



PROPOSAL TUGAS AKHIR - EC234701

**SISTEM DETEKSI KENDARAAN OVERDIMENSI
SECARA *REAL-TIME* DI GERBANG TOL
MENGUNAKAN SSD MOBILENETv2 BERBASIS
*EDGE DEVICE***

Ikhwanul Abiyu Dhiyya'ul Haq

NRP 5024 211 048

Dosen Pembimbing

Dr. Arief Kurniawan, S.T., M.T.

NIP 19740907 200212 1 001

Dr. Diah Puspito Wulandari, S.T., M.Sc.

NIP 19801219 200501 2 001

Program Studi Strata 1 (S1) Teknik Komputer

Departemen Teknik Komputer

Fakultas Teknologi Elektro dan Informatika Cerdas

Institut Teknologi Sepuluh Nopember

Surabaya

2025

[Halaman ini sengaja dikosongkan]



PRE FINAL PROJECT - EC234701

***REAL-TIME OVERDIMENSION VEHICLE DETECTION
SYSTEM AT TOLL GATES UTILIZING SSD
MOBILENETv2 ON EDGE DEVICES***

Ikhwanul Abiyu Dhiyya'ul Haq

NRP 5024 211 048

Advisor

Dr. Arief Kurniawan, S.T., M.T.

NIP 19740907 200212 1 001

Dr. Diah Puspito Wulandari, S.T., M.Sc.

NIP 19801219 200501 2 001

Undergraduate Study Program of Computer Engineering

Department of Computer Engineering

Faculty of Intelligent Electrical And Informatics Technology

Sepuluh Nopember Institute of Technology

Surabaya

2025

[Halaman ini sengaja dikosongkan]

LEMBAR PENGESAHAN

SISTEM DETEKSI KENDARAAN OVERDIMENSI SECARA *REAL-TIME* DI GERBANG TOL MENGGUNAKAN SSD MOBILENETv2 BERBASIS *EDGE DEVICE*

PROPOSAL TUGAS AKHIR

Diajukan untuk memenuhi salah satu syarat memperoleh gelar Sarjana Teknik pada Program Studi S-1 Teknik Komputer Departemen Teknik Komputer Fakultas Teknologi Elektro dan Informatika Cerdas Institut Teknologi Sepuluh Nopember

Oleh: **Ikhwanul Abiyu Dhiyya'ul Haq**
NRP. 5024 211 048

Disetujui oleh Tim Penguji Pra Tugas Akhir:

Dr. Arief Kurniawan, S.T., M.T.
NIP: 19740907 200212 1 001

(Pembimbing I)

.....

Dr. Diah Puspito Wulandari, S.T., M.Sc.
NIP: 19801219 200501 2 001

(Pembimbing II)

.....

Dr. Eko Mulyanto Yuniarno, S.T., M.T..
NIP: 19680601 199512 1 009

(Penguji I)

.....

Arta Kusuma Hernanda, S.T., M.T..
NIP: 19962023 11024

(Penguji II)

.....

-.
NIP: -

(Penguji III)

.....

Mengetahui,
Kepala Departemen Teknik Komputer FTEIC - ITS

Dr. Arief Kurniawan, S.T., M.T..
NIP. 19740907 200212 1 001

SURABAYA
Februari, 2025

[Halaman ini sengaja dikosongkan]

APPROVAL SHEET

REAL-TIME OVERDIMENSION VEHICLE DETECTION SYSTEM AT TOLL GATES UTILIZING SSD MOBILENETv2 ON EDGE DEVICES

PRE FINAL PROJECT

Submitted to fulfill one of the requirements
for obtaining a degree Bachelor of Engineering at
Undergraduate Study Program of Computer Engineering
Department of Computer Engineering
Faculty of Intelligent Electrical And Informatics Technology
Sepuluh Nopember Institute of Technology

By: Ikhwanul Abiyu Dhiyya'ul Haq
NRP. 5024 211 048

Approved by Pre Final Project Examiner Team:

Dr. Arief Kurniawan, S.T., M.T.
NIP: 19740907 200212 1 001

(Advisor I)

.....

Dr. Diah Puspito Wulandari, S.T., M.Sc.
NIP: 19801219 200501 2 001

(Co-Advisor II)

.....

Dr. Eko Mulyanto Yuniarno, S.T., M.T..
NIP: 19680601 199512 1 009

(Examiner I)

.....

Arta Kusuma Hernanda, S.T., M.T..
NIP: 19962023 11024

(Examiner II)

.....

-.
NIP: -

(Examiner III)

.....

Acknowledged,
Head of Computer Engineering Department FIEI - ITS

Dr. Arief Kurniawan, S.T., M.T..
NIP. 19740907 200212 1 001

SURABAYA
February, 2025

[Halaman ini sengaja dikosongkan]

PERNYATAAN ORISINALITAS

Yang bertanda tangan dibawah ini:

Nama Mahasiswa / NRP : Ikhwanul Abiyu Dhiyya'ul Haq / 5024 211 048
Departemen : Teknik Komputer
Dosen Pembimbing / NIP : Dr. Arief Kurniawan, S.T., M.T. / 19740907 200212 1 001

Dengan ini menyatakan bahwa Tugas Akhir dengan judul "SISTEM DETEKSI KENDARAAN OVERDIMENSI SECARA *REAL-TIME* DI GERBANG TOL MENGGUNAKAN SSD MOBILENETv2 BERBASIS *EDGE DEVICE*" adalah hasil karya sendiri, berfsifat orisinal, dan ditulis dengan mengikuti kaidah penulisan ilmiah.

Bilamana di kemudian hari ditemukan ketidaksesuaian dengan pernyataan ini, maka saya bersedia menerima sanksi sesuai dengan ketentuan yang berlaku di Institut Teknologi Sepuluh Nopember.

Surabaya, Februari 2025

Mengetahui
Dosen Pembimbing

Mahasiswa

Dr. Arief Kurniawan, S.T., M.T.
NIP. 19740907 200212 1 001

Ikhwanul Abiyu Dhiyya'ul Haq
NRP. 5024 211 048

[Halaman ini sengaja dikosongkan]

STATEMENT OF ORIGINALITY

The undersigned below:

Name of student / NRP : Ikhwanul Abiyu Dhiyya'ul Haq / 5024 211 048
Department : Computer Engineering
Advisor / NIP : Dr. Arief Kurniawan, S.T., M.T. / 19740907 200212 1 001

Hereby declared that the Final Project with the title of "*REAL-TIME OVERDIMENSION VEHICLE DETECTION SYSTEM AT TOLL GATES UTILIZING SSD MOBILENETv2 ON EDGE DEVICES*" is the result of my own work, is original, and is written by following the rules of scientific writing.

If in future there is a discrepancy with this statement, then I am willing to accept sanctions in accordance with provisions that apply at Sepuluh Nopember Institute of Technology.

Surabaya, February 2025

Acknowledged
Advisor

Student

Dr. Arief Kurniawan, S.T., M.T.
NIP. 19740907 200212 1 001

Ikhwanul Abiyu Dhiyya'ul Haq
NRP. 5024 211 048

[Halaman ini sengaja dikosongkan]

ABSTRAK

Nama Mahasiswa : Ikhwanul Abiyu Dhiyya'ul Haq
Judul Tugas Akhir : SISTEM DETEKSI KENDARAAN OVERDIMENSI SECARA *REAL-TIME* DI GERBANG TOL MENGGUNAKAN SSD MOBILENETv2 BERBASIS *EDGE DEVICE*
Pembimbing : 1. Dr. Arief Kurniawan, S.T., M.T.
2. Dr. Diah Puspito Wulandari, S.T., M.Sc.

Penggunaan truk sebagai moda transportasi barang di Indonesia terus meningkat, namun pelanggaran *ODOL (Overdimension Overloading)* pada truk menjadi salah satu penyebab utama kecelakaan lalu lintas. Penelitian ini bertujuan mengembangkan sistem deteksi kendaraan *overdimension* menggunakan teknologi *deep learning*, khususnya *Convolutional Neural Network (CNN)*, yang dapat berjalan secara *real-time* di *edge device*. Sistem ini juga diintegrasikan dengan sistem pemantauan lalu lintas dan dilengkapi fitur notifikasi otomatis kepada pihak berwenang saat pelanggaran terdeteksi. Dengan peningkatan akurasi dan efisiensi, sistem ini diharapkan mampu memberikan solusi efektif untuk mendeteksi kendaraan *ODOL* di berbagai lokasi.

