

**Pengenalan Rambu Lalu Lintas Menggunakan Model You
Only Look Once (YOLO) v8**

Laporan Tugas Akhir

Laporan Tugas Akhir (TA) ini disusun sebagai salah satu syarat untuk memperoleh gelar Sarjana (S1) pada Program Studi Teknik Informatika Universitas Islam Sultan Agung Semarang



DISUSUN OLEH:

JESITA REINANDRA PRIANDINI

NIM 32602000035

PROGRAM STUDI TEKNIK INFORMATIKA

FAKULTAS TEKNOLOGI INDUSTRI

UNIVERSITAS ISLAM SULTAN AGUNG

SEMARANG

2024

FINAL PROJECT

***TRAFFIC SIGNS RECOGNITION USING THE YOU ONLY LOOK ONE
(YOLO) V8 MODEL***

*This Final Assignment Report was prepared as one of the requirements for
obtaining a Bachelor's Degree (S1) in the Informatics Engineering Study
Program, Sultan Agung Islamic University, Semarang.*



Arranged By:

JESITA REINANDRA PRIANDINI

NIM 32602000035

MAJORING OF INFORMATICS ENGINEERING

INDUSTRIAL TECHNOLOGY FACULTY

SULTAN AGUNG ISLAMIC UNIVERSITY

SEMARANG

2024

LEMBAR PENGESAHAN PEMBIMBING

Laporan Tugas Akhir dengan judul "Pengenalan Rambu Lalu Lintas Menggunakan Model You Only Look Once (YOLO) V8" ini disusun oleh :

Nama : Jesita Reinandra Priandini

NIM : 32602000035

Program Studi : Teknik Informatika

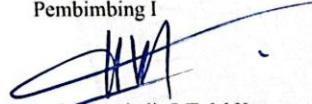
Telah disahkan oleh dosen pembimbing pada :

Hari : Selasa

Tanggal : 20 Agustus 2024

Mengesahkan,

Pembimbing I



Dedy Kurniadi, S.T., M.Kom
NIK. 210615048

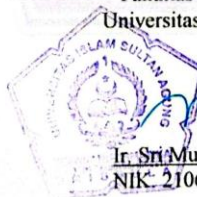

Pembimbing II



Ir. Sri Mulyono, M.Eng
NIK. 210616049

Mengetahui,

Ketua Program Studi Teknik Informatika
Fakultas Teknologi Industri
Universitas Islam Sultan Agung



Ir. Sri Mulyono, M.Eng
NIK. 210616049

LEMBAR PENGESAHAN PENGUJI

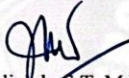
Laporan tugas akhir dengan judul “Pengenalan Rambu Lalu Lintas Menggunakan Model You Only Look Once (YOLO) V8” ini telah dipertahankan di depan dosen penguji Tugas Akhir pada :

Hari : Jum'at


Tanggal : 16 Agustus 2024

TIM PENGUJI

Penguji I


Badieyah, S.T.,M.Kom
NIK. 210615044

Penguji II


Imam Much Ibnu.S.,S.T.,M.Sc.,Ph.D
NIK. 210600017

SURAT PERNYATAAN KEASLIAN TUGAS AKHIR

Yang bertanda tangan dibawah ini :

Nama : Jesita Reinandra Priandini

NIM : 32602000035

Judul Tugas Akhir : Pengenalan Rambu Lalu Lintas Menggunakan Model You Only Look Once (YOLO) V8

Dengan bahwa ini saya menyatakan bahwa judul dan isi Tugas Akhir yang saya buat dalam rangka menyelesaikan Pendidikan Strata Satu (S1) Teknik Informatika tersebut adalah asli dan belum pernah diangkat, ditulis ataupun dipublikasikan oleh siapapun baik keseluruhan maupun sebagian, kecuali yang secara tertulis diacu dalam naskah ini dan disebutkan dalam daftar pustaka, dan apabila di kemudian hari ternyata terbukti bahwa judul Tugas Akhir tersebut pernah diangkat, ditulis ataupun dipublikasikan, maka saya bersedia dikenakan sanksi akademis. Demikian surat pernyataan ini saya buat dengan sadar dan penuh tanggung jawab.

Semarang, 31 Juli 2024

Yang Menyatakan,



Jesita Reinandra Priandini

PERNYATAAN PERSETUJUAN PUBLIKASI KARYA ILMIAH

Saya yang bertanda tangan dibawah ini :

Nama : Jesita Reinandra Priandini

NIM : 32602000035

Program Studi : Teknik Informatika

Fakultas : Teknologi Industri

Dengan ini menyatakan Karya Ilmiah berupa Tugas akhir dengan Judul : Pengenalan Rambu Lalu Lintas Menggunakan Model You Only Look Once (YOLO) V8. Menyetujui menjadi hak milik Universitas Islam Sultan Agung serta memberikan Hak bebas Royalti Non-Eksklusif untuk disimpan, dialihmediakan, dikelola dan pangkalan data dan dipublikasikan diinternet dan media lain untuk kepentingan akademis selama tetap menyantumkan nama penulis sebagai pemilik hak cipta. Pernyataan ini saya buat dengan sungguh-sungguh. Apabila dikemudian hari terbukti ada pelanggaran Hak Cipta/Plagiarisme dalam karya ilmiah ini, maka segala bentuk tuntutan hukum yang timbul akan saya tanggung secara pribadi tanpa melibatkan Universitas Islam Sultan agung.

Semarang, 30 Juli 2024


METERAI
TEMPEL
AC35FALX323254236
Jesita Reinandra Priandini

KATA PENGANTAR

Dengan mengucap syukur atas kehadiran Allah SWT yang telah memberikan rahmat dan karunia-Nya kepada penulis, sehingga dapat menyelesaikan Tugas Akhir dengan judul “Pengenalan Rambu Lalu Lintas Menggunakan Model You Only Look Once (YOLO) V8” ini guna memenuhi salah satu syarat menyelesaikan studi serta dalam memperoleh gelar sarjana (S1) pada Program Studi Teknik Informatika Fakultas Teknologi Industri Universitas Islam Sultan Agung Semarang.

Tugas Akhir ini disusun dan dibuat dengan adanya dukungan dari berbagai pihak, materi maupun teknis, oleh karena itu saya selaku penulis mengucapkan terima kasih kepada:

1. Rektor UNISSULA Bapak Prof. Dr. H. Gunarto, S.H., M.H yang mengizinkan penulis menimba ilmu di kampus ini.
2. Dekan Fakultas Teknologi Industri Ibu Dr. Ir. Novi Marlyana, S.T., M.T., IPU., ASEAN.Eng.
3. Dosen Pembimbing I penulis Bapak Dedy Kurniadi, S.T., M.Kom yang telah meluangkan waktu dan memberi ilmu dalam penyusunan tugas akhir ini.
4. Dosen Pembimbing II penulis Bapak Ir. Sri Mulyono, M.Eng yang telah memberikan bimbingan, dan saran dalam penyusunan tugas akhir ini.
5. Orang Tua penulis yang selalu memberikan doa, dukungan, dan motivasi agar tugas akhir ini berjalan dengan lancar.
6. Dan kepada semua pihak yang tidak dapat penulis sebutkan satu persatu.

Dengan segala kerendahan hati, penulis menyadari masih banyak kekurangan dari segi kualitas dan kuantitas maupun dari ilmu pengetahuan dalam penyusunan laporan, sehingga penulis mengharapkan adanya saran dan kritikan yang bersifat membangun demi kesempurnaan laporan ini di masa mendatang.

Semarang, 28 Juni 2024

Jesita Reinandra Priandini

DAFTAR ISI

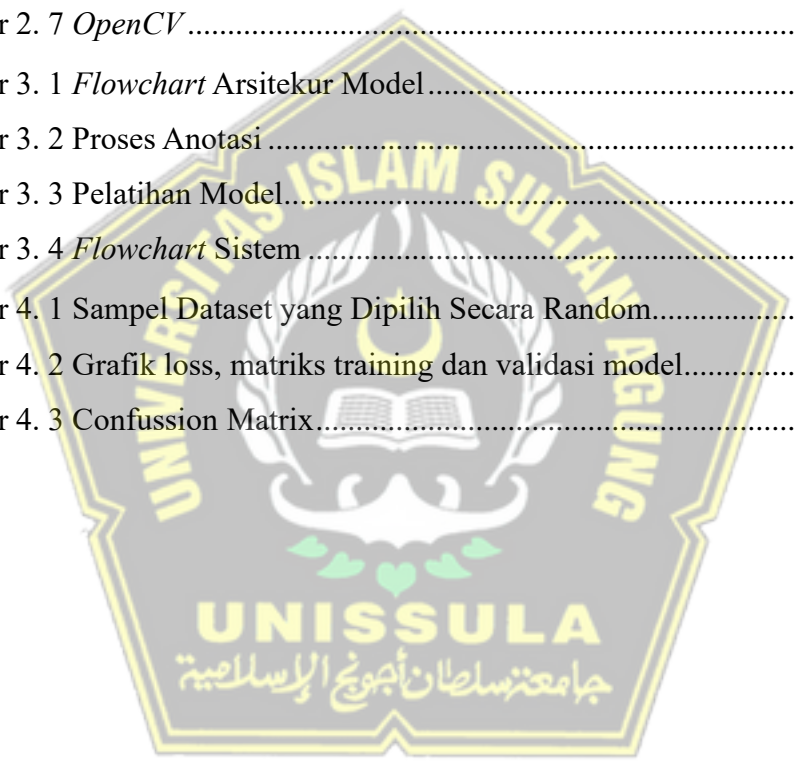
HALAMAN JUDUL	I
LEMBAR PENGESAHAN PEMBIMBING	III
LEMBAR PENGESAHAN PENGUJI.....	IV
SURAT PERNYATAAN KEASLIAN TUGAS AKHIR.....	V
SURAT PERSETUJUAN PUBLIKASI KARYA ILMIAH	VI
KATA PENGANTAR.....	VII
DAFTAR ISI.....	VIII
DAFTAR GAMBAR	X
DAFTAR TABEL	XI
ABSTRAK	XII
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Pembatasan Masalah	3
1.4 Tujuan Tugas Akhir	3
1.5 Manfaat.....	3
1.6 Sistematika Penulisan.....	3
BAB II TINJAUAN PUSTAKA DAN DASAR TEORI.....	5
2.1 Tinjauan Pustaka.....	5
2.2 Dasar Teori	7
2.2.1 Pengolahan Citra	7
2.2.2 Mobil <i>Autonomous</i>	8
2.2.3 Rambu Lalu Lintas	9
2.2.4 <i>Deep learning</i>	12
2.2.5 Deteksi Objek.....	13
2.2.6 <i>Convolutional Neural Network (CNN)</i>	13
2.2.7 <i>You Only Look Once (YOLO) V8</i>	17
2.2.8 <i>Confussion Matrix</i>	21
2.2.9 <i>OpenCV</i>	22
BAB III METODOLOGI PENELITIAN	24

3.1.	Metode Penelitian.....	24
3.2.	Studi Literatur.....	24
3.3.	Pengumpulan Data	24
3.4.	Metode Perancangan Arsitektur Model.....	25
3.5.	Analisis Kebutuhan	28
3.6.	Analisis Sistem.....	31
BAB IV HASIL ANALISIS PENELITIAN		32
4.1.	Hasil Pengumpulan Data	32
4.2.	Hasil Evaluasi.....	36
4.3.	Implementasi <i>Real-Time</i>	39
BAB V KESIMPULAN DAN SARAN		43
5.1.	Kesimpulan.....	43
5.2.	Saran.....	43
DAFTAR PUSTAKA.....		44



DAFTAR GAMBAR

Gambar 2. 1 Arsitektur CNN	13
Gambar 2. 2 <i>Convolutional Layer</i>	14
Gambar 2. 3 <i>Pooling Layer</i>	15
Gambar 2. 4 <i>Fullyconnected Layer</i>	16
Gambar 2. 5 Arsitektur YOLOv8.....	18
Gambar 2. 6 Hasil Penerapan YOLOv8.....	20
Gambar 2. 7 <i>OpenCV</i>	22
Gambar 3. 1 <i>Flowchart</i> Arsitektur Model.....	25
Gambar 3. 2 Proses Anotasi	26
Gambar 3. 3 Pelatihan Model.....	27
Gambar 3. 4 <i>Flowchart</i> Sistem	31
Gambar 4. 1 Sampel Dataset yang Dipilih Secara Random.....	35
Gambar 4. 2 Grafik loss, matriks training dan validasi model.....	37
Gambar 4. 3 Confussion Matrix.....	38



DAFTAR TABEL

Tabel 2. 1 Rambu lalu lintas.....	9
Tabel 4. 1 Jenis Rambu Dataset	32
Tabel 4. 2 Pembagian Dataset	35
Tabel 4. 3 Hasil Evaluasi.....	36
Tabel 4. 4 Hasil Implementasi.....	39



ABSTRAK

Mobil *Autonomous* merupakan kendaraan yang dapat berjalan secara mandiri tanpa bantuan manusia. Akan tetapi mobil ini memiliki kelemahan dalam mendeteksi rambu lalu lintas. Pengenalan rambu lalu lintas ini dimaksudkan untuk menambah keamanan mobil *autonomous* agar dapat mengenali rambu lalu lintas yang dilewatinya. Teknologi ini menggunakan model YOLOv8 yang merupakan pengembangan dari metode *Convolutional Neural Network* untuk mendeteksi dan mengklasifikasi rambu-rambu lalu lintas. Model YOLOv8 ini dipilih karena mempunyai nilai akurasi dan efisiensi yang tinggi. Penelitian ini menggunakan *dataset* dari *Roboflow* yang berjumlah 2390 gambar yang terdiri dari 17 jenis rambu lalu lintas di Indonesia. Hasil pelatihan pada penelitian ini menunjukkan model ini berkinerja secara optimal dengan nilai akurasi sebesar 97.90%, nilai *precision* senilai 0.978, *recall* senilai 0.989, MAP50 senilai 0.987, dan MAP50-95 senilai 0.825. Nilai tersebut menyatakan bahwasannya model dapat mengidentifikasi dan mengklasifikasikan rambu lalu lintas secara akurat dengan nilai yang cukup baik.

