

BAB II

LANDASAN TEORI

2.1 Teori Umum

2.1.1 Transportasi Umum

Transportasi umum di Indonesia adalah sarana penting bagi masyarakat untuk melakukan perjalanan dari satu tempat ke tempat lain. Dalam beberapa tahun terakhir, pemerintah Indonesia telah melakukan upaya untuk meningkatkan sistem transportasi umum di seluruh wilayah Indonesia. Kementerian Perhubungan terus menggenjot pendanaan kreatif (*creative financing*) pembangunan infrastruktur transportasi nasional melalui berbagai skema kerja sama, dengan berbagai pihak baik di dalam negeri maupun luar negeri. Pendanaan kreatif non-APBN melalui skema Kerja Sama Pemerintah dan Badan Usaha (KPBU) terus dilakukan untuk mempercepat pembangunan sarana dan prasarana transportasi. Namun, masih terdapat selisih yang signifikan antara kebutuhan investasi infrastruktur transportasi dengan kemampuan pendanaan APBN. Kebutuhan investasi tersebut mencapai Rp. 1.288 triliun, sementara sumber pendanaan APBN hanya dapat menutupi 40-65% dari total tersebut per tahunnya [22]. Berikut ini adalah beberapa tinjauan teori tentang transportasi umum di Indonesia:

A. Tingkat Ketergantungan Masyarakat Terhadap Transportasi Umum

Transportasi umum menjadi sarana penting bagi masyarakat Indonesia untuk melakukan aktivitas sehari-hari, seperti bekerja, rekreasi, berbelanja, dan lain-lain. Tingkat ketergantungan masyarakat terhadap transportasi umum dapat meningkatkan permintaan akan layanan transportasi umum yang berkualitas dan dapat mengakomodasi kebutuhan mobilitas masyarakat.

B. Solusi Untuk Meningkatkan Kualitas Transportasi Umum

Pemerintah Indonesia dapat meningkatkan kualitas transportasi umum dengan meningkatkan infrastruktur dan fasilitas, memperkuat regulasi yang

ada, dan memberikan insentif untuk investasi swasta di sektor transportasi. Selain itu, penggunaan teknologi dan inovasi dalam transportasi umum, seperti perhitungan jumlah penumpang, juga dapat menjadi solusi untuk meningkatkan kualitas *safety* transportasi umum di Indonesia.

2.1.2 Aspek Keselamatan Pada Transportasi Umum

Aspek keselamatan pada transportasi umum merujuk kepada berbagai faktor yang mempengaruhi keamanan penumpang dan pengguna jalan dalam mengakses dan menggunakan sistem transportasi umum. Kecelakaan lalu lintas disebabkan oleh tiga faktor yang berbeda, yaitu manusia, kendaraan, serta kondisi jalan dan lingkungan [23]. Terdapat beberapa faktor keselamatan yang dapat dipertimbangkan dalam transportasi umum meliputi kendaraan, sumber daya manusia, infrastruktur, pemeliharaan dan pengawasan, kesadaran dan ketaatan pengguna, sistem pengendalian operasional, dan teknologi.

a. Kendaraan

Kendaraan yang dijadikan sebagai transportasi umum harus terawat dan sesuai dengan standar keamanan dan kelayakan teknis. Semua komponen pada kendaraan seperti rem, ban, lampu, dan hal lainnya harus berfungsi dengan baik.

b. Pemeliharaan dan Pengawasan

Kendaraan harus diawasi secara teratur dan dilakukan pemeliharaan secara rutin. Pemerintah juga harus melakukan pengawasan dan penindakan terhadap pelanggaran yang dilakukan oleh oknum yang dapat menghiraukan keselamatan penumpang pada kendaraan tersebut.

c. Kesadaran dan Ketaatan Pengguna

Penumpang atau pengguna transportasi umum wajib memiliki kesadaran dan ketaatan dalam mematuhi aturan keselamatan, seperti menggunakan sabuk pengaman, tidak merokok, dan tidak membuang sampah sembarangan.

d. Sistem Pengendalian Operasional

Diperlukan pengendalian operasional yang ketat dan terstruktur agar tercipta koordinasi yang baik antara pihak yang terlibat dalam penyelenggaraan transportasi umum.

e. Teknologi

Penggunaan teknologi, seperti sistem informasi dan komunikasi, dapat meningkatkan keselamatan transportasi umum dengan memantau pergerakan kendaraan dan memberikan informasi real-time kepada pengguna.

2.1.3 *Over capacity* Penumpang Pada Transportasi Umum

Over capacity penumpang pada transportasi umum adalah kondisi di mana jumlah penumpang yang menggunakan sarana transportasi umum melebihi kapasitas maksimal yang direkomendasikan atau ditetapkan oleh pihak pengelola transportasi. *Over capacity* ini dapat terjadi karena beberapa faktor, seperti kepadatan penduduk yang tinggi di daerah tertentu, kurangnya opsi transportasi lain, dan kenaikan permintaan di jam sibuk. *Over capacity* penumpang dapat menyebabkan beberapa dampak negatif, seperti keamanan yang menurun, kenyamanan yang berkurang, penurunan efisiensi transportasi, dan bahkan kemungkinan kecelakaan. Dalam rangka pengoperasian bus Transjakarta pada masa pandemi, penumpang masih saja menghiraukan kenyamanan dan keselamatan mereka dengan memaksa masuk kedalam bus sehingga menimbulkan desakan dan membuat penumpang lain tidak merasa aman dan perilaku tersebut dapat mengurangi penerapan social distancing yang sangat dianjurkan pada masa itu [24]. Oleh karena itu, para pengelola

transportasi perlu mengidentifikasi dan menangani *over capacity* penumpang secepat mungkin untuk menghindari dampak negatif tersebut.

