

SISTEM DETEKSI DAN KLASIFIKASI JENIS KENDARAAN PADA PERSIMPANGAN BERSINYAL

Ubaidillah alhabsyi¹, Radimas Putra Muhammad D.L², Aryuanto Soetedjo³

Teknik Elektro S1 Institut Teknologi nasional, Malang, Indonesia

¹dilah.habsy@gmail.com, ²radimas@lecturer.itn.ac.id, ³aryuanto@lecturer.itn.ac.id

Abstract— Deteksi dan klasifikasi kendaraan sangat penting untuk berbagai tujuan seperti pemantauan lalu lintas, manajemen lalu lintas, dan keamanan lalu lintas. Jika jumlah kendaraan banyak dan sumber daya manusia terbatas, identifikasi kendaraan biasanya dicatat secara manual oleh petugas yang kurang efektif. DRI (Detection, Recognition, & Identifikasi), juga dikenal sebagai deteksi, klasifikasi, dan identifikasi kendaraan, adalah salah satu kemajuan teknologi yang saat ini berkembang pesat. Ini telah diterapkan secara efektif dalam sistem deteksi kendaraan, pengendalian lalu lintas, dan manajemen lalu lintas. Dengan menggunakan kamera CCTV untuk mendeteksi, mengklasifikasikan, dan menghitung algoritma YOLO (You Only Look Once), penelitian ini berusaha mengatasi masalah tersebut. Algoritma YOLOv5 dipilih oleh penulis karena memiliki pendeteksian objek yang lebih baik dan nilai akurasi yang lebih tinggi dibandingkan versi sebelumnya. Pustaka OpenCV, yang dibuat menggunakan bahasa pemrograman Python, akan membantu proses desain ini dengan menggunakan kumpulan data yang tersedia. Penulis menyimpulkan bahwa penelitian ini dilakukan dengan sangat baik dan memenuhi tuntutan objek penelitian dengan sangat akurat dan efisien. Hal ini dibuktikan dengan menguji sistem deteksi dan klasifikasi kendaraan dalam beberapa percobaan pelatihan.

Kata Kunci—Deteksi dan klasifikasi Kendaraan, YOLOv5

I. PENDAHULUAN

Ketika komputer mampu meniru cara berpikir dan berperilaku seperti manusia, kecerdasan buatan dapat menghasilkan aplikasi yang berguna untuk menyelesaikan berbagai masalah bagi manusia [1][2]. Saat ini, perkembangan kecerdasan buatan mencakup bidang komputasi seperti deteksi objek, yang melibatkan identifikasi objek dalam gambar dan video menggunakan teknologi AI [3]. Selain itu, penelitian di bidang visi komputer dan deep learning memungkinkan data secara otomatis direpresentasikan dalam bentuk teks, gambar, atau video [4].

Metode pengenalan objek berbasis region proposal dengan dua tahap dan metode regresi satu tahap merupakan dua pendekatan utama dalam *deep learning* untuk pengenalan objek. Kedua metode ini memanfaatkan *Convolutional Neural Networks* (CNN) [5]. Pendekatan region proposal bekerja dengan mengidentifikasi objek, mengklasifikasikan ulang menggunakan model pada berbagai lokasi dan skala, serta

memberikan skor pada gambar untuk dievaluasi. Sementara itu, metode regresi tidak melakukan klasifikasi ulang, sehingga prosesnya lebih cepat dibandingkan, misalnya, algoritma YOLO [6]. YOLO (*You Only Look Once*) V5, sebagai versi terbaru dari metode ini, memiliki kemampuan mendeteksi objek dengan cepat. YOLO didukung oleh dua framework, yaitu Darknet dan Darkflow, yang berbagi pustaka serupa namun menggunakan bahasa pemrograman yang berbeda. Salah satu penerapan deteksi objek yang berkembang pesat adalah identifikasi jenis kendaraan.

Orang sangat membutuhkan kendaraan untuk melakukan kegiatan sehari-hari mereka. Berbagai jenis kendaraan muncul sebagai hasil dari kemajuan transportasi. Ini termasuk kendaraan kecil seperti sepeda motor, sepeda, dan becak, kendaraan sedang seperti mobil, dan kendaraan besar seperti truk dan bus. Suatu sistem yang dapat mendeteksi dan mengklasifikasikan kendaraan sesuai dengan kebutuhan akan semakin dibutuhkan seiring dengan jumlah kendaraan yang ada. Metode YOLO Object Detection memiliki kemampuan untuk mendeteksi objek dengan hasil yang akurat dan dalam waktu yang sangat cepat [7].

Oleh sebab itu penulis ingin melakukan penelitian tentang pendeteksian objek berupa jenis kendaraan dengan metode YOLO Object Detection dengan judul “SISTEM DETEKSI DAN KLASIFIKASI JENIS KENDARAAN PADA PERSIMPANGAN BERSINYAL”.

II. TINJAUAN PUSTAKA

A. Convolutional Neural Network

CNN memproses dan mendeteksi objek pada gambar dengan memproses konvolusi. Salah satu dari dua arsitektur CNN adalah Lapisan Pengekstraksi Fitur (MLP) dan Lapisan Terhubung Sepenuhnya. Lapisan Pengekstraksi Fitur mengkodekan gambar menjadi fitur yang terdiri dari angka yang mewakili fitur tersebut, sedangkan Lapisan Terhubung Sepenuhnya menghubungkan semua aktivitas neuron di lapisan sebelumnya ke lapisan berikutnya [12]. Gambar 1 menunjukkan struktur CNN.

Gambar 3 Arsitektur YOLO

C. Citra

Citra adalah representasi atau representasi yang meniru suatu objek. Citra memiliki fitur yang tidak ada dalam data teks. image yang disimpan di perangkat elektronik juga disebut "citra". Dalam penelitian ini, ekstensi imageJPEG (*Joint Photographic Experts Group*) digunakan. JPEG adalah jenis file gambar grafik yang dikembangkan oleh suatu organisasi yang digunakan untuk proses gambar di komputer [13]. Ekstensi gambar JPEG adalah.jpg.

Pada umumnya gambar digunakan bila radiasi elektromagnetik yang dipancarkan atau dipantulkan dari suatu obyek tidak langsung direkam pada film. Sedangkan penginderaan jauh ialah ilmu dan seni untuk memperoleh informasi tentang obyek, daerah, atau gejala dengan jalan menganalisis data yang diperoleh dengan menggunakan alat tanpa kontak langsung terhadap obyek, daerah, atau gejala yang dikaji.[14]

D. Deteksi jenis kendaraan

Dengan menggunakan input gambar dan output klasifikasi, deteksi jenis kendaraan adalah proses klasifikasi pola. Hasilnya adalah gambar keluaran yang telah diproses sistem dan memiliki pola output dari proses pendeteksian [10]. Orang biasanya menggunakan kendaraan untuk kegiatan sehari-hari dan pindah dari tempat yang dekat maupun jauh. Mesin yang menggerakkan kendaraan memungkinkan mereka bergerak [12].

Dengan mengklasifikasi jenis kendaraan dengan gambar yang optimal bertujuan agar lalu lintas terhindar dari macet yang parah dan mengetahui jumlah kendaraan yang terdapat pada jalan tersebut, berikut tabel 1 pengkelasan kendaraan.

Tabel 1 Klasifikasi jenis kendaraan

Kelas	Jenis kendaraan	Tipikal kendaraan
Kendaraan Ringan	Kendaraan roda 2 (dua) dan 3 (tiga) dengan panjang tidak lebih dari 2,5 m	Sepeda motor, becak, sepeda
Kendaraan Sedang	mobil penumpang 4 (empat) tempat duduk, mobil penumpang 7 (tujuh) tempat duduk, dengan panjang tidak lebih dari 5,5 m	Sedan, jeep,
Kendaraan Berat	Bus sedang dan mobil angkutan barang 2 (dua) sumbu dengan panjang tidak lebih dari 12 m	Bus dan truk

Sumber: Panduan Kapasitas Jalan Indonesia 2023

E. Computer Vision

Computer vision erat kaitannya dengan pengolahan citra (*image processing*) dan visi mesin (*machine vision*). Teknologi ini dikembangkan dengan tujuan agar komputer atau mesin

dapat memahami dan menginterpretasikan informasi sensorik, mirip dengan cara kerja indera manusia dan otak.