Kata Kunci: Kendaraan *Autonomous*, Pengenalan Objek, YOLOv8, Rambu Lalu Lintas

ABSTRACT

Autonomous cars are vehicles that can run independently without human assistance. However, this car has a weakness in detecting traffic signs. The introduction of traffic signs is intended to increase the safety of autonomous cars so that they can recognize traffic signs as they pass. This technology uses the YOLOv8 model which is a development of the Convolutional Neural Network method for detecting and classifying traffic signs. This YOLOv8 model was chosen because it has a high level of accuracy and efficiency. This research uses a dataset from Roboflow, totaling 2390 images consisting of 17 types of traffic signs in Indonesia. The training results in this research show that this model performs optimally with an accuracy value of 97.90%, a precision value of 0.978, a recall value of 0.989, a MAP50 value of 0.987, and a MAP50-95 value of 0.825. This value states that the model can identify and classify traffic signs accurately with a fairly good value.

Keywords: Autonomous Vehicle, Object Recognition, YOLOv8, Traffic Signs

BAB I

PENDAHULUAN

1.1 Latar Belakang

Sejalan dengan kemajuan teknologi, manusia memegang peranan besar dalam memajukan teknologi yang ada pada saat ini. Teknik pembelajaran mesin (juga dikenal sebagai *machine learning*) memungkinkan mesin berfungsi seperti manusia. Metode ini dirancang untuk mempermudah tugas yang dilakukan manusia dengan memberikan kecerdasan buatan pada komputer (Sanjaya dan Septanto, 2022).

Melihat berkembangnya teknologi yang semakin cepat, berbagai pihak berupaya untuk menciptakan atau mengembangkan teknologi yang ada, terutama dalam aspek kemudahan dan keselamatan berkendara. *Autonomous Driving System (ADS)* kini sering dimanfaatkan oleh produsen mobil modern menjadi sistem bantuan mengemudi otomatis. Karena dapat membantu pengemudi dalam berkendara dengan lebih mudah, sistem ini memegang peranan penting. Dengan sistem ini, mobil dapat mengidentifikasi objek dan mengambil tindakan yang tepat berdasarkan kondisi sekitar. Sistem ini dapat melakukan pengoperasian secara otomatis tanpa memerlukan kendali pengemudi. Misalnya bisa memarkir mobil sendiri, berjalan tanpa ada orang yang memegang kemudi atau yang biasa disebut dengan *autopilot*. Akan tetapi kendaraan ini memiliki beberapa tantangan, antara lain: Pengenalan lampu lalu lintas, rambu lalu lintas, pejalan kaki, tdana jalur jalan yang tidak jelas, dan sebagainya (Kurniawati dan Akbar, 2023).

Karena itulah penelitian ini dilakukan guna meningkatkan fitur keamanan pengendara pada mobil *autonomous*. Teknologi pengenalan rambu lalu lintas adalah teknologi yang dirancang guna mendeteksi dan mengenali rambu lalu lintas secara *real time* melalui pengolahan citra *digital*. Pengenalan ini menggunakan model YOLOv8 untuk mengenali rambu lalu lintas. Teknologi ini kelak dapat mengidentifikasi rambu-rambu lalu lintas yang pada penelitian ini dibagi menjadi 17 kelas yaitu rambu petunjuk halte bus, dilarang masuk, dilarang belok kiri, dilarang belok kanan, dilarang putar balik, dilarang berhenti, petunjuk jalur kiri, petunjuk

masuk jalur kiri-kanan, dilarang parkir, petunjuk parkir, petunjuk penyeberangan pejalan kaki, peringatan pintu perlintasan kereta api, perintah berhenti, peringatan simpang tiga, peringatan lampu lalu lintas, petunjuk putar balik, dan peringatan. Serta dapat mengklasifikasikan rambu-rambu tersebut dengan bantuan alat kamera. Dengan menggunakan jaringan saraf konvolusional yang dalam, model YOLOv8 dapat belajar mengenali objek yang berbeda dan menemukannya secara akurat dalam sebuah gambar. Teknologi ini secara bersamaan dapat mendeteksi beberapa objek yang termasuk dalam kelas berbeda, sehingga sangat bermanfaat bagi aplikasi yang memerlukan pemrosesan *real-time* dan akurasi deteksi tinggi, seperti mengemudi otonom, pengawasan video, dan robotika.

Secara khusus, banyak penelitian telah dilakukan tentang pengenalan rambu lalu lintas. Pada masa kini beragam pendekatan yang dilakukan dan dikembangkan dengan menerapkan *deep learning*, seperti YOLOv5 (Nugroho dan Cahyono, 2022), *Support Vector Machine* (Lionirahmada dan Utamingrum, 2021), YOLO9000 (Kuncoro dkk., 2020), *Extreme Learning Machine* (Pauziah dan Herliana, 2021), dan *Convolutional Neural Network* (Akbar, 2022). Oleh karena itu, pada penelstian ini mengusung model YOLOv8 sebagai pengenalan rambu lalu lintas yang nantinya dapat diterapkan pada sebuah *prototipe* sistem mobil *autonomous* karena dapat menghasilkan akurasi, kecepatan, dan efisiensi yang optimal dalam mendeteksi suatu objek gambar.

1.2 Rumusan Masalah

Berdasarkan latar belakang tersebut penulis mengidentifikasi permasalahan yang akan dibahas yaitu:

1. Mengetahui bagaimana model YOLOv8 dapat mendeteksi rambu-rambu lalu lintas secara *real-time* melalui kamera.
2. Mengetahui bagaimana model YOLOv8 dapat mengklasifikasikan rambu-rambu lalu lintas yaitu rambu petunjuk halte bus, dilarang masuk, dilarang belok kiri, dilarang belok kanan, dilarang putar balik, dilarang berhenti, petunjuk jalur kiri, petunjuk masuk jalur kiri-kanan, dilarang parkir, petunjuk parkir, petunjuk penyeberangan pejalan kaki, peringatan pintu perlintasan kereta api, perintah

berhenti, peringatan simpang tiga, peringatan lampu lalu lintas, petunjuk putar balik, dan peringatan.

1.3 Pembatasan Masalah

Pembatasan masalah dari penulisan Tugas Akhir ini adalah:

1. Keterbatasan kondisi lingkungan tertentu, seperti siang hari dan kondisi cuaca normal. Kondisi ekstrim seperti hujan lebat, kabut tebal atau kegelapan malam tidak akan menjadi fokus.
2. Evaluasi model dilakukan dalam lingkungan pengujian tanpa integrasi secara langsung dengan mobil *autonomous* yang sesungguhnya.

1.4 Tujuan Tugas Akhir

Tujuan dari Tugas Akhir ini adalah mengimplementasikan model YOLOv8 untuk mendeteksi rambu lalu lintas, dan mengklasifikasikan informasi yang terkandung dalam rambu-rambu tersebut.

1.5 Manfaat

Manfaat yang diharapkan pada Tugas Akhir ini adalah guna meningkatkan keamanan dari kendaraan *autonomous* agar dapat mendeteksi rambu-rambu yang dilewatinya.

1.6 Sistematika Penulisan

Sistematika penulisan yang akan digunakan penulis untuk menyusun tugas akhir ini adalah sebagai berikut:

BAB I :PENDAHULUAN

Pada bab I, penulis menjelaskan urgensi penelitian yang diangkat, mulai dari latar belakang, membuat rumusan masalah, pembatasan masalah, tujuan, manfaat, dan sistematika penulisan

BAB II :TINJAUAN PUSTAKA DAN DASAR TEORI

Pada bab II, Penulis memuat dasar teori yang digunakan dan rujukan penelitian yang berisi peneliat sebelumnya yang menjadi

dasar dalam perancangan sistem yang dibuat dan membantu penulis dalam memahami teori tentang deteksi objek, klasifikasi, dan model *You Only Look Once* (YOLO) v8.

BAB III : METODE PENELITIAN

Pada bab III, penulis menjelaskan mulai dari tahap mendapatkan dataset, hingga pemodelan sistem.

BAB IV :HASIL DAN ANALIS

Pada bab IV, penulis meMAParkan hasil mulai dari hasil akhir sistem, pengujian sistem, dan implementasi sistem.

BAB V :KESIMPULAN DAN SARAN

Pada bab V, penulis meMAParkan kesimpulan yang diperoleh melalui hasil proses penelitian.



BAB II

TINJAUAN PUSTAKA DAN DASAR TEORI

2.1 Tinjauan Pustaka

Mobil *autonomous* merupakan kendaraan yang dapat berfungsi secara otomatis. Kendaraan tersebut dilengkapi oleh teknologi yang memungkinkannya merasakan lingkungan sekitarnya dan merespons kondisi yang terdeteksi oleh sistem. Menurut *National Highway Traffic Safety Administration* (NHTSA), mobil *autonomous* adalah kendaraan yang dapat mengendalikan kemudi, akselerasi, dan pengereman tanpa campur tangan pengemudi secara langsung. Mobil *autonomous* dapat menjalankan fungsi yang sama seperti pengemudi manusia berpengalaman (Sriratnasari dkk., 2019). Akan tetapi mobil ini memiliki kelemahan dalam mendeteksi rambu lalu lintas. Pengenalan rambu lalu lintas ini dimaksudkan untuk menambah keamanan mobil *Autonomous* agar dapat mengenali rambu lalu lintas yang dilewatinya. Model YOLOv8 ini dipilih karena memiliki tingkat akurasi dan efisiensi yang tinggi.

Sebelumnya telah dilakukan penelitian oleh Nugroho dan Cahyono (2022). Pada penelitian ini, masalah kecelakaan lalu lintas sering terjadi dikarenakan pengemudi tidak tertib dan sering melanggar rambu lalu lintas. Oleh karena itu, fitur ADAS (*Advanced Driving Assistant System*) dirancang dan diimplementasikan untuk deteksi lampu lalu lintas dan deteksi rambu lalu lintas. Algoritma YOLOv5 dapat diterapkan pada perangkat Raspberry Pi 4 sebagai pendeteksi objek guna mengenali rambu lalu lintas dan lampu lalu lintas. Rambu yang dimanfaatkan antara lain belok kanan, belok kiri, dilarang belok kanan, dilarang belok kiri, dilarang belok u, dilarang berhenti, dilarang parkir, lampu merah, lampu hijau, lampu kuning, putar u. Sistem ini telah berhasil diterapkan pada mobil dan mengeluarkan suara peringatan ketika mendeteksi suatu objek. Penelitian menghasilkan nilai *Mean Average Presisi* (mAP) sebesar 89,50%, sedangkan akurasi saat pengujian gambar lampu dan rambu lalu lintas sebesar 88,88%

Penelitian lain dilakukan oleh Lionirahmada dan Utamingrum (2021). Pada penelitian ini, masalah kecelakaan lalu lintas sering terjadi karena kurangnya

pemahaman masyarakat mengenai peraturan rambu lalu lintas. Oleh karena itu, dikembangkan sistem pengenalan rambu batas kecepatan untuk mengklasifikasikan jenis rambu batas kecepatan menggunakan ekstraksi fitur histogram fitur *Directional Gradient* (HOG) dan klasifikasi *Support Vector Machine* (SVM). Menggunakan mikrokontroler Raspberry Pi, kamera Pi, dan *speaker* sebagai media keluaran dalam bentuk audio, penelitian ini menerapkan histogram gradien arah dan mendukung klasifikasi mesin vektor, sehingga sistem berhasil mengenali tanda batas kecepatan. Maks 20 km, 25 km, 30 km, 40 km, 50 km, min 20 km. Penelitian ini menghasilkan nilai rata-rata akurasi yaitu 86,08%.

Menurut Kuncoro dkk (2020) salah satu implementasi dari visi komputer adalah mobil *self-driving*. Mobil *self-driving* merupakan sistem kendaraan yang dikontrol komputer yang dapat mengemudi, beradaptasi dengan lingkungannya, dan berfungsi tanpa campur tangan manusia. Akibatnya, kondisi lalu lintas menjadi semakin kompleks sehingga memerlukan kecerdasan buatan untuk memberikan informasi rambu dan membantu pengendalian kendaraan guna memastikan keselamatan berkendara. Penelitian ini menggunakan arsitektur YOLO9000 dengan tiga kelas data set yaitu rambu belok kanan, rambu belok kiri, dan rambu berhenti. Keakuratan penelitian ini cukup tinggi. Penelitian ini menghasilkan nilai akurasi yang cukup tinggi yaitu 93,41%.

