Kontribusi
				melalui jaringan lapisan kepala, dan mengeluarkan hasil prediksi melalui modul REP dan modul konvolusi.		kinerja yang baik dalam memprediksi jarak antara kendaraan. Model ini memiliki potensi untuk digunakan dalam berbagai aplikasi kendaraan otonom dan sistem keamanan berkendara.			menghindari tabrakan kendaraan.
4.	Huu-Huy Ngo	<i>Vehicle-Detection-based traffic density estimation at</i>	Melakukan estimasi kepadatan lalu lintas di	Dataset: Video CCTV di 15 jalan persimpangan di	Akurasi dari model yang diusulkan mencapai	Penelitian ini memberikan solusi efektif dalam estimasi	<i>Dataset</i> dan <i>testing</i> dalam penelitian ini cukup	Publikasi artikel kurang menyakinkan dan pembahasan	Mengembangkan teknologi YOLO

No	Penulis	Judul	Tujuan	Metodologi	Hasil	Catatan	Kelebihan	Kekurangan	Kontribusi
	(Ngo, 2023)	<i>road Intersections</i> (International Journal, 2023)	persimpangan berdasarkan deteksi kendaraan	provinsi Thai Nguyen, Vietnam. Model: Penelitian ini akan mendeteksi 5 jenis kendaraan jalan umum di Vietnam, termasuk: sepeda, sepeda motor, mobil, bus, dan truk. Pertama, sistem mengekstrak <i>frame-frame</i> berurutan dari video masukan untuk melakukan pemrosesan pada setiap <i>frame</i> terpisah. Tingkat	nilai mAP50 dan mAP50-95 sebesar 0.994 dan 0.915	kepadatan lalu lintas di persimpangan menggunakan deteksi kendaraan berdasarkan analisis video CCTV di 15 persimpangan di provinsi Thai Nguyen, Vietnam.	beragam, dan sudah menggunakan teknologi YOLO terbaru, yaitu YOLOv8	artikel kurang lengkap dan detail	terbaru untuk studi kasus kepadatan lalu lintas

No	Penulis	Judul	Tujuan	Metodologi	Hasil	Catatan	Kelebihan	Kekurangan	Kontribusi
				<p>pengambilan sampel adalah jumlah <i>frame</i> per detik (fps) dan diatur kustom. Selanjutnya, model <i>Convolutional Neural Networks</i> (CNN) digunakan untuk mendeteksi objek pada setiap <i>frame</i>. Model YOLOv8 digunakan untuk implementasi dan pengembangan sistem ini.</p>					
5.	Yu Zhang, Zhongyin Guo,	<i>Real-time Vehicle Detection Based</i>	Mengembangkan metode deteksi	Dataset: BIT <i>Vehicle_Dataset</i>	Hasil eksperimen menunjukkan	Eksperimen menunjukkan bahwa	Model yang diusulkan dapat	Model YOLO yang dikembangkan	Membangun metode perbaikan

No	Penulis	Judul	Tujuan	Metodologi	Hasil	Catatan	Kelebihan	Kekurangan	Kontribusi
	Jianqing Wu, Yuan Tian, Haotian Tang dan Xinming Guo (Y. Zhang et al., 2022)	<i>on Improved YOLO v5</i> (MDPI, Q2, September 2022)	kendaraan dalam berbagai skenario lalu lintas berdasarkan jaringan YOLOv5	Model: YOLOv5 menggunakan teknik <i>Flip-Mosaic</i> untuk meningkatkan deteksi objek. Dalam proses ini, tiga gambar acak dipilih dari <i>Dataset</i> , kemudian dilakukan operasi pembalikan dan penyisipan <i>Noise</i> acak untuk memperbaiki performa model. YOLOv5 mengatasi ketidakseimbangan sampel positif-negatif dengan	bahwa algoritma peningkatan data <i>Flip-Mosaic</i> yang dikombinasikan dengan YOLOv5 dapat meningkatkan akurasi deteksi kendaraan dan mengurangi tingkat deteksi palsu	kombinasi algoritma peningkatan data <i>Flip-Mosaic</i> dengan YOLOv5 dapat meningkatkan akurasi deteksi kendaraan dan mengurangi tingkat deteksi palsu.	mengurangi tingkat diteksi palsu	adalah YOLOv5, sedangkan saat artikel ini publish model YOLO sudah versi 7	deteksi kendaraan dalam berbagai skenario lalu lintas berdasarkan jaringan YOLO v5 yang telah diperbaiki.

No	Penulis	Judul	Tujuan	Metodologi	Hasil	Catatan	Kelebihan	Kekurangan	Kontribusi
				<p>menggunakan <i>Generalized IoU loss</i> sebagai fungsi kerugian <i>bounding box</i>. Untuk deteksi kendaraan terhadap oklusi, <i>GIoU loss</i> digantikan oleh <i>CIoU loss</i>.</p> <p>Perhitungan <i>frame Anchor</i> yang adaptif digunakan untuk menangani sifat multi-skala dari kendaraan non-kaku jarak jauh.</p> <p>Peningkatan ini secara kolektif</p>					

No	Penulis	Judul	Tujuan	Metodologi	Hasil	Catatan	Kelebihan	Kekurangan	Kontribusi
				meningkatkan kinerja deteksi objek.					
6.	Mujadded Al Rabbani Alif (Alif, 2024)	<i>YOLOv11 for Vehicle Detection: Advancements, Performance, and Applications in Intelligent Transportation Systems</i> (ArXiv, Oktober 2024)	Mengevaluasi performa YOLOv11 untuk deteksi kendaraan dalam konteks <i>Intelligent Transportation Systems (ITS)</i> .	<i>Dataset:</i> 1.321 gambar beresolusi 416×416, mencakup mobil, truk, bus, sepeda motor, dan sepeda, dipecah menjadi 70% train, 15% validasi, 15% test Model: Memperbaiki <i>backbone</i> dengan blok C3k2, SPPF, dan menambahkan C2PSA	mAP@[0.5:0.95]: dari 46.7% menjadi 48.5%		Arsitektur lebih matang: Penambahan C2PSA dan blok C3k2 meningkatkan efisiensi dan akurasi dalam deteksi kecil dan kompleks.	Performanya menurun di threshold tinggi: Untuk <i>Recall</i> tinggi, beberapa objek kecil bisa terlewat.	Menyediakan bukti bahwa arsitektur YOLOv11 memberi peningkatan akurasi dan kecepatan signifikan dibanding pendahulunya.

No	Penulis	Judul	Tujuan	Metodologi	Hasil	Catatan	Kelebihan	Kekurangan	Kontribusi
				(<i>Convolutional Block with Parallel Spatial Attention</i>) untuk mendukung deteksi objek kecil dan terhalang					
7.	Aniekan Essien, Ilias Petrounias, Pedro Sampaio dan Sandra Sampaio (Essien et al., 2021)	<i>A deep-learning model for urban traffic flow prediction with traffic events mined from Twitter</i> (Springer New York, Q1, Maret 2020)	Membangun model LSTM untuk prediksi kemacetan lalu lintas	Dataset: <i>Transport for Greater Manchester</i> (TfGM), Data cuaca dan curah hujan dari pengamatan, atau tweet diperoleh dari dua (2) akun pengguna Twitter informasi lalu lintas jalan raya (TfGM (@OfficialTfGM))	Model yang diusulkan dengan akurat menangkap pola <i>temporal</i> dan secara signifikan mengungguli model <i>Machine Learning</i> terkin. Seperti	Pendekatan ini menjanjikan untuk prediksi kemacetan, dengan integrasi data dari sumber yang berbeda menghasilkan model yang unggul dalam menangkap kompleksitas	<i>Dataset</i> yang digunakan cukup beragam sehingga memungkinkan hasil prediksi semakin akurat	Penelitian ini dilakukan pada tahun 2020, sehingga kurang dapat diandalkan jika digunakan sebagai referensi utama	Membuat model pengembangan metode LSTM, yaitu Bi-LSTM dalam prediksi kemacetan

No	Penulis	Judul	Tujuan	Metodologi	Hasil	Catatan	Kelebihan	Kekurangan	Kontribusi
				<p>dan <i>Waze Manchester</i> (@WazeTrafficMAN)</p> <p>Model: Tahap pertama melibatkan pemrosesan dan pra-pemrosesan data, pengumpulan, dan pembersihan <i>Dataset</i> lalu lintas, cuaca, dan tweet, dengan pembagian menjadi <i>Dataset</i> latih dan uji. Pada tahap kedua, dilakukan penggabungan data</p>	<p>yang ditunjukkan oleh grafik yang disajikan, model yang diusulkan tampil terbaik, dengan model SVR berada di posisi kedua.</p>	<p>dinamika lalu lintas.</p>			

No	Penulis	Judul	Tujuan	Metodologi	Hasil	Catatan	Kelebihan	Kekurangan	Kontribusi
				<p>menggunakan pendekatan DAI-DAO. Selanjutnya, tahap ketiga hingga kelima melibatkan <i>training</i> jaringan LSTM bi-direksional mendalam dengan menggunakan <i>Dataset</i> lalu lintas dan cuaca, serta <i>Dataset</i> lalu lintas, cuaca, dan tweet, masing-masing. <i>Output</i> dari setiap tahap adalah vektor keluaran yang</p>					

No	Penulis	Judul	Tujuan	Metodologi	Hasil	Catatan	Kelebihan	Kekurangan	Kontribusi
				digunakan dalam evaluasi prediksi arus lalu lintas.					
8.	Ayushi Chahal, Preeti Gulia, Nasib Singh Gill and Ishaani Priyadarsh ini (Chahal et al., 2023)	<i>A Hybrid Univariate Traffic Congestion Prediction Model for IoT-Enabled Smart City</i> (MDPI, Q2, April 2023)	Studi ini menyajikan model prediksi kemacetan lalu lintas cerdas yang menggunakan <i>Seasonal Auto-Regressive Integrated Moving Average</i> (SARIMA)	Dataset: <i>CityPulse EU FP7 project</i> Model: Pertama, <i>Dataset</i> diuraikan menggunakan "statsmodels" untuk memeriksa musiman, tren, dan nilai residual. Selanjutnya, komponen linear diberikan kepada model SARIMA, dengan parameter yang dipilih	Model ini memberikan nilai terendah untuk MAE, MSE, RMSE, dan MAPE, masing-masing sebesar 0,499, 0,337, 0,58, dan 0,03.	Studi ini memperkenalkan model prediksi kemacetan lalu lintas yang inovatif, menggabungkan SARIMA dan Bi-LSTM dengan <i>Dataset</i> dari proyek <i>CityPulse</i> EU FP7. Model ini menguraikan	Kombinasi antara 2 metode <i>timeseries</i> cukup menjanjikan untuk peningkatan akurasi data	Model <i>hybrid</i> ini perlu ditinjau lebih dalam lagi, mengingat dalam beberapa penelitian sebelumnya metode SARIMA tidak begitu efektif dibandingkan metode <i>Deep learning</i> terbaru	Membangun model <i>hybrid Deep learning</i> yang efektif untuk diteksi kemacetan lalu lintas

No	Penulis	Judul	Tujuan	Metodologi	Hasil	Catatan	Kelebihan	Kekurangan	Kontribusi
			<p>dan <i>Bidirectional Long Short-Term Memory</i> (Bi-LSTM).</p>	<p>menggunakan "pmdarima." Langkah ketiga melibatkan perhitungan nilai residual dan penggunaannya sebagai input untuk model Bi-LSTM, yang menangani komponen non-linear. Model Bi-LSTM menghasilkan prediksi untuk komponen non-linear tersebut. Pada langkah terakhir,</p>		<p><i>Dataset</i>, menerapkan SARIMA untuk komponen linear, dan Bi-LSTM untuk komponen non-linear. Prediksi dari kedua model digabungkan melalui BPNN.</p>			