Pengolahan citra adalah suatu sistem prosesing yang dilakukan dengan masukan berupa citra dan hasilnya juga berupa citra. Pengolahan citra ini dilakukan untuk memperbaiki kualitas citra, serta dapat mengambil informasi dari suatu citra, oleh karena itu *image processing* tidak dapat dilepaskan dengan bidang computer vision Sesuai dengan perkembangan computer vision itu sendiri.

Machine vision adalah teknologi dan metode untuk melakukan inspeksi dan analisis otomatis yang berbasis gambar, kontrol proses dan kontrol robot, yang biasanya digunakan pada skala industri. Machine vision itu sendiri mencakup berbagai teknologi seperti perangkat lunak dan perangkat keras, sistem terintegrasi, berbagai proses, metode, dan keahlian.

Tujuan dari machine vision untuk mengintegrasikan berbagai jenis teknologi untuk memecahkan berbagai masalah. Machine vision sering digunakan untuk kebutuhan produksi dan industri, seperti mendeteksi kecacatan pada produk.

Teknologi ini memungkinkan pemrosesan dan analisis gambar secara otomatis yang dapat diterapkan di berbagai bidang. Selain itu, mesin juga dilengkapi dengan kemampuan untuk mengintegrasikan analisis gambar otomatis dengan teknologi lain, seperti kombinasi perangkat keras dan perangkat lunak, untuk tujuan tertentu [6]

F. Bahasa Python

Python adalah sebuah bahasa pemrograman yang dapat digunakan pada beberapa platform (*Multi Platform*) dan bersifat terbuka (*Open Source*).

Guido Van Rossum mengembangkan bahasa pemrograman Python pertama kali di CWI, Belanda, pada tahun 1990. Bahasa ini dianggap sebagai bahasa tingkat tinggi, atau bahasa tingkat tinggi, dan dianggap sebagai bahasa berorientasi objek yang dinamis [8].

Dalam hal aturan penulisan kode program, pembangunan Python berbeda dengan pembangunan bahasa lain. Tidak seperti bahasa lain, Python memiliki indentasi, tipe data, tuple, dan dictionary.

Python adalah bahasa pemrograman dinamis yang mendukung paradigma pemrograman berorientasi objek. Python dapat digunakan untuk berbagai kebutuhan pengembangan perangkat lunak dan kompatibel dengan berbagai sistem operasi, seperti Linux, Windows, Unix, Symbian, dan lainnya.

Saat ini, Python menjadi salah satu bahasa pemrograman yang populer karena menawarkan sejumlah keunggulan, di antaranya:

- 1) Memiliki pustaka yang kaya dengan modul untuk berbagai kebutuhan.
- 2) Mendukung paradigma pemrograman berorientasi objek.
- 3) Memiliki sintaks yang sederhana dan mudah dipahami.
- 4) Dilengkapi dengan sistem pengelolaan memori otomatis.

- 5) Bersifat extensible dan embeddable, memungkinkan integrasi dengan bahasa lain, seperti menggabungkan modul Python dengan C++.

Adapun kekurangan bahasa Python sebagai berikut:

1. Beberapa penugasan terdapat di luar jangkauan kemampuan Python. Seperti bahasa pemrograman dinamis lainnya, Python tidak secepat atau efisien sebagai statis, tidak seperti bahasa pemrograman C.
2. Python cukup buruk untuk pengembangan platform mobile (Android/iOS).
3. Python merupakan interpreter sehingga bukan perangkat bantu terbaik untuk pengantar kinerja kritis. Jadi, Python bukan pilihan yang baik untuk tugas – tugas intensif memori.
4. Hampir mustahil untuk membuat game 3D grafis tinggi menggunakan Python.

Bahasa Python pertama kali dikembangkan oleh sekelompok programmer dengan dukungan dari Guido van Rossum dan Python Software Foundation. Yayasan ini merupakan organisasi independen yang memegang hak kekayaan intelektual atas bahasa Python sejak versi 2.1, untuk mencegah pengembangannya dikuasai oleh perusahaan komersial. Python telah berkembang hingga mencapai versi 2.6.1 dan 3.0 [15]

G. Pycharm

PyCharm adalah *Integrated Development Environment* (IDE) yang dirancang oleh JetBrains untuk mendukung pengembangan dengan bahasa pemrograman Python. Selain itu, JetBrains juga mengembangkan IDE lain untuk berbagai bahasa pemrograman, seperti IntelliJ IDEA untuk Java, PhpStorm untuk PHP, dan RubyMine untuk Ruby.

Kekurangan dan Kelebihan Pycharm:

Kelebihan Pycharm

- ✓ Pycharm merupakan jalan pintas produktif dalam pengembangan Python.
- ✓ Pycharm memiliki kemampuan untuk melihat seluruh kode sumber Python dengan satu klik
- ✓ Ketersediaan berbagai plugin
- ✓ Pycharm tergolong mudah digunakan.
- ✓ Pycharm memiliki dukungan komunitas yang sangat baik.
- ✓ Memfasilitasi pengembangan kode yang lebih cepat
- Lebih kuat, versi komersial tersedia
- ✓ Proses instalasi yang mudah

Kekurangan Pycharm

- ◆ Versi berbayar mahal
- ◆ Dapat menimbulkan masalah saat mencoba memperbaiki alat seperti venv
- ◆ Tidak cocok untuk pemula Python
- ◆ Membutuhkan banyak memori dan ruang penyimpanan

H. OpenCV

OpenCV (*Open Computer Vision*) adalah pustaka API (*Application Programming Interface*) sumber terbuka yang sangat cocok untuk pengolahan citra (*image processing*). OpenCV dikembangkan oleh Gary Bradsky untuk Intel pada

tahun 1999 dan resmi dirilis ke publik pada tahun 2000. Pada tahun 2005, OpenCV meraih penghargaan DARPA Grand Challenge. Saat ini, banyak algoritma pembelajaran mesin dan visi komputer yang didukung oleh OpenCV.

OpenCV mendukung berbagai bahasa pemrograman, termasuk C++, Python, Java, dan lainnya. Pustaka ini juga dapat digunakan pada berbagai platform, seperti Windows, Linux, macOS, Android, dan iOS. Dengan memanfaatkan antarmuka berbasis CUDA dan OpenCL, OpenCV mampu menjalankan operasi berkecepatan tinggi pada GPU. Kombinasi optimal antara OpenCV, API C++, dan bahasa pemrograman Python memungkinkan tercapainya performa tinggi tersebut.

OpenCV pada dasarnya terdiri dari lima pustaka utama, yaitu:

1. CV: Berisi algoritma untuk pengolahan citra dan visi komputer.
2. ML: Pustaka untuk machine learning.
3. Highgui: Mendukung antarmuka pengguna grafis (GUI), serta input/output gambar dan video.
4. CXCore: Menyediakan struktur data, dukungan untuk XML, dan fungsi grafis.
5. CvAux: Modul visi komputer sumber terbuka yang dikembangkan oleh Intel Corporation

Banyak fitur OpenCV yang dapat digunakan. Beberapa fitur utamanya adalah sebagai berikut:

1. /O Gambar dan Video: Fitur ini memungkinkan untuk membaca data gambar dari file atau umpan video langsung, serta menciptakan file gambar dan video.
2. Interface Pengolahan Visual dan Gambar: Fitur ini memungkinkan untuk melakukan eksperimen uji coba dengan berbagai algoritma penglihatan komputer yang umum, seperti deteksi garis, proyeksi elips,
3. OpenCV juga dapat melakukan operasi canggih seperti deteksi wajah, pengenalan wajah, dan aliran sinar optik, antara lain. AI (Artificial Intelligence) dan ML (Machine Learning).
4. OpenCV juga menyediakan library ML, yang tentunya sangat membantu komputer dalam mendeteksi, mengambil keputusan, dan bertindak terhadap objek.
5. Sampling gambar dan Substraksi: Fitur ini mencakup sampling random, rotasi, substraksi subregion gambar, dan fitur lainnya.
6. Fitur gambar biner OpenCV memungkinkan pembuatan dan analisis gambar biner.