Kata Kunci: *Overdimension, ODOL, deep learning, CNN, edge device*, sistem deteksi

[Halaman ini sengaja dikosongkan]

ABSTRACT

Name : Ikhwanul Abiyu Dhiyya'ul Haq
Title : *REAL-TIME OVERDIMENSION VEHICLE DETECTION SYSTEM AT TOLL GATES UTILIZING SSD MOBILENETv2 ON EDGE DEVICES*
Advisors : 1. Dr. Arief Kurniawan, S.T., M.T.
2. Dr. Diah Puspito Wulandari, S.T., M.Sc.

The use of trucks as a mode of transportation for goods in Indonesia continues to increase, but overdimension violations (ODOL) on trucks have become a major cause of traffic accidents. This study aims to develop an overdimension vehicle detection system using deep learning technology, specifically Convolutional Neural Network (CNN), that can operate in real-time on edge devices. The system is also integrated with traffic monitoring systems and equipped with an automatic notification feature to alert authorities when violations are detected. By improving accuracy and efficiency, this system is expected to provide an effective solution for detecting ODOL vehicles across various locations.

Keywords: Overdimension, ODOL, deep learning, CNN, edge device, detection system

[Halaman ini sengaja dikosongkan]

DAFTAR ISI

ABSTRAK	i
ABSTRACT	iii
DAFTAR ISI	v
DAFTAR GAMBAR	vii
DAFTAR TABEL	ix
1 PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	1
1.3 Tujuan	2
1.4 Batasan Masalah	2
1.5 Manfaat	2
1.6 Sistematika Penulisan	2
2 TINJAUAN PUSTAKA	5
2.1 Hasil Penelitian Terdahulu	5
2.1.1 Deteksi Kendaraan Truk pada Video Menggunakan Metode Tiny-YOLOv4	5
2.1.2 Sistem Deteksi Kendaraan <i>Overdimension</i> secara <i>Real-time</i> menggunakan <i>Deep Learning</i> di Jalan Raya	5
2.1.3 Implementasi Sistem Penghitung Kendaraan Otomatis Berbasis <i>Computer Vision</i>	6
2.2 Teori/Konsep Dasar	6
2.2.1 <i>Neural Network</i>	6
2.2.2 <i>SSD (Single Shot MultiBox Detector)</i>	10
2.2.3 <i>Edge Computing</i>	11
2.2.4 <i>MQTT (Message Queuing Telemetry Transport)</i>	13
2.2.5 <i>RESTful API</i>	14
2.2.6 <i>Flutter</i>	14
3 METODOLOGI	17

3.1	Data dan Peralatan	17
3.1.1	Data	17
3.1.2	Peralatan	17
3.2	Metode yang Digunakan	17
3.2.1	Akuisisi Data	18
3.2.2	<i>Labelling</i> Data	18
3.2.3	Praproses Data	19
3.2.4	<i>Training</i> Data	19
3.2.5	<i>Deployment</i> di <i>Edge Device</i>	19
3.2.6	Proses Deteksi	19
3.2.7	Penyimpanan Data ke <i>Cloud</i>	19
3.2.8	Pengiriman Notifikasi ke <i>Mobile Apps</i>	20
4	HASIL YANG DIHARAPKAN	21
4.1	Hasil yang Diharapkan dari Penelitian	21
4.2	Hasil Pendahuluan	21
5	JADWAL PENELITIAN	23
	DAFTAR PUSTAKA	25

DAFTAR GAMBAR

2.1	Diagram sederhana dari <i>Neural Network</i>	6
2.2	Grafik <i>Activation function Sigmoid</i>	7
2.3	Grafik <i>Activation function Tanh</i>	8
2.4	Grafik <i>Activation function ReLU</i>	8
2.5	Perbedaan antara <i>Feed Forward</i> dan <i>Backpropagation</i>	9
2.6	Arsitektur SSD	11
2.7	Jetson Nano Developer Kit	12
2.8	Beelink Gemini T34	13
3.1	Blok diagram metodologi	18
3.2	Ilustrasi pengelompokan kendaraan overdimensi atau tidak	18
3.3	Diagram basis data	20
4.1	Hasil <i>re-training dataset</i> dengan SSD MobileNetV2	21

[Halaman ini sengaja dikosongkan]

DAFTAR TABEL

5.1	Tabel <i>timeline</i> penelitian	23
-----	--	----

[Halaman ini sengaja dikosongkan]

BAB I

PENDAHULUAN

1.1 Latar Belakang

Penggunaan truk sebagai moda transportasi pengiriman barang di Indonesia terus meningkat setiap tahunnya. Berdasarkan data Badan Pusat Statistik tahun 2023 [1], truk merupakan kendaraan terbanyak ketiga setelah sepeda motor dan mobil penumpang, dengan jumlah yang signifikan di berbagai provinsi. Namun, penggunaan truk sebagai moda transportasi pengiriman barang juga menimbulkan dampak negatif, salah satunya adalah kecelakaan lalu lintas. Berdasarkan jurnal Analisis Pengaruh Kendaraan *ODOL* terhadap Tingkat Kecelakaan di Jalan Tol [2] didapatkan bahwa pengaruh kendaraan *ODOL* terhadap tingkat kecelakaan di jalan tol berhasil dibuktikan sebesar 32% dengan sisanya merupakan faktor lainnya. Kerugian dari kecelakaan *ODOL* juga menyebabkan penurunan kecepatan rata-rata kendaraan di jalan tol sebesar 12%.

Selain itu, pemeriksaan *ODOL* oleh UPPKB (Unit Pelaksana Penimbangan Kendaraan Bermotor) menyatakan bahwa pada tahun 2023, rata-rata kendaraan yang masuk dan diperiksa hanya berkisar di angka 5%. Dari jumlah tersebut, sebanyak 27,95% kendaraan terbukti melakukan pelanggaran. Mayoritas pelanggaran terkait dengan kelebihan muatan, mencapai 69%, sementara sisanya sebesar 31% berkaitan dengan pelanggaran dokumen. Kendaraan yang melanggar ketentuan muatan rata-rata membawa beban berlebih antara 5% hingga 20% dari kapasitas yang diizinkan [3]. Kondisi ini menunjukkan bahwa metode pemeriksaan manual memiliki keterbatasan dan tidak cukup efektif untuk mengurangi pelanggaran secara signifikan.

Oleh karena itu, dibutuhkan sistem otomatis yang mampu mendeteksi pelanggaran secara *real-time*, meningkatkan cakupan pemantauan, dan memberikan notifikasi langsung kepada pengguna. Sistem ini harus mampu mengidentifikasi kendaraan yang melanggar batas dimensi yang diizinkan, adapun teknologi yang digunakan adalah *deep learning* terkhusus *Convolutional Neural Network* (CNN). Implementasi model *deep learning* pada perangkat *edge device* akan memungkinkan sistem dapat berjalan secara *real-time* dan dapat diaplikasikan pada berbagai lokasi tanpa memerlukan infrastruktur yang rumit.

1.2 Rumusan Masalah

Pelanggaran *overdimension* pada kendaraan bermotor merupakan masalah serius yang dapat menimbulkan kerugian baik dari segi ekonomi maupun keselamatan. Namun, metode pemeriksaan manual yang saat ini digunakan oleh UPPKB memiliki keterbatasan dan tidak cukup efektif untuk mengurangi pelanggaran secara signifikan. Oleh karena itu, diperlukan sistem deteksi otomatis yang mampu mendeteksi kendaraan *overdimension* secara *real-time* dengan tingkat akurasi yang tinggi dan efisiensi yang optimal. Berdasarkan latar belakang tersebut, rumusan masalah yang diangkat dalam penelitian ini adalah sebagai berikut:

1. Bagaimana cara menerapkan sistem deteksi agar dapat dijalankan pada perangkat *edge device*?

2. Bagaimana cara membuat sistem deteksi yang *scalable* dan dapat diintegrasikan dengan sistem yang sudah ada?
3. Bagaimana cara menambahkan fitur notifikasi otomatis kepada pihak berwenang saat terjadi pelanggaran *overdimension*?

1.3 Tujuan

Tujuan utama dari penelitian ini yaitu:

1. Menerapkan sistem deteksi agar dapat dijalankan pada perangkat *edge device*.
2. Membuat sistem deteksi yang *scalable* dan dapat diintegrasikan dengan sistem yang sudah ada.
3. Menambahkan fitur notifikasi otomatis kepada pihak berwenang saat terjadi pelanggaran *overdimension*.