Pengelola transportasi dapat menggunakan teknologi untuk membantu mengelola transportasi dengan lebih efektif dan efisien dalam mengurangi *over capacity* penumpang. Contoh penggunaan *Machine Learning* dalam transportasi adalah membuat sebuah sistem untuk menghitung jumlah penumpang yang masuk dan keluar pada kendaraan transportasi. Fungsinya adalah agar tidak terjadinya *over capacity* pada kendaraan dan data tentang jumlah penumpang sehari-harinya menjadi hal yang dapat dijadikan pembuatan keputusan bagi pengelola transportasi. Dalam rangka mengatasi *over capacity* penumpang pada transportasi umum, penting bagi pengelola transportasi untuk memperhatikan dan memahami kebutuhan penumpang, serta terus melakukan penilaian dan evaluasi terhadap sistem transportasi yang ada agar dapat meningkatkan kualitas transportasi umum.

2.1.4 Object detection

Object detection adalah teknik yang digunakan untuk mengidentifikasi objek dalam gambar atau video dan menempatkan kotak batas (*bounding box*) di sekitar objek tersebut. Teknik *object detection* yang sering digunakan saat ini adalah menggunakan *deep learning*. *Deep learning* adalah sub-bidang dari *machine learning* yang menggunakan *neural network* untuk mempelajari pola dalam data. Jaringan saraf tiruan digunakan dalam *deep learning* karena kemampuannya untuk mempelajari fitur dari data secara otomatis dan menghasilkan output yang akurat. Dalam *object detection* terdapat 2 hal yang harus dilakukan yaitu: *object detection* dan klasifikasi objek. *Object detection* adalah proses untuk menemukan objek dalam gambar atau video. Klasifikasi objek adalah proses untuk mengidentifikasi jenis objek yang ditemukan. Pendeteksian sebuah objek perlu melakukan *labelling* dari beberapa *class* dataset, dengan melakukan pengklasifikasian objek dengan memanfaatkan model *Faster RCNN Inception v2* dan memanfaatkan COCO dataset [25].

Object detection dapat diimplementasikan dengan beberapa ragam bahasa pemrograman. Sebagai contoh seperti bahasa C++ dengan menggunakan library OpenCV. OpenCV adalah salah satu *library* yang populer untuk pemrosesan gambar dan menyediakan fitur untuk *object detection*. Selanjutnya juga dapat menggunakan bahasa Python. Untuk mengimplementasikan *object detection* menggunakan Python, dapat menggunakan beberapa library yang tersedia, seperti *OpenCV*, *TensorFlow*, *Keras*, atau *PyTorch*. Salah satu algoritma *deep learning* yang populer untuk *object detection* adalah YOLO (*You Only Look Once*). YOLO adalah algoritma *object detection* yang cepat dan akurat yang memproses gambar secara real-time. YOLO menggunakan jaringan syaraf konvolusi (*Convolutional Neural Network/CNN*) untuk melakukan *object detection* dan mengklasifikasikannya pada saat yang sama.

2.2 YOLO (*You Only Look Once*)

You Only Look Once (YOLO) adalah sebuah algoritma atau *framework object detection* yang populer di kalangan pengembang *computer vision*. YOLO menggunakan jaringan syaraf tiruan (*neural network*) untuk melakukan *object detection* dan klasifikasi objek pada saat yang sama dengan sangat cepat.

a) Konsep Dasar YOLO

Konsep dasar dari YOLO adalah bahwa untuk setiap piksel dalam gambar, jaringan CNN memprediksi *bounding box* dan probabilitas kelas untuk setiap objek yang terdeteksi. Dalam arsitektur YOLO, gambar dibagi menjadi beberapa sel (*grid cells*). Setiap sel bertanggung jawab untuk *object detection* yang berada di dalamnya. Setiap sel memprediksi beberapa *bounding box* dan probabilitas kelas untuk objek yang terdeteksi.

