

**PERANCANGAN SISTEM DETEKSI OBJEK DAN ESTIMASI JARAK
BERBASIS *CNN* UNTUK KENDARAAN ALAT BERAT
MENGUNAKAN SIMULATOR *UNITY***

(Skripsi)

Oleh

MUHAMMAD REZA PAHLEVI

2115031121



JURUSAN TEKNIK ELEKTRO

FAKULTAS TEKNIK

UNIVERSITAS LAMPUNG

BANDAR LAMPUNG

2025

ABSTRAK

PERANCANGAN SISTEM DETEKSI OBJEK DAN ESTIMASI JARAK BERBASIS *CNN* UNTUK KENDARAAN ALAT BERAT MENGUNAKAN SIMULATOR *UNITY*

Oleh

Muhammad Reza Pahlevi

Kendaraan alat berat di lingkungan industri seperti pertambangan dan konstruksi memiliki risiko kecelakaan tinggi akibat area *blind spot* (titik buta) yang luas, terutama di bagian depan kendaraan. Untuk meningkatkan keselamatan operasional, pada penelitian ini merancang dan mengimplementasikan simulator *UNITY* untuk deteksi objek dan estimasi jarak menggunakan *Convolutional Neural Network (CNN)*. Sistem ini diimplementasikan dalam lingkungan simulator *UNITY* untuk mendeteksi dua kelas objek utama, yaitu kendaraan alat berat dan manusia. *Faster R-CNN* yang dilatih menggunakan dataset yang telah disiapkan sebagai model deteksi pada sistem dengan mengevaluasi tiga metrik utama.

Pada penelitian ini, pengujian hasil akurasi deteksi menunjukkan nilai *mean Average Precision* sebesar 0,422, meskipun masih menghadapi kendala dalam mendeteksi objek kecil. Sementara pada pengujian estimasi jarak, sistem menunjukkan tingkat presisi dengan rata-rata *Percentage Error* sebesar 1,06% untuk kendaraan alat berat dan 2,7% untuk manusia, walaupun terjadi penurunan akurasi untuk objek manusia pada jarak yang jauh. Dari segi kecepatan, waktu inferensi sistem berhasil memenuhi target dengan rentang berkisar 537—832 ms. Sehingga, hasil penelitian ini membuktikan bahwa sistem deteksi berbasis *CNN* dan estimasi jarak memiliki potensi yang optimal untuk diterapkan di lingkungan yang dinamis.

Kata Kunci: Kendaraan Alat Berat, *Blind Spot*, Deteksi Objek, Estimasi Jarak, *Convolutional Neural Network*.

ABSTRACT

DESIGN OF A CNN-BASED OBJECT DETECTION AND DISTANCE ESTIMATION SYSTEM FOR HEAVY EQUIPMENT VEHICLES USING A UNITY SIMULATOR

By

Muhammad Reza Pahlevi

Heavy equipment vehicles in industrial environments, such as mining and construction, have a high risk of accidents due to large blind spot areas, particularly at the front of the vehicle. To enhance operational safety, this research designs and implements a UNITY simulator for object detection and distance estimation using a Convolutional Neural Network (CNN). This system is implemented within a UNITY simulation environment to detect two primary object classes: heavy equipment vehicles and humans. A Faster R-CNN model, trained on a prepared dataset, was used as the detection model for the system, with its performance evaluated using three key metrics.

In this study, detection accuracy tests showed a mean Average Precision (mAP) of 0.422, although it still faced challenges in detecting small objects. Meanwhile, in distance estimation tests, the system demonstrated a high level of precision with an average Percentage Error of 1.06% for heavy equipment vehicles and 2.7% for humans, although a decrease in accuracy was observed for human objects at greater distances. In terms of speed, the system's inference time successfully met its target, ranging from 537—832 ms. Therefore, the results of this study prove that the CNN-based detection and distance estimation system has significant potential for application in dynamic environments.

Keywords: Heavy Equipment Vehicles, Blind Spot, Object Detection, Distance Estimation, Convolutional Neural Network.

