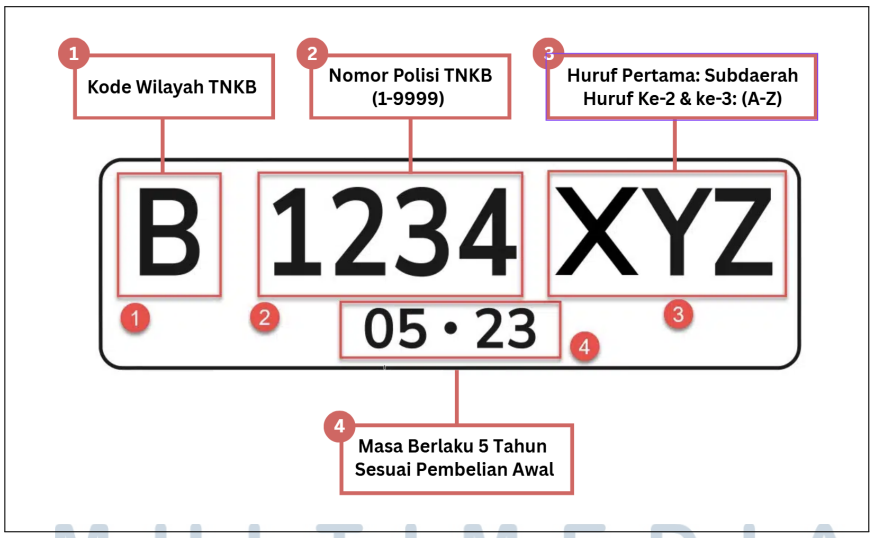
BAB 2
LANDASAN TEORI

2.1 Tinjauan Teori

2.1.1 Tanda Nomor Kendaraan Bermotor (TNKB)

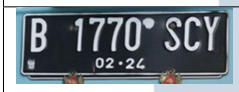
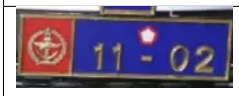
Tanda Nomor Kendaraan Bermotor (TNKB) merupakan Plat nomor kendaraan di Indonesia yang secara resmi dikenal sebagai memiliki format standar yang diatur oleh Korps Lalu Lintas (Korlantas) Polri seperti pada gambar 2.1 yang menggambarkan karakteristik plat nomor di Indonesia [5]. Struktur tersebut secara umum diawali dengan kode wilayah, yang terdiri dari satu atau dua huruf kapital untuk mengidentifikasi area registrasi kendaraan. Komponen tersebut kemudian diikuti oleh serangkaian nomor registrasi polisi yang terdiri dari satu hingga empat digit angka. Bagian akhir dari seri utama adalah kode seri akhir atau sub-daerah, yang merupakan kombinasi satu hingga tiga huruf. Selain seri utama tersebut, secara visual plat nomor juga mencantumkan informasi masa berlaku dalam format bulan dan tahun yang tercetak pada bagian bawah.



Gambar 2.1. Karakteristik Format Plat Nomor Kendaraan Pribadi di Indonesia

TNKB di Indonesia tidak hanya digunakan untuk kendaraan pribadi, tetapi memiliki variasi warna dan fungsi sesuai jenis kendaraan, sebagaimana dirangkum pada Tabel 2.1. Sistem identifikasi ini memungkinkan klasifikasi kendaraan dinas, listrik, umum, maupun kendaraan dengan status diplomatik atau kenegaraan khusus seperti DPR atau kedutaan besar.

Tabel 2.1. Gambar dan Jenis TNKB di Indonesia Berdasarkan Fungsi Kendaraan

Gambar TNKB	Jenis TNKB	Fungsi Kendaraan
	TNKB Hitam	Kendaraan mobil/motor pribadi lama
	TNKB Putih	Kendaraan mobil/motor pribadi baru
	TNKB Listrik	Kendaraan mobil/motor bertenaga listrik
	TNKB Umum	Kendaraan umum, truk, bus dan taksi
	TNKB Merah	Kendaraan instansi pemerintah
	TNKB Hijau	Kendaraan kawasan perdagangan bebas
	TNKB Diplomatik	Kendaraan kedutaan besar negara asing
	TNKB Pejabat	Kendaraan dengan keistimewaan khusus
	TNKB Polisi	Kendaraan milik Polisi Republik Indonesia
	TNKB TNI AD	Kendaraan milik TNI Angkatan Darat
	TNKB TNI AL	Kendaraan milik TNI Angkatan Laut
	TNKB TNI AU	Kendaraan milik TNI Angkatan Udara