Sebuah *bounding box* pada YOLO terdiri dari empat koordinat: *x*, *y*, lebar (*width*), dan tinggi (*height*). Setiap *bounding box* juga dikaitkan dengan probabilitas kelas yang menunjukkan kemungkinan bahwa objek dalam *bounding box* tersebut adalah suatu kelas tertentu.

b) Perbandingan Teknik *Object detection*

Kelebihan utama dari YOLO adalah kecepatannya. Dalam pengujian, YOLO mampu melakukan *object detection* pada kecepatan 45 *frame* per detik dengan akurasi yang tinggi. Kecepatan ini membuat YOLO sangat cocok untuk aplikasi real-time seperti *object detection* pada mobil, pengawasan video, dan lain sebagainya. Dalam jurnal *A Comparison of Object detection Techniques for Intelligent Transportation Systems*, peneliti membahas tentang perbandingan kecepatan dan akurasi antara YOLOv3, Faster R-CNN, dan SSD pada dataset MOTChallenge dan menemukan bahwa YOLOv3 memiliki kinerja yang lebih baik dalam hal kecepatan dan akurasi. Penelitian ini juga menunjukkan bahwa YOLOv3 lebih cocok untuk digunakan dalam sistem transportasi pintar [26]. Pada pengujian akurasi, model YOLOv3 berhasil mencapai akurasi deteksi yang lebih baik dengan nilai *Mean Average Precision* (mAP) sebesar 0,41 dibandingkan dengan model *Faster R-CNN* yang hanya mampu mencapai mAP 0,34. Selain itu, model YOLOv3 juga lebih unggul dalam hal kecepatan *object detection*, dengan waktu pengolahan sekitar 30-40 *frame* per detik, sementara model *Faster R-CNN* hanya mampu mencapai waktu pengolahan sekitar 10-20 *frame* per detik. Oleh karena itu, model YOLOv3 lebih cocok digunakan untuk aplikasi pengawasan lalu lintas yang membutuhkan *object detection* real-time [27].

Selain itu, YOLO memiliki performa yang lebih baik dalam mengatasi masalah *overlapping objects* (objek yang tumpang tindih) dibandingkan dengan algoritma *object detection* yang lain. Hal ini karena YOLO menggunakan *grid cell* untuk memprediksi objek dan *bounding box*.

2.2.1 *Precision, Recall, F1-Score*

Precision, Recall, dan F1-Score adalah metrik evaluasi yang digunakan dalam pengukuran kinerja sebuah model atau sistem dalam *object detection*. *Precision, Recall, dan F1-Score* dapat dihitung dari hasil visualisasi *Confusion Matrix* dari hasil penilaian kinerja pengklasifikasi. *Precision* adalah rasio prediksi positif benar dibandingkan dengan hasil prediksi positif secara keseluruhan [28]. *Recall* adalah rasio positif benar dibandingkan dengan semua data positif [28]. Sedangkan *F1-Score* adalah perbandingan *average precision* dan *recall* [28].

	Definition
True Positive (TP)	Jumlah tersebut dianggap positif.
True Negative (TN)	Jumlah tersebut dianggap negatif.
False Positive (FP)	Jumlah data positif yang tidak diklaim.
False Negatif (FN)	Jumlah data negatif yang tidak diklaim.

Gambar 2. 1 Penjelasan *Confusion Matrix*

Gambar 2.1 merupakan penjelasan *Confusion Matrix* yang memiliki 4 nilai yaitu *True Positive (TP)*, *True Negative (TN)*, *False Positive (FP)*, *False Negatif (FN)*. Dari keempat nilai tersebut, dapat menentukan nilai Precision, Recall, dan F1-Score. Terdapat beberapa rumus untuk melakukannya sebagai berikut:

$$Precision = \frac{TP}{TP+FP} \quad 1$$

$$Recall = \frac{TP}{TP+FN} \quad 2$$

$$F1-Score = \frac{2*Precision*Recall}{Precision+Recall} \quad 3$$

2.3 Tools

2.3.1 Python

Python adalah bahasa pemrograman yang mudah dipelajari dan kuat. Ia memiliki struktur data tingkat tinggi yang efisien dan pendekatan sederhana namun efektif terhadap pemrograman berorientasi objek. Sintaksis Python yang elegan dan tipenya yang dinamis, serta sifatnya yang terinterpretasi,

menjadikannya bahasa yang ideal untuk scripting dan pengembangan aplikasi cepat di banyak bidang pada sebagian besar platform [29]. Python didesain untuk memudahkan pengembangan software dengan penekanan pada kemudahan membaca kode dan sintaks yang jelas. Sebagai bahasa pemrograman, Python sangat mudah dipahami dibandingkan dengan bahasa pemrograman lainnya. Dengan sintaks yang jelas dan mudah dipahami, banyak user Python yang dapat mengembangkan kode lebih mudah dan cepat. Python dilengkapi dengan banyak modul dan paket yang dapat membantu dalam memudahkan pengembangan sebuah sistem atau aplikasi. Python banyak digunakan dalam ilmu data dan kecerdasan buatan, terutama karena ada banyak perpustakaan dan kerangka kerja yang tersedia yang memudahkan pengembangan aplikasi di bidang tersebut. Beberapa perpustakaan populer untuk ilmu data dan kecerdasan buatan di Python adalah NumPy, Pandas, Scikit-learn, TensorFlow, dan Keras.

2.3.2 Visual Studio Code

Visual Studio Code (VS Code) adalah sebuah *Integrated Development Environment (IDE)* yang dapat digunakan untuk mengembangkan berbagai jenis aplikasi, termasuk aplikasi pengolahan citra dan *object detection*. VS Code adalah IDE yang sangat populer di kalangan pengembang karena fitur-fiturnya yang dapat memudahkan pengembangan aplikasi [30].