Baik menggunakan kamera stereo maupun kamera single dengan berbagai sudut pandang, fitur pemodelan 3D sangat bermanfaat untuk mapping dan lokalisasi [15].

OpenCV dibuat untuk membantu komputer meniru kemampuan visual manusia. Ini memiliki fitur yang mendukung TensorFlow, Coffee, dan Torch / PyTorch [17]

I. Confusion matrix

Confusion Matrix yaitu metode yang digunakan untuk mengukur suatu kinerja pada metode klasifikasi. *Confusion matrix* mengandung informasi yang membandingkan hasil klasifikasi yang dilakukan oleh sistem dengan hasil klasifikasi

yang seharusnya. Pada matriks tersebut, terdapat empat nilai yang digunakan untuk perhitungan, yaitu:

- ✧ *True Positive (TP).*
- ✧ *True Negative (TN).*
- ✧ *False Positive (FP).*
- ✧ *False Negative (FN).*

Berikut adalah Tabel 2 yang menunjukkan matrix klasifikasi.

Tabel 2 Matriks klasifikasi

Confusion Matrix		True Class	
		Positive	Negative
Predicted Class	Positive	True Positive (TP)	False Positive (FP)
	Negative	False Negative (FN)	True Negative (TN)

Confusion matrix menghitung akurasi dengan menghitung kumulatif dari jumlah prediksi yang benar dibagi dengan jumlah dari prediksi yang benar ditambah dengan prediksi yang salah. Perhitungan dirumuskan sebagai berikut :

$$\text{Akurasi} = \frac{\sum \text{data uji yang dikenali}}{\sum \text{data uji coba}} * 100\%$$

Atau

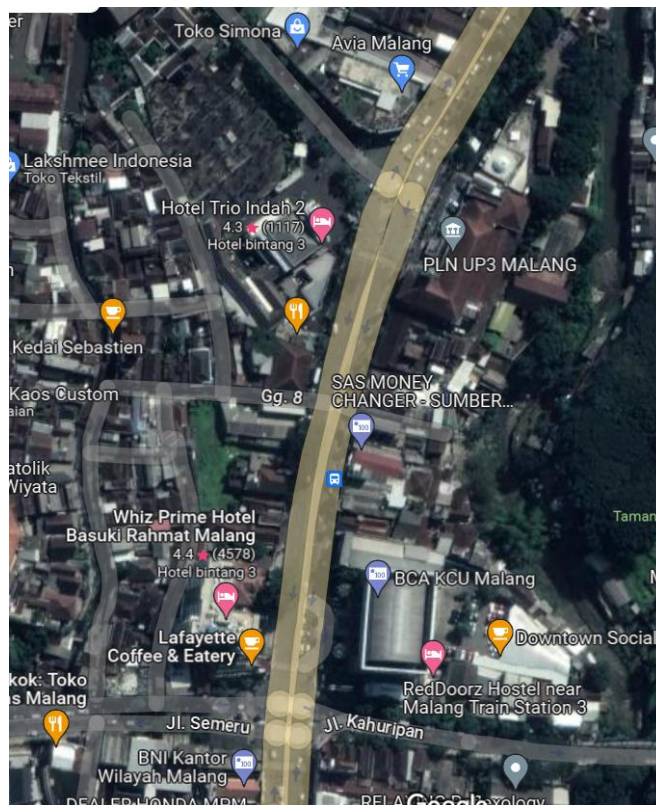
$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} * 100\%$$

III. METODE PENELITIAN

Dibagian ini penulis ingin membahas akan lokasi penelitian serta tahap – tahap metodologi penelitian, perancangan sistem.

A. Lokasi Penelitian

Dua simpang bersinyal yang berdekatan di daerah kayutangan yang mengalami kemacetan pada simpang bersinyal jalan raya malang-gempol dengan jarak 250 meter yaitu pada simpang tiga jalan (Raya Malang-Gempol) – jalan Brigjen Slamet Riadi dan simpang empat jalan (Raya Malang-Gempol) – jalan Kahuripan – jalan Semeru yang merupakan jalur menuju Alun-alun kota Malang dan daerah lain (Yutantinus Adrian,) seperti pada Gambar 3.

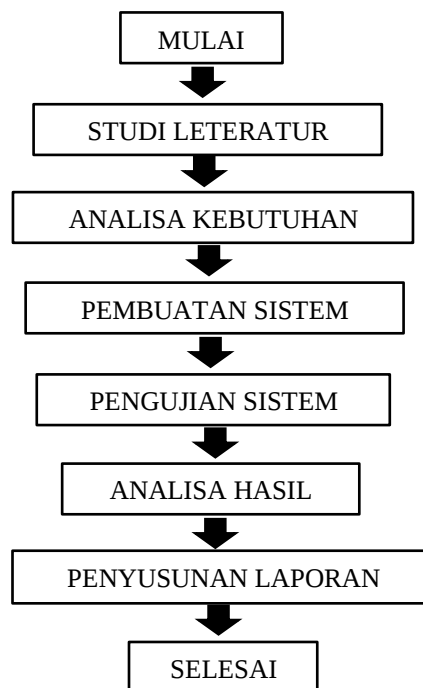


Gambar 3 Lokasi penelitian (sumber dari Google Maps)

Pada Gambar 3 memperlihatkan lokasi penelitian yang bersumber dari google maps tampak atas.

B. Diagram Alir Penelitian

Berikut diagram alir yang menjelaskan tahapan – tahapan pada penelitian yang akan dikerjakan :



Gambar 4 Diagram alir penelitian

Berikut penjelasan dari diagram alir pada Gambar 4 :

Mulai

Studi Literatur

Mencari referensi mengenai deteksi dan klasifikasi objek.

Analisa Kebutuhan

Menganalisa seluruh kebutuhan yang sesuai untuk perencanaan dan pembuatan sistem.

Pembuatan Sistem

Tahap membangun sistem deteksi dengan yolov5

Pengujian Sistem

Tahap pengetesan sistem dijalankan supaya dapat memahami keberhasilan sistem yang di jalankan.

Analisa Hasil

Pada tahap ini dilakukan analisis terhadap informasi yang diperoleh dari hasil pengujian sistem dan kelengkapan secara menyeluruh, sehingga dapat disimpulkan hasil dari penelitian yang telah dilakukan.

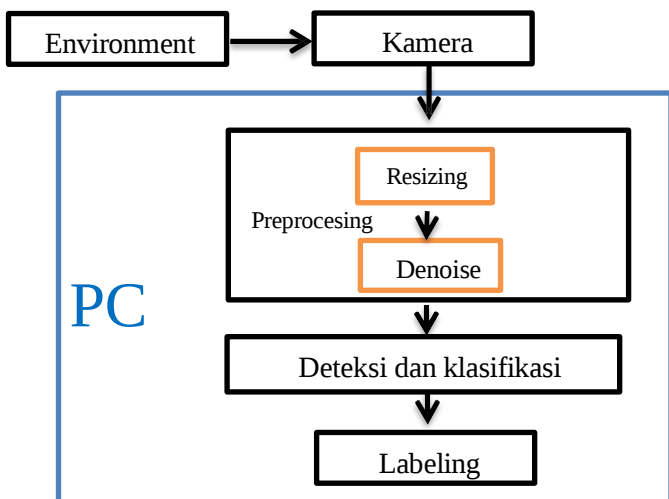
Penyusunan Laporan

Melakukan penyusunan laporan keseluruhan dari hasil riset yang sudah dicoba.