2.1.2 Computer Vision

Computer Vision adalah sebuah bidang interdisipliner dalam ilmu komputer dan kecerdasan buatan (*Artificial Intelligence*) yang bertujuan untuk memungkinkan mesin ”melihat”, menginterpretasi, dan memahami informasi visual dari dunia nyata, seperti gambar dan video. Tujuan fundamental dari *computer vision* adalah untuk mengotomatisasi tugas-tugas yang dapat dilakukan oleh sistem visual manusia. Untuk mencapai tujuan tersebut, *computer vision* memanfaatkan berbagai teori dan metode dari bidang lain, termasuk pengolahan sinyal, aljabar linear, probabilitas, dan optimisasi.Secara umum, tugas-tugas dalam ranah *computer vision* dapat diklasifikasikan ke dalam beberapa kategori utama yang mereplikasi kemampuan persepsi manusia [6]. Tugas-tugas tersebut antara lain *image classification* untuk mengenali kategori utama dari sebuah gambar, *object detection* untuk melokalisasi dan mengidentifikasi objek spesifik di dalam gambar, serta *image segmentation* untuk mempartisi gambar menjadi segmen-segmen piksel yang sesuai dengan setiap objek.

A Deep Learning

Deep learning adalah cabang fundamental dari *machine learning* yang didasarkan pada arsitektur Jaringan Syaraf Tiruan (*Artificial Neural Networks*) dengan banyak lapisan tersembunyi (*hidden layers*). Struktur berlapis yang dalam inilah yang memungkinkan model untuk secara hierarkis mempelajari representasi atau fitur dari data secara otomatis, mulai dari fitur tingkat rendah (seperti tepi dan warna) pada lapisan awal hingga fitur tingkat tinggi yang lebih kompleks dan abstrak (seperti bentuk objek) pada lapisan yang lebih dalam [7]. Kemampuan untuk belajar fitur secara mandiri dari data mentah inilah yang membuat *deep learning* sangat dominan dan berhasil mencapai performa canggih dalam berbagai tugas *computer vision*, termasuk deteksi objek dan segmentasi citra yang menjadi inti dari penelitian ini.

2.1.3 Zero-shot

Zero-shot learning merupakan pendekatan dalam bidang deep learning yang memungkinkan model menyelesaikan tugas baru tanpa melalui proses pelatihan ulang secara khusus terhadap data target. Pendekatan tersebut berada dalam ranah konsep *transfer learning*, yaitu strategi pemanfaatan model yang telah dilatih sebelumnya pada dataset berskala besar dan bersifat umum (*pre-trained model*) untuk menyelesaikan permasalahan pada domain yang berbeda secara langsung. Dalam konteks *zero-shot learning*, model yang telah memiliki kemampuan umum, seperti mengenali fitur visual dasar berupa tepi, tekstur, maupun bentuk, digunakan untuk melakukan prediksi atau segmentasi pada data yang belum pernah dipelajari secara eksplisit. Metode ini sangat bermanfaat dalam situasi yang tidak memungkinkan pengumpulan atau pelabelan data tambahan untuk pelatihan ulang, serta dapat menghemat sumber daya komputasi secara signifikan [8].

2.1.4 YOLOv8

YOLOv8 (*You Only Look Once version 8*) adalah model deteksi objek dari Ultralytics yang dirancang untuk kinerja *real-time* dengan akurasi tinggi. Dibandingkan versi sebelumnya, YOLOv8 memiliki arsitektur yang lebih efisien, fleksibilitas dalam tugas visi komputer seperti deteksi, segmentasi, dan klasifikasi, serta kemudahan dalam pelatihan dan implementasi [9]. Model tersebut banyak digunakan dalam berbagai aplikasi, seperti sistem pengawasan, pengenalan plat nomor kendaraan, dan analisis video. Dengan kemampuannya yang cepat dan akurat, YOLOv8 menjadi pilihan unggul dalam pemrosesan citra dan video berbasis kecerdasan buatan. Table 4.4 menunjukan perbandingan arsitektur pada versi YOLOv8 dengan arsitektur pada penelitian terdahulu.