Untuk melakukan *object detection* menggunakan VS Code, umumnya menggunakan bahasa pemrograman Python dan perpustakaan OpenCV. OpenCV adalah perpustakaan Python yang dapat digunakan untuk pemrosesan citra dan *object detection* [31]. OpenCV juga menyediakan berbagai algoritma untuk *object detection*, seperti *Haar Cascade*, HOG+SVM, dan YOLO.

Salah satu penelitian terkait penggunaan VS Code dan OpenCV untuk *object detection* adalah yang dilakukan oleh Feng. Dalam penelitiannya, mereka mengembangkan sebuah sistem *object detection* menggunakan YOLOv4 dan implementasinya menggunakan VS Code dan OpenCV [32]. Hasil dari penelitian ini menunjukkan bahwa sistem yang dikembangkan dapat melakukan *object detection* dengan akurasi yang tinggi dan dapat dijalankan secara real-time.

Selain itu, Agarwal juga menyebutkan bahwa *VS Code* dapat digunakan untuk mengembangkan model *object detection* berbasis neural network [30]. Model-model tersebut dapat diimplementasikan menggunakan bahasa pemrograman Python dan perpustakaan *Deep Learning* seperti *TensorFlow* atau *PyTorch*.

Dengan begitu, *Visual Studio Code* dikenal sebagai IDE yang sangat populer dan dapat digunakan untuk pengembangan aplikasi *object detection* dengan berbagai metode seperti *Haar Cascade*, HOG+SVM, YOLO, atau model-model berbasis *artificial neural network*. Dalam penggunaannya, *VS Code* umumnya digabungkan dengan bahasa pemrograman Python dan perpustakaan OpenCV atau *Deep Learning*.

2.3.3 Google Colaboratory

Google Colaboratory atau sering disingkat Colab adalah sebuah platform gratis yang disediakan oleh Google untuk memudahkan pengembangan aplikasi dan model machine learning di lingkungan cloud. Colab menggunakan Google Drive sebagai tempat penyimpanan dan pengelolaan proyek yang sedang dikerjakan. Platform ini akan digunakan untuk melakukan pelatihan model machine learning, termasuk pelatihan model untuk *object detection*.

Colab memiliki banyak keunggulan sebagai platform untuk pengembangan model *object detection* [33]. Salah satu contoh kelebihanannya adalah kemampuan Colab untuk menggunakan GPU secara gratis, sehingga mempercepat waktu pelatihan model dan pengembangan sistem *object detection*.

Selain itu, Kabir juga menunjukkan bahwa Colab dapat digunakan untuk pengembangan model *object detection* menggunakan beberapa arsitektur deep learning seperti YOLOv3, YOLOv4, Faster R-CNN, dan RetinaNet [33]. Dalam penelitian mereka, mereka menggunakan dataset COCO untuk melatih model dan mendapatkan hasil *object detection* yang akurat.

Hasil yang sejalan juga didapatkan oleh Mubarak yang melakukan penelitian penggunaan Google Colab dalam pelatihan model *object detection*

menggunakan arsitektur Faster R-CNN. Hasil dari penelitian mereka menunjukkan bahwa Colab dapat mengurangi waktu pelatihan model hingga 10 kali lipat dibandingkan dengan menggunakan CPU[34].

Selain itu, Colab juga menyediakan berbagai fitur seperti penggunaan notebook Jupyter, integrasi dengan TensorFlow dan Keras, serta kemampuan untuk mengimpor dataset dari Google Drive atau GitHub. Semua fitur tersebut dapat memudahkan *developer* dalam melakukan eksplorasi dan pengembangan model *object detection*.

2.3.4 Roboflow

Roboflow adalah sebuah platform yang menyediakan berbagai solusi untuk melakukan pengolahan data pada gambar dan video dalam konteks *Artificial Intelligence* atau *Machine Learning*. *Roboflow* dapat dimanfaatkan untuk membantu dalam memproses dataset berupa gambar dan video. Salah satu teknologi yang digunakan oleh *Roboflow* adalah *computer vision*. Teknologi ini digunakan untuk menganalisis data gambar dan video yang diunggah ke platform. Dengan memanfaatkan teknologi ini, pengguna dapat melakukan pengenalan terhadap sebuah objek, segmentasi gambar, dan analisis visual lainnya untuk memproses dataset. *Roboflow* membuat anotasi, persiapan, augmentasi, dan pengelolaan dataset visi komputer atau *computer vision* menjadi sangat sederhana dan sangat mempercepat proses dalam komputasi [35].

Seperti yang disebutkan oleh Ivan Goncharov, teknik data *augmentation* digunakan untuk memperkaya dataset dengan memanipulasi gambar atau video asli menjadi variasi baru yang lebih bervariasi dan mewakili kondisi dunia nyata. Selain itu, teknik *transfer learning* juga digunakan untuk mempercepat proses pelatihan model pada dataset yang baru. *Roboflow* juga menggunakan teknik *object detection* untuk mengenali dan memisahkan objek dalam gambar atau video, serta menyediakan fitur-fitur yang memudahkan pengguna untuk menjalankan workflow *machine learning* termasuk *preprocessing data* dan pelatihan model. Platform ini menggunakan teknologi dan konsep terbaru dalam

computer vision dan *machine learning* untuk membantu pengguna memproses dataset dan membangun model AI yang akurat dan efisien.