No	Penulis	Judul	Tujuan	Metodologi	Hasil	Catatan	Kelebihan	Kekurangan	Kontribusi
				<p>nilai prediksi dari SARIMA dan Bi-LSTM digabungkan menggunakan jaringan saraf berbasis <i>backpropagation</i> (BPNN) untuk memperoleh prediksi akhir. Model ini membedakan diri dengan tidak berspekulasi bahwa penggabungan nilai prediksi hanya dapat dilakukan melalui fungsi penambahan.</p>					

No	Penulis	Judul	Tujuan	Metodologi	Hasil	Catatan	Kelebihan	Kekurangan	Kontribusi
9.	Yujia Zhai, Yan Wan, dan Xiaoxiao Wang (Zhai et al., 2022)	<i>Optimization of Traffic Congestion Management in Smart Cities under Bidirectional Long and Short-term Memory Model</i> (Hindawi Limited, Q2, April 2022)	Membangun model prediksi aliran lalu lintas menggunakan metode Bi-LSTM	Dataset: Data lalu lintas di suatu persimpangan di Distrik Beilin, Xi'an, dipilih. Durasi pengukuran berlangsung selama 10 hari dari 1 hingga 10 November 2021, dengan interval statistik aliran lalu lintas setiap 2 jam. Sebanyak 120 set data diukur Model: Langkah pertama melibatkan pra-pemrosesan data	Perbandingan indikator evaluasi menunjukkan bahwa nilai koefisien determinasi Bi-LSTM adalah 0,795746, lebih tinggi daripada LSTM (0,778742). (MAPE) Bi-LSTM adalah 9,718624%, 9,722147%	Tahap eksperimen Bi-LSTM mencakup pembersihan dan penyaringan data, menentukan arah penelitian, pengumpulan dan rangkuman aliran lalu lintas, serta proses inialisasi dan <i>training</i> Bi-LSTM.	Mejelaskan secara detail kelebihan metode Bi-LSTM dibanding metode LSTM	<i>Dataset</i> yang digunakan terlalu sedikit sehingga hasil kurang dapat diandalkan. Selain itu metode yang digunakanpun sudah ada dalam penelitian sebelumnya	Melakukan prediksi aliran lalu lintas menggunakan <i>Dataset</i> baru

No	Penulis	Judul	Tujuan	Metodologi	Hasil	Catatan	Kelebihan	Kekurangan	Kontribusi
				<p>aliran lalu lintas, diikuti dengan <i>training set</i> menggunakan algoritma Bi-LSTM. Selanjutnya, 20% data aliran lalu lintas dipilih untuk membuat prediksi.</p> <p>Langkah eksperimental melibatkan pembersihan dan penyaringan data aliran lalu lintas asli, menentukan arah penelitian, mengumpulkan dan</p>	<p>dari LS. (MAE) Bi-LSTM (105.087415) lebih kecil daripada LSTM (106.156847) Tahap eksperimen bi-lstm ada disini</p>				

No	Penulis	Judul	Tujuan	Metodologi	Hasil	Catatan	Kelebihan	Kekurangan	Kontribusi
				<p>merangkum aliran lalu lintas dari semua jalur dalam arah yang dipilih, serta menghasilkan set data. Selain itu, proses inisialisasi dan <i>training</i> Bi-LSTM, termasuk fusi linear untuk hasil pemrosesan maju dan mundur, dijelaskan.</p> <p>Pemuatan algoritma Bi-LSTM dan nilai hasil prediksi, serta perolehan hasil akhir melalui</p>					

No	Penulis	Judul	Tujuan	Metodologi	Hasil	Catatan	Kelebihan	Kekurangan	Kontribusi
				pemrosesan fusi linear juga dibahas.					
10.	Ye Zheng, Xiaoming Li, LiuChang Xu dan Nu Wen (Zheng et al., 2022)	<i>A Deep learning-Based Approach for Moving Vehicle Counting and Short-term Traffic Prediction From Video Images</i> (Frontiers Media S.A., Q1, Mei 2022)	Menggabungkan metode pelacakan targetYOLOv4 dan perhitungan kendaraan bergerak yang baru serta jaringan memori jangka pendek yang ditingkatkan (LSTM) untuk	Dataset: Dataset voc2007 + 2012, dan Data lalu lintas yang dikumpulkan di persimpangan Jalan Qiaoxiang dan Nonglin di Distrik Futian, Shenzhen Model: Data volume kendaraan diambil dari 6 jalur, pengambilan data diambil 5 menit sekali, kolom waktu, nomor baris. Pagi saja, jam 7.00	Hasilnya menunjukkan bahwa algoritma yang diusulkan mengungguli metode sebelumnya dalam hal akurasi pelacakan deteksi target yang lebih tinggi sekitar 10% dan	Masih menggunakan YOLOv4, pengembangan belum dioptimalkan pada sisi keringanan proses	Artikel lengkap, detail, dan mudah dipahami	Model YOLO yang dikembangkan adalah YOLOv4, sedangkan saat artikel ini publish model YOLO sudah versi 7. Selain itu model <i>Deep learning</i> yang digunakan juga termasuk ketinggalan, di saat artikel ini publish model Bi-LSTM,	Membangun model perhitungan volume kendaraan <i>real-time</i> menggunakan pendekatan <i>computer vision</i> dan <i>Deep learning</i>

No	Penulis	Judul	Tujuan	Metodologi	Hasil	Catatan	Kelebihan	Kekurangan	Kontribusi
			prediksi aliran lalu lintas dengan kondisi cuaca.	– 9.00, data diambil 2 bulan (Kepadatan lalu lintas). Total 5616 <i>frame</i> , 85% <i>training</i> , 15% <i>testing</i> .	pengurangan sekitar setengah dari kesalahan prediksi lalu lintas, jika mempertimbangkan kondisi cuaca.			menunjukkan metode ini lebih baik dari LSTM untuk deteksi kendaraan sudah ada sebelumnya	
11.	Yang, Dongchuan, Li, Mingzhu, Guo, Ju-e, Du, Pei (Yang et al., 2024)	<i>An attention-based multi-input LSTM with sliding window-based two-stage decomposition</i>	Evaluasi efektifitas pendekatan <i>sliding window</i> dua-tahap (SWTSD) dalam mempersiapkan	Dataset: Empat Musim Dong Xinzhuang wind farm di Shaanxi, China Model: Sinyal mentah dibagi secara bergulir (<i>sliding window</i>)	Model <i>hybrid</i> SWTSD + MI-LSTM mengungguli semua benchmark model pada empat	SWTSD memastikan bahwa setiap <i>window</i> prediksi memiliki substruktur sinyal yang bersih dan	Dua-tahap dekomposisi + <i>sliding window</i> signifikan mengurangi <i>Noise</i> dan <i>leakage</i> , menghasilkan fitur lebih	Kompleksitas komputasi tinggi: dua tahap dekomposisi per <i>window</i> memerlukan banyak waktu/ <i>hardware</i> .	Memperkenalkan metode SWTSD (<i>sliding window</i> dua-tahap) dalam forecasting <i>wind speed</i> ,

No	Penulis	Judul	Tujuan	Metodologi	Hasil	Catatan	Kelebihan	Kekurangan	Kontribusi
		<i>for wind speed forecasting</i> (Applied Energy., Q1, Juli 2024)	an fitur sebelum melakukan prediksi	dan didekomposisi menggunakan ICEEMDAN untuk menghilangkan <i>Noise</i> dan mendapatkan <i>Intrinsic Mode Functions</i> (IMFs).	eksperimen komparatif	terpisah menurut skala.	bersih untuk LSTM.		memperbaiki masalah data <i>leakage</i> dan sinyal <i>Noise</i> .
12.	A. Abirami dan R. Kavitha (Abirami dan Kavitha, 2023)	<i>A novel automated Komodo Mlipir Optimization-based attention BiLSTM for early Detection of diabetic retinopathy</i>	Membangun sistem otomatisasi pendeteksian tahap awal Diabetic Retinopathy (DR) dari gambar fundus mata	Dataset: MESSIDOR dan IDRiD Model: BiLSTM dua arah dioptimasi parameter dengan KMA, ditambah <i>attention Layer</i> untuk menyoroti	Akurasi tinggi ~98.5% pada <i>Dataset</i> MESSIDOR dan IDRiD, dengan waktu komputasi ~50 detik		Optimasi otomatis <i>Hyperparameter</i> ABiLSTM lewat KMA → memaksimalkan performa dengan eksplorasi	Kompleksitas tinggi: kombinasi KMA + BiLSTM ± <i>attention</i> membutuhkan sumber daya komputasi cukup besar.	Memperkenalkan Komodo Mlipir <i>Algorithm</i> ke domain <i>Hyperparameter tuning deep learning</i> –

No	Penulis	Judul	Tujuan	Metodologi	Hasil	Catatan	Kelebihan	Kekurangan	Kontribusi
		(Signal, Image and Video Processing, Q2, Januari 2023)	dengan optimasi <i>Hyperparameter</i> menggunakan Komodo Mlipir <i>Algorithm</i> (KMA)	fitur penting temporal/spasial.			struktur yang luas.		pendekatan unik biologis.

2.2. Keaslian Penelitian

Penelitian ini menawarkan pendekatan inovatif melalui integrasi metode *computer vision* dan *deep learning* untuk prediksi kemacetan secara *real-time* di wilayah perkotaan. Salah satu keunikan utama dari penelitian ini terletak pada penggabungan deteksi visual kendaraan berbasis YOLO dengan model prediksi deret waktu Bi-LSTM, yang jarang dijumpai dalam penelitian serupa, khususnya dalam konteks lalu lintas perkotaan di Indonesia. Model deteksi dikembangkan untuk tetap akurat dalam kondisi lalu lintas padat dan pencahayaan yang kurang optimal, seperti yang umum terjadi di perkotaan.

Keunggulan lain dari penelitian ini adalah penggunaan Algoritma Komodo Mlipir sebagai metode optimasi parameter Bi-LSTM secara otomatis, yang tidak hanya meningkatkan akurasi, tetapi juga stabilitas prediksi dibandingkan dengan metode optimasi konvensional. Penelitian ini juga memanfaatkan teknik *sliding window* untuk membentuk representasi temporal dari lalu lintas dua arah secara dinamis, yang memungkinkan model menangkap pola pergerakan kendaraan secara lebih efektif dan berkelanjutan.