1.4 Batasan Masalah

Penelitian ini terfokus pada penggunaan model *deep learning* terkhusus *Convolutional Neural Network* (CNN) untuk mendeteksi kendaraan *overdimension* secara *real-time*. Adapun batasan masalah yang diangkat dalam penelitian ini adalah sebagai berikut:

1. Pembatasan pada model *deep learning* spesifik untuk deteksi kendaraan *overdimension*, tanpa mengeksplorasi model lain.
2. Pengujian awal sistem mungkin dilakukan dalam lingkungan terkendali, namun diharapkan dapat diaplikasikan pada berbagai lokasi dengan kondisi yang berbeda.

1.5 Manfaat

Adapun manfaat yang didapat pada pelaksanaan tugas akhir ini adalah sebagai berikut:

1. Memberikan kontribusi pada pengembangan teknologi deteksi kendaraan *overdimension* menggunakan model *deep learning*.
2. Memberikan solusi yang efektif dan efisien dalam mendeteksi kendaraan *overdimension* secara *real-time*.
3. Memberikan rekomendasi untuk penelitian selanjutnya dalam pengembangan sistem deteksi kendaraan *overdimension*.

1.6 Sistematika Penulisan

Dalam pembuatan laporan penelitian tugas akhir ini akan terbagi menjadi lima bagian bab yang meliputi:

1. **BAB I Pendahuluan**

Bab ini berisi penjelasan mengenai latar belakang yang mengarah pada permasalahan yang akan diangkat serta solusi yang diberikan. Selain itu terdapat pula tujuan dari penelitian serta batasan masalah dari cakupan yang akan dikerjakan.

2. **BAB II Tinjauan Pustaka**

Bab ini berisi penelitian terdahulu dengan topik yang berhubungan dengan penelitian yang akan dilakukan. Selain itu, pada bab ini dijelaskan juga mengenai teori - teori yang akan digunakan untuk membantu pengerjaan penelitian.

3. **BAB III Metodologi**

Bab ini berisi penjelasan mengenai rancangan dan metodologi penelitian secara sistematis serta pengimplementasiannya dalam setiap metode sehingga mendapatkan hasil dari penelitian.

4. **BAB IV Hasil yang Diharapkan**

Bab ini berisi penjelasan mengenai hasil yang diharapkan dari penelitian yang dilakukan. Selain itu, pada bab ini dijelaskan juga mengenai hasil pendahuluan yang telah dilakukan.

5. **BAB V Jadwal Penelitian**

Bab ini berisi penjelasan mengenai jadwal penelitian yang akan dilakukan. Selain itu, pada bab ini dijelaskan juga mengenai rencana kerja yang akan dilakukan.

[Halaman ini sengaja dikosongkan]

BAB II

TINJAUAN PUSTAKA

2.1 Hasil Penelitian Terdahulu

Penelitian ini mengambil referensi dari beberapa penelitian terdahulu dengan topik yang sama ataupun beririsan, serta dasar pengembangan dan inovasi yang akan pada penelitian ini. Berikut adalah beberapa penelitian terdahulu yang diambil sebagai referensi:

2.1.1 Deteksi Kendaraan Truk pada Video Menggunakan Metode Tiny-YOLOv4

Penelitian oleh Prismadika Yuniar et al. [4] mengkaji deteksi truk secara *real-time* dalam video pengamatan di jalan raya. Tantangan muncul karena truk memiliki dimensi serupa dengan bus dan struktur mekanis mirip pikap. Pada tahap awal, model YOLO asli hanya mampu mendeteksi bagian kepala dan bak truk tanpa memperhitungkan muatan. Oleh karena itu, dibuat *custom dataset* dengan *bounding box* untuk seluruh bagian truk beserta muatannya.

Tiny-YOLOv4 dipilih karena lebih ringan dan kompatibel dengan perangkat beragam. Pengujian pada tiga perangkat menunjukkan bahwa Laptop Acer E5 mencapai akurasi 98,2% pada 13 FPS, Laptop Lenovo Legion 5 98% pada 28 FPS, dan PC Dell Precision 3640 Tower 97,5% pada 38 FPS.

Hasil yang didapatkan yaitu FPS yang lebih tinggi cenderung menurunkan akurasi, dan perangkat dengan CPU di atas 3.0 GHz, RAM minimal 16 GB, dan GPU lebih dari 4 GB mampu menjalankan deteksi secara *real-time* dengan baik.

2.1.2 Sistem Deteksi Kendaraan *Overdimension* secara *Real-time* menggunakan *Deep Learning* di Jalan Raya

Penelitian ini dilakukan oleh Ahmad Hamayan Nabih [5]. Penelitian ini bertujuan untuk mengembangkan sistem deteksi kendaraan *overdimension* secara *real-time* menggunakan *deep learning* di jalan raya. Sistem ini menggunakan model YOLOv8 yang telah dimodifikasi untuk mendeteksi kendaraan *overdimension*. *Dataset* yang digunakan adalah *dataset* yang telah diambil dari jalan raya di Indonesia. Variasi yang digunakan yaitu variasi akurasi pada berbagai macam perangkat dan model YOLOv8.

Variasi perangkat meliputi PC dengan External VGA NVIDIA GeForce GTX 1650Ti, Laptop tanpa External VGA, dan Intel NUC i5 Gen 11. Hasil yang didapatkan yaitu PC dengan External VGA NVIDIA GeForce GTX 1650Ti mampu mendeteksi kendaraan *overdimension* dengan akurasi 85% - 90% pada 33 FPS - 63 FPS, Laptop tanpa External VGA dengan akurasi 87% - 92% pada 7 FPS - 10 FPS, dan Intel NUC i5 Gen 11 dengan akurasi 79% - 90% pada 2 FPS. Sedangkan variasi model YOLOv8 meliputi YOLOv8n dan YOLOv8s, dimana YOLOv8n di-*training* dengan 50 *epoch* dan YOLOv8s di-*training* dengan 75 *epoch*.

Kesimpulan yang didapatkan yaitu model YOLOv8n lebih baik dari segi FPS dibandingkan dengan YOLOv8s namun memiliki akurasi yang lebih rendah. Sedangkan model pem-

bandinginya, yaitu YOLOv8s memiliki akurasi yang lebih tinggi namun FPS yang lebih rendah. Kemudian juga ditemukan bahwa perangkat dengan spesifikasi yang lebih tinggi mampu mendeteksi kendaraan overdimension dengan akurasi yang lebih tinggi dan FPS yang lebih tinggi.

2.1.3 Implementasi Sistem Penghitung Kendaraan Otomatis Berbasis *Computer Vision*

Penelitian ini dilakukan oleh Dolly Indra et al. [6]. Penelitian ini bertujuan untuk mengembangkan sistem penghitung kendaraan otomatis di tempat pencucian kendaraan. Sistem ini menggunakan Raspberry Pi 4 sebagai *edge device* dan model SSD MobileNet V2 sebagai model deteksi. Sistem ini mampu menghitung jumlah kendaraan yang masuk dan keluar dari tempat pencucian kendaraan secara *real-time* yang terhubung dengan aplikasi *mobile*.

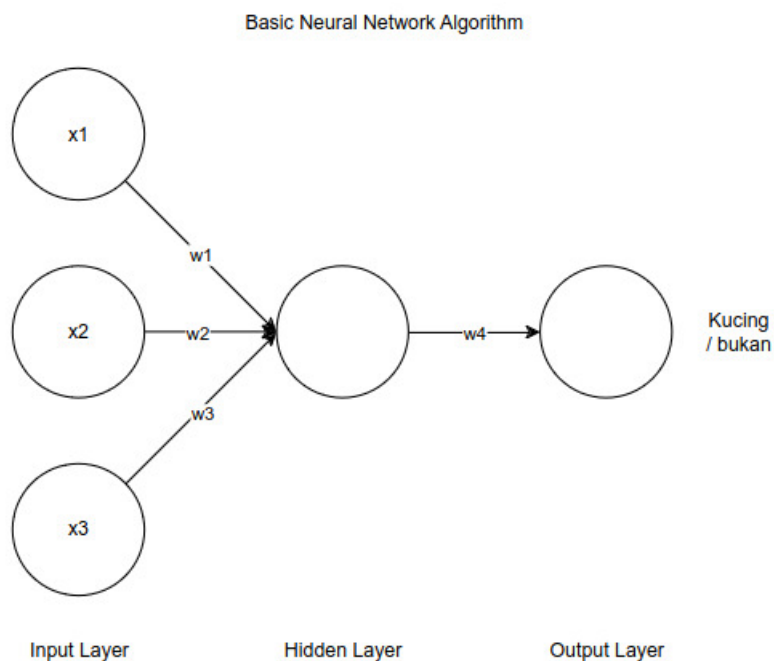
Hasil yang didapatkan yaitu sistem otomatis mampu mendeteksi kendaraan mobil dan motor dengan rata-rata tingkat akurasi sebesar 46,6 % dan aplikasi yang terdapat pada *smartphone* mampu menghitung jumlah kendaraan yang masuk dan keluar dari tempat pencucian kendaraan.