Dalam penelitian ini, *Roboflow* dimanfaatkan untuk melakukan preprocessing data, pelatihan model, hingga nanti pada saat melakukan evaluasi terhadap model dari *object detection* yang dilakukan. *Roboflow* akan sangat berguna untuk melatih model menggunakan model YOLOv8 untuk *object detection* penumpang dalam pembuatan sistem *object detection* menghitung jumlah penumpang.

2.4 Penulisan Terdahulu

Tabel 2. 1 Artikel Jurnal Terdahulu Pertama

JURNAL 1	
Judul Artikel Penelitian	<i>Implementation Counting and YOLO Object detection Methods for Identification Degree of Road Saturation</i>
Penulis	Rico Aditya Utama
Tahun	2022
Jurnal	<i>Journal of Systems Engineering and Information Technology</i>
Metode	<i>Computer Vision</i> dengan <i>YOLO Object Detector</i> berbasis <i>Deep Learning</i> .
Pembahasan dan Hasil	Hasil penelitian yang diperoleh adalah sistem yang dibuat cukup baik, didukung dari nilai kesalahan yang diperoleh sistem sekitar 3-4 % untuk data foto dan 10-11% untuk data video[18]

Tabel 2. 2 Artikel Jurnal Terdahulu Kedua

JURNAL 2	
Judul Artikel Penelitian	Implementasi Deteksi Real Time Klasifikasi Jenis Kendaraan di Indonesia Menggunakan Metode YOLOV5.
Penulis	Dadang Iskandar Mulyana, M Ainur Rofik
Tahun	2022

Jurnal	Jurnal Pendidikan Tambisa
Metode	YOLOv5
Pembahasan dan Hasil	Penelitian menggunakan dataset sebesar 1.332 gambar dengan beberapa kelas transportasi umum. Hasil penelitian ini menunjukkan bahwa YOLOV5 dapat mengenali objek secara konsisten dengan tingkat akurasi 90% [19]

Tabel 2. 3 Artikel Jurnal Terdahulu Ketiga

JURNAL 3	
Judul Artikel Penelitian	<i>Implementation of Object detection with You Only Look Once Algorithm in Limited Face-to-Face Times in Pandemic.</i>
Penulis	Yogi Yunefri, Sutejo, Yogi Ersan Fadrial, Keumala Anggraini, Maya Ramadhani, Roki Hardianto.
Tahun	2022
Jurnal	<i>Journal of Applied Engineering and Technological Science</i>
Metode	YOLO Tiny-v3
Pembahasan dan Hasil	Hasilnya menunjukkan nilai akurasi 76,316% untuk <i>menobject detection</i> manusia dan akurasi 94,444% dalam mendeteksi jarak peringatan tidak aman antar objek manusia. [17]

Tabel 2. 4 Artikel Jurnal Terdahulu Keempat

JURNAL 4	
Judul Artikel Penelitian	<i>YOLO with Adaptive Frame Control for Real Time Object detection Applications.</i>
Penulis	Jeonghun Lee, Kwang-il Hwang
Tahun	2022
Jurnal	<i>Multimedia Tools and Applications</i>

Metode	YOLO with AFC (<i>Adaptive Frame Control</i>)
Pembahasan dan Hasil	AFC dapat digunakan untuk dijadikan <i>add-on package</i> untuk <i>real-time processing</i> untuk melakukan <i>object detection</i> dengan berbagai algoritma bukan hanya YOLO.[16]

Tabel 2. 5 Artikel Jurnal Terdahulu Kelima

JURNAL 5	
Judul Artikel Penelitian	YOLO-FIRI: <i>Improved YOLOv5 for Infrared Image Object detection</i>
Penulis	Shasha Li, Yongjun Li, Yao Li, Mengjun Li, & Xiaorong Xu.
Tahun	2021
Jurnal	<i>IEEE Access</i>
Metode	YOLO-FIRI
Pembahasan dan Hasil	Hasil eksperimen pada dataset KAIST dan FLIR menunjukkan bahwa YOLO-FIRI memiliki peningkatan kualitatif dibandingkan dengan detektor canggih. Dibandingkan dengan YOLOv4, hasil rata-rata (mAP50) dari YOLO-FIRI meningkat 21% pada dataset KAIST, sedangkan kecepatan berkurang 62%, parameter menurun sebesar 89%, ukuran bobot berkurang lebih dari 94%, dan biaya komputasi berkurang sebesar 84%. Dibandingkan dengan YOLO-FIR, YOLO-FIRI memiliki sekitar 5% hingga 20% peningkatan AP, AR, mAP50, F1, dan mAP50:75 [15]

Tabel 2. 6 Artikel Jurnal Terdahulu Keenam

JURNAL 6	
Judul Artikel Penelitian	YOLO v3-Tiny: <i>Object detection and Recognition using one stage improved model.</i>
Penulis	Pranav Adarsh, Pratibha Rathi, Manoj Kumar.