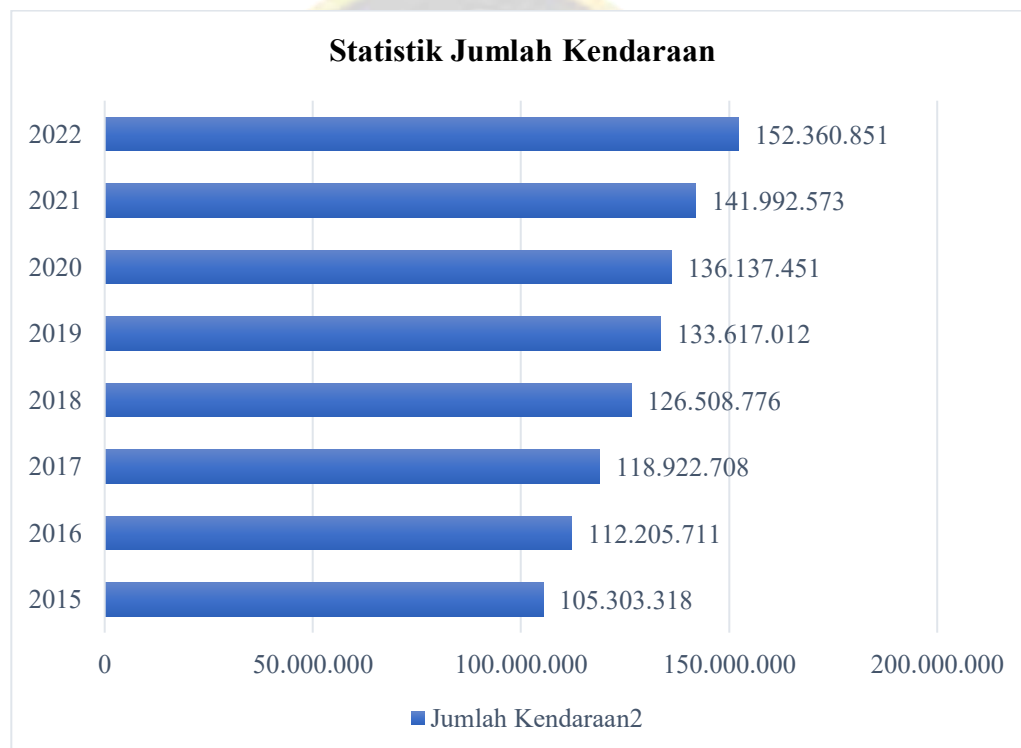
Data pelatihan dalam penelitian ini diperoleh dari rekaman CCTV lalu lintas di kawasan Fatmawati, Kota Semarang, sehingga mencerminkan kondisi riil jalanan kota dan menjadikan hasil penelitian lebih aplikatif dalam mendukung pengambilan keputusan yang cepat dan tepat oleh pemangku kebijakan lalu lintas. Dengan kombinasi teknik-teknik tersebut, penelitian ini diharapkan memberikan kontribusi nyata dalam pengembangan sistem prediksi lalu lintas berbasis *Visual-data-driven*, serta menjadi acuan bagi studi lanjutan di bidang manajemen lalu lintas cerdas berbasis AI.

2.3. Landasan Teori

2.3.1. Statistik Kendaraan Bermotor di Indonesia

Hubungan erat antara pertumbuhan jumlah kendaraan bermotor dan meningkatnya kemacetan lalu lintas semakin nyata seiring dengan bertambahnya pengguna kendaraan. Lonjakan jumlah pemilik kendaraan berdampak langsung pada peningkatan volume lalu lintas di jalan raya. Data yang dihimpun oleh Badan Pusat Statistik (BPS) Indonesia dari tahun 2015 hingga 2022 memberikan

gambaran komprehensif mengenai perubahan jumlah kendaraan, mencakup kategori seperti mobil penumpang, mobil bus, mobil barang, dan sepeda motor. Grafik statistik jumlah kendaraan per tahun dapat dilihat pada Gambar 2.1 (BPS, 2023).



Gambar 2.1 Statistik Jumlah Kendaraan Tahun 2015 s/d 2022.

Berdasarkan data yang dirilis oleh Badan Pusat Statistik (BPS) Indonesia, jumlah kendaraan bermotor di Indonesia menunjukkan tren peningkatan yang konsisten selama periode 2015 hingga 2022. Dalam rentang waktu tersebut, tercatat rata-rata pertumbuhan tahunan sebesar 5,43%, mencerminkan tingginya laju pertumbuhan kendaraan sebagai konsekuensi dari pertumbuhan populasi, aktivitas ekonomi, dan urbanisasi.

Jika dibandingkan secara langsung antara tahun 2015 dan 2022, jumlah kendaraan meningkat sebesar 44,69%, yang menunjukkan peningkatan signifikan dalam waktu tujuh tahun terakhir. Pertumbuhan ini berimplikasi langsung terhadap beban lalu lintas, konsumsi bahan bakar, serta potensi kemacetan, terutama di kawasan perkotaan.

Tabel 2.2 berikut menyajikan rincian persentase pertumbuhan jumlah kendaraan per tahun selama periode tersebut, yang dapat menjadi dasar analisis lebih lanjut mengenai tren mobilitas dan kebutuhan manajemen lalu lintas yang adaptif dan berbasis data (BPS, 2023).

Tabel 2.2 Presentase Jumlah Kendaraan Tahun 2015 s/d 2022.

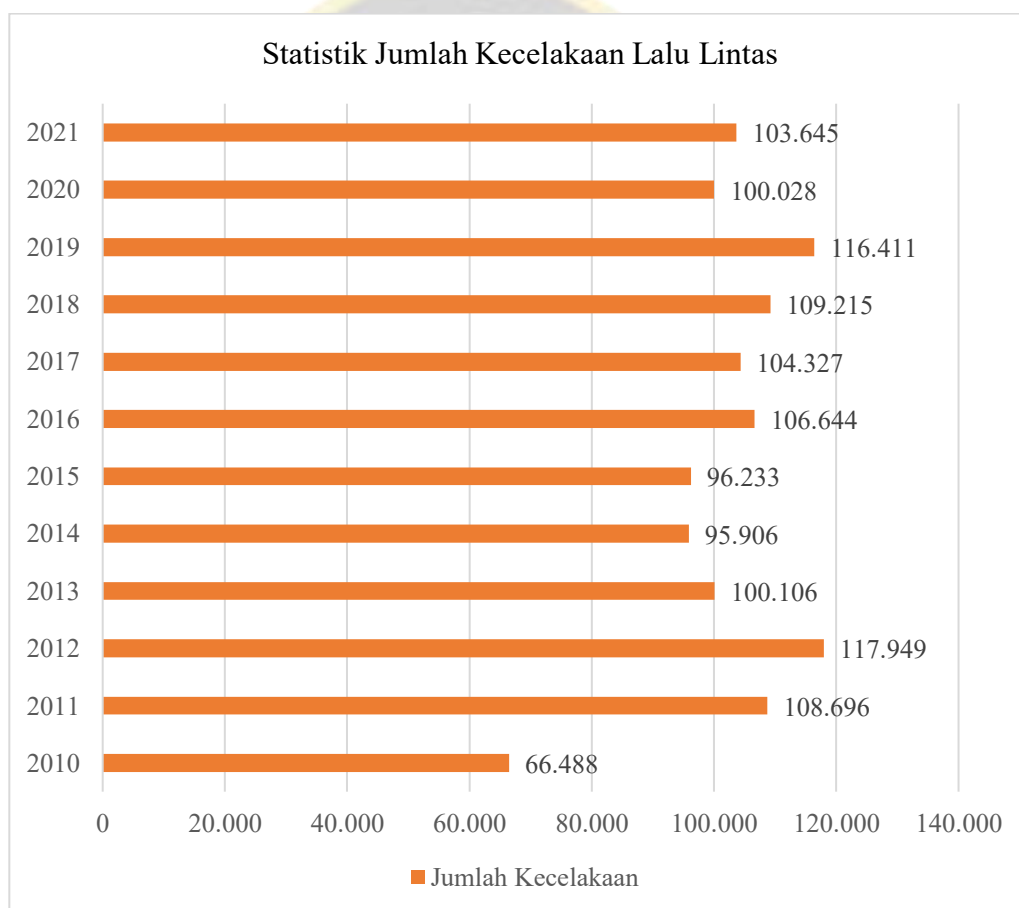
No	Tahun	Jumlah Kendaraan	Presentase Kenaikan tiap Tahun	Peningkatan Presentase sejak tahun 2015
1	2015	105.303.318	0,00%	0,00%
2	2016	112.205.711	6,55%	6,55%
3	2017	118.922.708	5,99%	12,93%
4	2018	126.508.776	6,38%	20,14%
5	2019	133.617.012	5,62%	26,89%
6	2020	136.137.451	1,89%	29,28%
7	2021	141.992.573	4,30%	34,84%
8	2022	152.360.851	7,30%	44,69%

Apabila pertumbuhan jumlah kendaraan terus berlangsung pesat tanpa disertai dengan pengembangan infrastruktur transportasi yang seimbang dan berkelanjutan, maka kemacetan lalu lintas menjadi konsekuensi yang tidak terelakkan. Ketidakseimbangan antara volume kendaraan dan kapasitas jalan yang tersedia akan menimbulkan kepadatan signifikan di berbagai ruas jalan, terutama di kawasan perkotaan dengan tingkat mobilitas tinggi.

Kondisi ini tidak hanya berdampak pada penurunan efisiensi waktu perjalanan dan produktivitas masyarakat, tetapi juga berkontribusi terhadap peningkatan konsumsi bahan bakar, pencemaran udara, serta risiko kecelakaan lalu lintas. Ketika arus lalu lintas tersendat dan kepadatan meningkat, potensi tabrakan, baik ringan maupun berat, turut mengalami eskalasi.

Tren ini tercermin dalam grafik statistik kecelakaan lalu lintas yang ditampilkan pada Gambar 2.2, yang memperlihatkan keterkaitan erat antara

pertumbuhan jumlah kendaraan dan peningkatan jumlah kecelakaan (BPS, 2023). Data ini menekankan pentingnya implementasi teknologi prediktif dan sistem manajemen lalu lintas berbasis data untuk mengantisipasi permasalahan yang semakin kompleks di masa depan.



Gambar 2.2 Statistik Jumlah Kecelakaan Lalu Lintas Tahun 2010 s/d 2021.

Data kecelakaan lalu lintas yang dihimpun selama periode 2010 hingga 2021 menunjukkan tren peningkatan yang cukup signifikan. Selama periode tersebut, rata-rata persentase kenaikan kecelakaan lalu lintas per tahun mencapai 5,21%, yang menunjukkan bahwa meskipun berbagai upaya keselamatan telah dilakukan, insiden kecelakaan masih mengalami peningkatan secara konsisten.

Secara kumulatif, dalam rentang waktu sebelas tahun tersebut terjadi kenaikan total sebesar 55,89% pada jumlah kecelakaan lalu lintas. Angka ini menjadi indikator penting bagi pemerintah dan pemangku kebijakan untuk mengevaluasi efektivitas sistem pengendalian lalu lintas dan strategi

keselamatan jalan yang telah diterapkan. Tabel 2.3 berikut menyajikan data persentase kecelakaan lalu lintas per tahun, yang dapat digunakan sebagai dasar untuk analisis tren dan penyusunan kebijakan mitigasi kecelakaan yang lebih tepat sasaran di masa mendatang (BPS, 2023).

Tabel 2.3 Presentase Jumlah Kecelakaan Lalu Lintas Tahun 2010 s/d 2021.

No	Tahun	Jumlah Kecelakaan	Presentase Kenaikan tiap Tahun	Peningkatan Presentase sejak tahun 2015
1	2010	66.488	0,00%	0,00%
2	2011	108.696	63,48%	63,48%
3	2012	117.949	8,51%	77,40%
4	2013	100.106	-15,13%	50,56%
5	2014	95.906	-4,20%	44,25%
6	2015	96.233	0,34%	44,74%
7	2016	106.644	10,82%	60,40%
8	2017	104.327	-2,17%	56,91%
9	2018	109.215	4,69%	64,26%
10	2019	116.411	6,59%	75,09%
11	2020	100.028	-14,07%	50,45%
12	2021	103.645	3,62%	55,89%

2.3.2. Transportasi Cerdas

Transportasi cerdas merupakan pendekatan inovatif dalam berbagai aspek transportasi, seperti infrastruktur canggih, mobilitas, pengendalian lalu lintas, dan sistem keamanan lalu lintas (Kothadiya *et al.*, 2021). Saat ini, lebih dari 50% populasi dunia tinggal di perkotaan, dan Program Pembangunan Perserikatan Bangsa-Bangsa memperkirakan bahwa pada tahun 2030, dua pertiga populasi global akan menetap di kota-kota, yang mengindikasikan bahwa kepadatan penduduk perkotaan akan meningkat dan berpotensi memperburuk permasalahan lalu lintas.