**PERANCANGAN SISTEM DETEKSI OBJEK DAN ESTIMASI JARAK
BERBASIS *CNN* UNTUK KENDARAAN ALAT BERAT
MENGUNAKAN SIMULATOR *UNITY***

Oleh

MUHAMMAD REZA PAHLEVI

2115031121

Skripsi

Sebagai salah satu syarat untuk mendapat gelar

SARJANA TEKNIK

Pada

Jurusan Teknik Elektro

Fakultas Teknik

Universitas Lampung



**FAKULTAS TEKNIK
UNIVERSITAS LAMPUNG
BANDAR LAMPUNG**

2025

Judul Skripsi : **PERANCANGAN SISTEM DETEKSI OBJEK
DAN ESTIMASI JARAK BERBASIS CNN
UNTUK KENDARAAN ALAT BERAT
MENGUNAKAN SIMULATOR UNITY**

Nama Mahasiswa : **Muhammad Reza Pahlevi**

No. Pokok Mahasiswa : **2115031121**

Program Studi : **Teknik Elektro**

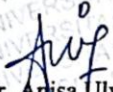
Fakultas : **Teknik**

MENYETUJUI

1. **Komisi Pembimbing**

Pembimbing Utama

Pembimbing Pendamping


Ir. Anisa Ulya Darajat, S.T., M.T
NIP. 199106102019032024


Herlinawati, S.T., M.T.
NIP. 197103141999032001

2. **Mengetahui**

**Ketua Jurusan
Teknik Elektro**

**Ketua Program Studi
Teknik Elektro**


Herlinawati, S.T., M.T.
NIP. 197103141999032001


Sumadi, S.T., M.T.
NIP. 197311042000031001

MENGESAHKAN**1. Tim Penguji****Ketua : Ir. Anisa Ulya Darajat, S.T., M.T.****Sekretaris : Herlinawati, S.T., M.T.****Penguji Bukan****Pembimbing : Dr. Eng. Ageng Sadnowo****Repelianto, S.T., M.T.****2. Dekan Fakultas Teknik****Dr. Eng. Ir. Helmy Fitriawan, S.T., M.Sc.****NIP. 197509282001121002****Tanggal Lulus Ujian Skripsi : 07 Agustus 2025**

SURAT PERNYATAAN

Dengan ini saya menyatakan bahwa dalam skripsi ini tidak terdapat karya yang pernah dilakukan orang lain dan sepanjang sepengetahuan saya tidak terdapat karya atau pendapat yang ditulis atau diterbitkan orang lain, kecuali secara teftulis diacu dalam naskah ini sebagaimana yang disebutkan di dalam daftar Pustaka. Selain itu, saya menyatakan pula bahwa skripsi ini dibuat oleh saya sendiri.

Apabila pernyataan saya tidak benar, maka saya bersedia dikenakan sanksi sesuai hukum yang berlaku.

Bandar Lampung, 18 Agustus 2025



Muhammad Reza Pahlevi
NPM. 2115031121

RIWAYAT HIDUP



Muhammad Reza Pahlevi lahir di Bandar Lampung pada tanggal 08 April 2003. Penulis lahir dari pasangan Ayah Hardi Kesuma dan Ibu Helda Puspa sebagai anak kelima dari lima bersaudara. Penulis mempunyai riwayat pendidikan di SD Negeri 2 Rawa Laut, SMP Negeri 2 Bandar Lampung, dan SMA Negeri 9 Bandar Lampung hingga lulus pada tahun 2021.

Penulis menjadi mahasiswa baru di Jurusan Teknik Elektro Fakultas Teknik Universitas Lampung pada tahun 2021 diterima melalui jalur Seleksi Bersama Masuk Perguruan Tinggi Negeri (SBMPTN).

Selama perkuliahan penulis aktif dalam kegiatan akademik dan non akademik. Penulis aktif dalam organisasi intra kampus, yaitu bergabung dalam HIMATRO UNILA sebagai anggota departemen pengembangan keteknikan, serta penulis juga aktif menjadi asisten Laboratorium Teknik Kendali dari Tahun 2023-2025. Penulis melaksanakan kerja praktik di PLN ULP Wayhalim pada tahun 2024 dengan judul laporan “Rancang Bangun Sistem Monitoring Energi Listrik Berbasis Arduino dengan Pengiriman Data Nirkabel NRF24L01”.

***“Hanya kepada Engkaulah kami
menyembah dan hanya kepada
Engkaulah kami memohon pertolongan.”***

(Q.S. Al-Fatihah: 5)

Sebuah pengingat untuk diri.