2.2 Teori/Konsep Dasar

Berikut adalah teori atau konsep dasar sebagai landasan teoritis dalam pelaksanaan penelitian ini.

2.2.1 *Neural Network*

Neural Network (NN), atau jaringan saraf, adalah fondasi dari banyak model *deep learning*, termasuk CNN. Bagian ini akan membahas konsep dasar NN mulai dari perceptron, fungsi aktivasi, mekanisme *feed forward*, *backpropagation*, *loss function*, dan *optimizer*.



Gambar 2.1: Diagram sederhana dari *Neural Network*

2.2.1.1 Perceptron

Perceptron merupakan dasar dari NN, yang diperkenalkan oleh Frank Rosenblatt pada tahun 1958 [7]. *Perceptron* adalah model jaringan saraf tiruan yang paling sederhana, yang memodelkan sebuah neuron dalam otak. Struktur dasar *perceptron* terdiri dari *input* x_1, x_2, \dots, x_n , *weights* (bobot) w_1, w_2, \dots, w_n , dan bias b . *Output* dari *perceptron* dihitung menggunakan rumus:

$$y = f(w \cdot x + b)$$

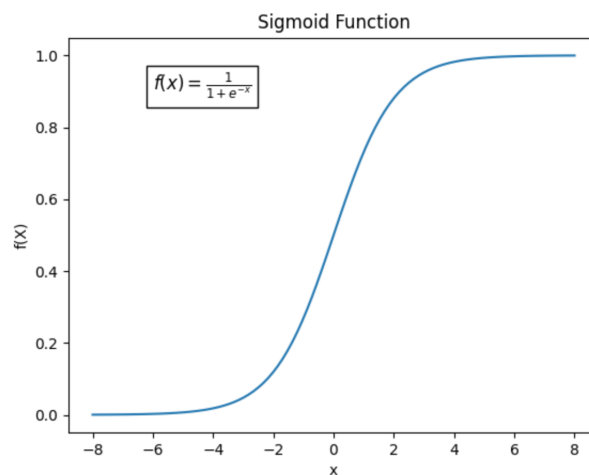
dimana f adalah fungsi aktivasi, $w \cdot x$ adalah *dot product* antara vektor bobot dan vektor input, dan b adalah bias.

2.2.1.2 Activation Function

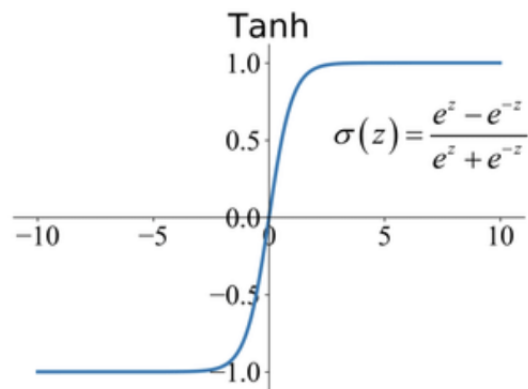
Activation function menentukan *output* dari sebuah *neuron* berdasarkan *input* total yang diterima. Beberapa fungsi aktivasi yang sering digunakan adalah:

- Sigmoid: $\sigma(x) = \frac{1}{1+e^{-x}}$. Fungsi ini menghasilkan *output* dalam rentang $(0, 1)$, yang berguna untuk tugas klasifikasi biner.
- Tanh: $\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$. Fungsi ini menghasilkan *output* dalam rentang $(-1, 1)$, yang membantu dalam mengatasi masalah *vanishing gradient*. Yaitu ketika nilai *input* yang besar atau kecil menyebabkan gradien yang sangat kecil.
- ReLU (Rectified Linear Unit): $ReLU(x) = \max(0, x)$. Fungsi ini menghasilkan *output* 0 jika *input* $x \leq 0$ dan x jika $x > 0$. ReLU adalah fungsi aktivasi yang paling umum digunakan dalam arsitektur NN modern karena sederhana dan efektif.

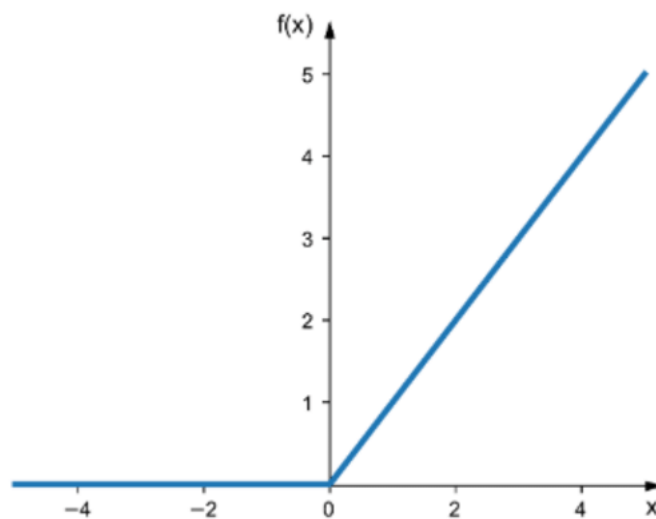
Berikut beberapa grafik dari *activation function* yang disebutkan di atas.



Gambar 2.2: Grafik *Activation function Sigmoid*



Gambar 2.3: Grafik *Activation function Tanh*



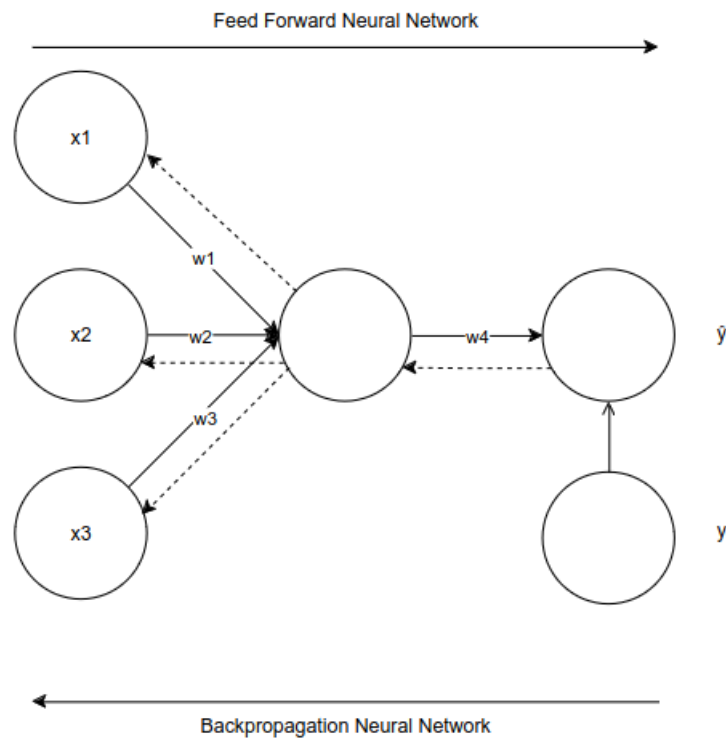
Gambar 2.4: Grafik *Activation function ReLU*

2.2.1.3 *Feed Forward*

Mekanisme *feed forward* adalah proses pengiriman input melalui lapisan-lapisan dari NN untuk mendapatkan *output*. Setiap neuron dalam lapisan menerima *input* dari semua neuron di lapisan sebelumnya, mengaplikasikan bobot dan bias, dan menghasilkan *output* menggunakan fungsi aktivasi.

2.2.1.4 *Backpropagation*

Backpropagation adalah metode yang digunakan untuk memperbarui bobot jaringan dengan cara mengoptimalkan fungsi kerugian. Proses ini melibatkan penghitungan gradien fungsi kerugian terhadap setiap bobot di jaringan dengan menggunakan aturan rantai, dan secara iteratif mengatur ulang bobot untuk meminimalkan *loss*.