Kota pintar menawarkan solusi terhadap berbagai tantangan perkotaan, seperti pencemaran lingkungan, kelangkaan sumber daya, kemacetan lalu lintas, dan keamanan publik, dengan transportasi cerdas sebagai salah satu komponen utamanya (Yan et al., 2020). Transportasi cerdas memanfaatkan teknologi komunikasi terbaru untuk meningkatkan efisiensi pergerakan, pemantauan lokasi, interaksi antara kendaraan dan elemen lalu lintas lainnya, serta keselamatan lalu lintas secara umum, yang berkontribusi terhadap penurunan angka kecelakaan dengan kecenderungan menuju nol kematian (Evtiukov et al., 2020).

Teknologi ini juga berperan penting dalam meningkatkan kualitas hidup masyarakat di lingkungan perkotaan. Implementasi transportasi cerdas melibatkan penggunaan perangkat dan sensor dalam sistem kendali kendaraan, seperti sistem navigasi mobil, manajemen sinyal lalu lintas, pengenalan nomor kendaraan, serta pemantauan kecepatan (Babar dan Arif, 2019).

Pengendalian lalu lintas merupakan aspek utama dalam menciptakan sistem transportasi cerdas yang efisien. Dengan pengelolaan yang terkoordinasi, kondisi jalan raya dapat lebih teratur, sehingga mengurangi kemacetan dan menurunkan angka kecelakaan. Namun, penerapan sistem transportasi cerdas yang berbasis analisis data video dari kamera CCTV masih menghadapi berbagai tantangan. Dalam desain saat ini, data video yang diperoleh harus dikirim kembali ke Pusat Kontrol Transportasi untuk dianalisis lebih lanjut (Liu et al., 2022).

2.3.3. Kamacetan Lalu Lintas

Kemacetan lalu lintas merupakan kondisi pada jaringan jalan ketika permintaan lalu lintas melebihi kapasitas jaringan yang tersedia, sehingga menyebabkan penurunan kinerja sistem transportasi. Kondisi ini ditandai oleh kecepatan kendaraan yang menurun, waktu tempuh yang semakin panjang, ketidakpastian waktu kedatangan, serta meningkatnya antrean kendaraan, terutama pada kawasan perkotaan dengan tingkat mobilitas yang tinggi (Bansal & Graham, 2023).

Dalam perspektif jaringan jalan, kemacetan tidak hanya dipengaruhi oleh jumlah kendaraan, tetapi juga oleh interaksi kompleks antara arus lalu lintas, kapasitas jalan, sistem pengendalian lalu lintas, dan perilaku pengguna jalan. Hubungan ini sering direpresentasikan melalui konsep *congestion technology* yang menggambarkan perubahan kinerja jaringan jalan, seperti arus dan kecepatan, seiring dengan meningkatnya intensitas penggunaan jaringan. Sejumlah penelitian menunjukkan bahwa peningkatan kapasitas jalan tidak selalu efektif dalam mengurangi kemacetan, karena pada kondisi tertentu justru dapat memicu peningkatan volume lalu lintas, sehingga kecepatan rata-rata jaringan tidak mengalami peningkatan yang signifikan (Cvetek et al., 2021).

Kemacetan lalu lintas juga berdampak luas terhadap aspek ekonomi, sosial, dan lingkungan, antara lain meningkatnya konsumsi bahan bakar, emisi polutan, pemborosan waktu perjalanan, penurunan produktivitas, serta menurunnya kualitas hidup masyarakat perkotaan. Kompleksitas permasalahan ini mendorong perlunya pendekatan yang lebih adaptif dan berbasis data dalam memahami serta mengelola kondisi lalu lintas (Shorbagy et al., 2024).

Untuk merepresentasikan tingkat kemacetan secara kuantitatif, berbagai indikator kinerja lalu lintas digunakan, seperti kecepatan kendaraan, tingkat okupansi penggunaan jalan, volume atau jumlah kendaraan, serta waktu tempuh. Di antara indikator tersebut, kecepatan kendaraan dipandang sebagai indikator yang paling prioritas karena secara langsung mencerminkan kondisi aktual pergerakan lalu lintas dan mudah diinterpretasikan sebagai tingkat pelayanan jalan. Selanjutnya, okupansi penggunaan jalan berfungsi sebagai indikator pendukung yang menggambarkan tingkat pemanfaatan ruang jalan, sedangkan volume kendaraan digunakan sebagai indikator tambahan untuk menunjukkan intensitas permintaan lalu lintas. Dengan urutan prioritas tersebut, penilaian kemacetan diharapkan lebih representatif terhadap kondisi lapangan dan relevan untuk mendukung sistem estimasi serta prediksi kemacetan lalu lintas berbasis data.

2.3.4. *Computer Vision*

Computer vision (CV) merupakan teknik yang diakui sebagai alat untuk mengekstraksi informasi dari berbagai jenis data, baik foto maupun video, serta menangani tugas deteksi objek pada berbagai skala (Illarionova dan Shadrin, 2022). Dalam beberapa tahun terakhir, dengan berkembangnya teori *Deep learning* serta meningkatnya kemampuan perangkat keras GPU, teknologi *computer vision* juga mengalami kemajuan pesat. Penerapan *computer vision* dalam berbagai bidang, termasuk transportasi, memiliki manfaat praktis yang signifikan dalam mengurangi ketergantungan pada tenaga kerja manusia.

Deteksi objek merupakan salah satu cabang fundamental dalam pengolahan citra digital dan *computer vision*, serta menjadi komponen inti dalam sistem pemantauan cerdas untuk berbagai aplikasi. Meskipun *Dataset* telah dipersiapkan pada bagian sebelumnya, tugas deteksi kendaraan masih menghadapi tantangan dalam mengenali objek berukuran kecil. Kamera pemantauan lalu lintas umumnya dipasang pada ketinggian 8 hingga 12 meter, sehingga kendaraan yang berada jauh dari kamera cenderung tampak kecil dalam rekaman. Selain itu, gambar yang dihasilkan sering kali mengalami efek bayangan serta memiliki informasi yang terbatas, yang menyebabkan kesulitan dalam deteksi objek (Zhang et al., 2022).

Selain itu, *computer vision* juga memainkan peran sentral dalam sistem pengenalan dan pelacakan objek di sistem transportasi cerdas masa depan, termasuk dalam pengembangan mobil pintar. Beberapa manfaat utama dari mobil pintar mencakup aspek berikut (Huu et al., 2022):

- a. Mobil pintar dapat mengurangi kecelakaan lalu lintas yang disebabkan oleh kesalahan manusia, seperti stres, mabuk, atau kelelahan.
- b. Mobil pintar dapat membantu penyandang disabilitas, lansia, atau individu yang tidak memiliki izin mengemudi untuk bepergian jarak jauh dengan aman.
- c. Industri transportasi mengalami transformasi signifikan, yaitu barang dapat dikirim secara otomatis dengan lebih cepat, akurat, dan biaya yang lebih rendah, terutama untuk pengiriman jarak jauh.

- d. Transportasi umum, seperti taksi dan bus, dirancang untuk beralih ke mobil pintar yang dapat mengangkut banyak penumpang hampir sepanjang waktu, meningkatkan efisiensi operasional dengan meminimalkan waktu tunggu dan jarak tempuh pelanggan. Selain itu, mobil pintar berkontribusi pada perbaikan tata ruang perkotaan dengan mengurangi kebutuhan lahan parkir, mengurangi kepadatan lalu lintas, serta menekan tingkat kemacetan, yang pada akhirnya berdampak positif terhadap lingkungan.

Salah satu model algoritma *computer vision* yang banyak digunakan adalah *regression-based Algorithms*. Algoritma ini menentukan koordinat posisi serta probabilitas klasifikasi objek dalam gambar secara langsung, tanpa perlu melakukan tahap deteksi awal. Kemampuan ini memungkinkan proses deteksi menjadi sangat cepat dengan hasil yang dapat diperoleh secara *real-time*. Beberapa contoh *regression-based Algorithms* yang populer meliputi *You Only Look Once (YOLO)*, *Single Shot MultiBox Detector (SSD)*, *Retina Neural Network (RetinaNet)*, dan *Center point Detection Network (CenterNet)* (Arkin *et al.*, 2023). Dalam sebuah penelitian yang membahas deteksi objek untuk identifikasi tablet secara *real-time* di farmasi rumah sakit, YOLOv3 menunjukkan keunggulan dalam kecepatan deteksi sambil tetap mempertahankan evaluasi *mean Average Precision (mAP)* yang memuaskan dibandingkan dengan *RetinaNet* dan *Single Shot MultiBox Detector (SSD)* (Arkin *et al.*, 2023).

2.3.5. YOLO

. YOLO, singkatan dari "*You Only Look Once*," adalah sebuah metode dalam pemrosesan citra komputer dan *computer vision* yang digunakan untuk mendeteksi objek dalam gambar atau video dengan cepat dan akurat (Gillani *et al.*, 2022). YOLO menggabungkan deteksi objek dan klasifikasi objek dalam satu langkah, menjadikannya salah satu pendekatan terdepan dalam bidang deteksi objek *real-time*.

Salah satu pengembang terkemuka dalam pengembangan model YOLO adalah *Ultralytics*, yang secara aktif berkontribusi dalam pengembangan

berbagai versi YOLO serta menghadirkan fitur-fitur baru yang relevan untuk kebutuhan industri dan penelitian. Berikut ini adalah daftar perkembangan model YOLO yang diadaptasi oleh *Ultralytics*:

1. YOLOv2 (2016)
YOLOv2 memperkenalkan inovasi seperti *Batch normalization*, *Anchor Boxes*, dan *dimension clusters*, yang meningkatkan akurasi deteksi (Redmon dan Farhadi, 2017).
2. YOLOv3 (2018)
YOLOv3 menambahkan *backbone* Darknet-53, penggunaan *multi-scale prediction*, dan *residual Blocks* untuk meningkatkan deteksi objek pada berbagai ukuran (Redmon dan Farhadi, 2018).
3. YOLOv4 (2020)
YOLOv4, yang dikembangkan oleh komunitas, membawa fitur seperti *Mosaic data augmentation*, *CSPDarknet53*, dan *bag of freebies* untuk pelatihan (Bochkovskiy et al., 2020).
4. YOLOv5 (2020)
YOLOv5, dikembangkan langsung oleh *Ultralytics*, dikenal karena kemudahan penggunaan, otomatisasi *Hyperparameter*, dan ekspor model yang fleksibel (*Ultralytics*, 2022).
5. YOLOv6 (2022)
YOLOv6 fokus pada kecepatan inferensi dan akurasi *real-time* untuk industri, terutama pada sistem robotika (*Ultralytics*, 2023).
6. YOLOv7 (2022)
YOLOv7 menambahkan kemampuan baru seperti *task-aligned regression*, *pose estimation*, dan model yang efisien untuk perangkat keras terbatas (C. Wang et al., 2023.).
7. YOLOv8 (2023)
Dirilis oleh *Ultralytics*, YOLOv8 merupakan arsitektur baru yang mendukung deteksi objek, segmentasi, klasifikasi, pelacakan, dan estimasi pose dalam satu *framework* (Yaseen, 2024).

8. YOLOv9 (2024)

YOLOv9 memperkenalkan teknik PGI (*Programmable Gradient Information*) dan GELAN (*Generalized Efficient Layer Aggregation Network*) untuk efisiensi tinggi dan performa unggul (C.-Y. Wang et al., 2024).