Tabel 2.2. Perbandingan Arsitektur YOLOv3, YOLOv5, dan YOLOv8

Fitur	**YOLOv3**	**YOLOv5**	**YOLOv8**
Tahun Rilis	2018	2020	2023
Framework	Darknet (C)	PyTorch	PyTorch (Ultralytics)
Arsitektur	Darknet-53	CSP-Darknet	C2f Module (*anchor-free*)
Anchor-Free	Tidak	Tidak	Ya
Dukungan Segmentasi	Tidak	Ya (YOLOv5-seg)	Ya (*native*)
Ukuran Model (*Medium*)	∼240 MB	∼85 MB	∼50 MB
mAP@0.5 (COCO)	∼33%	∼45–50%	∼53%+
Dukungan Format Ekspor	ONNX, CoreML, TFLite	ONNX, TorchScript	ONNX, TorchScript, TFLite, CoreML

2.1.5 BYTETrack

BYTETrack adalah algoritma pelacakan multi-objek (*Multi-Object Tracking*) yang efisien dan akurat, digunakan untuk mengidentifikasi serta melacak objek yang sama dari satu *frame* ke *frame* berikutnya dalam video [10]. Algoritma tersebut memanfaatkan strategi pemadanan berbasis Intersection over Union (IoU) dan Kalman Filter untuk memperkirakan pergerakan objek, sehingga dapat mempertahankan identitas objek meskipun terjadi tumpang tindih (*occlusion*) atau perubahan posisi. BYTETrack unggul dalam kecepatan pemrosesan dan ketepatan pelacakan, menjadikannya pilihan ideal untuk aplikasi seperti deteksi dan pelacakan plat nomor kendaraan Indonesia [11].

2.1.6 Segment Anything Model 2 (SAM2)

Segment Anything Model 2 (SAM2) merupakan sebuah model segmentasi gambar yang dirancang untuk mengidentifikasi dan memisahkan objek dalam gambar dengan tingkat akurasi yang tinggi [12]. SAM2 menggunakan pendekatan berbasis pembelajaran mendalam yang memanfaatkan arsitektur *neural network* untuk melakukan segmentasi secara otomatis. Model tersebut mampu mengenali berbagai objek dalam satu gambar tanpa memerlukan anotasi khusus untuk setiap objek, sehingga memberikan fleksibilitas yang lebih besar dalam aplikasi segmentasi. Cara kerja SAM2 berfokus pada pemrosesan citra yang mendalam, di mana model tersebut dilatih menggunakan dataset yang sangat besar dan beragam. Proses pelatihan melibatkan penggunaan teknik augmentasi data untuk meningkatkan variasi dalam dataset, sehingga model dapat belajar untuk mengenali objek dalam berbagai kondisi pencahayaan, sudut pandang, dan latar belakang. Setelah pelatihan, SAM2 dapat melakukan segmentasi dengan memberikan input gambar dan menghasilkan peta segmentasi yang menunjukkan area yang terpisah untuk setiap objek yang terdeteksi.

2.1.7 Segmentasi

Segmentasi merupakan proses dalam pengolahan citra digital yang bertujuan untuk membagi suatu gambar menjadi beberapa bagian atau objek yang lebih bermakna. Proses tersebut dilakukan dengan cara mengelompokkan piksel berdasarkan karakteristik tertentu, seperti warna, intensitas, tekstur, atau bentuk, sehingga dapat mempermudah analisis dan pemrosesan lanjutan.

2.1.8 Streamlit

Streamlit adalah *framework open-source* berbasis Python yang digunakan untuk membangun antarmuka pengguna (UI) interaktif dalam aplikasi data secara cepat dan sederhana. *Framework* tersebut sangat efisien dalam pengolahan dan penyajian data secara *real-time*, sehingga banyak digunakan dalam penelitian, analisis data, dan *deployment* model kecerdasan buatan. Keunggulan utama Streamlit adalah kemampuannya untuk mengintegrasikan berbagai pustaka data sains serta mendukung pengembangan aplikasi berbasis web tanpa memerlukan konfigurasi *backend* yang kompleks.

2.2 Penelitian Terdahulu

Dalam penelitian ini, kajian terhadap penelitian terdahulu dilakukan untuk memahami metode yang telah digunakan dalam deteksi dan pengaburan plat nomor kendaraan.