Dalam salah satu penelitian yang dilakukan oleh Pauziah dan Herlina (2021) jenis rambu lalu lintas sangat beragam, dan tidak semua pengguna jalan mengetahui rambu-rambu lalu lintas yang ada, maka seringkali berujung pada kecelakaan lalu lintas dan pelanggaran lain yang tidak hanya merugikan diri sendiri tetapi orang lain. Mengingat permasalahan tersebut maka penelitian ini dilakukan dengan tujuan untuk membangun sistem pengenalan rambu sesuai jenis rambu jalan sehingga pengguna jalan dapat mengenali rambu jalan tersebut. Penelitian ini menggunakan teknik pengolahan gambar cepat yang disebut *Extreme Learning Machine* (ELM). Penelitian ini memperoleh suatu model yang dapat mengenali gambar rambu lalu lintas dengan akurasi, presisi, dan recall yang tinggi. Penelitian ini memperoleh nilai akurasi sebesar 97%.

Disamping itu penelitian lain dilakukan oleh Akbar (2022) mengenai pengenalan rambu lalu lintas yang memanfaatkan *Convolutional Neural Network* dengan Algoritma *Stochastic Gradient Descent* (SGD). Menurut Akbar Output *Traffic Sign Recognition* (TSR) dapat dimasukkan ke *Driver Assistance Systems* (DAS), *Advance Driver Assistance Systems* (ADAS), *Autonomous Driving System* (ADS) dan sistem lainnya. Keakuratan keluaran TSR berdampak besar bagi pengguna jalan. Keputusan yang diambil oleh sistem ini mempunyai konsekuensi yang mematikan, dengan skenario terburuk mengakibatkan kecelakaan lalu lintas. Penelitian ini memperoleh hasil dari penelitian yaitu sistem mampu mendeteksi dan mengklasifikasikan rambu lalu lintas dengan akurasi 97,33%.

2.2 Dasar Teori

2.2.1 Pengolahan Citra

Pengolahan citra merupakan pengolahan gambar/citra secara spesifik, citra yang dimaksud merupakan citra diam dan citra bergerak, serta dalam ilmu pasti, pengolahan citra adalah fungsi kontinu dari ketajaman cahaya dua dimensi. Berikut teknik pengolahan citra yang digunakan dalam memproses citra digital diantaranya:

- a. Citra *Greyscale* adalah warna abu-abu suatu piksel mengandungi nilai piksel lain, atau citra yang hanya mempunyai warna hitam dan putih, gambar ini memiliki guratan warna kroma 8-bit (256) bercampur dengan skala abu-abu. Gambar berwarna meliputi 3 jenis bit warna yang mempunyai perbedaan ragam warna, 8bit mempunyai 256 ragam warna, 16bit mempunyai 65.536 ragam warna dan 24bit mempunyai 16.777.216 ragam warna.
- b. Citra Biner adalah citra yang memiliki dua kekuatan warna yaitu hitam putih yang memiliki nilai 0 dan 255, dengan terlebih dahulu mengubah warna citra *Red Green Blue* menjadi warna *greyscale* sehingga bisa mempercepat proses pengolahan citra dan mengurangi biaya pengolahan.
- c. *Thresholding*, adalah metode yang digunakan untuk menyorot gambar skala abu-abu dan *biner* dengan tujuan untuk mengidentifikasi dengan jelas area objek dan area latar belakang gambar. Biasanya, gambar ini digunakan untuk pemrosesan lebih mendalam pengenalan objek dan ekstraksi fitur.

- d. Deteksi *Canny Edge* adalah *operator* deteksi modern optimal yang ditemukan oleh Marr dan Hildreth. Deteksi Canny Edge digunakan untuk mendeteksi kontur dan bentuk dalam citra. Teknik tersebut digunakan untuk mendeteksi objek dalam citra (Habibi dkk., 2021).

2.2.2 Mobil *Autonomous*

Mobil *Autonomous* adalah kendaraan yang dapat berjalan secara mandiri tanpa bantuan manusia. Hal tersebut mencakup semua aktivitas berkendara, seperti mewaspadaikan lingkungan sekitar, memantau sistem penting, dan mengendalikan kendaraan, termasuk navigasi dari asal ke tujuan. Konsep mobil *autonomous* adalah untuk memberikan layanan yang aman dan mudah digunakan bagi pengguna mobil (Kuncoro dkk., 2020).

Menurut *Society of Automotive Engineers (SAE)*, kendaraan memiliki enam tingkat otonomi:

1. Level 0: Kendaraan dikendalikan sepenuhnya oleh manusia.
2. Level 1: Kendaraan dilengkapi dengan sistem bantuan pengemudi yang dapat mengontrol fungsi seperti pengereman dan perpindahan gigi.
3. Level 2: Kendaraan dapat mengontrol beberapa fungsi secara bersamaan, namun tetap memerlukan perhatian manusia.
4. Level 3: Kendaraan dapat mengendalikan semua fungsi dalam kondisi tertentu, namun manusia harus bersedia mengambil kendali.
5. Level 4: Kendaraan dapat mengontrol semua fungsi dan memantau lingkungan dalam kondisi tertentu, sehingga manusia tidak perlu menarik perhatian pada dirinya sendiri.
6. Level 5: Kendaraan sepenuhnya otonom dan tidak memerlukan bantuan manusia.

Beberapa teknologi yang digunakan pada mobil self-driving antara lain

1. Sensor dan kamera

Kendaraan *autonomous* memantau lingkungannya dengan menggunakan sensor seperti radar, GPS, kamera, dan LiDAR (deteksi dan jangkauan cahaya). Sensor dan kamera pada mobil *autonomous* dimanfaatkan guna mengenali rambu lalu lintas dan memberikan informasi kepada pengemudi.

2. *Artificial Intelligence* (AI)

Teknologi AI digunakan untuk menganalisis data dari sensor dan mengambil keputusan tentang pergerakan kendaraan.



3. Pemrosesan Data

Data dari sensor diproses oleh sistem komputer yang kuat untuk mengambil keputusan tentang pergerakan kendaraan, termasuk: pengatur kecepatan, peralihan, pengereman.







2.2.3 Rambu Lalu Lintas

Kata rambu dalam Kamus Besar Bahasa Indonesia (KBBI) berarti tanda atau sesuatu yang menyatakan sesuatu. Rambu dapat dipahami sebagai sesuatu yang memberi perintah, larangan, menunjukkan dan memberikan keterangan yang perlu diketahui dan ditaati. Undang-undang Nomor 22 Tahun 2009 tentang Lalu Lintas dan Angkutan mengatur bahwasannya rambu lalu lintas merupakan suatu bagian perlengkapan jalan yang berupa lambang, huruf, angka, frasa dan/atau kombinasi yang dapat digunakan untuk memperingatkan, melarang, memerintahkan, atau memberi petunjuk kepada penggunanya (Christoffel dkk, 2019).

Tabel 2. 1 Rambu lalu lintas

No	Nama Rambu	Gambar Rambu
1	Petunjuk Halte Bus	
2	Petunjuk Masuk Jalur Kiri-Kanan	

No	Nama Rambu	Gambar Rambu
3	Dilarang Masuk	
4	Petunjuk Penyebrangan Pejalan Kaki	
5	Dilarang Belok Kiri	
6	Peringatan Simpang Tiga	
7	Berhenti	
8	Dilarang Berhenti	

No	Nama Rambu	Gambar Rambu
9	Peringatan Lampu Lalu Lintas	
10	Petunjuk Putar Balik	
11	Dilarang Parkir	
12	Dilarang Belok Kanan	
13	Peringatan	
14	Dilarang Putar Balik	

No	Nama Rambu	Gambar Rambu
15	Peringatan Pintu Perlintasan Kereta Api	
16	Petunjuk Jalur Kiri	
17	Petunjuk Parkir	

2.2.4 Deep learning

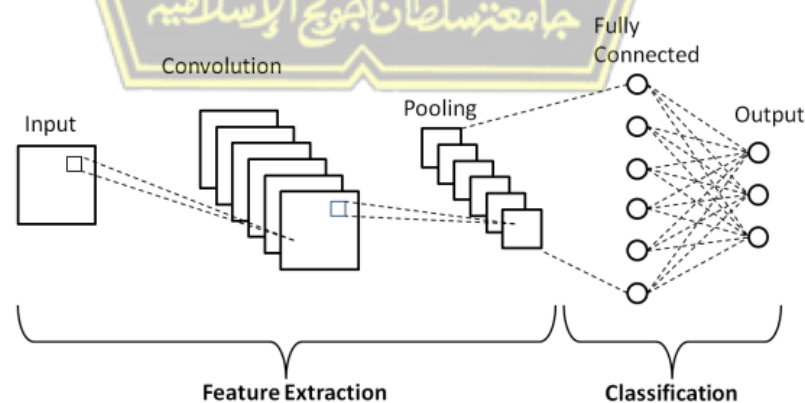
Deep Learning atau pembelajaran mendalam adalah komponen dari *Machine Learning* dan pada dasarnya merupakan jaringan saraf yang memiliki tiga lapisan atau lebih. Meskipun jaringan saraf ini mencoba untuk mensimulasikan perilaku otak manusia, kemampuannya guna "belajar" dari sejumlah data besar masih jauh dari maksimal. Meskipun jaringan neural satu lapis dapat memberikan perkiraan kasar, lapisan tersembunyi tambahan dapat membantu pengoptimalan dan akurasi. Pembelajaran mendalam mendukung beberapa aplikasi dan layanan *Artificial Inteligent* (AI) atau kecerdasan buatan yang menaikkan otomatisasi dan menjalankan tugas analitis dan fisik secara otomatis. Teknologi pembelajaran mendalam digunakan dalam produk dan layanan sehari-hari (contohnya asisten digital, *remote AC* yang dihidupkan menggunakan suara, dan pendeteksi penipuan kartu kredit) serta teknologi baru (seperti mobil tanpa pengemudi) (Putra dkk., 2022).

2.2.5 Deteksi Objek

Deteksi objek adalah kumpulan tugas dari *computer vision* dalam gambar digital. Tugas *Computer Vision* dapat dibagi menjadi tiga bagian: klasifikasi gambar, lokalisasi objek, dan deteksi objek. Klasifikasi gambar adalah algoritma yang membuat daftar kategori objek dalam sebuah gambar. Lokalisasi objek merupakan algoritma yang mendeteksi keberadaan objek dalam suatu gambar dan menampilkan lokasinya dalam bentuk kotak pembatas. Deteksi objek merupakan algoritma yang menggunakan kotak pembatas untuk menampilkan lokasi objek dan tipe kelas objek dalam suatu gambar. Oleh karena itu, tugas deteksi objek adalah mengembalikan beberapa objek yang terdeteksi, beserta informasi tentang kelas objek, tingkat kepercayaan, dan koordinat setiap objek. Baru-baru ini, pengenalan objek berbasis gambar telah terbukti akurat dan tepat pada data pelatihan (Atria, 2023).

2.2.6 *Convolutional Neural Network* (CNN)

Convolutional Neural Network (CNN) adalah arsitektur khusus dalam pembelajaran mendalam yang bermanfaat untuk memproses data gambar dan tugas-tugas visi komputer. Tidak seperti jaringan saraf biasa, dimana semua neuron terhubung satu sama lain, CNN menggunakan pendekatan yang lebih efisien dengan menghubungkan setiap *neuron* hanya ke bagian tertentu dari lapisan sebelumnya (Sugianto dkk, 2022).



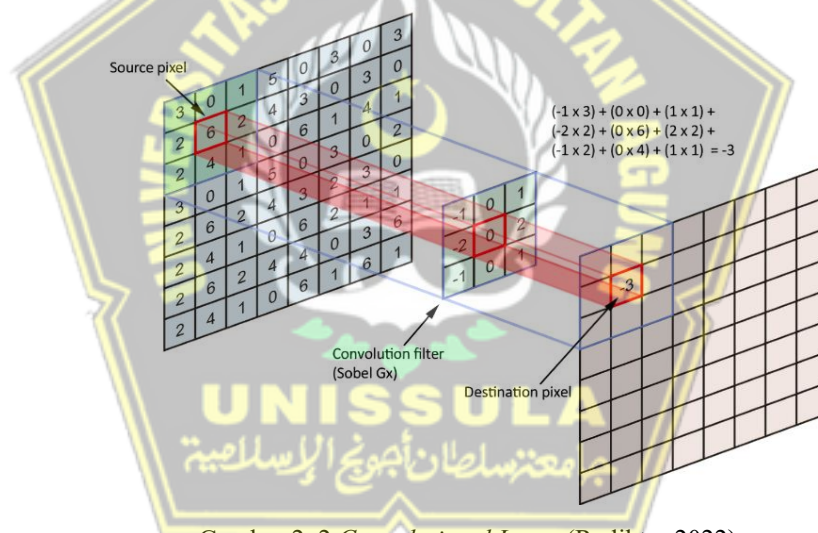
Gambar 2. 1 Arsitektur CNN (Radikto, 2022)

Gambar 2.1 merupakan arsitektur dari *Convolutional Neural Network*. Secara umum, CNN dibagi menjadi tiga lapisan: lapisan *convolution*, lapisan

polling, lapisan *Rectified Linear Unit* (ReLU), lapisan *fully connected* dan lapisan *softmax activation*. Berikut penjelasan masing-masing lapisan:

a. Lapisan *Convolution*

Lapisan *convolution* atau lapisan konvolusional merupakan lapisan pertama CNN dan merupakan kumpulan *filter* yang membantu memeriksa masukan. Setiap *filter* pada lapisan konvolusional mempunyai panjang, tinggi, dan ketebalan tertentu yang sama dengan data masukan. Lapisan konvolusional secara signifikan meningkatkan kompleksitas model dengan mengoptimalkan keluaran. Dioptimalkan dengan tiga parameter: kedalaman, ukuran langkah (mengontrol cara filter bergerak ketika diimplementasikan ke data *input*), dan pengaturan *zero padding* (penambahan piksel pada nilai tertentu akan dilakukan).