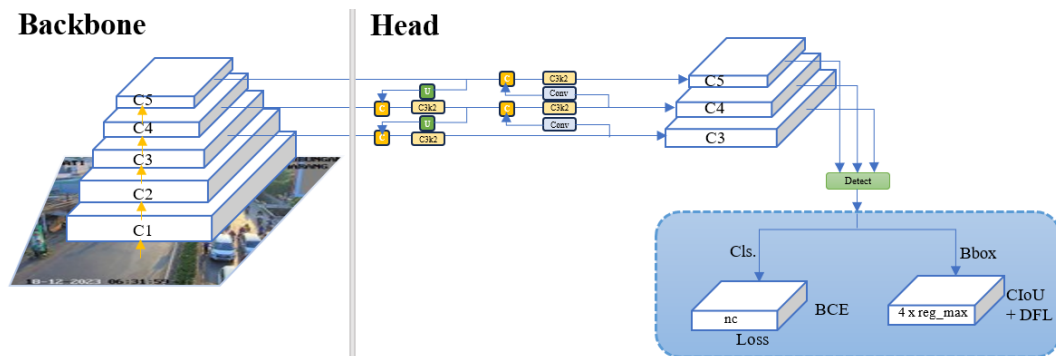
9. YOLOv10 (2024)

Dikembangkan oleh Tsinghua *University* menggunakan kerangka kerja *Ultralytics*, versi ini mengenalkan *End-to-end Detection Head* yang menghilangkan proses NMS (Livine, 2024).

10. YOLO11 (2025) (Terbaru)

Dirilis oleh *Ultralytics*, YOLO11 adalah model mutakhir yang mendukung berbagai tugas visi komputer secara efisien: deteksi, segmentasi, klasifikasi, pelacakan, hingga estimasi pose (Khanam dan Hussain, 2024).

Versi YOLO yang secara resmi dikembangkan oleh *Ultralytics* adalah YOLOv5, YOLOv8, dan YOLO11. Ketiga versi ini merupakan hasil pengembangan murni dari tim *Ultralytics*, bukan lanjutan dari pengembang sebelumnya seperti Joseph Redmon atau komunitas independen lainnya. YOLOv5 menjadi titik awal kontribusi *Ultralytics* dalam pengembangan YOLO, menghadirkan kemudahan penggunaan dan efisiensi yang tinggi. Selanjutnya, YOLOv8 diperkenalkan dengan arsitektur baru yang lebih fleksibel dan mendukung berbagai tugas visi komputer seperti deteksi objek, segmentasi, pelacakan, klasifikasi, hingga estimasi pose. Terbaru, YOLO11 melanjutkan inovasi dengan peningkatan performa dan efisiensi, serta kemampuan mendukung berbagai domain aplikasi secara lebih luas. Ketiga versi ini dikembangkan dalam ekosistem *Ultralytics* yang terintegrasi, dengan dukungan dokumentasi resmi, perangkat lunak *Python*, serta pembaruan berkelanjutan. Untuk modul arsitektur umum dari YOLO11 atau YOLO versi terbaru saat ini dapat dilihat pada Gambar 2.3.



Gambar 2.3 Arsitektur YOLOv11.

Arsitektur YOLO terdiri dari dua bagian utama, yaitu *backbone* dan *head*. *Backbone* adalah komponen penting dalam arsitektur YOLO yang bertugas mengekstraksi fitur dari citra masukan pada berbagai skala, dilakukan dengan menyusun lapisan konvolusional dan blok khusus untuk menghasilkan peta fitur (*feature maps*) dengan resolusi berbeda-beda. YOLO11 mempertahankan struktur serupa dengan versi sebelumnya, menggunakan lapisan konvolusional awal untuk menurunkan resolusi citra secara bertahap sambil meningkatkan jumlah saluran (*channels*). Peningkatan utama pada YOLO11 adalah pengenalan blok C3k2, yang menggantikan blok C2f dari versi sebelumnya. Blok C3k2 ini merupakan implementasi yang lebih efisien secara komputasi dari *Cross Stage Partial (CSP) Bottleneck*, dengan menggunakan dua konvolusi kecil sebagai pengganti satu konvolusi besar seperti yang terdapat pada YOLOv8. Notasi "k2" menunjukkan ukuran kernel yang lebih kecil, yang berkontribusi terhadap proses yang lebih cepat namun tetap mempertahankan performa.

Selain itu, YOLO11 tetap menggunakan blok *Spatial Pyramid Pooling - Fast (SPPF)* dari versi sebelumnya, namun kini menambahkan blok baru bernama C2PSA (*Cross Stage Partial with Spatial Attention*). Blok ini memperkuat mekanisme atensi spasial (*spatial attention*) dalam peta fitur, memungkinkan model untuk lebih fokus pada area penting dalam citra. Dengan melakukan pooling fitur secara spasial, C2PSA membantu YOLO11 memusatkan perhatian pada area-area yang relevan, sehingga meningkatkan akurasi deteksi terhadap objek dengan ukuran dan posisi yang bervariasi.

Komponen *neck* dalam YOLO11 berfungsi menggabungkan fitur dari berbagai skala dan meneruskannya ke bagian *head* untuk prediksi akhir. Proses ini mencakup *upsampling* dan *concatenation* dari peta fitur yang berasal dari berbagai level, sehingga informasi multi-skala dapat ditangkap secara efektif. YOLO11 menggantikan blok C2f pada bagian *neck* dengan C3k2 untuk mempercepat dan mengoptimalkan proses agregasi fitur. Selain itu, perhatian lebih terhadap area spesifik juga diperkuat melalui modul C2PSA yang ditambahkan di bagian *neck*, membedakannya dari YOLOv8 yang tidak memiliki mekanisme ini.

Bagian *head* bertanggung jawab menghasilkan prediksi akhir berupa deteksi objek dan klasifikasi. Bagian ini memproses peta fitur dari *neck*, dan mengeluarkan koordinat kotak pembatas serta label kelas dari objek dalam citra. YOLO11 memanfaatkan beberapa blok C3k2 di bagian *head* untuk memproses fitur secara efisien. Blok C3k2 di sini fleksibel tergantung parameter *c3k*: jika *c3k = False*, modul berperilaku seperti blok C2f dengan struktur *Bottleneck* standar; sedangkan jika *c3k = True*, digunakan struktur C3 untuk ekstraksi fitur yang lebih kompleks dan dalam. Ciri khas utama C3k2 adalah proses yang lebih cepat karena menggunakan dua konvolusi kecil serta efisiensi parameter yang lebih baik dibandingkan versi CSP *Bottleneck* sebelumnya. Selain itu, ditambahkan pula blok C3k yang lebih fleksibel dengan ukuran kernel yang dapat disesuaikan, meningkatkan kemampuan dalam mengekstraksi detail dari citra.

Untuk menyempurnakan peta fitur, bagian *head* juga dilengkapi dengan beberapa blok CBS (*Convolution-BatchNorm-SiLU*). Blok ini bertugas mengekstraksi fitur penting secara akurat, menstabilkan aliran data melalui *Batch normalization*, serta meningkatkan performa model dengan menggunakan fungsi aktivasi *Sigmoid Linear Unit* (SiLU). CBS menjadi komponen penting baik dalam ekstraksi fitur maupun proses deteksi. Pada akhir setiap cabang deteksi, terdapat serangkaian lapisan *Conv2D* yang mengurangi peta fitur ke jumlah keluaran yang dibutuhkan untuk prediksi koordinat *bounding box* dan klasifikasi objek. Lapisan *Detect* terakhir akan mengonsolidasikan hasil ini,

termasuk koordinat kotak pembatas untuk lokalisasi objek, skor objektivitas untuk mengindikasikan keberadaan objek, dan skor kelas untuk menentukan jenis objek yang terdeteksi.

2.3.6. Varian Ringan YOLO: YOLOv5n, YOLOv8n, dan YOLO11n

Evolusi model You Only Look Once (YOLO) menunjukkan tren peningkatan efisiensi arsitektur dalam menangani tugas deteksi objek secara real-time. Untuk memberikan pemahaman komprehensif mengenai posisi masing-masing model dalam hirarki teknologi deteksi objek, dijelaskan perbandingan teknis terhadap arsitektur yang digunakan. Perbandingan Arsitektur YOLOv5n, YOLOv8n, dan YOLO11n disajikan pada Tabel 2.4 di bawah ini.

Tabel 2.4 Perbandingan Arsitektur Model YOLOv5n, YOLOv8n, dan YOLOv11n.

Fitur / Metrik	YOLOv5n	YOLOv8n	YOLO11n
<i>Box Prediction</i>	<i>Anchor-based</i>	<i>Anchor-free</i>	<i>Anchor-free</i>
<i>Backbone</i>	CSP-Darknet	C2f Module	C3k2
Jumlah Parameter	±1.9 M	±3.2 M	±2.6 M

Secara fundamental, perbedaan ketiga arsitektur ini berfokus pada transisi metode lokalisasi objek dan optimasi komponen ekstraksi fitur (*backbone*). Pada generasi awal, YOLOv5n menerapkan arsitektur *CSP-Darknet* dengan pendekatan *Anchor-based*. Metode ini mengharuskan model menggunakan sekumpulan kotak acuan (*prior boxes*) dengan dimensi yang telah ditentukan sebelumnya untuk mendeteksi keberadaan objek. Meskipun pendekatan ini sangat stabil dan efisien dalam penggunaan memori, ketergantungan pada *anchor* sering kali membatasi fleksibilitas model dalam mengenali objek yang memiliki variasi rasio aspek yang sangat beragam atau ekstrem.

Pengembangan berlanjut pada YOLOv8n yang memperkenalkan perubahan paradigma melalui implementasi strategi *Anchor-free*. Dengan menghilangkan kotak acuan, model dapat memprediksi titik pusat objek secara

langsung, yang secara signifikan meningkatkan akurasi deteksi pada objek berukuran kecil serta mempercepat proses konvergensi saat pelatihan. Selain itu, YOLOv8n memperbarui struktur *backbone* menggunakan C2f Module. Penambahan modul ini bertujuan untuk memperkaya aliran gradien informasi dalam jaringan, sehingga meskipun jumlah parameter meningkat menjadi ± 3.2 M, kemampuan ekstraksi fitur menjadi jauh lebih representatif dibandingkan sebelumnya.

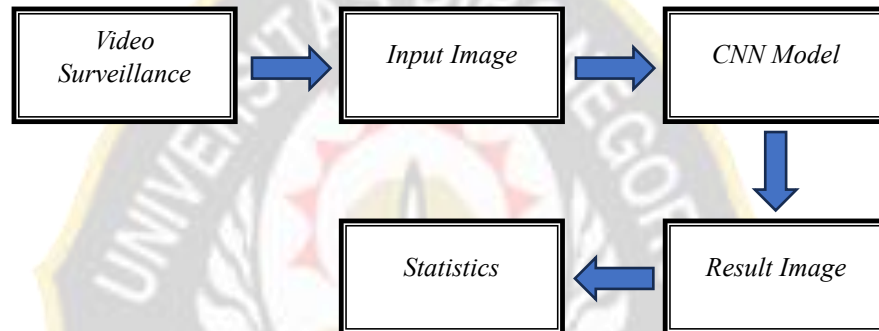
Puncak optimasi saat ini direpresentasikan oleh YOLO11n, yang dirancang untuk memberikan performa superior dengan beban komputasi yang lebih rendah. Arsitektur ini menyempurnakan struktur *backbone* dan *neck* melalui penggunaan modul C3k2. Inovasi utama pada YOLO11n terletak pada kemampuannya mengurangi jumlah parameter menjadi ± 2.6 M tanpa mengorbankan kedalaman fitur atau akurasi deteksi. YOLO11n secara khusus dioptimalkan untuk memaksimalkan *throughput* pada perangkat keras modern, menjadikannya pilihan paling efisien dalam hal keseimbangan antara kecepatan pemrosesan fitur per lapisan dan penggunaan memori.