2.2.1 Chan et al. (2020)

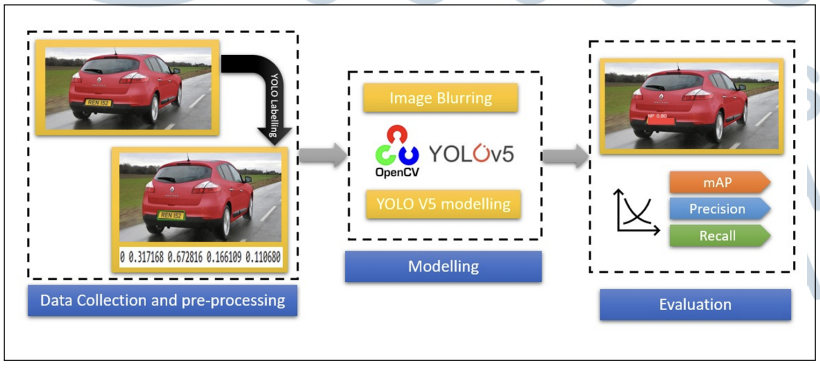
Penelitian yang dilakukan oleh Chan et al. (2020) bertujuan untuk mengembangkan sistem deteksi dan pengaburan plat nomor kendaraan secara *real-time* menggunakan teknik *deep learning*. Penelitian tersebut berfokus pada pembuatan *dataset* plat nomor kendaraan di Uni Eropa yang disebut *THI License Plate Dataset* (TLPD), yang terdiri dari lebih dari 17.000 gambar kendaraan dan 18.000 plat nomor yang dilabeli. Dataset tersebut mencakup berbagai kondisi lingkungan seperti siang, malam, dan cuaca salju, serta variasi sudut dan jarak kamera. Selain itu, penelitian tersebut memperkenalkan alat pelabelan cepat bernama FastLabel untuk mempermudah proses anotasi data. Dua model deteksi plat nomor, yaitu Fast-Yolo dan Tiny-Yolov3, dilatih menggunakan dataset tersebut dan menunjukkan kinerja yang baik dengan presisi dan recall mencapai 93% pada kondisi siang dan malam, serta 87% pada kondisi salju. Penelitian tersebut juga memvalidasi kemampuan model dalam mendeteksi plat nomor secara *real-time* dengan kecepatan 30 *frame* per detik.

Model yang digunakan dalam penelitian tersebut adalah Fast-Yolo dan Tiny-Yolov3, yang merupakan varian dari arsitektur YOLO (*You Only Look Once*). Fast-Yolo menggunakan *transfer learning* dengan bobot awal dari model UFPR, sedangkan Tiny-Yolov3 menggunakan bobot awal dari model Tiny-Yolov3 asli. Proses pelatihan dilakukan selama 30.000 *epoch* dengan *batch size* 128 untuk Fast-Yolo dan 512 untuk Tiny-Yolov3. Ukuran gambar yang digunakan adalah 416x416 piksel untuk Fast-Yolo dan 608x608 piksel untuk Tiny-Yolov3. Model tersebut dioptimalkan dengan *fine-tuning* untuk meningkatkan akurasi deteksi. Arsitektur YOLO terdiri dari tiga komponen utama: *backbone* untuk ekstraksi fitur, *neck* untuk agregasi fitur, dan head untuk prediksi *bounding-box*. Hasil penelitian menunjukkan bahwa Tiny-Yolov3 memiliki performa yang lebih baik dibandingkan Fast-Yolo, terutama dalam mendeteksi plat nomor pada berbagai kondisi lingkungan.

2.2.2 Shringarpure, Darshan Vijay (2023)

Penelitian yang dilakukan oleh Darshan Vijay Shringarpure (2023) berfokus pada deteksi dan pengaburan plat nomor kendaraan menggunakan teknik *deep learning* dengan model YOLOv5. Tujuan utama penelitian adalah menjaga privasi data dengan mengaburkan plat nomor kendaraan yang terdeteksi dalam gambar atau video. Penelitian tersebut menggunakan dua dataset, yaitu dataset publik dari Kaggle yang berisi 432 gambar plat nomor kendaraan dan dataset UFPR yang terdiri dari 4500 gambar kendaraan dengan variasi kondisi cuaca dan sudut pandang. Hasil penelitian menunjukkan akurasi deteksi sebesar 99% dan akurasi pengaburan 100% untuk gambar, serta 94% untuk video dengan kecepatan 30 frame per detik. Penelitian ini juga mengembangkan metode untuk memverifikasi keakuratan pengaburan dengan memastikan bahwa plat nomor yang telah diaburkan tidak dapat lagi dideteksi oleh model.