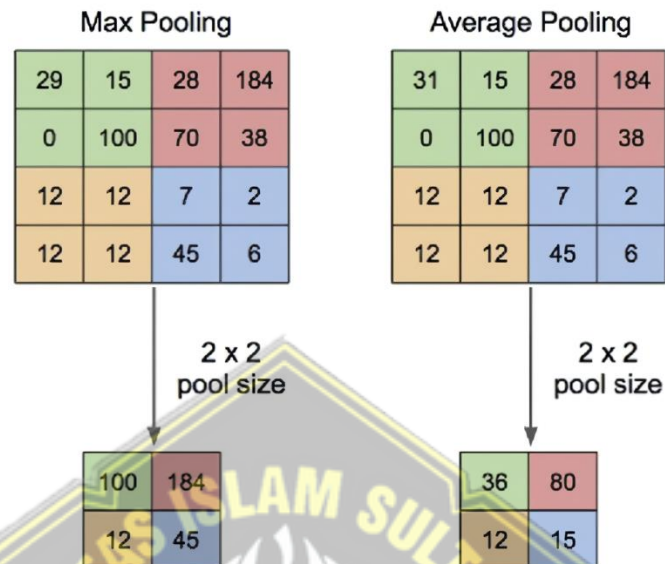


Gambar 2. 2 *Convolutional Layer* (Radikto, 2022)

b. Lapisan *Pooling*

Tahap selanjutnya setelah tahap *convolution* merupakan tahap *pooling* yang digunakan mengurangi jumlah parameter dari masukan (mengurangi ukuran matriks). Lapisan *pooling* CNN mengambil fitur peta atau peta aktivasi sebagai masukan dan meneruskannya melalui berbagai jenis operasi statistik berdasarkan nilai piksel disekitarnya. Keuntungan lapisan *pooling* adalah mempunyai kemampuan untuk mengurangi dimensi volume peta fitur keluaran secara bertahap, sehingga membantu mengontrol *overfitting*. Pada gambar 2.3

ada dua jenis metode pengumpulan: (*average polling*) pengumpulan rata-rata dan (*max polling*) pengumpulan maksimum.



Gambar 2. 3 *Polling Layer* (Radikto, 2022)

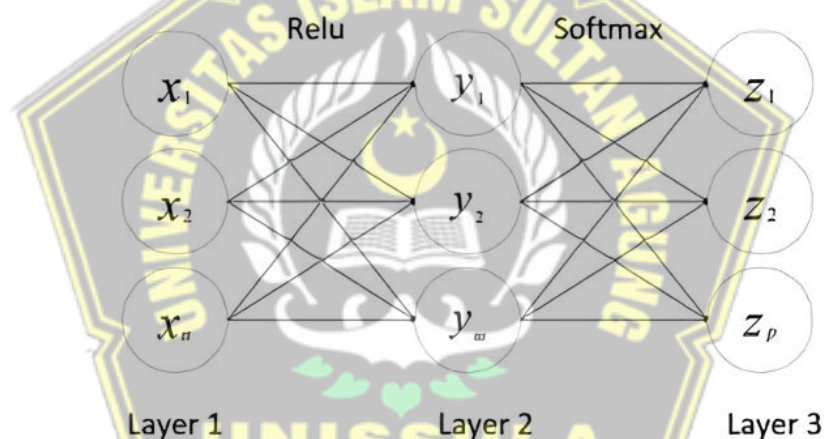
Max Pooling adalah teknik pengumpulan lain yang digunakan di CNN untuk mengurangi ukuran matriks fitur. Tujuan utama *Max Pooling* adalah untuk mempercepat proses komputasi dan mengurangi kemungkinan *overfitting* dengan mengurangi dimensi matriks fitur sekaligus menjaga informasi penting. Cara kerja *Max Pooling* adalah *Max Pooling* pertama-tama menggunakan filter yang bergerak secara horizontal dan vertikal melalui matriks fitur. Kemudian, untuk setiap *patch* yang terdeteksi oleh filter, nilai maksimum semua elemen dalam *patch* tersebut dihitung. Nilai maksimum ini digunakan untuk menggantikan nilai asli pada matriks fitur. Ini menciptakan peta fitur baru dengan dimensi yang lebih kecil.

Average Pooling adalah salah satu teknik pengumpulan yang digunakan dalam jaringan saraf konvolusional untuk mengurangi ukuran matriks fitur yang dihasilkan oleh lapisan konvolusional. Tujuan utama dari *average pooling* adalah untuk mengurangi dimensi matriks fitur tanpa kehilangan informasi penting, sehingga mempercepat proses komputasi dan mengurangi kemungkinan *overfitting*. Cara kerja *Average Pooling* adalah pertama menggunakan filter yang bergerak secara horizontal dan vertikal melalui

matriks fitur. Kemudian, untuk setiap *patch* yang terdeteksi oleh filter, nilai rata-rata semua elemen dalam *patch* tersebut dihitung. Nilai rata-rata ini digunakan untuk menggantikan nilai asli pada matriks fitur. Ini menciptakan peta fitur baru dengan dimensi yang lebih kecil.

c. Lapisan *fully connected*

Lapisan *fully connected* merupakan lapisan yang menerima masukan dari keluaran lapisan penyatuan sebelumnya dan menghasilkan peta fitur dalam bentuk larik multidimensi. Oleh karena itu, peta fitur harus diubah menjadi vektor agar dapat digunakan. Lapisan ini digunakan untuk menggabungkan fitur-fitur yang diekstraksi oleh lapisan konvolusi dan lapisan polling untuk membuat prediksi akhir seperti klasifikasi gambar.



Gambar 2.4 *Fullyconnected Layer* (Radikto, 2022)

Setiap *neuron* dalam struktur konvolusional harus diubah menjadi satu informasi sebelum memasuki proses lapisan yang terhubung sepenuhnya. Jika proses ini dilakukan hingga data informasi spasial hilang dan tidak dapat dipulihkan, lapisan yang terhubung sepenuhnya hanya dapat diterapkan di terminal *neuron*. Pada gambar 2.4, *layer 1* memproses *transfer feed* dengan beralih ke *layer 2* yang menggunakan aktivasi ReLu. Selanjutnya *layer 2* memproses klasifikasi menggunakan fungsi *softmax*.

Setiap neuron di CNN menggunakan fungsi aktivasi untuk memperkenalkan nonlinier ke model, hal tersebut memungkinkan jaringan mempelajari pola yang lebih kompleks. Fungsi aktivasi yang biasa digunakan di CNN

a. *Rectified Linear Unit (ReLU)*

Rectified Linear Unit (ReLU) merupakan fungsi aktivasi linier yang biasa dimanfaatkan pada CNN. Tugas dari ReLu yaitu memutuskan apakah akan mengaktifkan neuron dari nilai bobot masukan.

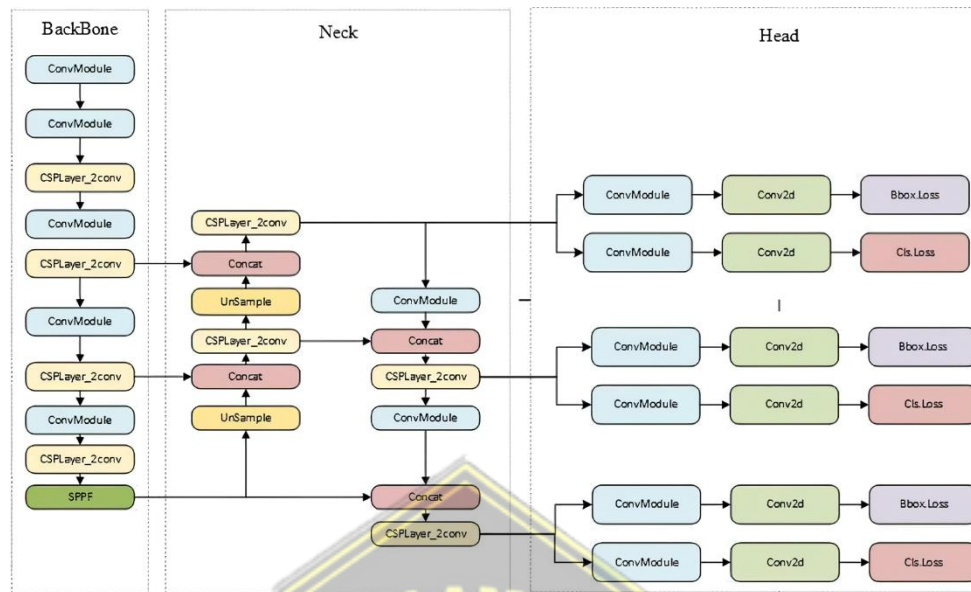
b. *Softmax Activation*

Softmax activation adalah bentuk yang mirip dengan regresi logistik yang bekerja pada lebih dari dua kelas dalam proses klasifikasi. Softmax digunakan untuk mengonversi keluaran lapisan terakhir menjadi distribusi probabilitas yang mendasarinya.

2.2.7 *You Only Look Once (YOLO) V8*

YOLO (*You Only Look Once*) adalah deteksi objek dan segmentasi gambar yang diluncurkan pada tahun 2015 oleh Joseph Redmon dan Ali Farhadi di Universitas Washington. Versi terbaru, YOLOv8, yang dikembangkan oleh Ultralytics mewakili pengembangan terkini dalam model deteksi objek, menghadirkan fitur-fitur baru dan peningkatan performa dan fleksibilitas. YOLOv8 mampu menyelesaikan berbagai tugas kecerdasan buatan, termasuk menawarkan fleksibilitas kepada pengguna dalam berbagai aplikasi dan domain, termasuk deteksi, segmentasi, estimasi *pose*, pelacakan, dan klasifikasi. Ultralytics YOLOv8 adalah model deteksi objek dan segmentasi gambar *real-time* terancang yang mengintegrasikan pengembangan terkini dalam pembelajaran mendalam dan visi komputer.

YOLOv8 adalah model terkini YOLO yang diproduksi pada tahun 2023, dengan peningkatan kecepatan dan akurasi yang signifikan dibandingkan versi sebelumnya. YOLOv8 melanjutkan keberhasilan model sebelumnya dan mempublikasikan fitur dan peningkatan performa, fleksibilitas, dan efisiensi. YOLOv8 memanfaatkan arsitektur *Convolutional Neural Network* untuk mendeteksi objek dengan cepat dan akurat dalam satu tahap.



Gambar 2. 5 Arsitektur YOLOv8 (Zhang, 2023)

Arsitektur YOLOv8 Ultralytics berpijak pada konsep deteksi objek satu tahap, memungkinkan mendeteksi objek dalam gambar secara efisien dan akurat. Arsitektur ini mengintegrasikan banyak pembaruan dan peningkatan dibandingkan model YOLO sebelumnya (Armin dkk., 2023).