Gambar 2.5: Perbedaan antara *Feed Forward* dan *Backpropagation*

2.2.1.5 Loss Function dan Optimizer

Loss function mengukur seberapa baik model NN melakukan prediksi dibandingkan dengan nilai sebenarnya. *Loss function* yang umum meliputi:

- **Gradient Descent:** Metode optimisasi yang digunakan untuk meminimalkan fungsi kerugian dengan mengupdate bobot jaringan berdasarkan gradien fungsi kerugian.
- **Stochastic Gradient Descent (SGD):** Versi sederhana dari *Gradient Descent* yang memperbarui bobot jaringan berdasarkan gradien dari satu sampel data.
- **Momentum:** Metode optimisasi yang membantu percepatan konvergensi dengan menambahkan momentum pada proses *Gradient Descent*.
- **RMSprop:** Metode optimisasi yang menyesuaikan laju pembelajaran untuk setiap parameter berdasarkan gradien rata-rata kuadrat sebelumnya.
- **Adam:** Algoritma optimisasi yang menggabungkan konsep dari *Momentum* dan *RMSprop* untuk memperbarui bobot jaringan secara adaptif.

Optimisasi adalah proses mencari parameter model yang meminimalkan fungsi kerugian. Algoritma optimisasi yang sering digunakan antara lain *Stochastic Gradient Descent* (SGD), *Adam*, dan *RMSprop*. Adapun pada penelitian ini, digunakan algoritma Adam sebagai *optimizer* untuk memperbarui bobot jaringan karena kemampuannya dalam menyesuaikan laju pembelajaran secara adaptif.

2.2.1.6 CNN (*Convolutional Neural Network*)

CNN (*Convolutional Neural Networks*) memperluas konsep NN dengan memasukkan lapisan konvolusi, yang menggunakan filter untuk mengekstraksi fitur spasial dari data *input*. Berikut merupakan beberapa konsep dasar dalam CNN:

- ***Convolutional Layer***: Menggunakan filter atau *kernel* yang 'meluncur' di atas gambar untuk menghasilkan *feature map*.
- ***Stride***: Jumlah piksel yang dilewati filter pada setiap gerakan. Stride yang lebih besar menghasilkan *feature map* yang lebih kecil.
- ***Padding***: Penambahan piksel di sekitar input untuk memungkinkan filter beroperasi di tepi gambar.
- ***Pooling Layer (Subsampling)***: Mengurangi dimensi spasial dari *feature map* untuk mengurangi jumlah parameter dan komputasi. Contohnya adalah *max pooling* dan *average pooling*.
- ***Fully Connected Layer***: Lapisan di mana setiap neuron terhubung ke semua neuron di lapisan sebelumnya, biasanya digunakan di akhir jaringan untuk mengklasifikasikan fitur yang diekstraksi oleh lapisan konvolusi.

CNN merupakan salah satu arsitektur *deep learning* yang paling banyak digunakan dalam pengolahan citra. CNN pertama kali diperkenalkan oleh Yan LeCun et al. [8] dalam penelitian tentang pengenalan tulisan tangan menggunakan LeNet-5. Model ini menggabungkan operasi konvolusi, *subsampling (pooling)*, dan *fully connected layers* untuk mengekstraksi fitur secara hierarkis dari gambar *input*.

Perkembangan CNN semakin pesat dengan adanya inovasi pada arsitektur. Salah satu inovasi signifikan adalah *Residual Neural Network (ResNet)* yang diperkenalkan oleh Kaiming He et al. [9]. ResNet memperkenalkan konsep *skip connection* yang memungkinkan model untuk belajar representasi yang lebih baik dari data. Hal ini memungkinkan model untuk dilatih dengan kedalaman yang lebih besar tanpa mengalami masalah *vanishing gradient*.

Selain itu, metode modern seperti normalisasi *batch (BatchNorm)* yang dikembangkan oleh Ioffe et al. [10] dan Szegedy et al. [11] telah menjadi standar dalam *training* CNN. *BatchNorm* membantu jaringan untuk melakukan konvergensi lebih cepat dengan menormalkan distribusi setiap *batch* selama pelatihan. Penggunaan konvolusi 1x1, seperti pada arsitektur *Inception*, juga memperkenalkan cara baru untuk mengurangi jumlah parameter sambil mempertahankan kapasitas jaringan.

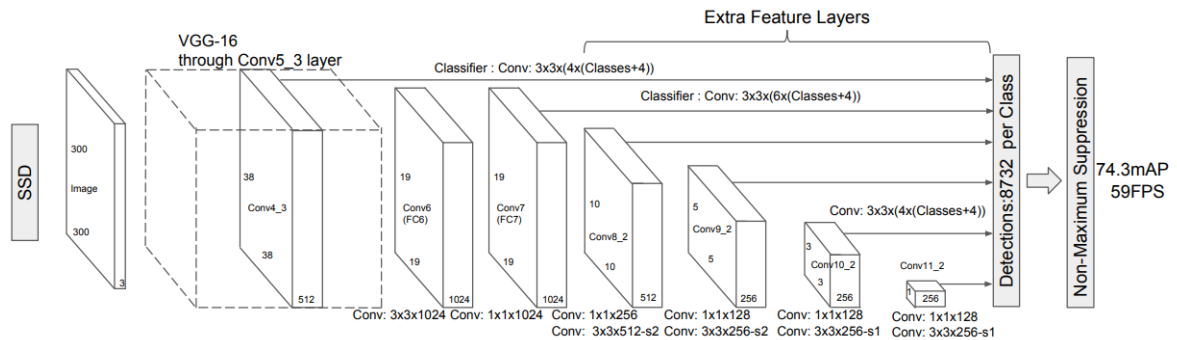
Penerapan CNN telah meluas ke berbagai domain, termasuk deteksi objek, segmentasi citra, dan pengenalan pola. Dalam konteks penelitian ini, CNN digunakan untuk mendeteksi kendaraan overdimensi berdasarkan karakteristik visualnya. Pendekatan ini sesuai dengan studi sebelumnya yang menunjukkan kemampuan CNN dalam mengolah data visual dengan tingkat akurasi yang tinggi.

2.2.2 SSD (*Single Shot MultiBox Detector*)

Single Shot MultiBox Detector (SSD) adalah arsitektur deteksi objek yang dikembangkan oleh Liu et al. [12]. SSD memperkenalkan pendekatan *single shot* yang memungkinkan model

untuk melakukan deteksi objek dan klasifikasi dalam satu langkah. Hal ini berbeda dengan pendekatan *two-stage* seperti Faster R-CNN yang memerlukan dua tahap untuk deteksi objek.

SSD menggunakan *multi-scale feature maps* untuk mendeteksi objek pada berbagai skala. Arsitektur ini terdiri dari *base network* (biasanya menggunakan arsitektur VGG16 atau ResNet) yang diikuti oleh beberapa lapisan konvolusi untuk menghasilkan *feature maps* pada berbagai resolusi. Setiap lapisan *feature map* digunakan untuk memprediksi *bounding box* dan kelas objek yang terdapat dalam gambar.



Gambar 2.6: Arsitektur SSD

SSD telah terbukti efektif dalam deteksi objek pada berbagai *dataset*, termasuk PASCAL VOC dan COCO. Keunggulan utama SSD adalah kemampuannya untuk melakukan deteksi objek secara cepat dan akurat dalam satu langkah. Dalam penelitian ini, SSD digunakan sebagai model deteksi untuk mengidentifikasi kendaraan overdimensi pada video pengamatan jalan raya.

2.2.3 Edge Computing

Edge computing adalah paradigma komputasi yang memungkinkan pemrosesan data dilakukan di dekat sumber data, seperti perangkat *IoT* (*Internet of Things*) atau sensor. Hal ini bertentangan dengan komputasi tradisional yang memerlukan pengiriman data ke pusat data atau cloud untuk diproses. *Edge computing* memungkinkan pemrosesan data yang lebih cepat, responsif, dan efisien dengan memanfaatkan sumber daya lokal.

Keuntungan utama dari *edge computing* adalah kemampuannya untuk mengurangi latensi dan mempercepat respons sistem [13]. Dengan melakukan pemrosesan data di dekat sumber data, *edge computing* memungkinkan aplikasi untuk merespons lebih cepat terhadap perubahan kondisi lingkungan. Hal ini sangat penting dalam konteks aplikasi *real-time* seperti deteksi objek pada video pengamatan jalan raya.

Selain itu, *edge computing* juga membantu mengurangi beban jaringan dan *cloud* dengan melakukan pemrosesan data secara lokal. Hal ini dapat mengurangi biaya operasional dan meningkatkan efisiensi sistem secara keseluruhan. Dalam konteks penelitian ini, *edge computing* digunakan untuk menjalankan model deteksi objek secara *real-time* pada perangkat lokal tanpa memerlukan koneksi internet.

Contoh implementasi *edge computing* yang digunakan dalam penelitian ini adalah Jetson Nano dan Beelink Gemini T34 yang berfungsi sebagai *edge device* untuk menjalankan model deteksi objek secara *real-time*.