Model yang digunakan dalam penelitian tersebut adalah YOLOv5, yang merupakan salah satu algoritma deteksi objek tercepat dan paling stabil. Arsitektur YOLOv5 terdiri dari tiga komponen utama: *Backbone* (CSPDarknet) untuk ekstraksi fitur, *Neck* (PANet) untuk agregasi fitur, dan *Head* untuk prediksi *bounding-box*. Proses pelatihan model dilakukan dengan 25 *epoch*, *batch size* 32, dan ukuran gambar 1980x1980 piksel untuk meningkatkan akurasi deteksi pada gambar beresolusi tinggi. Setelah deteksi plat nomor berhasil, pengaburan dilakukan dengan memodifikasi fungsi YOLOv5, khususnya pada metode CV2.rectangle yang mengisi *bounding-box* dengan warna yang sama dengan batasnya, sehingga plat nomor benar-benar tersamarkan. Verifikasi pengaburan dilakukan dengan memastikan bahwa plat nomor yang telah diaburkan tidak lagi terdeteksi oleh model.

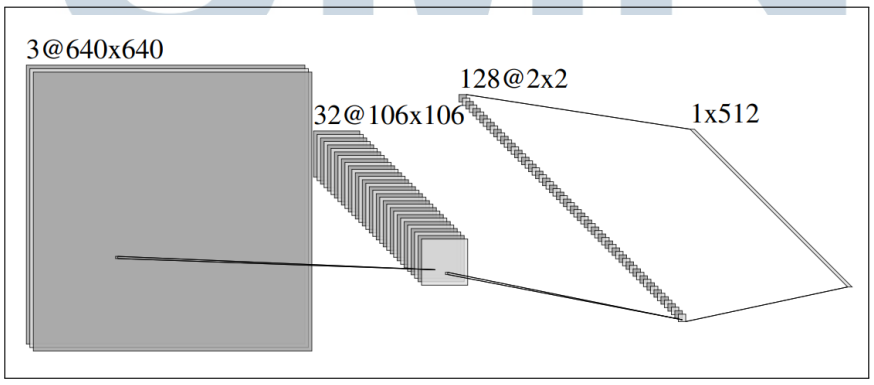


Gambar 2.2. Arsitektur YOLOv5 yang digunakan pada penelitian oleh Shringarpure
Sumber: [13]

2.2.3 Sarin P, Wei P (2024)

Penelitian oleh Sarin dan Wei (2024) [14] dalam proyek akademis CS231n *Deep Learning* di Universitas Stanford menyajikan pendekatan yang berbeda dalam menangani tantangan anonimisasi plat nomor. Fokus utama dari penelitian tersebut adalah untuk mengembangkan sistem segmentasi plat nomor yang berkinerja tinggi dengan skenario *low-data*, yaitu melatih model dari awal (*from scratch*) menggunakan dataset yang jumlahnya kurang dari 1% dari yang biasa digunakan untuk melatih model-model canggih seperti YOLO. Pendekatan tersebut secara fundamental berbeda dengan strategi *fine-tuning* yang mengandalkan model pra-terlatih (*pre-trained*). Penelitian tersebut juga secara kritis membahas isu etis terkait penggunaan data dan sistem pengawasan dalam ranah *computer vision*.

Metodologi utama yang digunakan bukanlah *transfer learning* dari model YOLO, melainkan perancangan sebuah arsitektur *Convolutional Neural Network* (CNN) kustom yang relatif sederhana. Seperti diilustrasikan pada Gambar 2.3, arsitektur *baseline* mereka menggunakan serangkaian lapisan konvolusi, ReLU, dan *max-pooling* untuk mengubah citra masukan berukuran 640x640 menjadi sebuah representasi fitur (*embedding*) berdimensi 512. *Embedding* tersebut kemudian dilewatkan ke sebuah lapisan linear dan fungsi aktivasi *softmax* untuk menghasilkan prediksi probabilitas segmentasi untuk setiap piksel. Untuk mengatasi masalah ketidakseimbangan kelas, mereka juga mengimplementasikan sebuah fungsi *loss* kustom yang memberikan bobot penalti lebih besar pada kesalahan segmentasi di area plat. Pendekatan ini menunjukkan bahwa dengan perancangan arsitektur dan fungsi *loss* yang cermat, performa yang kompetitif dapat dicapai bahkan dengan sumber daya data yang sangat terbatas.



Gambar 2.3. Diagram Arsitektur CNN Kustom pada Penelitian Sarin dan Wei
Sumber: [14]