Arsitektur YOLOv8 pada Gambar 2.5 terdiri dari jaringan *backbone*, *neck*, dan *head*. *Backbone* juga dikenal sebagai *ekstraktor* fitur tulang punggung, berperan untuk mengekstraksi fitur-fitur penting dari masukan. *Backbone* menggunakan struktur CSPDarknet yang terdiri dari layer *convolutional* dan *polling*. Komponen tersebut menangkap pola sederhana dari lapisan pertama, yaitu tepi dan tekstur. Jaringan tulang punggung menggunakan kombinasi dari berbagai lapisan konvolusi untuk mengekstrak fitur dari gambar masukan dan menyempurnakan fitur tersebut menggunakan serangkaian lapisan konvolusi, *concat layer* dan *upsample* di bagian leher. *Neck* (leher) berperan menjadi penghubung antara tulang punggung dan kepala, menjalankan operasi fusi fitur dan menggabungkan informasi kontekstual. *Head* (kepala) adalah komponen terakhir dari jaringan dan berperan guna mencetak keluaran jaringan, contohnya kotak pembatas dan nilai keyakinan pada deteksi objek. Fungsi *head* membuat kotak pembatas yang terdapat pada objek yang mungkin ada pada gambar, memberikan nilai kepercayaan pada setiap *bounding box* yang menyatakan kemungkinan

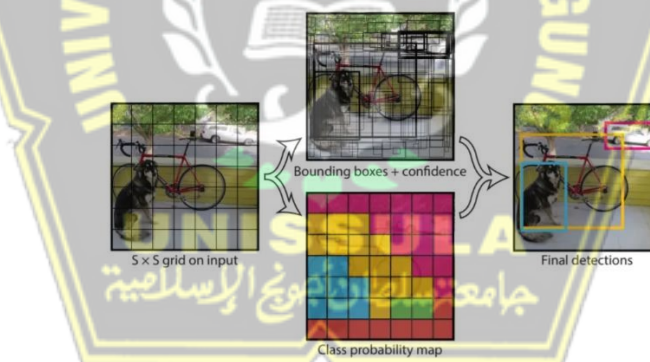
keberadaan objek, dan mengurutkan objek pada kotak pembatas berdasarkan kategori (Yanto, Aziz dan Irmawati, 2023).

Arsitektur YOLOv8 merupakan evolusi dari YOLO versi sebelumnya dan menghadirkan beberapa peningkatan dan manfaat baru. Salah satu manfaat utama YOLOv8 yaitu peningkatan kinerja dan akurasi deteksi objek. Hal ini diperoleh dengan menambahkan lebih banyak lapisan pemrosesan gambar dan menggunakan teknik pembelajaran mesin yang lebih canggih. Perbedaan lain antara YOLOv8 dan YOLO versi sebelumnya adalah penggunaan *anchor boxes*. YOLOv8 menyajikan metode *anchor-free* yang lebih sederhana dan efisien dibandingkan dengan model YOLO versi sebelumnya berbasis *anchor-based*. Hal ini menjadikan YOLOv8 lebih mudah diterapkan dan memungkinkan deteksi objek lebih cepat. Selain itu, YOLOv8 juga mempunyai kemampuan deteksi objek yang lebih baik dan akurat dibandingkan YOLO versi sebelumnya. Hal ini disebabkan adanya penambahan lapisan pemrosesan gambar dan penggunaan teknik pembelajaran mesin yang lebih canggih. Secara keseluruhan, arsitektur YOLOv8 merupakan peningkatan dari YOLO versi sebelumnya, menawarkan keunggulan signifikan dalam hal performa, akurasi, dan kemampuan deteksi objek (Yanto dkk., 2023).

Arsitektur YOLOv8 adalah model visi komputer yang dapat disesuaikan dengan berbagai kebutuhan spesifik. YOLOv8 memiliki beberapa Teknik dalam menangani variasi bentuk objek dalam gambar. Berikut beberapa fitur arsitektur yang membuat YOLOv8 menonjol:

1. YOLOv8 memanfaatkan jaringan saraf utama (*backbone*) yang kuat guna mengekstrak fitur dari gambar. Pada hal ini, Ultralytics menjadikan arsitektur CSPDarknet53 sebagai jaringan saraf utamanya, yang diakui dengan kemampuannya menangkap representasi gambar yang mendalam.
2. YOLOv8 menggunakan teknik pelatihan adaptif yang mengadaptasi model terhadap perubahan bentuk objek pada gambar. Teknik ini mencakup penyeimbangan fungsi kerugian selama pelatihan dan penggunaan teknik seperti Adam, yang berkontribusi terhadap peningkatan akurasi, konvergensi lebih cepat, dan peningkatan performa model secara keseluruhan.

3. YOLOv8 memanfaatkan teknik augmentasi data baru yang membantu mengatasi aspek variasi gambar, seperti resolusi rendah dan oklusi, dalam situasi deteksi objek dunia nyata yang kondisinya kurang ideal. Oleh karena itu, model dapat mengatasi perubahan bentuk objek pada gambar yang tidak ideal.
4. YOLOv8 dapat melakukan analisis gambar tingkat lanjut melalui segmentasi semantik baru dan kemampuan prediksi kelas. Oleh karena itu, selain kemampuan pengenalan objek intinya, model dapat mendeteksi aktivitas, warna, tekstur, dan bahkan hubungan antar objek. Hal ini memungkinkan model menangani perubahan bentuk objek dalam gambar dengan lebih baik.
5. YOLOv8 memanfaatkan lapisan deteksi untuk memprediksi kotak pembatas dan kelas objek. Dalam arsitektur ini, gambar dibagi menjadi beberapa grid, dan model memprediksi beberapa kotak pembatas beserta probabilitas kelas bagi setiap sel *grid*. Deteksi multiskala memungkinkan model mendeteksi objek pada skala berbeda, sehingga tidak terbatas untuk objek yang memiliki ukuran tertentu.



Gambar 2. 6 Hasil Penerapan YOLOv8 (Carolina, 2023)

6. YOLOv8 menerapkan teknik seleksi loncat yang mempermudah integrasi informasi lebih detail dari lapisan sebelumnya ke dalam lapisan-lapisan yang lebih dalam. Hal ini membantu meningkatkan potensi model untuk mengenali objek, khususnya objek kecil atau kompleks. Oleh karena itu, arsitektur YOLOv8 meraih keseimbangan terbaik antara kecepatan dan akurasi, yang menjadi penyelesaian yang baik bagi tugas deteksi objek yang rumit. (Armin dkk., 2023)

2.2.8 Confussion Matrix

Dalam mengevaluasi performa model pembelajaran mesin digunakan *Confussion Matrix* dengan menunjukkan prediksi dari klasifikasi aktual dan klasifikasi prediksi sehingga didapat nilai Akurasi, *Precision*, *Recall*, dan *Mean Average Precision* (MAP). Matriks konfusi memiliki empat klasifikasi, antara lain *true negative* (TN), *true positive* (TP), *false negative* (FN), dan *false positive* (FP), yang diperoleh melalui nilai aktual dan prediksi. TP adalah jumlah sampel positif yang berhasil diklasifikasikan dengan benar. TN adalah jumlah sampel negatif yang berhasil diklasifikasikan dengan benar. FP adalah jumlah sampel negatif yang salah diklasifikasikan sebagai positif. FN adalah jumlah sampel positif yang salah diklasifikasikan sebagai *negative* (Maulana dkk, 2024).

Akurasi adalah ukuran yang menentukan seberapa dekat hasil yang diukur atau diprediksi dengan nilai sebenarnya atau dengan nilai yang diterima sebagai kebenaran. Dalam konteks pengukuran, presisi mengacu pada tingkat kesesuaian antara hasil yang diperoleh dan nilai standar atau yang diharapkan.

$$\text{Akurasi} = \frac{TP}{TP+FP+FN} \quad (1)$$

Untuk mendapatkan *precision*, dapat mengukur berapa banyak total prediksi positif model yang benar-benar positif. *Precision* membantu menunjukkan informasi mengenai keakuratan model dalam mengklasifikasikan kejadian sebagai benar. Semakin tinggi *precision*nya, semakin sedikit alarm palsu yang diperoleh model. Akan tetapi, *precision* harus tetap memperhatikan *recall* (sensitivitas).

$$\text{Precision} = \frac{TP}{TP+FP} \quad (2)$$

Recall atau sensitivitas (tingkat positif sebenarnya) adalah metrik evaluasi sebagai bagian dari klasifikasi model. Penarikan kembali menghitung seberapa baik suatu model dapat mendapati semua contoh positif yang sesungguhnya dalam kumpulan data. Metrik ini menjelaskan wawasan mengenai kapabilitas model dalam mengenali seluruh kasus positif yang ada.

$$\text{Recall} = \frac{TP}{TP+FN} \quad (3)$$

Mean Average Precision (MAP) merupakan metrik yang digunakan untuk mengukur kinerja sistem informasi seperti pendeteksi objek dalam menghitung

akurasi dokumen terbaik. Nilai MAP menunjukkan kemampuan sistem dalam mengidentifikasi dokumen relevan dan memprioritaskan dokumen paling relevan di urutan teratas. MAP menggabungkan beberapa sub-metrik seperti matriks konfusi, *Intersection over Union* (IoU), *Recall*, dan *Precision*. Oleh karena itu, peta dapat digunakan untuk mengevaluasi kinerja model dalam tugas-tugas seperti pengenalan objek dan pengambilan informasi.

$$\text{MAP} = \frac{1}{N} \sum_{i=1}^N AP_i \quad (4)$$

Keterangan:

AP = nilai rata-rata presisi ke i

N = jumlah kelas yang dievaluasi

Beberapa variasi MAP yang digunakan, antara lain:

1. MAP50

MAP50 adalah versi mAP yang paling umum digunakan, dimana suatu model dianggap benar jika *Intersection Over Union* (IoU) antara prediksi dan kebenaran dasar lebih besar dari 0,5. MAP50 sering digunakan dalam kontes deteksi objek seperti COCO dan PASCAL VOC, dimana ambang batas ini dianggap cukup untuk menentukan keberhasilan deteksi.

2. MAP50-95

MAP50-95 Ini adalah varian yang lebih ketat dimana mAP dihitung dengan mempertimbangkan berbagai ambang batas IoU dari 0,5 hingga 0,95 dalam interval 0,05. MAP50-95 mengevaluasi model pada berbagai tingkat kesulitan pengenalan objek, sehingga memberikan gambaran performa model yang lebih komprehensif.

2.2.9 OpenCV



Gambar 2. 7 OpenCV (www.opencv.org)

Open Source Computer Vision Library (*OpenCV*) merupakan perpustakaan *open source* yang diproduksi oleh Intel yang berfokus pada penyederhanaan

pemrograman terkait gambar digital. *OpenCV* memiliki berbagai fitur termasuk deteksi wajah, pelacakan wajah, deteksi wajah, pemfilteran Kalman, banyak jenis metode AI, dan pustaka visi komputer tingkat rendah terkait dalam bahasa pemrograman C/C++ algoritma. Selain itu, *OpenCV* dikembangkan dengan Python, Java, dan Matlab (Danrekha dan Huda, 2021).

OpenCV adalah alat canggih untuk pemrosesan *realtime*. Selain itu, ini adalah alat lintas *platform*, sehingga dapat diintegrasikan dengan bahasa pemrograman untuk melakukan fungsi pemrosesan gambar dan pengenalan. *OpenCV* memiliki algoritma yang tidak dipatenkan dan dapat digunakan tanpa batasan hukum.

OpenCV pertama kali diperkenalkan secara resmi oleh *Inter Research* pada tahun 1999 sebagai kelanjutan dari bagian proyek yang berjudul Aplikasi Intensif CPU, Penelusuran Sinar Waktu Nyata, dan Dinding Tampilan 3D. Kontributor utama proyek ini mencakup staf pengoptimalan Intel Russia dan tim Intel *Performance Center*. *OpenCV* memiliki banyak fitur yang dapat digunakan. Fitur utama *OpenCV* adalah:

- a. Gambar dan Video dengan antarmuka memungkinkan membaca data gambar melalui file atau umpan video *realtime*, dan memungkinkan membuat file gambar dan video.
- b. Visi komputer umum dan pemrosesan gambar digital (untuk API tingkat rendah dan menengah). Pemrosesan ini memungkinkan bereksperimen dengan berbagai algoritma visi komputer standar. Ini termasuk deteksi garis, tepi, puncak, proyeksi elips, piramida gambar guna pemrosesan gambar multiskala, pencocokan templat, dan banyak transformasi (*Fourier*, transformasi kosinus diskrit, transformasi jarak).
- c. Modul visi komputer tingkat tinggi.
- d. Metode AI dan pembelajaran mesin untuk aplikasi visi komputer sering kali membutuhkan pembelajaran mesin atau teknik AI lainnya, di antaranya terdapat dalam paket pembelajaran mesin *OpenCV*.

BAB III

METODOLOGI PENELITIAN

3.1. Metode Penelitian

Penelitian ini mengaplikasikan deteksi model YOLOv8. Dalam penelitian ini model tersebut digunakan mendeteksi, memproses, dan mengklasifikasi rambu-rambu lalu lintas yaitu rambu petunjuk halte bus, dilarang masuk, dilarang belok kiri, dilarang belok kanan, dilarang putar balik, dilarang berhenti, petunjuk jalur kiri, petunjuk masuk jalur kiri-kanan, dilarang parkir, petunjuk parkir, petunjuk penyeberangan pejalan kaki, peringatan pintu perlintasan kereta api, perintah berhenti, peringatan simpang tiga, peringatan lampu lalu lintas, petunjuk putar balik, dan peringatan.