Tahun	2020
Jurnal	<i>International Conference on Advanced Computing & Communication Systems (ICACCS)</i>
Metode	YOLO v3-Tiny
Pembahasan dan Hasil	<i>Two stage detector</i> lebih fokus pada akurasi sedangkan <i>one stage detector</i> fokus kepada kecepatan. Hasil perbandingan menunjukkan bahwa YOLO v3-Tiny meningkatkan kecepatan <i>object detection</i> sambil memastikan akurasi hasilnya.[14]

Tabel 2. 7 Artikel Jurnal Terdahulu Ketujuh

JURNAL 7	
Judul Artikel Penelitian	<i>Object detection through Modified YOLO Neural Network</i>
Penulis	Tanvir Ahmad, Yinglong Ma, Muhammad Yahya, Belal Ahmad, Shah Nazir, Amin ul Haq.
Tahun	2020
Jurnal	<i>Hindawi Scientific Programming</i>
Metode	<ul style="list-style-type: none"> • YOLO v1 • R-CNN
Pembahasan dan Hasil	<i>Neural network YOLOv1</i> berbasis <i>object detection</i> dengan memodifikasi fungsi kerugian dan menambahkan lapisan penyatuan piramida spasial dan modul awal dengan kernel konvolusi 1 x 1. jaringan baru dilatih dengan metode <i>end-to-end</i> , dan percobaan ekstensif pada a dataset Pascal VOC yang menunjukkan keefektifan jaringan baru yang ditingkatkan, dengan deteksi hasil menjadi 65,6% dan 58,7%, masing-masing. Hasil dari jaringan yang diusulkan telah dibandingkan dengan yang ada R-CNN dan YOLOv1 [13]

Tabel 2. 8 Artikel Jurnal Terdahulu Kedelapan

JURNAL 8	
Judul Artikel Penelitian	Tinier-YOLO: <i>A Real-Time Object detection Method for Constrained Environments.</i>
Penulis	Wei Fang, Lin Wang, Peiming Ren
Tahun	2019
Jurnal	<i>IEEE Access</i>
Metode	<ul style="list-style-type: none"> • Tiny-YOLOV3 • Tinier-YOLO
Pembahasan dan Hasil	Hasil penelitian menunjukkan model ukuran 8,9MB (hampir 4× lebih kecil dari Tiny-YOLO-V3) sekaligus mencapai performa real-time 25 FPS pada Jetson TX1 dan mAP 65,7% pada PASCAL VOC dan 34,0% pada COCO[12]

Tabel 2. 9 Artikel Jurnal Terdahulu Kesembilan

JURNAL 9	
Judul Artikel Penelitian	<i>A Simple Vehicle Counting System Using Deep Learning with YOLOv3 Model</i>
Penulis	Muhammad Fachrie
Tahun	2020
Jurnal	Ikatan Ahli Informatika Indonesia
Metode	YOLO V3
Pembahasan dan Hasil	Hasil penelitian menunjukkan bahwa hasil deteksi model YOLOv3 untuk melacak pergerakan kendaraan tertinggi sebesar 97,72% dengan video zoom frontside-1x [20].

Tabel 2. 10 Artikel Jurnal Terdahulu Kesepuluh

JURNAL 10	
-----------	--

Judul Artikel Penelitian	<i>YOLO for Penguin Detection and Counting Based on Remote Sensing Images</i>
Penulis	Jiahui Wu, Wen Xu, Jianfeng He, Musheng Lan.
Tahun	2023
Jurnal	<i>Multidisciplinary Digital Publishing Institute</i>
Metode	YOLOPd
Pembahasan dan Hasil	Hasil penelitian <i>object detection</i> untuk menghitung jumlah penguin memiliki akurasi rata-rata 94,6% dari 18 gambar [21].

Salah satu penelitian terdahulu pertama yang berjudul “Implementasi Deteksi Real Time Klasifikasi Jenis Kendaraan di Indonesia Menggunakan Metode YOLOV5” melakukan pembuatan sistem *penobject detection* menggunakan metode YOLOV5 untuk mendeteksi jenis kendaraan di jalan raya. Penelitian ini menggunakan dataset sebesar 1.332 gambar dengan kelas bajaj, becak, bus, mobil, mobil molen, mobil pik’up, sepeda, sepeda motor, dan truk. Hasil penelitian ini menunjukkan bahwa YOLOV5 dapat mengenali objek secara konsisten dengan tingkat akurasi 90% [19].

Penelitian terdahulu kedua yang berjudul “*Implementation Counting and Yolo Object detection Methods for Identification Degree of Road Saturation*” melakukan identifikasi tingkat kemacetan dengan memperhatikan parameter derajat kejenuhan jalan berdasarkan Manual Kapasitas Jalan Indonesia (MKJI). Metode yang digunakan adalah menggabungkan teknik *Computer Vision* dengan *YOLO Object Detector* berbasis *Deep Learning*. Hasil penelitian yang diperoleh adalah sistem yang dibuat cukup baik, didukung dari nilai kesalahan yang diperoleh sistem sekitar 3-4 % untuk data foto dan 10-11% untuk data video [18].