Selesai

C. Perancangan Sistem

Rancangan sistem ini bertujuan untuk membuat sistem deteksi objek berupa kendaraan serta klasifikasi menjadi 3 class, untuk proses kerja sistem akan di jelaskan pada diagram blok di bawah ini

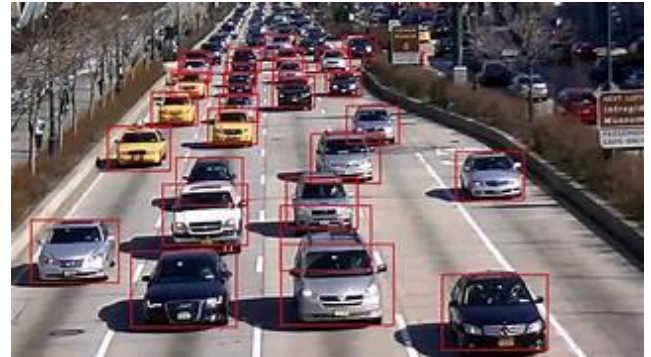


Gambar 5 Diagram blok

Deskripsi diagram blok pada gambar 5 :

- Environment yaitu lingkuan \ tempat objek (lalu lintas)
- Kamera sebagai pengambilan video
- Preprocessing adalah pemrosesan data sebelum dilakukan deteksi dan klasifikasi, dalam prosesnya terdapat dua bagian yaitu resizing dan Denoise.
- Resizing yaitu mengubah ukuran gambar tanpa memotong gambar, yang biasa nya akan mempengaruhi ukuran file dan kualitas gambar.

- Denoise merupakan suatu proses untuk mereduksi atau mengurangi noise pada sebuah citra digital.[18]
- Deteksi dan klasifikasi yaitu proses pendeteksian objek kendaraan setelah itu melakukan proses klasifikasi menjadi 3 (tiga) kelas, yaitu kendaraan ringan (sepeda, becak, motor) panjang tidak lebih dari 2,5 m, kendaraan sedang (mobil penumpang) panjang tidak lebih dari 5,5 m, kendaraan berat (truk, bus) panjang tiak lebih dari 12 m.
- Labeling untuk melakukan pelabelan pada data yang telah dibagi sesuai kelas nya dan muncul di monitor PC seperti gambar 6.



Gambar 6 Contoh lapeblng

IV. HASIL DAN PEMBAHASAN

A. Perlengkapan percobaan

Bagian di sini membahas perihal perlengkapan percobaan yang akan di laksanakan di awali dengan Alat dan bahan sebagai eksekutor dalam penelitian yang dilakukan mulai dari tahap perancangan sampai dengan pelaporan. Berikut adalah alat dan bahan yang digunakan dalam penelitian dapat dilihat pada Tabel 3.

Tabel 3 Perlengkapan

No	Jenis perangkat	Perangkat	Spesifikasi	Keterangan
1.	Perangkat Keras	PC/ Laptop	Intel® Core™ i3- 6006U CPU @ 2.00 GHz RAM 8.00 GB	Perangkat utama untuk pengujian project
2.	Perangkat Lunak	Python	Versi 3.6.8	Bahasa Pemograman utama yang digunakan menglolah seluruh program deteksi objek
		YOLO	V5	Library dari python yang digunakan untuk melatih data dan deteksi objek
		Google Colab	rasi	Perangkat utama untuk pelatihan dan pengujian project
		Roboflow	Web application	Web aplikasi yang digunakan untuk preprocessing data

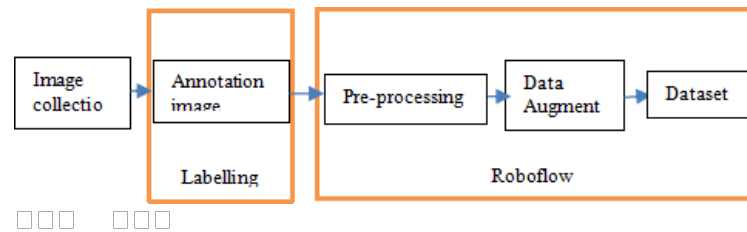
		Open Cv	Versi 4.1.2	Untuk mengolah kamera dan pemrosesan data input
		Matplotlib	3.2.2	Pustaka yang digunakan untuk memvisualisasi data array ke dalam bentuk 2d atau untuk memplot data loss hasil training
		Numpy	1.18.5	Pustaka yang digunakan untuk proses aritmatika dalam bentuk array
		PyYAML	5.3	Pustaka yang digunakan untuk menyimpan file ke dalam format YAML pada python agar file tersebut dapat dibaca manusia
		Scipy	1.4.1	Pustaka python berupa operator scientific yang digunakan untuk cluster
		Torch	1.6.0	Torch merupakan struktur data dan operasi aritmatika untuk tensor multidimensi, dimana modulmodul pada PyTorch digunakan pada YOLOv5
		Anaconda 3	Versi 2020.2.0.0	Software untuk membuat environment project
		Virtual studio		Software yang digunakan untuk membuat halaman report
		Sistem Operasi	Windows 10 64 Bit	Sistem operasi yang digunakan pada perangkat keras.

B. Pembuatan sistem deteksi

Proses pembuatan sistem deteksi objek pada persimpangan bersinyal, pada kasus ini menggunakan dataset kendaraan dan memiliki 3 class atau golongan. Pada proses pertama akan dilakukan membangun dataset, kemudian pada proses kedua akan melakukan training dataset menggunakan model custom yolov5 serta hasil training dataset atau model weight Yolov5 yang akan disave dengan nama best.pt yang berfungsi untuk proses deteksi. Proses ketiga yaitu proses deteksi objek dan klasifikasi class menggunakan model yang telah dilatih.

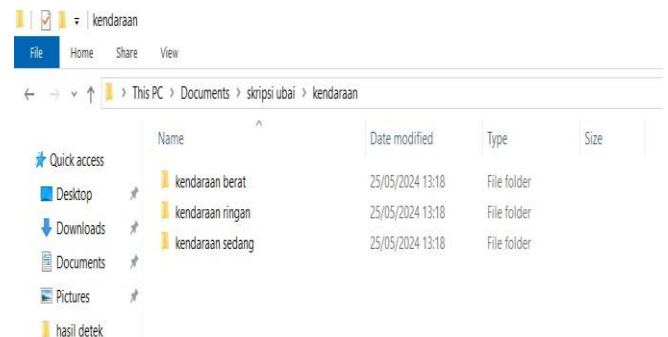
1. Pengumpulan dataset

Dalam proses membangun atau mengumpulkan dataset untuk data latih awal diperlukan Image collection dengan format JPG/Img, kemudian gambar tersebut akan diberi label di setiap gambar menggunakan LabelImg. Hasil dari labelling akan digabung dalam satu folder serta diupload ke dalam roboflow untuk proses pre-processing image atau gambar. Hasil dari pre- processing yaitu berupa file YML. Berikut Diagram Alir Membangun Dataset dapat dilihat pada Gambar 7.



Gambar 7 Diagram alir membangun dataset

Image collection atau pengumpulan gambar yang akan dilabeling berikut contoh pengumpulan gambar yang telah di pisahkan menjadi tiga kelas dapat dilihat pada gambar 8.



Gambar 8 Pengumpulan gambar kendaraan

Pembagian kelas menjadi tiga yaitu kendaraan ringan, kendaraan sedang, kendaraan berat.dan untuk jumlah perkelas nya memiliki 300 gambar seperti tabel 4 dibawah ini.

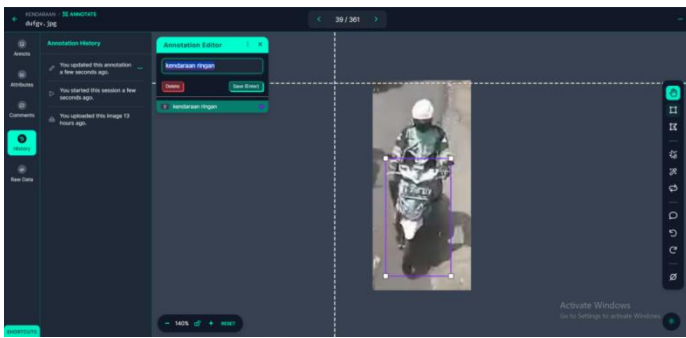
Tabel 4 Jumlah dataset perclass

No	Class	Jumlah data	Contoh
1.	Kendaraan ringan	300 foto jpg	Becak ,sepeda , motor
2.	Kendaraan sedang	300 foto jpg	Mobil, jep
3.	Kendaraan berat	300 foto jpg	Truk , bus

Proses pelabeling menggunakan roboflow :

Kendaraan ringan

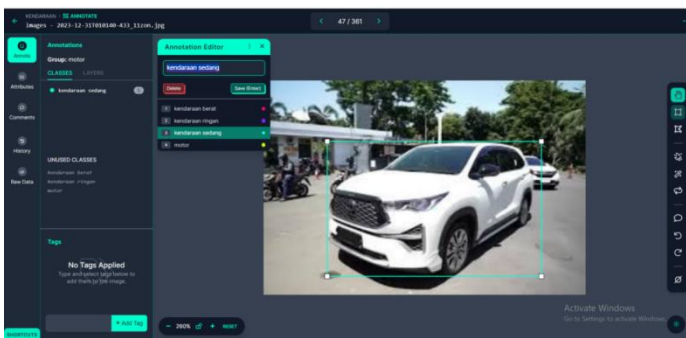
Penentuan kendaraan ringan mengikuti ketentuan dari tabel 1 yaitu Kendaraan roda 2 (dua) dan 3 (tiga) dengan panjang tidak lebih dari 2,5 m, contoh proses pelabelan kendaraan ringan dapat di liat pada gambar 9.