3.2. Studi Literatur

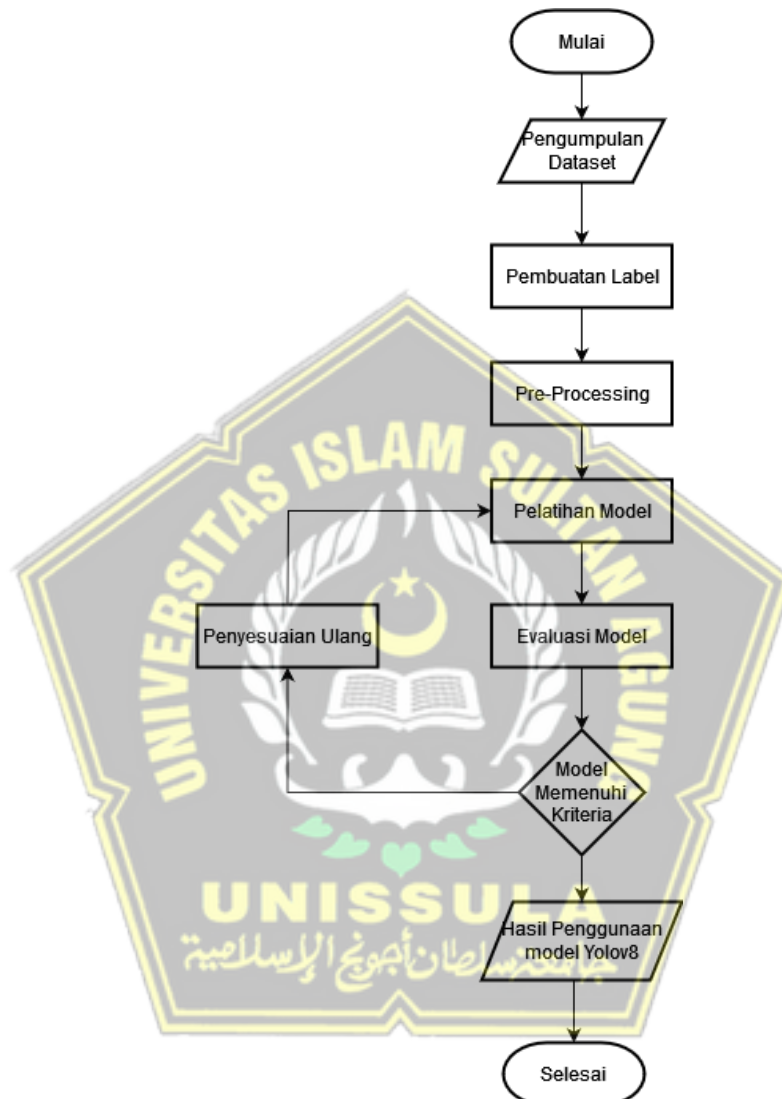
Pada tahap ini peneliti mengumpulkan artikel-artikel yang berkaitan dengan penelitian sebelumnya tentang deteksi dan klasifikasi objek menggunakan metode YOLOv8 dari jurnal, internet dan hal-hal lain yang mendukung penelitian ini.

3.3. Pengumpulan Data

Dataset yang digunakan dalam sistem ini adalah *dataset* yang diambil dari situs *Roboflow*. Jumlah *dataset* yang digunakan berjumlah 2390 gambar rambu yang terdiri dari 17 kelas yaitu rambu petunjuk halte bus, dilarang masuk, dilarang belok kiri, dilarang belok kanan, dilarang putar balik, dilarang berhenti, petunjuk jalur kiri, petunjuk masuk jalur kiri-kanan, dilarang parkir, petunjuk parkir, petunjuk penyeberangan pejalan kaki, peringatan pintu perlintasan kereta api, perintah berhenti, peringatan simpang tiga, peringatan lampu lalu lintas, petunjuk putar balik, dan peringatan. Dalam setiap gambar yang ada dalam *dataset* tersebut memiliki resolusi gambar masing-masing sebesar 416x416 *pixel*.

3.4. Metode Perancangan Arsitektur Model

Pada tahap ini penulis mendefinisikan alur sistem dalam bentuk *flowchart* pada gambar 3.1 dan selanjutnya memberikan gambaran aliran sistem.



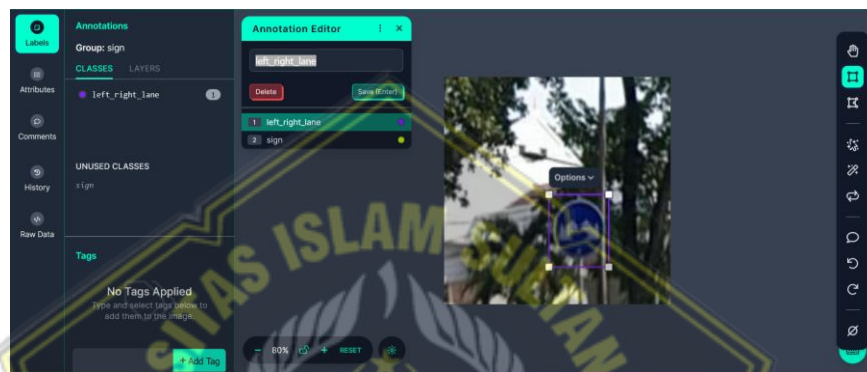
Gambar 3. 1 *Flowchart* Arsitektur Model

Metode ini melakukan beberapa tahapan yang perlu dilakukan dalam penelitian ini:

1. Pembuatan Label Data

Pada tahap *labelling* data, data awal berupa data mentah diproses melalui proses anotasi atau pelabelan. Anotasi objek merupakan *markup* dengan menentukan kotak pembatas yang berisi nama kelas yang sesuai dengan setiap objek. *Bounding Box* (Kotak Pembatas) tersebut memuat anotasi nama kelas

yang tepat dengan objeknya. Tahap ini penting saat melatih model deteksi objek untuk menyediakan model informasi mengenai tempat dan jenis objek pada gambar. Setiap gambar yang ditempatkan dalam kumpulan data diberi anotasi sesuai dengan format YOLO. Objek diberi label dengan format YOLO. Keluaran pada tahap pelabelan ini berupa file data berformat TXT yang memuat rincian kotak pembatas. Proses anotasi ini dapat dilakukan secara langsung di *Roboflow*.



Gambar 3. 2 Proses Anotasi

Gambar 3.2 merupakan proses pembuatan label kelas. Objek rambu lalu lintas diberi label sesuai dengan nama rambu lalu lintasnya yang ditandai dengan *bounding box* (kotak pembatas) yang mengelilingi gambar rambu lalu lintas. Setelah semua data sudah dianotasi kemudian data dibagi menjadi data *train*, *val*, dan *test*. Setelah data terbagi dataset dapat diunduh dengan format YOLOv8 (TXT dan YAML) yang nantinya akan digunakan untuk proses pelatihan.

2. *Pre-processing* Gambar

Pada tahap *pre-processing* dilakukan proses *resize*. *Resize* yang dapat disebut juga dengan mengubah ukuran merupakan langkah penting dalam prapemrosesan gambar untuk deteksi objek dengan model pembelajaran mendalam. Proses ini mengubah ukuran gambar ke dimensi yang dibutuhkan oleh model. YOLOv8 mungkin memerlukan gambar masukan berukuran 416x416 piksel. Artinya setiap gambar yang diproses harus diubah ukurannya menjadi 416 x 416 piksel. Proses ini dilakukan dengan mendefinisikan

'imgsz=416' pada coding yang digunakan.. Mengubah ukuran yang sesuai akan membantu model mengenali objek dengan lebih akurat dan efisien.

3. Pelatihan Model

Tahap pelatihan model dalam penelitian ini dilakukan dengan menggunakan *Google Colab*. Pada tahap pembuatan model pada penelitian ini, peneliti menggunakan model YOLOv8. Model dilatih menggunakan kumpulan data rambu lalu lintas yang diperoleh dari *Roboflow* guna mengklasifikasikan beragam jenis rambu lalu lintas.

```
# Build from YAML and transfer weights
Final_model = YOLO('yolov8n.yaml').load('yolov8n.pt')

# Training The Final Model
Result_Final_model = Final_model.train(data="D:\\jesi\\roadsign\\data.yaml",
epochs=50, imgsz = 416, batch = 64 ,lr0=0.0001, dropout= 0.15, device = 'cpu']
```

Gambar 3. 3 Pelatihan Model

Gambar 3.3 merupakan tahap pelatihan untuk model YOLOv8. Sebagai langkah awal, model YOLOv8 ditempatkan melalui file konfigurasi 'yolov8n.yaml' dan file bobot model terlatih 'yolov8n.pt'. Model tersebut kemudian dilatih menggunakan *dataset*. Pada penelitian ini pengenalan rambu lalu lintas berbasis YOLOV8 mengaplikasikan nilai *hyperparameter learning rate* (lr0) sebesar 0,0001, *image size* sebesar 416x416 piksel, *batch* sebesar 64, *dropout* sebesar 15%, dan jumlah *epoch* sebanyak 50. Kombinasi hyperparameter ini digunakan agar mendapat nilai matrik yang tinggi pada hasil pelatihan model. Proses tersebut menghasilkan *frame* dengan *bounding box* pada gambar *dataset*. Data gambar yang terlihat dapat dideteksi dan diklasifikasi sesuai dengan proses pelatihan yang telah dilakukan. Sedangkan gambar yang tertutup benda seperti daun dan tiang tidak dapat terdeteksi oleh sistem.

4. Evaluasi Model

Setelah proses pelatihan selesai, model YOLOv8 diuji pada kumpulan data yang belum pernah dilihat sebelumnya. Performa model YOLOv8 dievaluasi menggunakan matriks konfusi yang dihasilkan selama proses

training. Metrik layaknya Akurasi, *Precision*, *Recall*, dan *Mean Average Precision* (MAP) digunakan untuk mengevaluasi performa model.

5. Hasil Penggunaan Model YOLOv8

Pada tahap hasil akan memberikan keluaran berupa gambar dengan bounding box yang mengelilingi rambu lalu lintas dan terdapat nama gambar rambu di atasnya sesuai dengan gambar rambu yang dideteksi. Hal tersebut bertujuan untuk mengklasifikasikan gambar rambu sesuai dengan nama rambu lalu lintasnya.

3.5. Analisis Kebutuhan

Penelitian ini menganalisis perangkat lunak yang diperlukan agar pengembangan aplikasi dapat berfungsi sesuai yang diharapkan mulai dari proses input hingga hasil akhir. Berikut daftar perangkat yang digunakan dalam pengembangan sistem:

1. Komputer

Komputer yang digunakan pada penelitian ini adalah laptop Avita Pura NS15A6 AMD Ryzen 3 3200U *with Radeon Vega Mobile Gfx* 2.60 GHz. Ram 8GB dengan 64-bit *operating system*.

2. Kamera

Kamera yang digunakan dalam penelitian ini adalah *webcam external* dari Logitech C270. Kamera ini memiliki resolusi maksimal 720p/30fps dengan kamera *mega pixel* 0.9 dan jangkauan mikofon maksimal 1 meter.

3. Python 3.11.9

Python merupakan bahasa pemrograman tingkat tinggi yang fleksibel yang berfokus pada keterbacaan dan kesederhanaan. Diproduksi oleh Guido van Rossum dan pertama kali tersedia pada tahun 1991, *Python* menjadi salah satu bahasa pemrograman paling terkenal dan banyak digunakan di seluruh dunia. Penelitian ini menggunakan bahasa pemrograman *Python* versi 3.11.9 karena memiliki perpustakaan yang lengkap dan bersifat *open source*.

4. Anaconda

Anaconda merupakan distribusi *open source* dari bahasa pemrograman *Python* yang dimanfaatkan guna mengembangkan ilmu data, pembelajaran mesin, komputasi ilmiah, dan aplikasi terkait lainnya. Dalam penelitian ini, menggunakan Anaconda untuk membuat lingkungan *virtual*. Anaconda memungkinkan membuat lingkungan *virtual* terisolasi untuk berbagai proyek.

5. Google Colab

Google Colab atau *Google Colaboratory* merupakan layanan yang disediakan oleh *Google* yang dapat digunakan pengguna untuk menulis dan menjalankan kode *Python* di *browser*. *Colab* berguna sebagai pembelajaran mesin, analisis data, dan pendidikan karena memberikan akses gratis ke sumber daya komputasi canggih seperti GPU dan TPU. Dalam penelitian ini, menggunakan *Google Colab* untuk melatih model dan mendapatkan nilai akurasi tertinggi. *Google Colab* digunakan karena menyediakan T4 GPU yang dapat mempercepat pemrosesan *training* data.

6. Visual Studio Code (VSCode)

Visual Studio Code (VSCode) merupakan *editor* teks untuk kode sumber yang dibuat oleh *Microsoft*. Dalam penelitian ini, *VSCode* digunakan untuk membuat dan menjalankan program. Karena *VSCode* memiliki integrasi *Git* bawaan yang memungkinkan pengguna mengelola repositori, melakukan, mendorong dan menarik, dan melihat riwayat perubahan kode langsung dari *editor*. Dalam penelitian ini *VSCode* digunakan untuk proses deteksi *realtime*.

7. Open Source Computer Vision (OpenCV)

Penelitian ini menggunakan *Open Source Computer Vision Library (OpenCV)* untuk manipulasi gambar dan *video*. Ini membantu membaca, melihat dan menyimpan *video* dan menyediakan berbagai fitur pengeditan *video*.