2.3.7. YOLO untuk Deteksi Volume Kendaraan

Salah satu versi YOLO yang dikembangkan untuk deteksi volume kendaraan adalah YOLOv8. Gambaran umum dari implementasi YOLOv8 untuk mendeteksi volume kendaraan dapat dilihat pada Gambar 2.4 (Ngo, 2023). Data masukan berupa video pengawasan di persimpangan jalan, dengan tujuan penelitian ini untuk mendeteksi 5 jenis kendaraan umum di Vietnam, yaitu sepeda, sepeda motor, mobil, bus, dan truk.

Proses dimulai dengan mengekstrak *frame* berturut-turut dari video masukan untuk diproses secara terpisah, dengan tingkat sampling yang dapat diatur secara kustom berdasarkan jumlah *frame per second* (fps). Selanjutnya, model *Convolutional Neural Network* (CNN) digunakan untuk mendeteksi objek pada setiap *frame*, dengan YOLOv8 menjadi pilihan untuk implementasi dan pengembangan sistem. Model ini diakui sebagai salah satu model deteksi objek paling canggih dan akurat.

Hasil *training* YOLOv8 yang baru memungkinkan deteksi 5 jenis kendaraan jalan yang ditargetkan. Gambar hasil akhir menunjukkan lokasi kendaraan dengan menggunakan *bounding box*, sehingga sistem dapat menghitung jumlah masing-masing jenis kendaraan dan total lalu lintas di persimpangan.



Gambar 2.4 Implementasi YOLOv8 untuk Mendeteksi Volume Kendaraan.

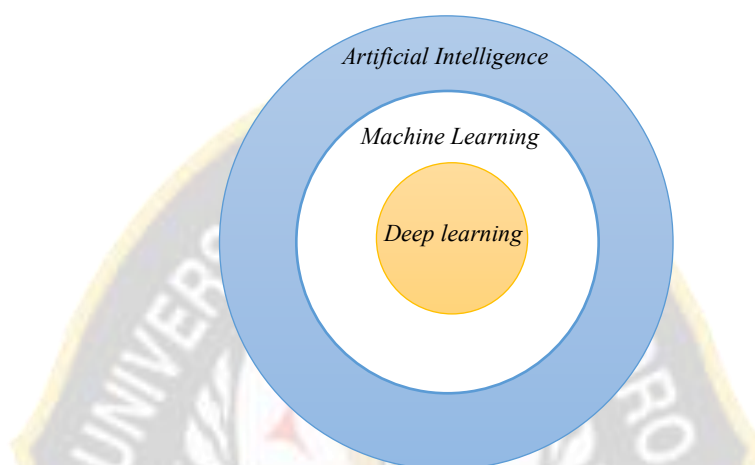
Hasil akhir dari proses implementasi YOLOv8 untuk mendeteksi volume kendaraan mencakup informasi statistik yang melibatkan analisis, perbandingan, dan evaluasi kepadatan lalu lintas di beberapa persimpangan di provinsi Thai Nguyen, Vietnam. Penelitian ini menggunakan aplikasi C-ThaiNguyen sebagai alat untuk mengumpulkan dan menganalisis data lalu lintas. Model YOLOv8, yang telah dilatih dengan tingkat akurasi tinggi menggunakan *Dataset Veh5*, menunjukkan kinerja yang sangat baik dengan nilai mAP50 sebesar 0.994 dan mAP50-95 sebesar 0.915.

2.3.8. Artificial Intelligence

Artificial Intelligence (AI) merupakan bidang ilmu komputer yang didedikasikan untuk pembuatan sistem yang melakukan tugas-tugas yang membutuhkan kecerdasan manusia. Secara luas, AI dapat diartikan sebagai penggabungan kecerdasan manusia ke dalam mesin. Dalam AI, mesin menyelesaikan tugas berdasarkan aturan dan algoritma yang ditetapkan.

AI adalah istilah umum untuk setiap program komputer yang memiliki sentuhan kecerdasan manusia, dengan lingkup yang mencakup *Machine Learning* dan *Deep learning*. Irisan antara *Artificial Intelligence*, *Machine*

Learning , dan *Deep learning* dapat dilihat pada Gambar 2.5 (Pesapane et al., 2018).



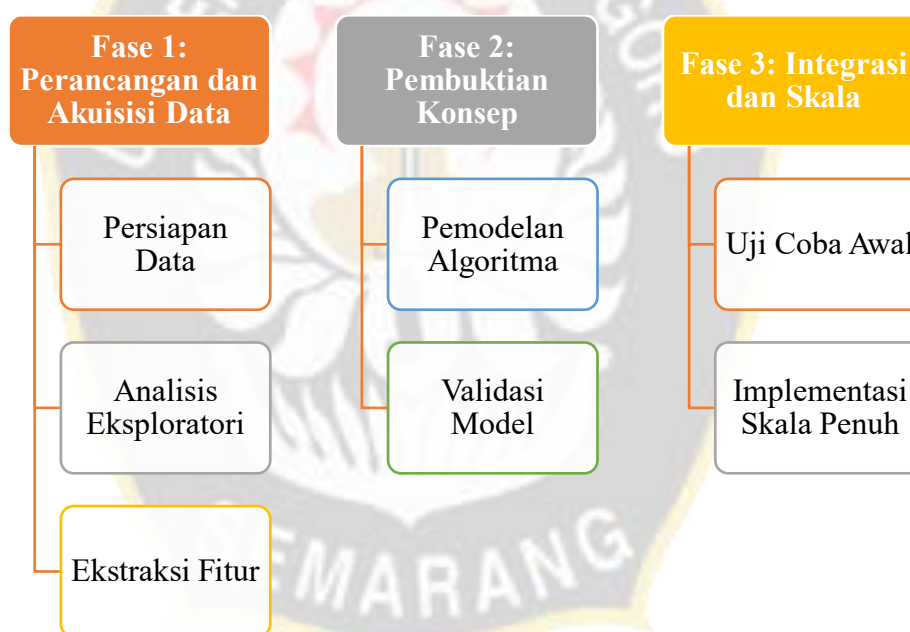
Gambar 2.5 Irisan *Artificial Intelligence*, *Machine Learning*, dan *Deep Learning*.

Artificial Intelligence (AI) dan penerapannya memiliki peran besar dalam bidang pencitraan *dermatologis*, mencakup berbagai aspek seperti akuisisi citra, pemrosesan citra, interpretasi hasil, pelaporan, dan perencanaan tindak lanjut (Jakhar dan Kaur, 2020). Pesatnya perkembangan *Artificial Intelligence* (AI) memberikan peluang yang belum pernah ada sebelumnya untuk meningkatkan kinerja berbagai industri dan bisnis, termasuk sektor transportasi.

Inovasi yang diperkenalkan oleh AI mencakup metode komputasi yang sangat canggih yang meniru cara kerja otak manusia. Penerapan AI di bidang transportasi ditujukan untuk mengatasi tantangan meningkatnya permintaan perjalanan, emisi CO₂, masalah keamanan, dan degradasi lingkungan. Mengingat ketersediaan sejumlah besar data kuantitatif dan kualitatif, serta AI di era digital ini, mengatasi masalah tersebut dengan cara yang lebih efisien dan efektif menjadi lebih bisa diterapkan.

Contoh metode AI yang diterapkan di bidang transportasi adalah *Artificial Neural Networks* (ANN), *Artificial Algorithms* (GA), *Simulated Annealing* (SA), *Artificial Immune System* (AIS), *Ant Colony Optimizer* (ACO), *Bee Colony Optimization* (BCO), dan *Fuzzy Logic Model* (FLM). Keberhasilan penerapan AI membutuhkan pemahaman yang baik tentang hubungan antara AI dan variabel data, serta karakteristik dari sistem transportasi itu sendiri.

Selain itu, penerapan AI sangat menjanjikan bagi otoritas transportasi untuk menentukan cara menggunakan teknologi ini untuk menciptakan peningkatan cepat dalam mengurangi kemacetan, membuat waktu tempuh lebih dapat diandalkan bagi pelanggan, serta meningkatkan ekonomi dan produktivitas aset vital mereka (Abduljabbar et al., 2019). Salah satu kemampuan AI adalah membuat prediksi. Untuk membangun model prediksi yang kuat, penting untuk melalui tiga fase, seperti yang terlihat pada Gambar 2.6 (Abduljabbar et al., 2019).



Gambar 2.6 Fase Pembangunan Model Prediksi dengan AI.

Pada Pada fase 1, semua sumber data harus dievaluasi dan digunakan untuk menggabungkan model lanjutan ke dalam industri. Fase ini sangat penting karena menjadi tahap awal untuk penilaian kinerja aset yang akan diimplementasikan. Penilaian yang tepat di fase ini membantu memahami kondisi awal dan merencanakan strategi perbaikan yang diperlukan.

Pada fase pembuktian konsep, banyak model dapat dipilih untuk penilaian kinerja yang lebih kritis. Tahap ini bertujuan untuk mengidentifikasi mode kegagalan yang mungkin terjadi serta waktu yang diperlukan untuk keseluruhan siklus hidup proyek. Proses ini memungkinkan pengujian berbagai

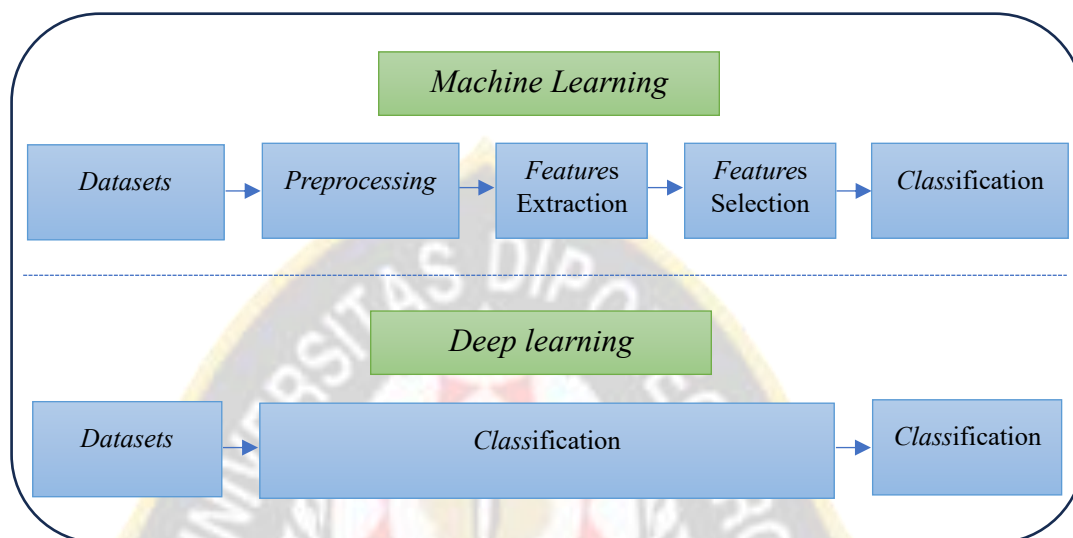
model untuk memastikan bahwa model yang dipilih dapat memberikan hasil yang optimal dalam jangka panjang.

Selain itu, fase terakhir dari diagram mewakili informasi prediksi *real-time* pada kinerja aset. Dalam fase ini, model yang digunakan harus terus diperbarui dan diskalakan untuk memberikan hasil terbaik. Oleh karena itu, pembaruan berkelanjutan dan adaptasi terhadap perubahan kondisi sangat penting untuk menjaga kinerja sistem.

2.3.9. Deep learning

Deep learning (DL) adalah salah satu cabang dari *Machine Learning* (ML) yang menggunakan *Neural Networks* dengan *multiple hidden layers* untuk mempelajari representasi data yang semakin kompleks (Kufel *et al.*, 2023). Dalam beberapa tahun terakhir, paradigma komputasi DL dianggap sebagai standar terbaik dalam komunitas *Machine Learning* (ML). Pendekatan ini telah memperoleh pengakuan luas karena kemampuannya dalam menangani tugas-tugas kognitif yang kompleks.