Gambar 9 Pelabelan kendaraan ringan

Kendaraan sedang

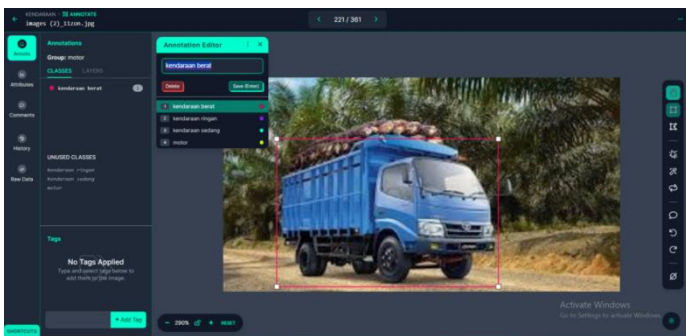
Penentuan kendaraan ringan mengikuti ketentuan dari tabel 1 yaitu mobil penumpang 4 (empat) tempat duduk, mobil penumpang 7 (tujuh) tempat duduk, dengan panjang tidak lebih dari 5,5 m, contoh proses pelabelan kendaraan ringan dapat di lihat pada gambar 10.



Gambar 10 Pelabelan kendaraan sedang

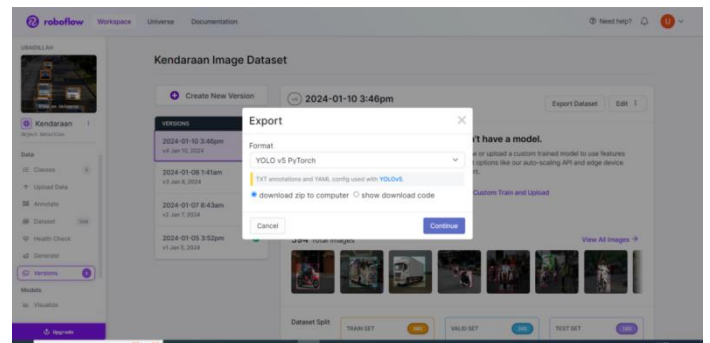
Kendaraan berat

Penentuan kendaraan ringan mengikuti ketentuan dari tabel 1 yaitu Bus sedang dan mobil angkutan barang 2 (dua) sumbu dengan panjang tidak lebih dari 12 m, contoh proses pelabelan kendaraan ringan dapat di lihat pada gambar 11.



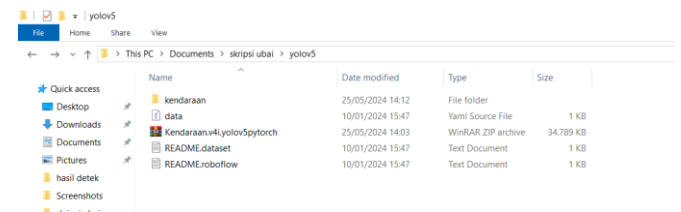
Gambar 11 Pelabelan kendaraan berat.

Setelah melakukan proses pelabelan pada dataset dapat di Export download zip to computer contoh proses download dapat di lihat pada gambar 12.



Gambar 12 Download dataset

Setelah selesai mendownload Dataset zip lalu di extract sehingga menghasilkan file seperti pada gambar 13.

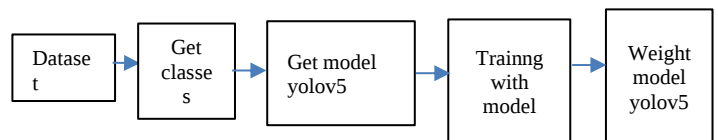


Gambar 13 Extract dataset

Setelah proses pendownloadan dataset maka akan melakukan proses ditraining dataset di google colab.

1. Training dataset

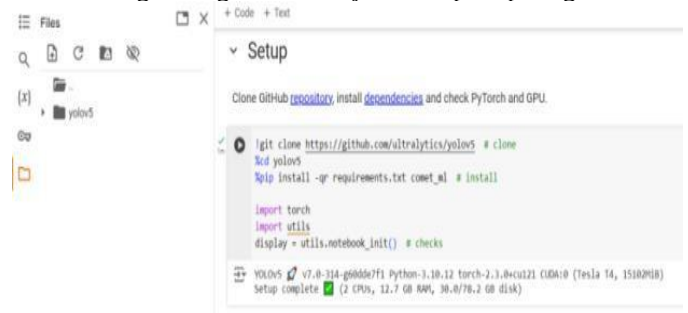
Untuk melakukan training dataset dapat mengikuti tahap-tahapan yang terdapat pada Diagram Alir Training Dataset pada gambar 14.



Gambar 14 Diagram Alir Training Dataset

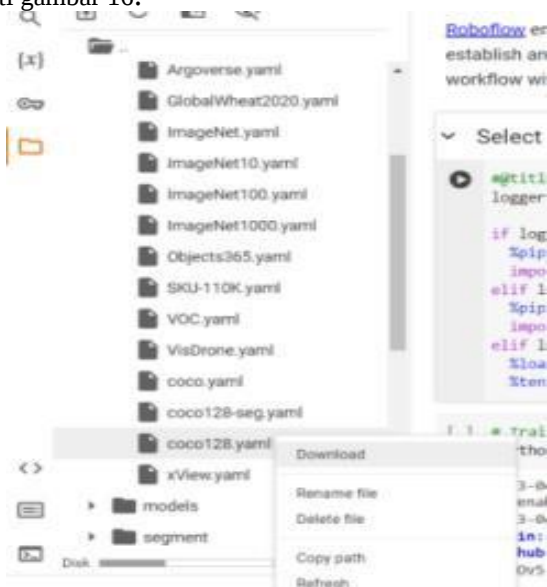
Pada diagram alir training dataset proses awal masukan dataset sehingga akan membentuk class yang berfungsi untuk membuat model yolov5 proses selanjutnya mentraining dataset menggunakan google collaboration.

Untuk melakukan training dataset di google collab terlebih dahulu menghubungkan ke yolov5 seperti pada gambar 15.



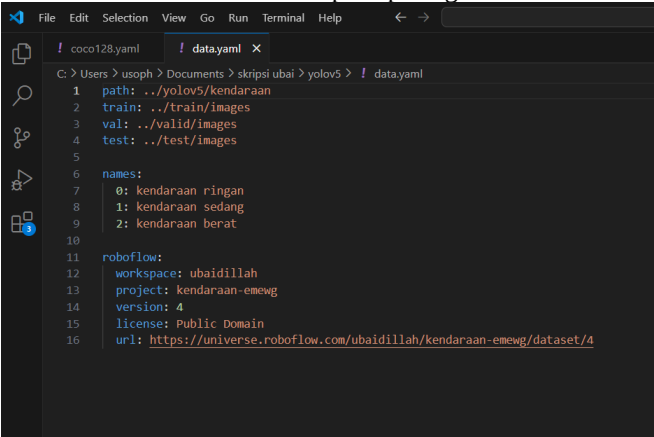
Gambar 15 Penghubungan yolov5

Setelah berhasil terhubung lalu melakukan proses pendownload data coco128, contoh proses pendownloadan seperti gambar 16.