8. PyTorch 2.3.0

PyTorch adalah perpustakaan pembelajaran mendalam sumber terbuka yang dikembangkan oleh *AI Research Lab* (FAIR) Facebook. Dalam penelitian ini, perpustakaan PyTorch 2.3.0 digunakan untuk membangun dan melatih model jaringan saraf guna melakukan komputasi numerik yang efisien. Paket Pytorch yaitu *torchvision* dimanfaatkan untuk mempermudah proses pengolahan citra dan deteksi objek dalam model *deep learning*.

9. Ultralytics

Pustaka Ultralytics adalah implementasi resmi Ultralytics untuk YOLOv5 dan YOLOv8. Ultralytics berfokus pada pengembangan model YOLO yang digunakan untuk berbagai aplikasi visi komputer, termasuk deteksi objek, segmentasi gambar, dan klasifikasi. Pustaka ini digunakan untuk melatih, memvalidasi, dan menyimpulkan model YOLO.

10. *Library* Matplotlib

Matplotlib adalah perpustakaan plot untuk visualisasi data. *Library* ini berguna untuk menampilkan hasil deteksi objek dalam gambar. Matplotlib memungkinkan membuat beragam jenis grafik, meliputi grafik garis, grafik batang, histogram, dan plot sebar.

11. *Library* Pdanas

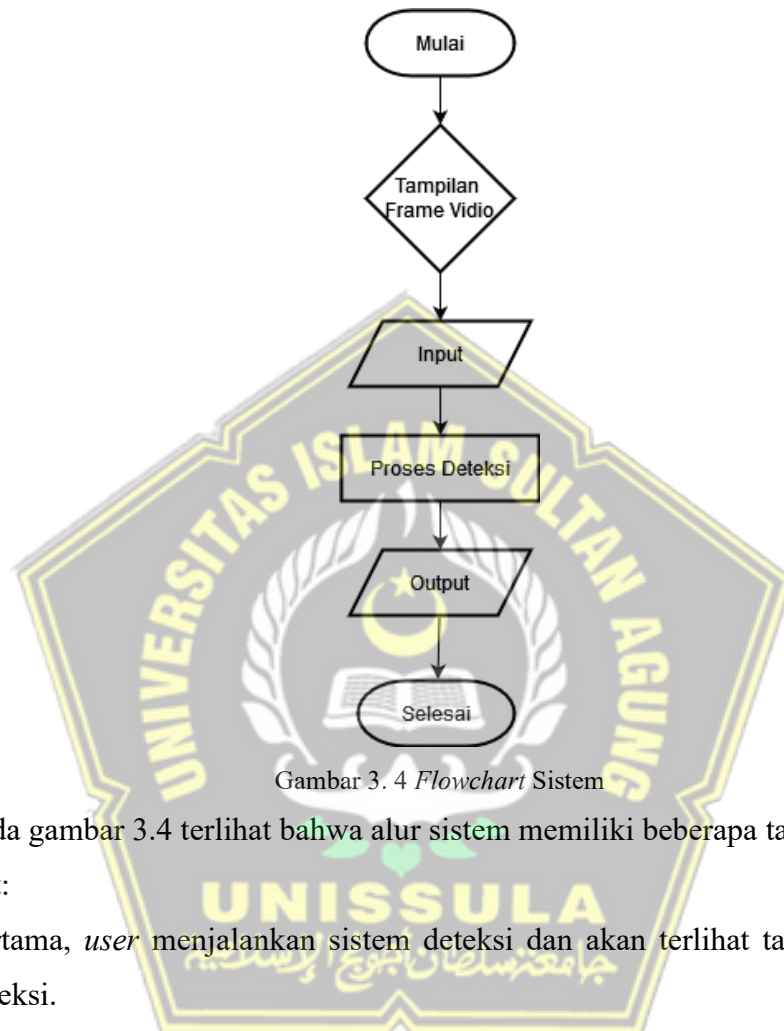
Pdanas merupakan perpustakaan Python yang sangat terkenal guna manipulasi dan analisis data. Pdanas menghadirkan struktur data yang mudah digunakan dan berkinerja tinggi, seperti *DataFrame*, dan berbagai fungsi untuk melakukan manipulasi data yang kompleks dan efisien. Penelitian ini menggunakan Pdanas untuk memuat dan menyimpan data dalam format CSV.

12. *Library* Numpy

NumPy (*Numerical Python*) adalah pustaka Python yang dapat melakukan operasi numerik dengan efisiensi tinggi. NumPy diintegrasikan dengan OpenCV untuk melakukan operasi pemrosesan gambar yang lebih kompleks.

3.6. Analisis Sistem

Pada tahap analisis sistem, dilakukan analisis untuk membuat alur kerja sistem dengan menggunakan diagram alur seperti gambar 3.4.



Gambar 3.4 *Flowchart* Sistem

Pada gambar 3.4 terlihat bahwa alur sistem memiliki beberapa tahapan sebagai berikut:





- Pertama, *user* menjalankan sistem deteksi dan akan terlihat tampilan sistem deteksi.
- Kemudian, *user* melakukan proses *input* dimana kamera akan menangkap gambar rambu lalu lintas yang ada.
- Sistem akan memproses dan mendeteksi gambar rambu yang ditangkap kamera sesuai dengan model yang sudah dilatih dan divalidasi.
- Jika gambar memiliki kecocokan dengan data yang telah dilatih dan divalidasi maka sistem akan memberikan hasil berupa nama rambu lalu lintas dalam *bounding box* yang terdapat dalam sistem. Jika tidak ada kecocokan antara gambar yang diinputkan dengan data yang dilatih maka sistem tidak akan bisa mendeteksi gambar tersebut.

BAB IV HASIL ANALISIS PENELITIAN

4.1. Hasil Pengumpulan Data

Dalam penelitian ini menggunakan *dataset* yang mencakup 2390 gambar yang terdiri dari 17 jenis rambu lalu lintas. Jenis rambu lalu lintas tersebut terdiri dari:

Tabel 4. 1 Jenis Rambu Dataset

No	Nama Rambu	Gambar Rambu	Jumlah Gambar
1	Petunjuk Halte Bus		120
2	Petunjuk Masuk Jalur Kiri-Kanan		117
3	Dilarang Masuk		144
4	Petunjuk Penyebrangan Pejalan Kaki		111

No	Nama Rambu	Gambar Rambu	Jumlah Gambar
5	Dilarang Belok Kiri		157
6	Peringatan Simpang Tiga		170
7	Berhenti		144
8	Dilarang Berhenti		161
9	Peringatan Lampu Lalu Lintas		156
10	Petunjuk Putar Balik		78

No	Nama Rambu	Gambar Rambu	Jumlah Gambar
11	Dilarang Parkir		146
12	Dilarang Belok Kanan		146
13	Peringatan		175
14	Dilarang Putar Balik		168
15	Peringatan Pintu Perlintasan Kereta Api		164
16	Petunjuk Jalur Kiri		120

No	Nama Rambu	Gambar Rambu	Jumlah Gambar
17	Petunjuk Parkir		113

Dataset yang digunakan dibagi menjadi data *training*, *validation*, dan *testing*.

Tabel 4.2 merupakan pembagian dataset.

Tabel 4. 2 Pembagian Dataset

No	Data	Jumlah Gambar
1	<i>Training</i>	1687
2	<i>Validation</i>	526
3	<i>Testing</i>	177

Dataset gambar yang digunakan dalam penelitian ini memiliki resolusi setiap gambar 416x416 piksel. Gambar 4.1 merupakan beberapa sampel gambar dari *dataset* yang digunakan yang dipilih secara random oleh sistem.



Gambar 4. 1 Sampel *Dataset* yang Dipilih Secara Random

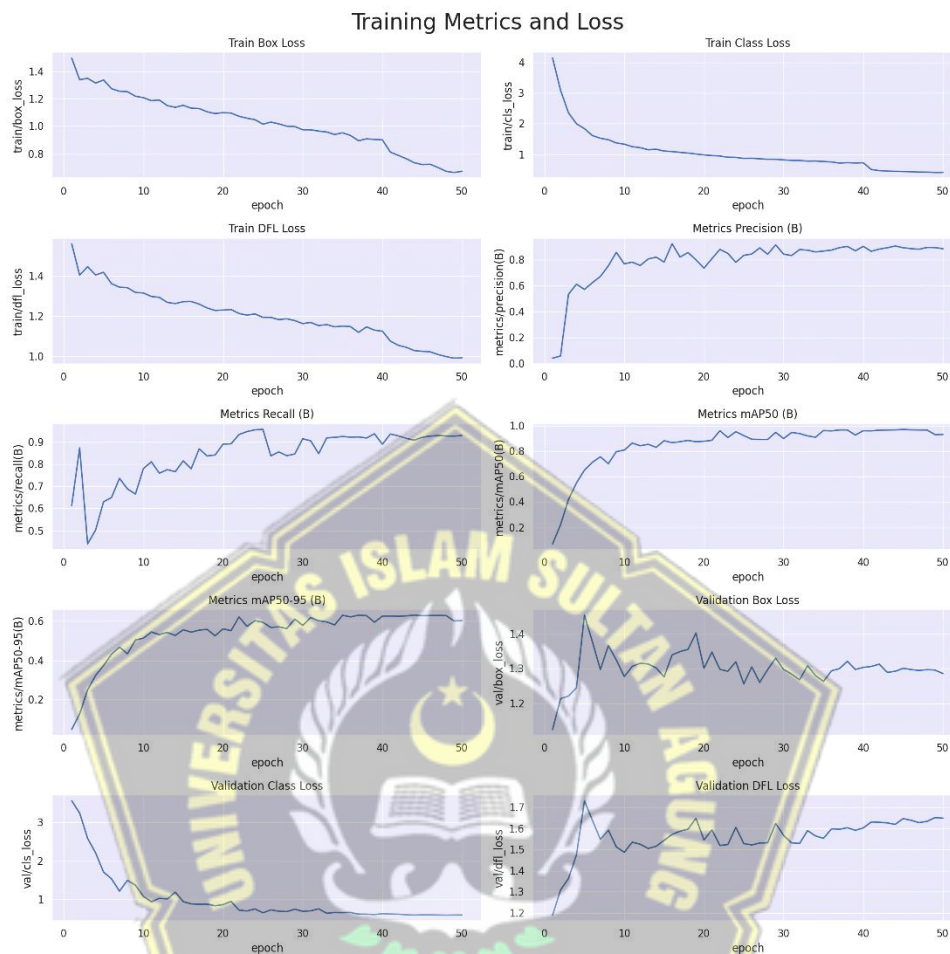
4.2. Hasil Evaluasi

Tahap evaluasi pada model deteksi objek YOLOv8 sangatlah penting. Pertama, model diuji akurasi guna mengenali objek pada gambar yang tidak diketahui. Biasanya, metrik layaknya *Precision*, *Recall*, dan *Mean Average Precision* (MAP) dimanfaatkan guna menghitung kinerja Model. Setelah model cukup akurat, label prediksi akan diuji. Artinya, memeriksa apakah model tidak hanya mendeteksi eksistensi objek pada gambar dengan benar, tetapi juga mengidentifikasi kelas objek tersebut dengan benar. Pada penelitian ini hasil dari evaluasi model adalah sebagai berikut:

Tabel 4. 3 Hasil Evaluasi

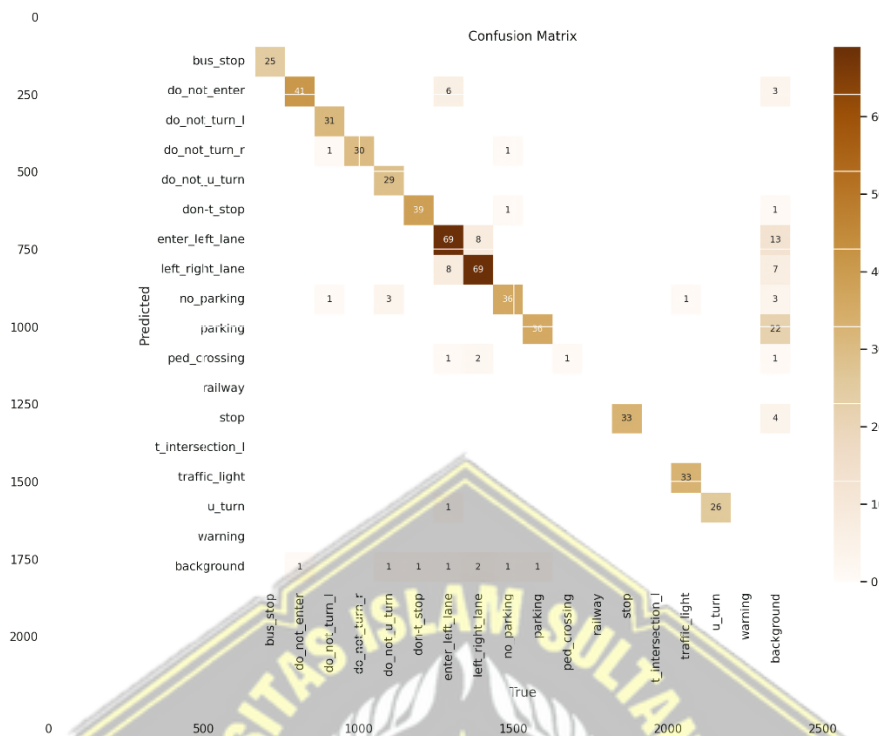
Data	Akurasi	<i>Precision</i>	<i>Recall</i>	MAP50	MAP50-95
<i>Data Training</i>	97.90%	0.978	0.989	0.987	0.825
<i>Data Validation</i>	90.50%	0.905	0.908	0.965	0.631
<i>Data Testing</i>	81.80%	0.818	0.777	0.789	0.403

Hasil pada tabel 4.3 menunjukkan nilai yang diperoleh mendekati nilai 1 yang berarti bahwa model deteksi objek berfungsi dengan optimal dan dapat mendeteksi serta mengklasifikasi gambar rambu lalu lintas dengan akurasi yang cukup tinggi. Hal ini membuktikan kualitas kumpulan data yang dipergunakan pada pelatihan dan efektivitas arsitektur model YOLOv8 cukup baik.