SANWACANA

Puji Syukur Alhamdulillah penulis ucapkan kepada Allah SWT yang telah memberikan nikmat, Rahmat dan hidayah-Nya sehingga penulis diberikan kelancaran dan kemudahan untuk menjalankan perkuliahan, penelitian, dan penulis naskah skripsi yang berjudul **“PERANCANGAN SISTEM DETEKSI OBJEK DAN ESTIMASI JARAK BERBASIS CNN UNTUK KENDARAAN ALAT BERAT MENGGUNAKAN SIMULATOR *UNITY*”** hingga selesai.

Selama proses menyelesaikan skripsi ini, penulis menyakini banyak mendapatkan bimbingan, masukan, bantuan, dukungan, kritik dan saran dari berbagai pihak. Untuk itu, penulis ingin mengucapkan rasa terima kasih kepada:

1. Ibu Prof. Dr. Ir Lusmeilia Afriani, D.E.A., I.P.M., selaku Rektor Universitas Lampung.
2. Bapak Dr. Eng. Ir. Helmy Fitriawan, S.T., M.Sc., selaku Dekan Fakultas Teknik Universitas Lampung.
3. Ibu Herlinawati, S.T., M.T. selaku Ketua Jurusan Teknik Elektro Fakultas Teknik Universitas Lampung dan selaku Pembimbing Pendamping tugas akhir, yang bersedia meluangkan waktu, tenaga, pikiran, serta selalu memberi bantuan, dukungan, motivasi yang tidak akan pernah saya lupakan. Terima kasih ibu, atas arahan dan nasihat yang selalu diberikan selama perkuliahan dan proses penyusunan skripsi ini.
4. Bapak Sumadi, S.T., M.T. selaku Ketua Program Studi Teknik Elektro Fakultas Teknik Universitas Lampung dan telah memberikan motivasi kepada penulis dalam menyelesaikan studi.
5. Ibu Ir. Anisa Ulya Darajat, S.T., M.T. selaku Pembimbing Utama tugas akhir dan Dosen Pembimbing Akademik, yang bersedia meluangkan waktu, tenaga, pikiran, serta selalu memberi bantuan, dukungan, motivasi yang tidak akan pernah saya lupakan. Terima kasih ibu, atas arahan dan nasihat yang selalu diberikan selama perkuliahan dan proses penyusunan skripsi ini;
6. Bapak Dr. Eng. Ageng Sadnowo Repelianto S. T., M. T. selaku Penguji tugas akhir, yang bersedia meluangkan waktu, tenaga, pikiran, serta selalu

memberi bantuan, dukungan, motivasi yang tidak akan pernah saya lupakan. Terima kasih bapak, atas arahan dan nasihat yang selalu diberikan selama perkuliahan dan proses penyusunan skripsi ini.

7. Ibu Ir. Yetti Yuniati S.T., M. T. selaku Kepala Laboratorium Pengukuran Besaran Listrik. Terima kasih atas ilmu, bimbingan dan diberikannya tempat untuk menyelesaikan tugas akhir ini.
8. Seluruh Dosen dan karyawan Jurusan Teknik Elektro Universitas Lampung, berkat ilmu yang telah diajarkan kepada penulis selama penulis menjalani masa studi di perkuliahan.
9. Pak Baiqodar, S.T. selaku PLP Laboratorium Pengukuran Besaran Listrik. Terima kasih atas ilmu dan diberikannya tempat untuk menyelesaikan tugas akhir ini.
10. Kedua orang tua yang tercinta dan tersayang, Juragan dan Ibu Juragan. Terimakasih atas segala cinta dan kasih sayangnya, yang selalu melangitkan doa-doa terbaik, memberikan dukungan, semangat, nasihat, arahan, motivasi, dan selalu bertanya kapan selesai skripsi. Mamah dan papah alasan utama penulis tidak pernah menyerah didalam berbagai kesulitan.
11. Bang Rendy, Angguman, Bang Rey, dan Ences. Terima kasih atas segala motivasi, arahan, dukungan, hiburan dan semangat.
12. Keluarga El Pance, yang telah memberikan banyak motivasi, saran, dan hiburan bermain PES.
13. Keluarga Valorant, yang telah memberikan banyak motivasi, saran, dan hiburan bermain Valo.
14. Keluarga rekan-rekan di Laboratorium Pengukuran Besaran Listrik yang selalu memberikan dukungan, pertolongan, canda tawa, dalam setiap proses apapun selama menjadi asisten laboratorium PBL.
15. Keluarga besar Angkatan 2021, yang telah memberikan banyak motivasi, nilai-nilai sosial, dan bantuan dalam berbagai hal.
16. Keluarga besar HIMATRO UNILA, yang telah menjadi wadah dalam mengembangkan nilai-nilai organisasi bagi penulis.