2.2.3.1 Jetson Nano



Gambar 2.7: Jetson Nano Developer Kit

Jetson Nano adalah perangkat komputasi keluaran NVIDIA yang didedikasikan dalam pengembangan *machine learning* dan *edge computing*. Jetson Nano memiliki kemampuan untuk dapat menjalankan beberapa *neural networks* secara paralel. Kemampuan ini memungkinkan Jetson Nano untuk digunakan dalam *image classification*, *object detection*, *segmentation*, dan *speech processing* [14]. Perangkat ini dapat menjadi solusi dalam menjalankan model *machine learning* atau *deep learning* pada perangkat portable dan hemat energi.

Perangkat ini dilengkapi dengan 128 NVIDIA CUDA *cores*. Ditenagai oleh prosesor *Quad-core ARM Cortex-A57 MPCore*, platform ini menyediakan fungsionalitas komputasi yang solid dengan memori 4 GB 64-bit LPDDR4 yang beroperasi pada 1600MHz, memberikan *bandwidth* 25.6 GB/s. Untuk penyimpanan, Jetson Nano dilengkapi dengan 16 GB eMMC 5.1, yang memberikan ruang yang cukup untuk aplikasi dan data pengguna. Jetson Nano dapat mengkodekan video dengan kecepatan 250MP/sec, mendukung format seperti 1x 4K pada 30fps (HEVC), 2x 1080p pada 60fps (HEVC), dan sebagainya. Sementara itu, untuk decode video, perangkat ini menawarkan kemampuan hingga 500MP/sec, dengan dukungan untuk format seperti 1x 4K pada 60fps (HEVC) dan 2x 4K pada 30fps (HEVC). Jetson Nano juga dilengkapi dengan 12 jalur kamera (3x4 atau 4x2) MIPI CSI-2 D-PHY 1.1, yang mendukung kecepatan hingga 1.5 Gb/s per pasangan, memberikan fleksibilitas dalam pengembangan aplikasi berbasis kamera. Dalam hal konektivitas, perangkat ini menawarkan Gigabit Ethernet dan M.2 Key E, serta kemampuan tampilan melalui HDMI 2.0 dan eDP 1.4. Jetson Nano juga dilengkapi dengan 4x USB 3.0 dan USB 2.0 Micro-B, serta berbagai opsi konektivitas lainnya seperti GPIO, I2C, I2S, SPI, dan UART, semuanya dalam form factor mekanis 69.6 mm x 45 mm dengan konektor tepi 260-pin [14].

2.2.3.2 Beelink Gemini T34



Gambar 2.8: Beelink Gemini T34

Beelink Gemini T34 adalah mini PC yang dilengkapi dengan prosesor Intel Apollo Lake N3450. Prosesor ini memiliki 4 core dan 4 thread dengan kecepatan hingga 2.2 GHz. Beelink Gemini T34 juga dilengkapi dengan Intel HD Graphics 500 yang mendukung resolusi hingga 4K. Mini PC ini memiliki RAM 8 GB LPDDR3 dan penyimpanan 128 GB eMMC SSD. Beelink Gemini T34 memiliki berbagai port konektivitas seperti HDMI, USB 3.0, USB 2.0, dan Gigabit Ethernet. Mini PC ini juga dilengkapi dengan konektivitas Wi-Fi 5 dan Bluetooth 4.0. Beelink Gemini T34 memiliki dimensi 11.9 cm x 11.9 cm x 1.79 cm dan berat 0.4 kg, sehingga mudah dipindahkan dan ditempatkan di berbagai lokasi. [15]

Beelink Gemini T34 dapat digunakan sebagai *edge device* untuk menjalankan model deteksi objek secara *real-time*. Mini PC ini memiliki performa yang cukup untuk menjalankan aplikasi deteksi objek dengan akurasi tinggi dan latensi rendah. Dengan konektivitas Wi-Fi dan Bluetooth, Beelink Gemini T34 dapat terhubung ke jaringan lokal dan perangkat lain untuk mentransfer data deteksi objek. Dengan demikian, Beelink Gemini T34 merupakan solusi yang ideal untuk implementasi *edge computing* dalam deteksi objek pada video pengamatan jalan raya.

2.2.4 MQTT (*Message Queuing Telemetry Transport*)

MQTT (*Message Queuing Telemetry Transport*) adalah protokol komunikasi yang ringan dan efisien untuk pertukaran pesan antara perangkat. MQTT dirancang untuk digunakan dalam kondisi jaringan yang tidak stabil atau berkapasitas rendah, seperti pada perangkat IoT atau *edge device*.

MQTT menggunakan model *publish-subscribe* yang memungkinkan perangkat untuk berkomunikasi secara asinkron. Dalam model ini, perangkat yang mengirim pesan disebut *publisher*, sedangkan perangkat yang menerima pesan disebut *subscriber*. Pesan dikirim ke topik tertentu dan dapat diterima oleh satu atau lebih *subscriber* yang berlangganan topik tersebut.

Keunggulan utama dari MQTT adalah kemampuannya untuk mengirim pesan secara efisien dan andal di berbagai kondisi jaringan. MQTT menggunakan protokol TCP/IP yang ringan dan memiliki overhead yang rendah, sehingga cocok untuk digunakan pada perangkat dengan sumber daya terbatas. Selain itu, MQTT juga mendukung koneksi yang tahan lama dan *quality of service* (QoS) yang dapat disesuaikan.

Dalam penelitian ini, MQTT digunakan sebagai protokol komunikasi antara *edge device* dan *cloud* untuk mentransfer data deteksi objek secara *real-time*. MQTT memungkinkan perangkat untuk berkomunikasi secara efisien dan responsif tanpa memerlukan koneksi internet yang stabil. [16]

2.2.5 RESTful API

RESTful API (*Representational State Transfer Application Programming Interface*) adalah arsitektur desain yang digunakan untuk membangun layanan web yang ringan, fleksibel, dan mudah diakses. RESTful API menggunakan prinsip REST yang memisahkan antara data dan tampilan, serta memanfaatkan metode HTTP untuk berkomunikasi antara klien dan server.

RESTful API menggunakan 4 metode HTTP utama untuk berinteraksi dengan *resource*:

- **GET:** Mengambil data dari *resource*.
- **POST:** Membuat data baru di *resource*.
- **PUT:** Memperbarui data yang ada di *resource*.
- **DELETE:** Menghapus data dari *resource*.

RESTful API menggunakan URI (*Uniform Resource Identifier*) untuk mengidentifikasi sumber daya yang diakses dan format data yang diinginkan. Data dikirimkan dalam format yang umum seperti JSON atau XML untuk memudahkan pertukaran data antara klien dan server. [17]

Keunggulan utama dari RESTful API adalah kemudahan penggunaan, fleksibilitas, dan skalabilitas. RESTful API memungkinkan pengembang untuk membangun layanan web yang dapat diakses dari berbagai platform dan bahasa pemrograman. Dalam penelitian ini, RESTful API digunakan untuk mengintegrasikan antara aplikasi Flutter dengan *cloud* untuk mentransfer data deteksi objek secara *real-time*.

2.2.6 Flutter

Flutter adalah *framework* pengembangan aplikasi *open-source* yang dikembangkan oleh Google [18]. Flutter memungkinkan pengembang untuk membuat aplikasi *mobile* dengan antarmuka pengguna yang kaya dan responsif. Flutter menggunakan bahasa pemrograman Dart yang dikembangkan oleh Google sebagai bahasa utamanya.

Keunggulan utama dari Flutter adalah kemampuannya untuk membuat aplikasi *mobile* yang konsisten di berbagai platform, termasuk Android dan iOS. Flutter menggunakan *widget* sebagai komponen dasar untuk membangun antarmuka pengguna. *Widget* Flutter bersifat

deklaratif, yang berarti pengembang mendefinisikan bagaimana antarmuka pengguna harus terlihat berdasarkan kondisi saat ini.

Flutter juga menyediakan berbagai *plugin* dan *package* yang memperluas fungsionalitas aplikasi. Pengembang dapat menggunakan *plugin* Flutter untuk mengakses fitur perangkat seperti kamera, lokasi, dan sensor. Selain itu, Flutter juga menyediakan *package* untuk mengintegrasikan aplikasi dengan layanan *cloud*, *database*, dan API eksternal.

Dalam penelitian ini, Flutter digunakan sebagai platform pengembangan aplikasi *mobile* untuk menampilkan hasil deteksi objek secara *real-time*. Aplikasi Flutter akan berkomunikasi dengan *cloud* untuk menerima data deteksi objek dan menampilkannya dalam antarmuka pengguna yang responsif.