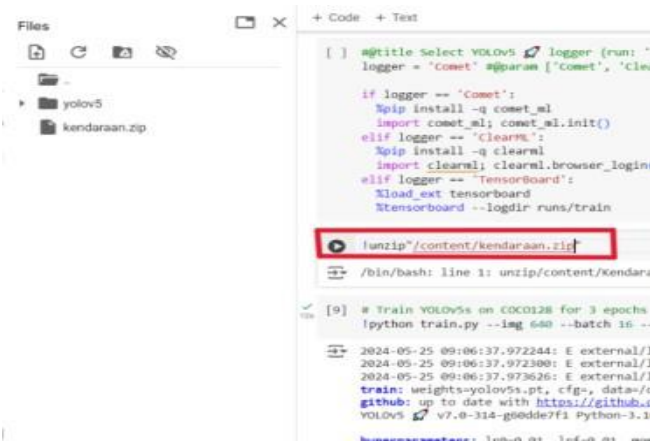
Gambar 16 Proses download data coco128

Setelah mendownload data coco128, lalu membuka data coco128 dan data secara bersamaan untuk mengubah data sesuai data coco128 seperti pada gambar 17.



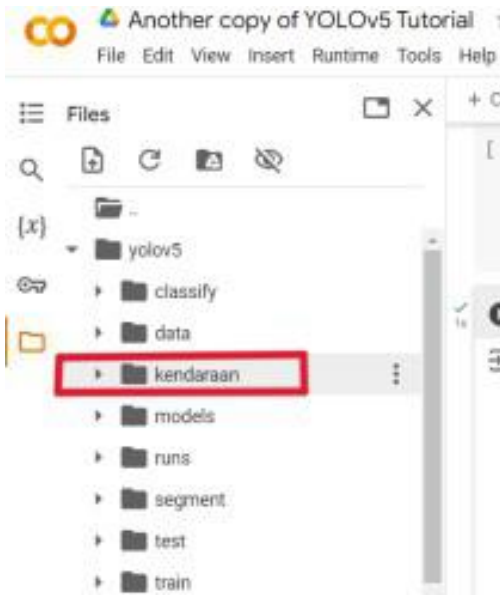
Gambar 17 Penyesuaian data dengan data coco128

Kemudian masukan data kendaraan ke google colab dan beri perintah (!unzip "/content/kendaraan.zip") seperti gambar 18.



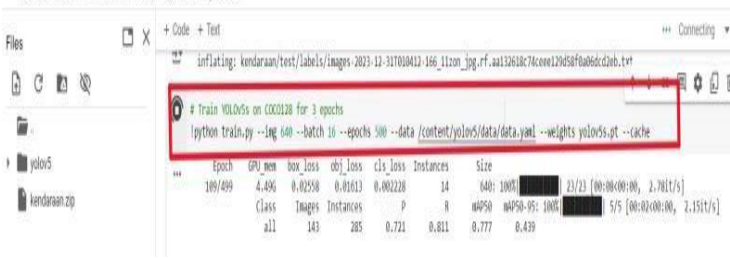
Gambar 18 Proses unzip folder kendaraan

Setelah itu Data kendaraan akan masuk ke dalam data yolov5 seperti gambar 19.



Gambar 19 Setelah melakukan unzip folder kendaraan

Setelah selesai bisa dilakukan proses training seperti gambar 20.

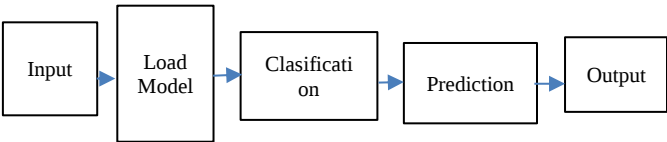


Gambar 20 Proses training dataset

Setelah dilakukan training dataset seperti gambar 20 akan menghasilkan file pt. dilakukan detection objek atau pengujian model, pengujian dapat menggunakan video/gambar.

2. Pengujian sistem deteksi

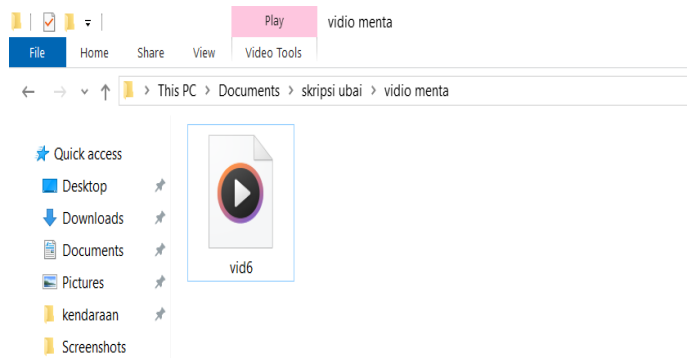
Pada tahap pengujian menggunakan Input berupa video lalu lintas pada persimpangan. Berikut Diagram Alir Pengujian terdapat pada gambar 21.



Gambar 21 Diagram Alir Pengujian

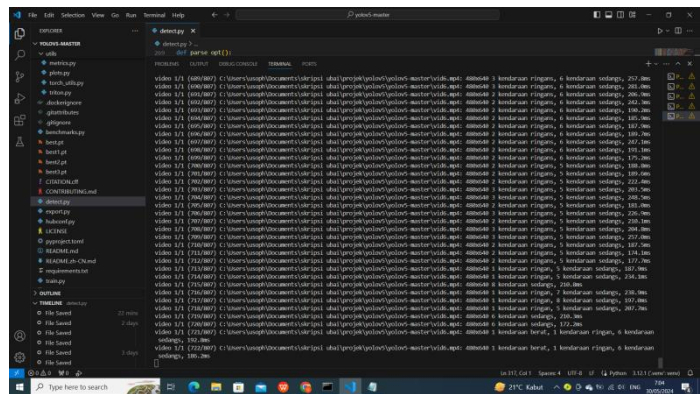
Pada diagram alir pengujian terdapat beberapa proses, Proses awal masukan input kemudian meload model lalu akan memproses klasifikasi class serta memunculkan nilai prediksi yang berupa bounding box tiap class nya

Berikut Input vidio lalu lintas yang akan digunakan sebagai objek pendeteksiian pada gambar 22.



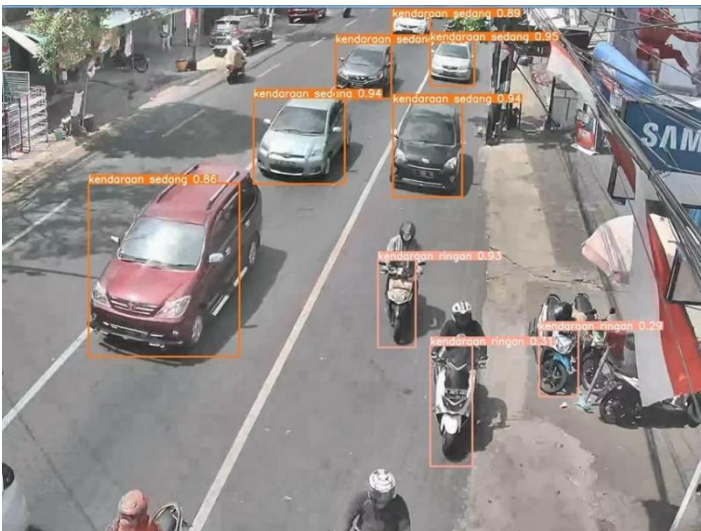
Gambar 22 Input vidio

Setelah menyelesaikan semua proses pengumpulan dataset, training dataset dan mendapatkan file.pt maka dapat selakukan proses akhir yaitu proses pendeksiian kendaraan. Berikut proses pendeteksiian kendaraan terhadap input vidio dapat di lihat pada gambar 23.



Gambar 23 Proses pendeteksiian

Berikut Hasil pendeteksiian kendaraan menggunakan yolov5 dapat dilihat pada gambar 24.



Gambar 24 Hasil deteksi dan klasifikasi kendaraan

Berdasarkan gambar 24 hasil deteksi berhasil dengan sangat baik dan berhasil mendeteksi kendaraan sesuai kelas yang ditentukan yaitu kendaraan ringan yang ditunjang dengan label berwarna merah mudah sedangkan kendaraan sedang berwarna jingga, meskipun mendeteksi kendaraan ringan yang sedang parkit, atau di luar jalan lalu lintas.

C. Tabel perbandingan

Perbandingan dilakukan sebanyak tiga kali menggunakan perbedaan jumlah dataset yang berjumlah 150 dataset, 300 dataset, 450 dataset, 600 dataset, 750 dataset, dan 900 dataset.