Gambar 4. 2 Grafik *loss*, matriks *training* dan validasi model

Gambar 4.2 adalah Kurva *Training Metrics* dan *Loss* adalah visualisasi yang digunakan untuk memantau performa model selama proses pelatihan. Kurva ini membantu memahami bagaimana model belajar dari data dan membantu mengidentifikasi masalah seperti *overfitting* atau *underfitting*. Selama proses pelatihan dan validasi, grafik kerugian (mencakup *Box loss*, *Class loss*, *DFL Loss*) dan grafik metrik (misalnya *Precision*, *Recall*, *MAP50* dan *MAP50-95*) akan ditampilkan. Sesuai yang dipaparkan pada Gambar 4.2, bagan menunjukkan penurunan tingkat kerugian yang konsisten dan peningkatan nilai metrik yang stabil selama 50 *epoch*. Hal ini menunjukkan performa model yang baik dalam mengklasifikasikan berbagai jenis rambu lalu lintas.




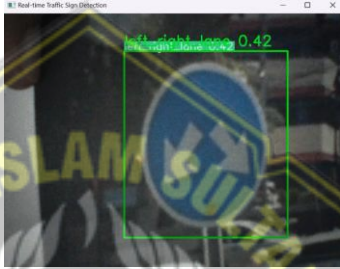









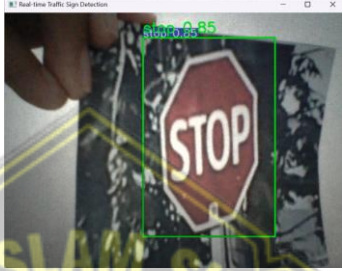
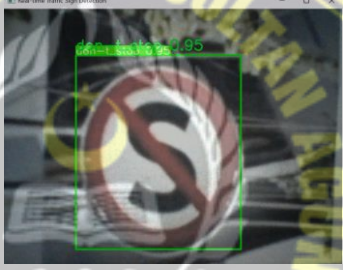


Gambar 4.3 *Confusion Matrix*






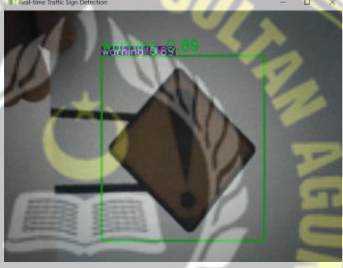

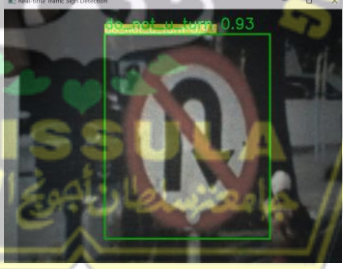


Gambar 4.3 merupakan gambar tabel *Confusion Matrix* adalah alat evaluasi penting yang memberikan gambaran rinci tentang kinerja model klasifikasi dengan memperlihatkan total perkiraan data yang benar dan salah untuk setiap kelas dalam bentuk matriks. Matriks konfusi membuktikan bahwa model dapat mengklasifikasi beragam jenis rambu lalu lintas secara tepat. Matriks tersebut diberi kode warna, yaitu warna gelap menggambarkan nilai yang lebih tinggi. Bagian warna di bagian kanan menggambarkan nilai dari 0 sampai 50. Bagian diagonal dari kiri atas ke kanan bawah berwarna gelap, menggambarkan perkiraan yang tepat dari model. Dengan demikian, hasil membuktikan bahwasannya model berhasil untuk tugas klasifikasi rambu lalu lintas.





4.3. Implementasi *Real-Time*

Tabel 4. 4 Hasil Implementasi

No	Gambar Rambu	Implementasi <i>Real-Time</i>	Hasil
1			Tepat
2			Tepat
3			Tepat
4			Tepat
5			Tepat

No	Gambar Rambu	Implementasi <i>Real-Time</i>	Hasil
6			Tepat
7			Tepat
8			Tepat
9			Tepat
10			Tepat

No	Gambar Rambu	Implementasi <i>Real-Time</i>	Hasil
11			Tepat
12			Tepat
13			Tepat
14			Tepat
15			Tepat

No	Gambar Rambu	Implementasi <i>Real-Time</i>	Hasil
16			Tepat
17			Tepat

Tabel 4.4 menunjukkan ketepatan hasil implementasi *realtime* gambar rambu lalu lintas yang telah dilakukan oleh penulis. Hasil tersebut menunjukkan sistem dapat mengidentifikasi gambar rambu lalu lintas dan dapat mengklasifikasikan gambar rambu lalu lintas melalui *bounding box* yang mengelilingi gambar dan label kelas dengan tepat sesuai informasi yang terkandung dalam rambu. Serta *confidence score* atau skor kepercayaan yang juga ditampilkan disamping label kelas. Akan tetapi poses tersebut hanya dapat sesuai jika gambar rambu tidak tertutup benda apapun seperti daun, tiang, dll.

BAB V

KESIMPULAN DAN SARAN

5.1. Kesimpulan

Dari hasil penelitian ini, dapat disimpulkan bahwa pengenalan rambu lalu lintas menggunakan model YOLOv8 berhasil mengenali jenis rambu lalu lintas yang berisi 17 jenis rambu yang berbeda secara *realtime*. Hasil pelatihan menunjukkan model YOLOv8 berkinerja optimal dengan nilai akurasi sebesar 97.90%, nilai *precision* senilai 0.978, *recall* senilai 0.989, MAP50 senilai 0.987, dan MAP50-95 senilai 0.825. Nilai tersebut membuktikan bahwasannya model dapat mengidentifikasi dan mengklasifikasikan rambu lalu lintas dengan akurasi yang cukup tinggi. Dengan demikian, penelitian ini membuktikan bahwasannya terdapat kemampuan besar untuk penggunaan model YOLOv8 dalam pengenalan rambu lalu lintas dan klasifikasi rambu lalu lintas.

5.2. Saran

Berdasarkan penelitian yang telah dilakukan peneliti memiliki saran untuk pengembangan maupun penelitian selanjutnya, yaitu sebagai berikut:

1. Penelitian ini dapat menjadi referensi bagi penelitian selanjutnya yang akan mengimplementasikan model di perangkat *Edge AI* seperti *Jetson Nano*, untuk simulasi *Autonomous Vehicle*.
2. Penelitian ini hanya berfokus pada 17 jenis rambu lalu lintas yang berbeda. Penelitian di masa depan dapat memperluas cakupannya dengan menambahkan jenis rambu lalu lintas yang ada. Lebih banyak data memberi model semakin besar peluang untuk mempelajari beragam jenis rambu lalu lintas, yang pada akhirnya dapat meningkatkan akurasi dan performa model secara keseluruhan.
3. Penelitian ini hanya berfokus pada satu kondisi cuaca yaitu cuaca normal. Penelitian selanjutnya dapat mengembangkan dan memperluas cakupan keadaan seperti malam hari, hujan, dan berkabut untuk meningkatkan robusta model YOLOv8 dalam berbagai situasi nyata.

DAFTAR PUSTAKA

- Akbar, M. (2022) 'Pengenalan Rambu Lalu-lintas menggunakan Convolutional Neural Network (Studi Kasus: Rambu Lalu-lintas Indonesia)', *InfoTekJar : Jurnal Nasional Informatika dan Teknologi Jaringan*, 6(2), pp. 271–276. Available at: <https://doi.org/10.30743/infotekjar.v6i2.4564>.
- Danrekha, M. Z. dan Huda, Y. (2021) 'Deteksi Warna Manggis Menggunakan Pengolahan Citra dengan Opencv Python', *Voteteknika (Vocational Teknik Elektronika dan Informatika)*, 9(4), p. 27. doi: 10.24036/voteteknika.v9i4.114251.
- Armin, E. U. dkk. (2023) 'Performa Model YOLOv8 untuk Deteksi Kondisi Mengantuk pada pengendara mobil', *BRAHMANA: Jurnal Penerapan Kecerdasan Buatan*, 5(1), pp. 67–76.
- Atria, R. (2023) 'Pendeteksi Masker pada Gambar Menggunakan Model Deep Learning Yolo-v2 dengan ResNet-50', 10(5), pp. 4967–4973.
- Christoffel, J. M., Tulenan, V. dan Sengkey, R. (2019) 'Aplikasi Augmented Reality Pengenalan Rambu Lalu Lintas menggunakan Metode User Defined Target', *Jurnal Teknik Informatika*, 14(3), pp. 349–356.
- Habibi, M. N. dkk. (2021) 'Implementasi Advanced Driver Assistance System Menggunakan Raspberry Pi Dengan Metode Shape Detection Dalam Mengenali Rambu-Rambu Lalu Lintas Di Indonesia (Implementation Advanced Driver Assistance System Using Raspberry Pi With Shape Detection Method for', 8(5), pp. 6212–6221.
- Kediri, U. N. P. dan Penelitian, T. (2022) 'Seminar Nasional Inovasi Teknologi IMPLEMENTATION OF FACE RECOGNITION FOR ATTENDANCE USING YOLO V3 METHOD Teknik Informatika , Fakultas Teknik , Universitas Nusantara PGRI Kediri Teknik Mesin , Fakultas Teknik , Universitas Nusantara PGRI Kediri Seminar N', pp. 50–55.
- Kuncoro, N. C., Wibowo, S. A. dan Usman, K. (2020) 'Analisis Kinerja Prototipe Traffic Sign Recognition Untuk Sistem Autonomous Car Menggunakan You Only Look Once Performance Analysis of Prototyping Traffic Sign

- Recognition for Autonomous Car System By Using You Only Look Once', 7(3), pp. 8872–8878.
- Kurniawati, A. dan Akbar, H. (2023) 'Deteksi Lampu Lalu Lintas Menggunakan YOLO untuk Autonomous Car', *Journal of Computer Engineering, Network, dan Intelligent Multimedia*, 1(1), pp. 28–43. Available at: <https://jcenim.com/index.php/jcenim/article/view/6>.
- Lionirahmada, M. dan Utaminingrum, F. (2021) 'Early Warning Sistem Rambu Pembatas Kecepatan menggunakan Histogram Oriented Gradient dan Klasifikasi SVM berbasis Raspberry Pi', 5(9), pp. 4061–4068. Available at: <http://j-ptiik.ub.ac.id>.
- Maulana, I., Rahaningsih, N. dan Suprpti, T. (2024) 'Analisis Penggunaan Model Yolov8 (You Only Look Once) Terhadap Deteksi Citra Senjata Berbahaya', *JATI (Jurnal Mahasiswa Teknik Informatika)*, 7(6), pp. 3621–3627. doi: 10.36040/jati.v7i6.8271.
- Nugroho, A. dan Cahyono, M. R. A. (2022) 'Implementasi Object Recognition Pada Rambu-Rambu Dan Lampu Lalu Lintas Dengan Raspberry Pi Dengan Algoritma Yolov5', *Sebatik*, 26(2), pp. 549–556. doi: 10.46984/sebatik.v26i2.2047.
- Pauziah, R. dan Herliana, A. (2021) 'Implementasi Deteksi Rambu Lalu Lintas Menggunakan Metode Extreme Learning Machine', *E-Prosiding Sistem Informasi*, x(x), pp. 36–41.
- Putra, I. K., Dewanta, F. dan Astuti, S. (2022) 'Sistem Pintu Gerbang Otomatis Menggunakan Deep Learning Object Detection Automatic Gate System Using Deep Learning Object Detection', *e-Proceeding of Engineering*, 8(6), pp. 3965–3971.
- Sanjaya, C. dan Septanto, H. (2022) 'Pengembangan Model Pendeteksian Gambar Rambu Lalu Lintas dengan Metode Faster R-CNN dengan Library Tensorflow', 8(1), pp. 862–868.
- Sriratnasari, S. R. dkk. (2019) 'Integrated Smart Transportation using IoT at DKI Jakarta', in *2019 International Conference on Information Management dan Technology (ICIMTech)*. IEEE, pp. 531–536.

Yanto, Y., Aziz, F. dan Irmawati, I. (2023) 'Yolo-V8 Peningkatan Algoritma Untuk Deteksi Pemakaian Masker Wajah', *JATI (Jurnal Mahasiswa Teknik Informatika)*, 7(3), pp. 1437–1444. doi: 10.36040/jati.v7i3.7047.