Penelitian terdahulu ketiga yang berjudul “*Implementation of Object detection with You Only Look Once Algorithm in Limited Face-to-Face Times in Pandemic*” melakukan pembuatan sistem yang dapat memantau jarak fisik menggunakan metode YOLO Tiny V3 dengan *menobject detection* manusia. Hasilnya

menunjukkan nilai akurasi 76,316% untuk *object detection* manusia dan akurasi 94,444% dalam mendeteksi jarak peringatan tidak aman antar objek manusia [17].

Penelitian terdahulu keempat yang berjudul “*YOLO v3-Tiny: Object detection and Recognition Using One Stage Improved Model*” melakukan perbandingan antara *two stage detector* seperti RCNN, Fast RCNN, dan Faster RCNN dengan *one stage detector* seperti YOLO v1, YOLO v2, YOLO v3m dan SSD. Hasilnya menunjukkan bahwa *two stage detector* lebih fokus kepada akurasi sedangkan *one stage detector* terfokus kepada kecepatan. Hasil perbandingan juga menunjukkan bahwa YOLO v3-Tiny meningkatkan kecepatan *object detection* sambil memastikan akurasi hasilnya [14].

Penelitian terdahulu kelima yang berjudul “*YOLO-FIRI: Improved YOLOv5 for Infrared Image Object detection*” melakukan sebuah pengembangan terhadap YOLOv5 untuk melakukan *object detection* bergambar *infrared* dengan YOLO-FIRI. Hasil eksperimen dari penelitian yang dilakukan pada dataset KAIST dan FLIR menunjukkan bahwa YOLO-FIRI memiliki peningkatan kualitatif dibandingkan dengan detektor canggih. Dibandingkan dengan YOLOv4, hasil rata-rata (mAP50) dari YOLO-FIRI meningkat 21% pada dataset KAIST, sedangkan kecepatan berkurang 62%, parameter menurun sebesar 89%, ukuran bobot berkurang lebih dari 94%, dan biaya komputasi berkurang sebesar 84%. Dibandingkan dengan YOLO-FIR, YOLO-FIRI memiliki sekitar 5% hingga 20% peningkatan AP, AR (*recall* rata-rata), mAP50, F1, dan mAP50:75 [15]

Penelitian terdahulu keenam yang berjudul “*Tinier-YOLO: A Real-Time Object detection Method for Constrained Environments*” melakukan *object detection* secara *realtime* pada *constrained environments* dengan memanfaatkan metode Tinier-YOLO. Hasil penelitian menunjukkan model ukuran 8,9MB (hampir 4× lebih kecil dari Tiny-YOLO-V3) sekaligus mencapai performa real-time 25 FPS pada Jetson TX1 dan mAP 65,7% pada PASCAL VOC dan 34,0% pada COCO [12].

Penelitian terdahulu ketujuh yang berjudul “*YOLO with adaptive frame control for real-time object detection applications*” melakukan penunjukan masalah yang berkaitan dengan pemrosesan *real-time object detection* dengan YOLO terkait dengan *network cameras* dan kemudian mengusulkan sebuah YOLO arsitektur dengan *Adaptive Frame Control (AFC)* yang dapat mengatasi masalah secara lebih efisien. AFC dapat digunakan untuk dijadikan *add-on package* untuk *real-time processing* untuk melakukan *object detection* dengan berbagai algoritma bukan hanya YOLO [16].

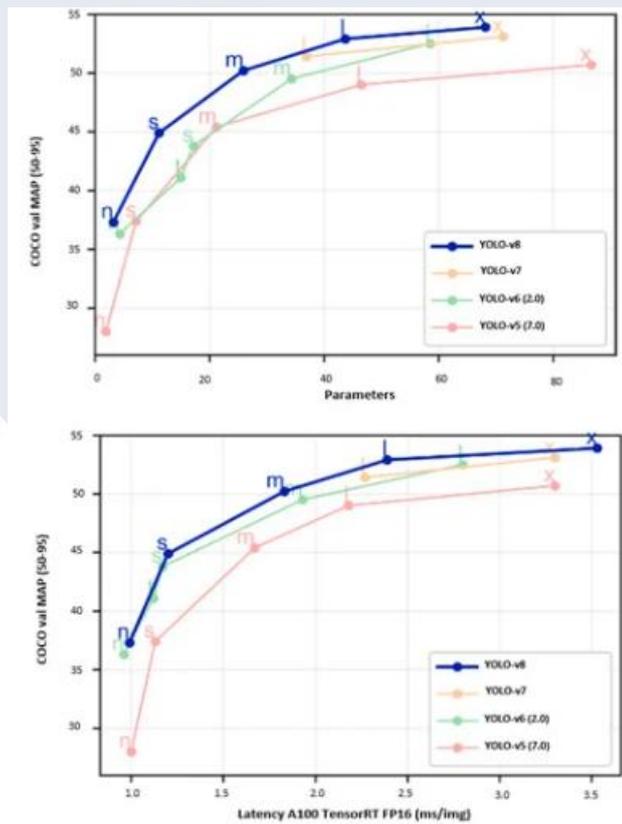
Penelitian terdahulu kedelapan yang berjudul “*Object detection through Modified YOLO Neural Network*” melakukan *neural network* YOLOv1 berbasis *object detection* dengan memodifikasi fungsi kerugian dan menambahkan lapisan penyatuan piramida spasial dan modul awal dengan kernel konvolusi 1 x 1. jaringan baru dilatih dengan metode *end-to-end*, dan percobaan ekstensif pada a dataset Pascal VOC yang menunjukkan keefektifan jaringan baru yang ditingkatkan, dengan deteksi hasil menjadi 65,6% dan 58,7%, masing-masing. Hasil dari jaringan yang diusulkan telah dibandingkan dengan yang ada R-CNN dan YOLOv1 [13].