[Halaman ini sengaja dikosongkan]

BAB III

METODOLOGI

Penelitian ini dilaksanakan sesuai dengan metodologi yang tertera pada bab ini. Metodologi yang digunakan dalam penelitian ini adalah sebagai berikut:

3.1 Data dan Peralatan

Adapun tata pendukung berupa data dan peralatan yang digunakan dalam pelaksanaan penelitian tugas akhir ini adalah sebagai berikut:

3.1.1 Data

Data yang akan digunakan dalam penelitian ini adalah data citra yang diperoleh dari hasil pengambilan gambar secara manual menggunakan kamera di gerbang tol. Data citra tersebut akan digunakan sebagai data latih dan data uji dalam proses pelatihan model dan pengujian model. Data citra tersebut akan diolah dan diolah menggunakan teknik visi komputer untuk mendeteksi apakah kendaraan yang melintas adalah kendaraan yang *overdimension* atau tidak.

3.1.2 Peralatan

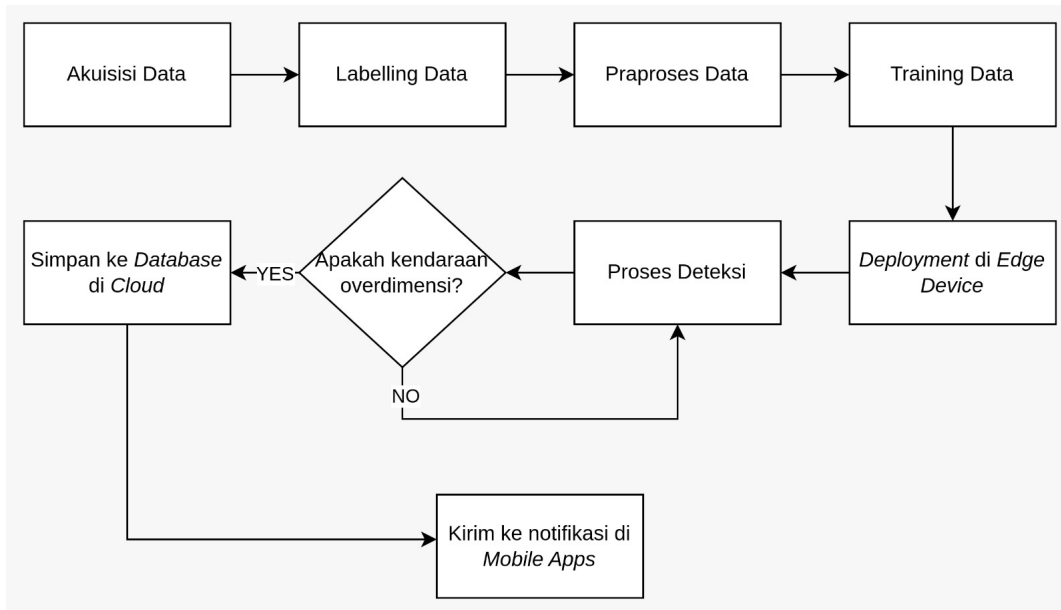
Adapun peralatan yang digunakan dalam penelitian tugas akhir ini adalah sebagai berikut:

- Visual Studio Code
- Jetson Nano Developer Kit
- Beelink Gemini T34
- Tripod
- Webcam
- *Server Cloud*
- *Smartphone Android*
- Laptop

Tugas akhir ini akan dilaksanakan sesuai dengan desain sistem beserta implementasinya yang akan dibahas pada bab ini. Desain sistem ini merupakan konsep dasar perancangan dan pembuatan program pada tugas akhir ini. Desain sistem ini direpresentasikan dalam bentuk blok diagram yang diselesaikan secara bertahap dan menyeluruh.

3.2 Metode yang Digunakan

Adapun tugas akhir ini merupakan pengembangan dari teknologi visi komputer yang kemudian diimplementasikan pada *Jetson Nano Developer Kit* dan *Beelink Gemini T34*. Secara umum, pelaksanaan tugas akhir ini didasari oleh blok diagram yang dapat dilihat pada **Gambar 3.1**.



Gambar 3.1: Blok diagram metodologi

3.2.1 Akuisisi Data

Pada tahap ini, data citra yang diperoleh dari hasil pengambilan gambar secara manual menggunakan kamera di gerbang tol. Data citra tersebut akan digunakan sebagai data latih dan data uji dalam proses pelatihan model dan pengujian model. Data citra tersebut akan diolah dan diolah menggunakan teknik visi komputer untuk mendeteksi apakah kendaraan yang melintas adalah kendaraan yang *overdimension* atau tidak.

3.2.2 Labelling Data

Pada tahap ini, data citra yang telah diperoleh akan dilakukan *labelling*. *Labelling* data dilakukan dengan cara memberikan label pada data citra yang telah diperoleh. Label yang diberikan adalah label yang menunjukkan apakah kendaraan yang melintas adalah kendaraan yang *overdimension* atau tidak.



Gambar 3.2: Ilustrasi pengelompokan kendaraan overdimensi atau tidak

3.2.3 Praproses Data

Pada tahap ini, data citra yang telah dilabeli akan dilakukan praproses. Praproses data dilakukan dengan cara melakukan augmentasi data. Augmentasi data dilakukan dengan cara melakukan rotasi, pergeseran, dan perubahan warna pada data citra. Hal ini dilakukan untuk memperbanyak variasi data citra yang akan digunakan dalam proses pelatihan model.

3.2.4 Training Data

Pada tahap ini, data citra yang telah dilabeli dan dipreproses akan dilakukan *training model*. *Training model* dilakukan dengan cara membagi data citra menjadi dua bagian, yaitu data latih dan data validasi. Data latih digunakan untuk melatih model sedangkan data validasi digunakan untuk menguji model. Model yang digunakan adalah model *SSD MobileNetV2*.

3.2.5 Deployment di Edge Device

Pada tahap ini, model yang telah dilatih akan diimplementasikan pada *Jetson Nano Developer Kit* dan *Beelink Gemini T34*. Implementasi model dilakukan dengan cara mengubah model yang telah dilatih menjadi model yang dapat dijalankan pada *Jetson Nano Developer Kit* dan *Beelink Gemini T34*. Model yang diimplementasikan adalah model *SSD MobileNetv2*.

3.2.6 Proses Deteksi

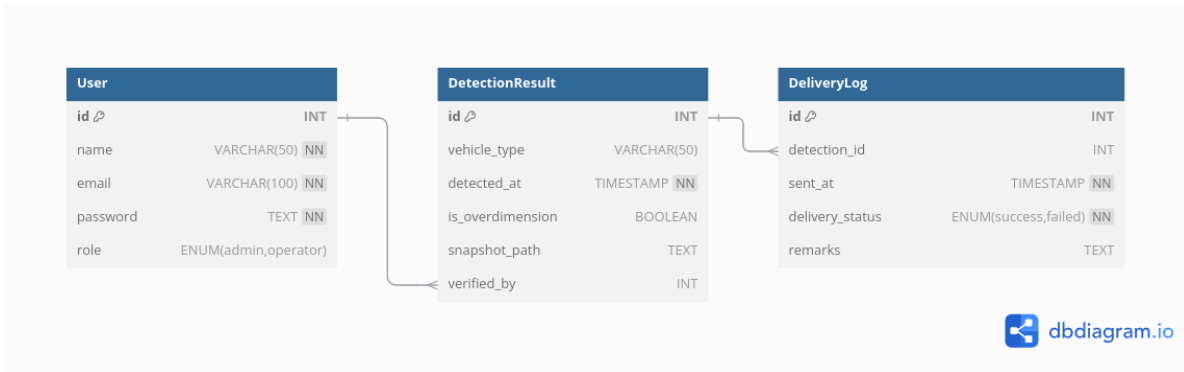
Pada tahap ini, model yang telah diimplementasikan akan digunakan untuk mendeteksi apakah kendaraan yang melintas adalah kendaraan yang *overdimension* atau tidak. Proses deteksi dilakukan dengan *live inferencing* pada *Jetson Nano Developer Kit* dan *Beelink Gemini T34*. Hasil deteksi akan ditampilkan pada layar monitor.

3.2.7 Penyimpanan Data ke Cloud

Pada tahap ini, hasil deteksi yang telah dilakukan akan disimpan ke *server cloud*. Proses penyimpanan data dilakukan dengan cara mengirimkan data hasil deteksi ke *database cloud*. Data hasil deteksi yang disimpan yaitu:

- ID
- Jenis kendaraan
- Waktu deteksi
- *Overdimension* atau tidak
- Gambar kendaraan

Diagram basis data yang digunakan dapat dilihat pada **Gambar 3.3**.