Berikut hasil perbandingan yang telah dilakukan.

Perbandingan error pada kelas kendaraan ringan dapat dilihat pada tabel 5.

Tabel 5 Perbandingan hasil deteksi kendaraan ringan.

Jumlah dataset	Epoch	Kendaraan ringan	Error
150 foto jpg	500 kali	57	11
300 foto jpg	500 kali	57	10
450 foto jpg	500 kali	57	8
600 foto jpg	500 kali	57	5
750 foto jpg	500 kali	57	3
900 foto jpg	500 kali	57	0

Perbandingan error pada kelas kendaraan sedang dapat dilihat pada tabel 6.

Tabel 6 Perbandingan hasil deteksi kendaraan sedang

Jumlah dataset	Epoch	Kendaraan sedang	Error
150 foto jpg	500 kali	20	6
300 foto jpg	500 kali	20	5
450 foto jpg	500 kali	20	3
600 foto jpg	500 kali	20	2
750 foto jpg	500 kali	20	1
900 foto jpg	500 kali	20	0

Perbandingan error pada kelas kendaraan sedang dapat dilihat pada tabel 7.

Tabel 7 Perbandingan hasil deteksi kendaraan berat

Jumlah dataset	Epoch	Kendaraan berat	Error
150 foto jpg	500 kali	0	21
300 foto jpg	500 kali	0	19
450 foto jpg	500 kali	0	14
600 foto jpg	500 kali	0	10
750 foto jpg	500 kali	0	8
900 foto jpg	500 kali	0	4

Berdasarkan Tabel diatas dapat di simpulkan bahwa jumlah dataset sangat mempengaruhi hasil deteksi yang di lakukan semakin banyak dataset yang di siap kan maka hasil deteksi akan semakin sedikit error atau kendaraan yang tidak kedeteksi.

D. Perhitungan Confusion matrix

- Skenario 1

digunakan 150 data dari Dataset yang terdiri dari 50 data pada setiap kelas dan epoch 500, Proses pengujian dilakukan pada setiap dataset secara individu. Hasil pengujian tersebut kemudian dicatat dan dapat dilihat dalam tabel di bawah ini.

Tabel 8 Hasil Prediksi Proses Skenario 1

Matrix		Prediksi objek	
		benar	salah
objek	Kendaraan ringan	39	11
	Kendaraan sedang	44	6
	Kendaraan berat	29	21

Berdasarkan Tabel 8 prediksi di atas, dapat disimpulkan bahwa hasilnya cukup baik. Untuk menghitung akurasi, dapat digunakan confusion matrix seperti di bawah ini.

$$Accuracy = \frac{\text{jumlah predik benar}}{\text{jumlah data}} * 100\%$$

$$Accuracy = \frac{112}{150} * 100\%$$

$$Accuracy = 74,6\%$$

Dari hasil perhitungan dengan menggunakan confusion matrix, diperoleh nilai akurasi deteksi kendaraan sebesar 74,6%.

- Skanario 2

Digunakan 300 dataset yang terdiri dari 100 data pada tiap kelas dan epoch 500. Proses pengujian dilakukan pada setiap dataset secara individu. Hasil pengujian tersebut kemudian dicatat dan dapat dilihat dalam tabel di bawah ini.

Tabel 9 Hasil Prediksi Proses Skenario 2

Matrix		Prediksi objek	
		benar	salah
objek	Kendaraan ringan	90	10
	Kendaraan sedang	95	5

	Kendaraan berat	81	19
--	-----------------	----	----

Berdasarkan Tabel 9 prediksi di atas, dapat disimpulkan bahwa hasilnya cukup baik. Untuk menghitung akurasi, dapat digunakan confusion matrix seperti di bawah ini.

$$Accuracy = \frac{\text{jumlah predik benar}}{\text{jumlah data}} * 100\%$$

$$Accuracy = \frac{266}{300} * 100\%$$

$$Accuracy = 88,6\%$$

Dari hasil perhitungan dengan menggunakan confusion matrix, diperoleh nilai akurasi deteksi kendaraan sebesar 88,6%.

- Skanario 3

Digunakan 450 dataset yang terdiri dari 150 data pada tiap kelas dan epoch 500. Proses pengujian dilakukan pada setiap dataset secara individu. Hasil pengujian tersebut kemudian dicatat dan dapat dilihat dalam tabel di bawah ini.

Tabel 10 Hasil Prediksi Proses Skenario 3

Matrix		Prediksi objek	
		benar	salah
objek	Kendaraan ringan	142	8
	Kendaraan sedang	147	3
	Kendaraan berat	136	14

Berdasarkan Tabel 10 prediksi di atas, dapat disimpulkan bahwa hasilnya cukup baik. Untuk menghitung akurasi, dapat digunakan confusion matrix seperti di bawah ini.

$$Accuracy = \frac{\text{jumlah predik benar}}{\text{jumlah data}} * 100\%$$

$$Accuracy = \frac{425}{450} * 100\%$$

$$Accuracy = 94,4\%$$

Dari hasil perhitungan dengan menggunakan confusion matrix, diperoleh nilai akurasi deteksi kendaraan sebesar 94,4%.

- Skanario 4

Digunakan 600 dataset yang terdiri dari 200 data pada tiap kelas dan epoch 500. Proses pengujian dilakukan pada setiap dataset secara individu. Hasil pengujian tersebut kemudian dicatat dan dapat dilihat dalam tabel di bawah ini.

Tabel 11 Hasil Prediksi Proses Skenario 4

Matrix		Prediksi objek	
		benar	salah
objek	Kendaraan ringan	195	5

	Kendaraan sedang	198	2
	Kendaraan berat	190	10

Berdasarkan Tabel 11 prediksi di atas, dapat disimpulkan bahwa hasilnya cukup baik. Untuk menghitung akurasi, dapat digunakan confusion matrix seperti di bawah in.

$$\text{Accuracy} = \frac{\text{jumlah predik benar}}{\text{jumlah data}} * 100\%$$

$$\text{Accuracy} = \frac{583}{600} * 100\%$$

$$\text{Accuracy} = 97,1\%$$

Dari hasil perhitungan dengan menggunakan confusion matrix, diperoleh nilai akurasi deteksi kendaraan sebesar 97,1%.

- Skanario 5

Digunakan 750 dataset yang terdiri dari 250 data pada tiap kelas dan epoch 500. Proses pengujian dilakukan pada setiap dataset secara individu. Hasil pengujian tersebut kemudian dicatat dan dapat dilihat dalam tabel di bawah ini.

Tabel 12 Hasil Prediksi Proses Skenario 5

Matrix		Prediksi objek	
		benar	salah
objek	Kendaraan ringan	247	3
	Kendaraan sedang	249	1
	Kendaraan berat	242	8

Berdasarkan Tabel 12 prediksi di atas, dapat disimpulkan bahwa hasilnya cukup baik. Untuk menghitung akurasi, dapat digunakan confusion matrix seperti di bawah in.

$$\text{Accuracy} = \frac{\text{jumlah predik benar}}{\text{jumlah data}} * 100\%$$

$$\text{Accuracy} = \frac{738}{750} * 100\%$$

$$\text{Accuracy} = 98,4\%$$

Dari hasil perhitungan dengan menggunakan confusion matrix, diperoleh nilai akurasi deteksi kendaraan sebesar 98,4%.

- Skanario 6

Digunakan 900 dataset yang terdiri dari 300 data pada tiap kelas dan epoch 500. Proses pengujian dilakukan pada setiap dataset secara individu. Hasil pengujian tersebut kemudian dicatat dan dapat dilihat dalam tabel di bawah ini.

Tabel 13 Hasil Prediksi Proses Skenario 6

Matrix		Prediksi objek	
		benar	salah
objek	Kendaraan ringan	300	0
	Kendaraan sedang	300	0
	Kendaraan berat	294	4

Berdasarkan Tabel 13 prediksi di atas, dapat disimpulkan bahwa hasilnya cukup baik. Untuk menghitung akurasi, dapat digunakan confusion matrix seperti di bawah in.

$$\text{Accuracy} = \frac{\text{jumlah predik benar}}{\text{jumlah data}} * 100\%$$

$$\text{Accuracy} = \frac{895}{900} * 100\%$$

$$\text{Accuracy} = 99,5\%$$

Dari hasil perhitungan dengan menggunakan confusion matrix, diperoleh nilai akurasi deteksi kendaraan sebesar 99,5%.