Selain itu, DL secara bertahap telah menjadi pendekatan komputasi yang paling banyak digunakan dalam bidang ML, sehingga mencapai hasil yang luar biasa pada sejumlah tugas kognitif kompleks. Hal ini menunjukkan bahwa DL tidak hanya menyamai, tetapi dalam beberapa kasus juga mengungguli kinerja manusia. Kemampuan DL untuk mempelajari pola-pola dalam data yang sangat besar membuatnya sangat efektif dalam berbagai aplikasi, termasuk pengenalan citra dan pemrosesan bahasa alami. Salah satu keuntungan utama dari DL adalah kemampuannya untuk memproses dan mengklasifikasikan jumlah data yang besar secara efisien. DL memungkinkan pembelajaran dan klasifikasi dilakukan dalam satu langkah yang terintegrasi. Perbedaan antara tahap DL dan ML dapat dilihat pada Gambar 2.7 (Alzubaidi *et al.*, 2021).



Gambar 2.7 Perbedaan *Machine Learning* dan *Deep learning*.

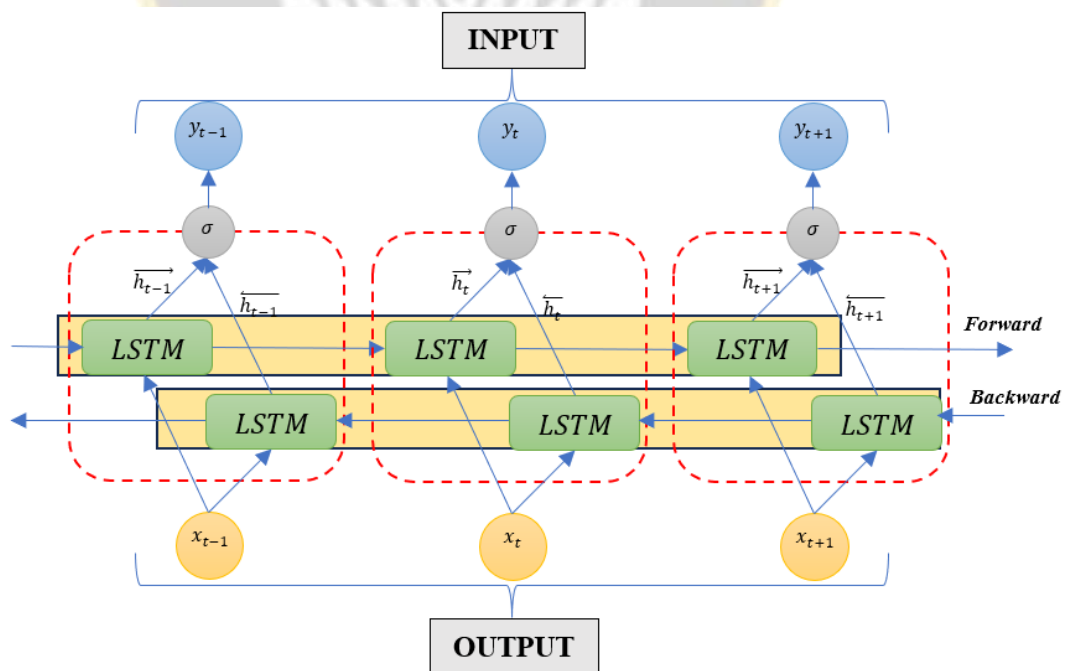
Teknik-teknik *Deep learning* (DL) dapat dikelompokkan ke dalam tiga kategori utama: *unsupervised*, *partially supervised* (*semi-supervised*), dan *supervised*. Setiap kategori ini memiliki metode dan aplikasi yang berbeda, yang disesuaikan dengan kebutuhan spesifik dalam proses pembelajaran. *Deep Reinforcement Learning* (DRL), yang juga dikenal sebagai *Reinforcement Learning* (RL), merupakan jenis teknik pembelajaran lainnya yang sebagian besar dianggap masuk ke dalam kategori teknik pembelajaran *supervised*.

Pada metode *unsupervised*, proses pembelajaran dapat dilakukan tanpa menggunakan label pada data. Teknik-teknik DL dalam kategori ini berhasil mengurangi kompleksitas dimensi non-linear dan pengelompokan data. Beberapa teknik DL yang sukses dalam hal ini termasuk *restricted Boltzmann machines*, *auto-encoders*, dan *Generative Adversarial Networks* (GANs), yang merupakan teknik terbaru dan sangat relevan dalam bidang ini.

Recurrent Neural Networks (RNNs), termasuk pendekatan GRUs dan LSTM, juga sering digunakan untuk pembelajaran *unsupervised* dalam berbagai aplikasi. Pendekatan ini sangat efektif dalam menangani data yang memiliki ketergantungan temporal atau urutan, seperti data deret waktu dan teks (Alzubaidi et al., 2021).

2.3.10. Metode Bi-LSTM

Bi-LSTM merupakan singkatan dari "*Bidirectional Long Short-term Model*" dan merupakan versi pengembangan dari LSTM (*Long Short-Term Memory*). Kelemahan utama dari LSTM adalah bahwa model ini hanya menggunakan nilai-nilai data yang telah tersedia sebelumnya untuk membuat prediksi, sementara data masa depan tidak berperan dalam pengambilan keputusan. Bi-LSTM mengatasi hal ini dengan meningkatkan hasil prediksi melalui penggunaan data baik dari masa lalu maupun masa depan, sehingga memungkinkan model untuk mengalirkan data ke arah maju maupun ke arah mundur. Metode ini membantu model prediktif dalam menggunakan informasi dari kedua arah untuk melatih model dan membuat keputusan yang lebih tepat. Baik observasi masa lalu maupun hasil masa depan disimpan dalam sel-sel memori, yang berperan dalam pengambilan keputusan yang lebih akurat (Wang *et al.*, 2022). Gambar 2.8 menunjukkan arsitektur Bi-LSTM (Chahal *et al.*, 2023).



Gambar 2.8 Arsitektur Bi-LSTM.

Arsitektur Bi-LSTM terdiri dari dua *Layer LSTM*, yaitu *Layer forward* dan *Layer backward*. *Layer forward* memproses input data dari arah depan, sedangkan *Layer backward* memproses input data dari arah belakang. *Output* dari kedua *Layer* kemudian digabungkan untuk menghasilkan *output* akhir. Adapun definisi dari masing-masing variabel adalah sebagai berikut:

x : Input data berupa vektor yang terdiri dari nilai-nilai dari data sekuensial.

$\longrightarrow_{h_{t-1}}$: menunjukkan bahwa *hidden state* dari *Layer backward* pada waktu t akan diteruskan ke *hidden state* dari *Layer backward* pada waktu $t+1$.

$\longrightarrow_{h_{t+1}}$: menunjukkan bahwa *hidden state* dari *Layer forward* pada waktu t akan diteruskan ke *hidden state* dari *Layer forward* pada waktu $t+1$.

$\longleftarrow_{h_{t-1}}$: menunjukkan bahwa *hidden state* dari *Layer backward* pada waktu t akan digunakan untuk menghasilkan *output* akhir. Hal ini dilakukan agar *output* akhir dapat dipengaruhi oleh informasi dari input data yang telah diproses pada waktu t .

$\longleftarrow_{h_{t+1}}$: menunjukkan bahwa *hidden state* dari *Layer forward* pada waktu t akan digunakan untuk menghasilkan *output* akhir. Hal ini dilakukan agar *output* akhir dapat dipengaruhi oleh informasi dari input data yang akan diproses pada waktu $t+1$.

y : *Output* akhir berupa vektor yang terdiri dari nilai-nilai dari hasil prediksi.

Nilai yang diprediksi pada waktu t direpresentasikan sebagai ' yt ' diberikan dalam Persamaan 2.3, dihitung dengan menggabungkan nilai data aliran maju dan mundur. Representasi matematis dari aliran data maju dan mundur model Bi-LSTM diberikan dalam Persamaan 2.1 dan 2.2 (Cai et al., 2019).

$$\overrightarrow{ht} = \sigma(W_{\overrightarrow{hx}}xt + W_{\overrightarrow{hh}}(\overrightarrow{h_{t-1}}) + b_{\overrightarrow{h}}) \quad (2.1)$$

Keterangan variabel:

- \overrightarrow{ht} : *hidden state* pada arah maju di waktu ke- t .

- x_t : input pada waktu ke- t (misalnya jumlah kendaraan tiap jenis pada menit ke- t).
- $W_{\overrightarrow{hx}}$: matriks bobot antara input dan *hidden layer* untuk arah maju.
- $W_{\overrightarrow{hh}}$: matriks bobot antara *hidden state* sebelumnya dan *hidden state* saat ini.
- $b_{\overrightarrow{h}}$: bias pada arah maju.
- σ : fungsi aktivasi sigmoid logistik, yang membatasi keluaran antara 0 dan 1 untuk menormalkan sinyal.

(2.2)

$$\overleftarrow{h}_t = \sigma(W_{\overleftarrow{hx}}x_t + W_{\overleftarrow{hh}}(\overleftarrow{h}_{t+1}) + b_{\overleftarrow{h}})$$

Keterangan variabel:

- \overleftarrow{h}_t : *hidden state* pada arah mundur di waktu ke- t .
- $W_{\overleftarrow{hx}}$: matriks bobot antara input dan *hidden layer* untuk arah mundur.
- $W_{\overleftarrow{hh}}$: matriks bobot antara *hidden state* berikutnya dan *hidden state* saat ini (karena arah mundur memproses urutan dari masa depan ke masa lalu).
- $b_{\overleftarrow{h}}$: bias pada arah mundur.
- σ : fungsi sigmoid yang sama seperti pada arah maju.

(2.3)

$$y_t = W_{\overrightarrow{yh}}\overrightarrow{h}_t + W_{\overleftarrow{yh}}\overleftarrow{h}_t + b_y$$

Keterangan variabel:

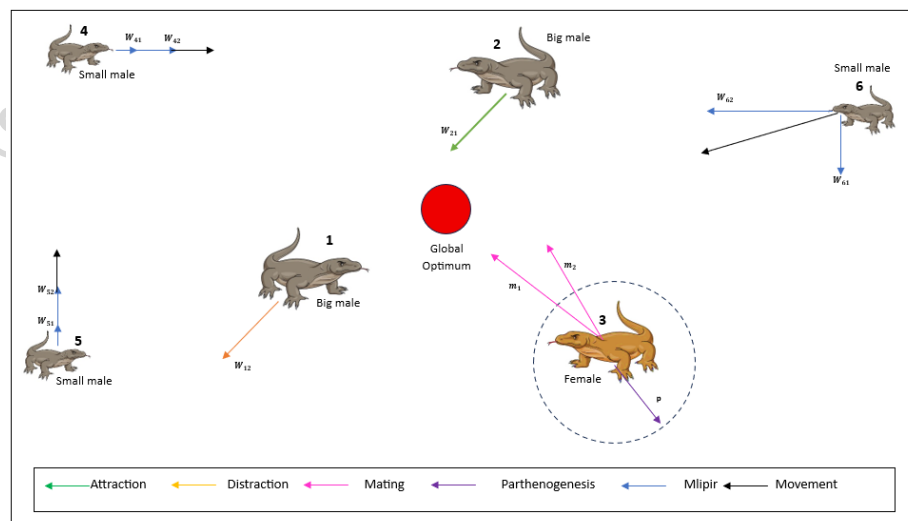
- y_t : nilai prediksi pada waktu ke- t , misalnya jumlah kendaraan hasil estimasi model.
- $W_{\overrightarrow{yh}}$: bobot dari *hidden state* arah maju ke lapisan output.
- $W_{\overleftarrow{yh}}$: bobot dari *hidden state* arah mundur ke lapisan output.
- b_y : bias pada lapisan output.
- $\overrightarrow{h}_t, \overleftarrow{h}_t$: *hidden state* hasil proses dua arah yang digabungkan secara linear untuk membentuk prediksi akhir.