Penelitian terdahulu kesembilan yang berjudul “*A Simple Vehicle Counting System Using Deep Learning with YOLOv3 Model*” melakukan pelacakan pergerakan kendaraan. Metode yang digunakan adalah dengan menggunakan YOLO V3 dan hasil penelitian menunjukkan sistem berhasil melakukan *object detection* dengan akurasi tertinggi sebesar 97.72% dengan zoom fronside-1x [20].

Penelitian terdahulu kesepuluh yang berjudul “*YOLO for Penguin Detection and Counting Based on Remote Sensing Images*” melakukan *object detection* dengan metode YOLOPd untuk menghitung jumlah penguin dari gambar jarak jauh udara. Hasil penelitian *object detection* untuk menghitung jumlah penguin memiliki akurasi rata-rata 94,6% dari 18 gambar [21].

Berdasarkan hasil penelitian terdahulu yang sudah terlampir, penelitian ini berbeda dengan penelitian yang telah dilakukan oleh para peneliti sebelumnya dikarenakan penelitian ini menggunakan metode YOLOv8 untuk melakukan

pendeteksian objek berupa manusia untuk dihitung jumlah objek tersebut saat memasuki ke dalam transportasi atau bus. Objek manusia dianggap sebagai penumpang bus yang akan dilakukan *object detection* nya saat memasuki pintu masuk bus. Perbedaan YOLOv8 dengan penelitian sebelumnya adalah YOLOv8 baru diluncurkan pada tanggal 10 Januari 2023 dan merupakan YOLO *version* terbaru untuk saat ini [36].

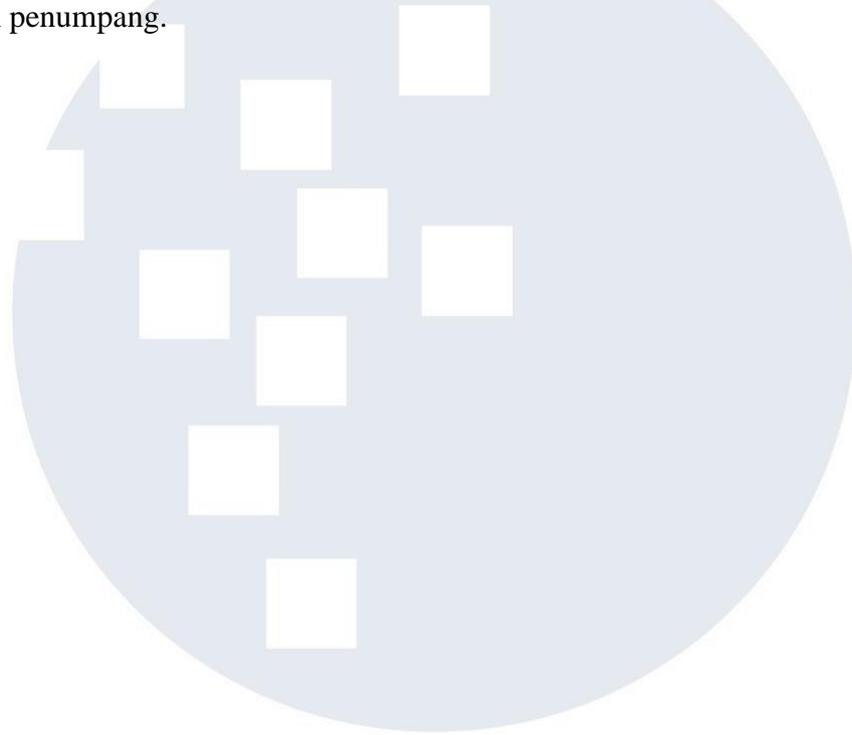


Gambar 2. 2 Perbandingan YOLOv8 dengan YOLO lainnya

Sumber: [48]

Gambar 2.2 menunjukkan bahwa ketika membandingkan YOLO-v8 dengan YOLO-v5 dan YOLO-v6 yang dilatih pada resolusi gambar 640, semua varian YOLO-v8 menghasilkan *output* yang lebih baik dengan jumlah parameter yang serupa, menunjukkan reformasi arsitektur yang efisien dibandingkan dengan YOLO *version* sebelumnya [37].

Oleh karena itu, penelitian ini dapat dikatakan baru dikarenakan belum terlalu banyak penelitian yang menggunakan model YOLOv8 dalam penelitiannya, terlebih dalam melakukan deteksi objek serta implementasi dalam menghitung jumlah penumpang.



UMMN

UNIVERSITAS
MULTIMEDIA
NUSANTARA