Gambar 3.3: Diagram basis data

Struktur basis data yang digunakan terdiri dari tiga tabel utama, yaitu:

1. **Tabel *User***: Tabel ini digunakan untuk menyimpan informasi pengguna sistem, seperti nama, email, kata sandi, dan peran pengguna (administrator atau operator). Struktur ini memungkinkan sistem untuk mengatur akses berdasarkan peran pengguna, sehingga keamanan dan tanggung jawab penggunaan sistem dapat terjaga.
2. **Tabel *DetectionResult***: Tabel ini merupakan inti dari sistem, yang menyimpan informasi hasil deteksi kendaraan. Data yang dicatat meliputi jenis kendaraan, waktu deteksi, status overdimensi, serta jalur penyimpanan gambar kendaraan. Selain itu, tabel ini juga memiliki kolom untuk mencatat identitas pengguna yang memverifikasi hasil deteksi.
3. **Tabel *DeliveryLog***: Tabel ini berfungsi mencatat log pengiriman data hasil deteksi ke cloud. Informasi yang disimpan meliputi waktu pengiriman, status pengiriman (berhasil atau gagal), dan keterangan tambahan, seperti pesan kesalahan jika pengiriman mengalami kegagalan. Tabel ini penting untuk melacak riwayat pengiriman dan mendukung analisis jika terjadi kendala teknis.

3.2.8 Pengiriman Notifikasi ke *Mobile Apps*

Pada tahap ini, hasil deteksi yang telah disimpan ke *server cloud* akan dikirimkan notifikasi ke aplikasi *mobile*. Proses pengiriman notifikasi dilakukan dengan cara mengirimkan notifikasi ke aplikasi *mobile* yang terhubung dengan *server cloud*. Notifikasi yang dikirimkan berupa notifikasi apakah kendaraan yang melintas adalah kendaraan yang *overdimension* atau tidak.

BAB IV

HASIL YANG DIHARAPKAN

4.1 Hasil yang Diharapkan dari Penelitian

Berdasarkan hasil pengujian yang telah dilakukan dalam tugas akhir ini, diharapkan penelitian ini menghasilkan suatu sistem deteksi kendaraan *overdimension* dengan model yang di-*deploy* di *edge device* dengan akurasi dan kecepatan yang tinggi. Sistem ini diharapkan dapat digunakan pada gerbang tol untuk mendeteksi kendaraan *overdimension* yang melintas.

4.2 Hasil Pendahuluan

Sampai saat ini, telah dilakukan studi literatur mengenai deteksi kendaraan *overdimension* dan *object detection*. Selain itu, telah dilakukan proses *re-training* dari *dataset* milik Ahmad Hamayan Nabih [5] dengan menggunakan model SSD MobileNetV2 di Jetson Nano. Hasil dari *re-training* dapat dilihat pada **Gambar 4.1**.



Gambar 4.1: Hasil *re-training dataset* dengan SSD MobileNetV2

Dari hasil di atas, dapat dilihat bahwa model yang dihasilkan sudah mampu mendeteksi kendaraan *overdimension* dengan cukup baik dan stabil di 35 - 40 FPS.

[Halaman ini sengaja dikosongkan]

BAB V

JADWAL PENELITIAN

Tabel 5.1: Tabel *timeline* penelitian

Kegiatan	Minggu															
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Pengambilan data																
Praproses & pengolahan data																
Testing & Analisa data																
Evaluasi penelitian & dokumentasi																

Pada *timeline* yang tertera di Tabel 5.1, terdapat empat kegiatan utama yang akan dilakukan selama 16 minggu. Kegiatan-kegiatan tersebut adalah sebagai berikut:

1. Pengambilan data (Minggu 1-3)

Pada tahap ini, dilakukan pengambilan data yang akan digunakan dalam penelitian ini. Data yang diambil adalah data kendaraan *overdimension* yang melintas di gerbang tol.

2. Praproses & pengolahan data (Minggu 3-8)

Pada tahap ini, dilakukan praproses dan pengolahan data yang telah diambil. Data yang telah diambil akan diproses dan diolah agar dapat digunakan dalam pelatihan model.

3. Testing & Analisa data (Minggu 7-11)

Pada tahap ini, dilakukan pengujian model yang telah dilatih. Model yang telah dilatih akan diuji dengan data yang belum pernah dilihat sebelumnya. Selain itu, dilakukan analisa terhadap hasil pengujian.

4. Evaluasi penelitian & dokumentasi (Minggu 11-16)

Pada tahap ini, dilakukan evaluasi terhadap penelitian yang telah dilakukan. Selain itu, dilakukan dokumentasi terhadap penelitian yang telah dilakukan.

[Halaman ini sengaja dikosongkan]

DAFTAR PUSTAKA

- [1] B. P. Statistik. "Jumlah kendaraan bermotor menurut provinsi dan jenis kendaraan (unit)." Accessed: 16 September 2024, Accessed: Sep. 16, 2024. [Online]. Available: <https://its.id/m/statistikbpsjumlahkendaraanbermotor2023>.
- [2] E. Oktarinda, N. Prihutomo, and E. Maulani, "Analisis pengaruh kendaraan ODOL terhadap tingkat kecelakaan di jalan tol," *Construction and Material Journal*, vol. 2, no. 1, 2020.
- [3] K. P. D. Hubdat. "Ditjen hubdat gelar fgd transformasi digital dalam pengawasan kendaraan odol." Accessed: 13 October 2024, Accessed: Oct. 13, 2024. [Online]. Available: <https://hubdat.dephub.go.id/id/publikasi/ditjen-hubdat-gelar-fgd-transformasi-digital-dalam-pengawasan-kendaraan-odol/>.
- [4] P. Y. Putra, A. S. Arifianto, Z. E. Fitri, and T. D. Puspitasari, "Deteksi kendaraan truk pada video menggunakan metode tiny-yolo v4," *Jurnal Informatika Polinema*, vol. 9, no. 2, 2023.
- [5] A. H. Nabih, "Sistem deteksi kendaraan overdimension secara real-time menggunakan deep learning di jalan raya," Undergraduate thesis, Institut Teknologi Sepuluh Nopember, 2024.
- [6] D. Indra, Herman, and F. S. Budi, "Implementasi sistem penghitung kendaraan otomatis berbasis computer vision," *Komputika: Jurnal Sistem Komputer*, vol. 12, no. 1, 2023.
- [7] F. Rosenblatt, "The perceptron: A probabilistic model for information storage and organization in the brain," *Psychological Review*, vol. 65, no. 6, p. 386, 1958.
- [8] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.
- [9] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2015.
- [10] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *arXiv preprint arXiv:1502.03167*, 2015.
- [11] C. Szegedy, W. Liu, Y. Jia, et al., "Going deeper with convolutions," *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1–9, 2015.
- [12] W. Liu et al., "Ssd: Single shot multibox detector," *arXiv preprint arXiv:1606.02228*, vol. 9905, no. 2, 2016.
- [13] A. W. Services. "Apa itu komputasi edge?" Accessed: 3 December 2024. [Online]. Available: <https://aws.amazon.com/id/what-is/edge-computing/>.
- [14] NVIDIA, *Jetson nano*, <https://www.nvidia.com/en-us/autonomous-machines/embedded-systems/jetson-nano/>, Accessed: 17 Dec 2024, 2024.
- [15] Juragantablet, *Mini pc beelink gemini t34 8/256gb ssd windows 10 wifi hdmi 4k usb 3.0*, <https://tokopedia.link/se5YnR9woPb>, Accessed: 17 Dec 2024, 2024.

- [16] U. Hunkeler, H. L. Truong, and A. Stanford-Clark, “Mqtt-s — a publish/subscribe protocol for wireless sensor networks,” in *2008 3rd International Conference on Communication Systems Software and Middleware and Workshops (COMSWARE '08)*, 2008, pp. 791–798. DOI: [10.1109/COMSWA.2008.4554519](https://doi.org/10.1109/COMSWA.2008.4554519).
- [17] C. Rodriguez et al., “Rest apis: A large-scale analysis of compliance with principles and best practices,” Jun. 2016, pp. 21–39, ISBN: 978-3-319-38790-1. DOI: [10.1007/978-3-319-38791-8_2](https://doi.org/10.1007/978-3-319-38791-8_2).
- [18] Google. “Flutter - build apps for any screen.” Accessed: 4 December 2024. [Online]. Available: <https://flutter.dev/>.