2.3.11. Sliding window

Teknik *sliding window* digunakan untuk mengubah data deret waktu (*time-series*) ke dalam format yang dapat digunakan oleh model pembelajaran mesin, terutama dalam konteks prediksi nilai di masa depan berdasarkan data historis (C. Chen et al., 2022). Pendekatan ini menawarkan beberapa keunggulan penting. Pertama, model dapat menangkap pola atau tren dalam data tanpa menghilangkan urutan waktu. Kedua, penggunaan jendela yang saling tumpang tindih memperbanyak jumlah sampel pelatihan, sehingga model bisa belajar dari beragam pola data. Ketiga, pemanfaatan data historis mendukung peningkatan akurasi dalam prediksi deret waktu. Melalui teknik ini, diharapkan model Bi-LSTM mampu memahami pola arus lalu lintas dengan baik dan memperkirakan jumlah kendaraan secara efektif di masa mendatang.

2.3.12. Algoritma Komodo Mlipir

Algoritma Komodo Mlipir (KMA) merupakan algoritma metaheuristik yang diilhami oleh perilaku khas komodo dalam berburu dan mempertahankan hidup. Kata “*Mlipir*” yang berasal dari bahasa Jawa menggambarkan gerakan perlahan tapi pasti, yang merepresentasikan strategi komodo saat mendekati dan menangkap mangsa. Strategi ini diadaptasi sebagai pendekatan untuk menyelesaikan permasalahan optimasi yang kompleks (Suyanto et al., 2022b). Gambar 2.9 menggambarkan visualisasi dari konsep algoritma ini.



Gambar 2.9 Konsep Algoritma Komodo Mlipir.

KMA memadukan kemampuan eksplorasi dan eksploitasi secara unik. Eksplorasi dilakukan melalui pergerakan yang bersifat adaptif, memungkinkan algoritma menjelajahi ruang solusi secara menyeluruh. Sebaliknya, eksploitasi terfokus pada peningkatan kualitas solusi dengan memanfaatkan solusi terbaik yang telah diperoleh sebelumnya. Melalui mekanisme ini, KMA mampu menghindari jebakan pada solusi lokal dan secara konsisten mengarah pada solusi optimal (Kusuma et al., 2023).

Algoritma ini dirancang untuk memberikan hasil yang efisien dan stabil. Proses iteratif yang diterapkan tidak hanya meningkatkan kualitas solusi, tetapi juga lebih hemat sumber daya komputasi jika dibandingkan dengan algoritma lain seperti *Artificial Algorithm* (GA) atau *Particle Swarm Optimization* (PSO). Inovasi dalam pendekatan ini menjadikan KMA sangat berpotensi dalam menyelesaikan tantangan optimasi nyata, termasuk untuk kasus prediksi lalu lintas berbasis data waktu (*temporal*).

Keistimewaan KMA terletak pada fleksibilitasnya dalam menghadapi berbagai jenis permasalahan optimasi. Algoritma ini meniru proses seleksi alami seperti yang dilakukan komodo dalam memilih mangsa berdasarkan peluang keberhasilan. Selain itu, disisipkan elemen gangguan acak yang terkontrol untuk menjaga keragaman solusi tanpa kehilangan fokus terhadap pencarian solusi optimal. KMA sangat sesuai diterapkan dalam konteks *Machine Learning*, terutama dalam penyesuaian parameter model seperti LSTM, yang performanya sangat bergantung pada konfigurasi parameter yang akurat. Deskripsi langkah algoritmik dari KMA dapat ditemukan pada Algoritma 1.

Algoritma 1: Algoritma Komodo Mlipir

Hasil: k_{best} sebagai solusi optimum global

Tetapkan n, p , dan d sebagai jumlah individu Komodo, proporsi jantan besar, dan laju *Mlipir*, masing-masing.

Inisialisasi n individu dengan dimensi m

While	<p><i>StoppingCriterion</i> = <i>False</i> do</p> <ul style="list-style-type: none"> • Untuk setiap Komodo, hitung kualitasnya, lalu urutkan; • Berdasarkan peringkat mereka dan proporsi p, bagi populasi menjadi tiga grup: <ul style="list-style-type: none"> • q jantan besar dengan kualitas tertinggi, • 1 betina dengan kualitas sedang, • dan s jantan kecil dengan kualitas rendah menggunakan persamaan 2.4 dan 2.5. • Untuk setiap jantan besar, pindahkan menggunakan persamaan 2.7, dan pertahankan q jantan besar berkualitas tertinggi (posisi terbaik) untuk bertahan di generasi berikutnya. • Perbarui betina dengan cara: <ul style="list-style-type: none"> • kawin dengan jantan besar pemenang menggunakan persamaan 2.8, atau • melakukan partenogenesis menggunakan persamaan 2.10. • Untuk setiap jantan kecil, pindahkan menggunakan persamaan 2.11, dan pertahankan semua posisi baru mereka untuk bertahan di generasi berikutnya. • Perbarui ukuran populasi n menggunakan persamaan 2.13. • Pilih Komodo dengan kualitas tertinggi dari ketiga grup sebagai solusi terbaik sejauh ini, k_{best}
End	

Dalam implementasinya, setiap individu komodo diwakili oleh vektor *real* berdimensi m yang menunjukkan posisinya di ruang pencarian. Sebagai contoh, populasi yang terdiri atas enam komodo (k_1, k_2, \dots, k_6) di ruang berdimensi dua ($m = 2$) hanya memiliki atribut posisi tanpa komponen kecepatan. Hal ini membuat pergerakan mereka lebih fleksibel karena tidak bergantung pada inersia (Suyanto et al., 2022b). Populasi (n) ini dibagi ke dalam tiga kategori utama:

- *Big males*: individu dengan performa tertinggi,
- *Female*: individu dengan kualitas sedang,
- *Small males*: individu dengan performa terendah.

Proporsi masing-masing kelompok diatur oleh parameter p (umumnya $p = 0.5$), sesuai Persamaan 2.4 dan 2.5. Ketiga kelompok ini minimal memiliki dua anggota agar interaksi tetap dapat terjadi. Dengan q adalah jumlah *Big males*, s

adalah jumlah *Small males*, dan setidaknya setiap kelompok memiliki dua individu untuk interaksi.

$$q = \lfloor (p - 1) n \rfloor, \quad (2.4)$$

$$s = n - q, \quad (2.5)$$

Keterangan variabel:

- q : jumlah *Big males*
- s : jumlah *Small males*
- n : adalah ukuran total populasi,
- simbol $\lfloor \rfloor$ menunjukkan pembulatan ke bawah (floor).

Setiap kelompok minimal berisi dua individu agar proses interaksi antar-komodo tetap dapat terjadi.

Big males saling memengaruhi dengan mekanisme tarik dan pengalihan berdasarkan nilai kualitas. Individu *big male* yang kualitasnya rendah akan bergerak menuju yang lebih tinggi, sementara individu terbaik dapat mengalami tarikan atau gangguan acak dengan probabilitas 0,5. Gerakan ini dijelaskan melalui Persamaan 2.6 dan 2.7. Strategi ini dikenal sebagai *High Exploitation Low Exploration* (HILE) yang memungkinkan pencarian intensif di sekitar kandidat solusi terbaik.

$$w_{ij} = \begin{cases} r_1(k_j - k_i), & \text{if } f(k_j) < f(k_i) \text{ or } r_2 < 0.5 \\ r_1(k_i - k_j), & \text{jika tidak} \end{cases} \quad (2.6)$$

$$k'_i = k_i + \sum_{j=1}^q w_{ij}, \text{ dimana } j \neq i, \quad (2.7)$$

Keterangan variabel:

- k_i, k_j : posisi individu ke- i dan ke- j .
- r_1, r_2 : bilangan acak dalam $[0,1]$,
- $f(k)$: fungsi *fitness* (nilai kualitas solusi),
- w_{ij} : vektor perpindahan,
- k'_i : posisi baru hasil pergerakan.

Untuk *Female*, aksi eksploitasi dan eksplorasi masing-masing memiliki probabilitas 0,5. Dalam eksploitasi, *Female* akan melakukan perkawinan dengan *big male* terbaik, menghasilkan dua anak (Persamaan 2.8).

$$k'_{il} = r k_{il} + (1 - r) k_{jl}, \quad (2.8)$$

$$k'_{jl} = r k_{jl} + (1 - r) k_{il},$$

Sedangkan dalam eksplorasi, *Female* menjalankan partenogenesis, yakni menghasilkan keturunan sendiri dengan menambahkan gangguan acak kecil (Persamaan 2.9 dan 2.10). Dengan $\alpha = 0.1$ membatasi variasi dalam radius 10% dari ruang pencarian.

$$(k_{i1}, k_{i2}, \dots, k_{im}) \rightarrow (k'_{i1}, k'_{i2}, \dots, k'_{im}), \quad (2.9)$$

$$k'_{ij} = k_{ij} + (2r - 1)\alpha |ub_j - lb_j|, \quad (2.10)$$

Keterangan variabel Eksploitasi dan Eksplorasi:

- r, rl : bilangan acak $[0, 1]$,
- $\alpha = 0,1$: menentukan intensitas gangguan (10% dari rentang ruang pencarian),
- ub_j dan lb_j : batas atas dan bawah untuk dimensi ke- j ,
- k'_{ij} : posisi baru *Female* setelah gangguan acak.

Small males menerapkan strategi *Mlipir*, yaitu mengikuti *Big males* hanya pada sebagian dimensi tertentu secara acak, sesuai dengan parameter (d) (Persamaan 2.11 dan 2.12). Strategi ini disebut sebagai *Low Exploitation High Exploration* (LIHE), yang dirancang untuk mendukung eksplorasi menyeluruh di ruang pencarian.

$$w_{ij} = \begin{cases} \sum_{l=1}^m r_l (k_{jl} - k_{il}), & \text{jika } r_2 < d \\ 0, & \text{lainnya} \end{cases} \quad (2.11)$$

$$k'_i = k_i + \sum_{j=1}^q w_{ij}, \text{ dimana } j \neq i, \quad (2.12)$$

Keterangan variabel:

- d : probabilitas mengikuti dimensi tertentu dari *Big males*,

- w_{ij} : kontribusi perpindahan per dimensi,
- strategi ini membantu menjaga keberagaman populasi dan mencegah konvergensi prematur.

Ukuran populasi (n) tidak statis, tetapi diatur secara adaptif selama proses *iterasi*. Jika terjadi peningkatan *fitness* secara berurutan, maka jumlah individu dikurangi. Sebaliknya, jika tidak ada perbaikan, maka jumlah individu ditambah. Mekanisme ini dirumuskan melalui Persamaan 2.13. dengan a adalah jumlah individu yang dihapus atau ditambahkan.

$$n' = \begin{cases} n - a, & \text{if } \delta f_1 > 0 \text{ dan } \delta f_2 > 0 \\ n + a, & \text{if } \delta f_1 = 0 \text{ and } \delta f_2 = 0 \end{cases} \quad (2.13)$$

Keterangan variabel:

- a : jumlah individu yang dihapus atau ditambahkan,
- $\delta f_1, \delta f_2$: perubahan nilai *fitness* pada dua iterasi terakhir.

Gabungan dari gerakan *Big males*, mekanisme reproduksi *Female*, dan strategi *Mlipir* dari *Small males* memberikan keseimbangan optimal antara eksplorasi dan eksploitasi. Penyesuaian ukuran populasi juga berperan penting dalam meningkatkan kemampuan adaptif algoritma ini untuk menangani berbagai tantangan optimasi secara efektif.