Hasil dari perhitingan Confusion matrix Skanario 1, Skanario 2, Skanario 3, Skanario 4, Skanario 5, dan Skanario 6 menghasilkan accuracy yang cukup baik serta berbeda-beda dikarenakan dataset yang digunakan berjumlah berbeda. Dapat dilihat pada Tabel 14.

Tabel 14 hasil perhitungan accuracy

	dataset	accuracy
Skanario 1	150	74,6%
Skanario 2	300	88,6%
Skanario 3	450	94,4%
Skanario 4	600	97,1%
Skanario 5	750	98,4%
Skanario 6	900	99,5%

Berdasarkan Tabel 14 dapat mengambil kesimpulan bahwa jumlah dataset sangat mempengaruhi hasil accuracy, semakin banyak dataset maka accuracy semakin tinggi dan semakin sedikit jumlah dataset maka accuracy akan rendah. Seperti skanario 1 yang memiliki dataset 150 menghasilkan accuracy sebesar 74,6%, sedangkan skanario 3 yang memiliki dataset 900 menghasilkan accuracy lebih tinggi sebesar 99,5%.

V. KESIMPULAN

A. Kesimpulan

Berdasarkan hasil penelitian yang telah dilakukan, maka kesimpulan yang didapat yaitu:

1. Sistem deteksi dan klasifikasi kendaraan berhasil mendeteksi dan klasifikasi jenis class yang diharapkan dan berhasil menyimpan hasil deteksi.
2. Semakin banyak dataset maka semakin bagus dan konsisten hasil deteksi klasifikasi kendarannya.
3. Berdasarkan perhitungan menggunakan confutson matrix, Sistem deteksi dan klasifikasi jenis kendaraan pada persimpangan bersinyal memiliki nilai akurasi sebesar 74,6%, memiliki nilai akurasi sebesar 88,6% pada 300 dataset, memiliki nilai akurasi sebesar 94,4% pada 450

dataset, memiliki nilai akurasi sebesar 97,2% pada 500 dataset, memiliki nilai akurasi sebesar 98,4 pada dataset 750. dan memiliki nilai akurasi sebesar 99,5%.

B. Saran

Berdasarkan penelitian yang telah dilakukan, terdapat beberapa saran untuk pengembangan sistem selanjutnya, antara lain:

1. Melakukan pengembangan dataset sehingga sistem dapat mendeteksi dan klasifikasi jenis kendaraan lebih baik.
2. Melakukan pengembangan pada sistem deteksi dan klasifikasi jenis kendaraan supaya dapat dilakukan pendeteksian secara real-time.

VI. REFERENSI

- [1] M. B. M. Abi Rachman Warsil, M. S. (2017). "(Yolo) Untuk Automated Teller Machine (Atm).", Maj. Ilm. UNIKOM, vol. 17, 69–76.
- [2] A. Asni b and T. Sihotang, "Ekstraksi Ciri Menggunakan Algoritme Discrete Wavelet Transform (Dwt) Dan Principal Component Analysis (Pca) Pada Warna Kulit Wajah," J. Tek. Elektro Uniba (JTE Uniba), vol. 3, no. 2, pp. 19–23, 2019.
- [3] M. Affandes and A. Ramadani, "Deteksi Gerak Menggunakan Background Subtraction Dan Deteksi Tepi Sobel," Semin. Nas. Teknol. dan Rekayasa, no. 255, pp. 1–6, 2017.
- [4] L. M. rifatul Azizah, S. F. Umayah, S. Riyadi, C. Damarjati, and N. A. Utama, "Deep learning implementation using convolutional neural network in mangosteen surface defect detection," Proc. - 7th IEEE Int. Conf. Control Syst. Comput. Eng. ICCSCE 2017, vol. 2017-Novem, no. November, pp. 242–246, 10.1109/ICCSCE.2017.8284412.
- [5] H. Ji, Z. Gao, T. Mei, and B. Ramesh, "Vehicle Detection in Remote Sensing Images Leveraging on Simultaneous Super-Resolution," IEEE Geosci. Remote Sens. Lett., vol. 17, no. 4, pp. 676–680, 2020, 10.1109/LGRS.2019.2930308.
- [6] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, Real Time Object Detection.," IEEE Conf. Comput. Vis. Pattern Recognit., pp. 779–788, 2016.
- [7] S. Jupiyandi, F. R. Saniputra, Y. Pratama, M. R. Dharmawan, and I. Cholissodin, "Pengembangan Deteksi Citra Mobil untuk Mengetahui Jumlah Tempat Parkir Menggunakan CUDA dan Modified YOLO," J. Teknol. Inf. dan Ilmu Komput., vol. 6, no. 4, p. 413, 2019, doi: 10.25126/jtiik.2019641275.
- [8] R. Himilda and R. A. Johan, "Klasifikasi Jenis Kendaraan Menggunakan Metode Extreme Learning Machine (Vehicle Classification Using Extreme Learning Machine Method)," vol. 2, no. 4, pp. 237–243, 2021
- [9] W. Swastika, A. Kurniawan, and H. Setiawan, "Deteksi dan Klasifikasi Merek Mobil untuk Penentuan Iklan Billboard Menggunakan Convolution Neural Network," J. Teknol. Inf. dan Ilmu Komput., vol. 7, no. 4, p. 701, 2020, doi: 10.25126/jtiik.2020742183.
- [10] J. S. W. Hutaeruk, T. Matulatan, and N. Hayaty, "Deteksi Kendaraan secara Real Time menggunakan Metode YOLO Berbasis Android," J. Sustain. J. Has. Penelit. dan Ind. Terap., vol. 9, no. 1, pp. 8–14, 10.31629/sustainable.v9i1.1401, 2020, doi:
- [11] F. T. A. Muhammad Alfin Jimly Asshiddiqie, Basuki Rahmat, " Deteksi Tanaman Tebu Pada Lahan Pertanian Menggunakan Convolutional Neural" Inform. dan Sist. Inf., vol. 1, no. 1, pp. 229–237, 2020
- [12] K. A. Shianto, K. Gunadi, and E. Setyati, "Deteksi Jenis Mobil Menggunakan Metode YOLO Dan Faster R-CNN," J. Infra, vol. 7, no. 1, pp. 157–163, 2019.
- [13] A. Asni b and T. O. Dana, "Identifikasi Wajah Dengan Segmentasi Warna Kulit Menggunakan Metode Viola Jones," J. Tek. Elektro Uniba (JTE Uniba), vol. 4, no. 1, pp. 1–6, 2019, doi: 10.36277/jteuniba.v4i1.47.
- [14] B. Pradana, "Implementasi metode low pass filtering untuk mereduksi noise pada objek citra digital," vol. 9, pp. 93–99, 2015.
- [15] T. C. A. Zulkhaidi and E. Maria, "Pengenal Pola Bentuk Wajah dengan OpenCV," J. Teknol. Inf., vol. 3, no. 2, pp. 181–186, 2019.
- [16] M. Muthi'ah, "Maratul Muthi'ah-Rancang Bangun Database dan Migrasi Data Melalui teknik ORM untuk Aplikasi Dashboard Pemantauan Mytra," Tek. Inform., vol. 1, 2015.
- [17] Munir, R. 2004. Pengolahan Citra Digital. Informatika. Bandung

VII. BIODATA PENULIS

Ubaidillah alhabsyi, lahir di Jember 25 Desember 1997. Penulis menyelesaikan Pendidikan Menengah Kejuruan di SMK Darussholihin tahun 2017. kemudian 2019 penulis melanjutkan studi diperguruan swasta Institut Teknologi Nasional Malang prodi Teknik Elektro S-1 dengan memilih konsentrasi Teknik Elektronika. Akhir kata penulis mengucapkan rasa syukur yang sebesar - besarnya atas terselesaikannya skripsi ini yang berjudul "Sistem deteksi dan klasifikasi jenis kendaraan pada persimpangan bersinyal"