DAFTAR ISI

	Halaman
ABSTRAK	0
DAFTAR ISI.....	i
DAFTAR GAMBAR.....	i
DAFTAR TABEL	i
BAB I. PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah	3
1.3 Tujuan Penelitian.....	4
1.4 Manfaat Penelitian.....	4
1.5 Batasan Masalah	4
1.6 Sistematika Penulisan	5
BAB II . TINJAUAN PUSTAKA.....	7
2.1 Penelitian Terkait.....	7
2.2 Pengolahan Citra	8
2.3 Arsitektur <i>Neural Network</i>	9
2.4 <i>Convolutional Neural Network</i>	10
2.4.1 <i>Convolutional Layer</i>	11
2.4.2 <i>Pooling Layer</i>	12
2.4.3 <i>Activation Function</i>	13
2.4.4 <i>Fully Connected Layer</i>	13
2.5 Deteksi Objek dengan <i>CNN</i>	13
2.6 <i>Faster R-CNN</i>	14
2.6.1 <i>Feature Extraction</i>	15

2.6.2	<i>Region Proposal Network (RPN)</i>	15
2.6.3	<i>Region of Interest (RoI) Pooling</i>	17
2.7	Metrik Evaluasi Deteksi Objek.....	19
2.7.1	<i>Loss</i>	19
2.7.2	<i>Intersection over Union</i>	20
2.7.3	<i>Precision dan Recall</i>	20
2.7.3	<i>Average Precision dan mean Average Precision</i>	21
2.8	Konfigurasi <i>Faster RCNN</i>	21
BAB III. METODOLOGI PENELITIAN		23
3.1	Waktu dan Tempat Penelitian	23
3.2	Alat dan Bahan	23
3.3	Metode Penelitian	23
3.3.1	Konsep Sistem.....	24
3.3.2	Perancangan Sistem.....	25
3.4	Perancangan Metode <i>CNN</i>	27
3.5	Denah Peletakan Kamera.....	33
3.6	Mengukur Jarak dengan Satu Kamera.....	34
3.7	Perancangan Sketsa Simulator.....	35
3.8	Pengujian Sistem	36
3.9	Pengujian Validasi Jarak Deteksi.....	36
3.10	Pengujian Akurasi Deteksi Objek	37
3.11	Pengujian Kinerja Model <i>CNN</i>	37
BAB IV. HASIL DAN PEMBAHASAN		39
4.1	Implementasi Perancangan dan Prinsip Kerja	39
4.2	Hasil Pengujian Model	40

4.3	Kecepatan Proses Sistem Pendeteksian dan Pengukuran	
	Jarak Kendaraan Alat Berat dan Manusia	52
4.4	Pengujian Akurasi Estimasi Jarak Objek Kendaraan Alat Berat.....	53
4.4.1	Metodologi Pengujian Akurasi Jarak	54
4.4.2	Hasil Pengujian Akurasi Jarak pada Zona <i>Blind Spot</i> (1-6 Meter)	56
4.4.3	Hasil Pengujian Akurasi Jarak di Luar Zona <i>Blind Spot</i> (>6 Meter) .	59
V.	KESIMPULAN DAN SARAN	64
5.1	Kesimpulan.....	64
5.2	Saran	64
	DAFTAR PUSTAKA	65

DAFTAR GAMBAR

Gambar 2. 1 Arsitektur <i>CNN</i> [11]	10
Gambar 2. 2 Arsitektur <i>Faster RCNN</i>	14
Gambar 3.1 Konsep Perancangan Sistem Deteksi Kendaraan Alat Berat dan	
Manusia.....	24
Gambar 3. 2 Blok Diagram Sistem	25
Gambar 3. 3 Blok Diagram <i>Training Model</i>	26
Gambar 3. 4 Diagram Blok Sistem Deteksi Objek	27
Gambar 3. 5 Diagram Alir <i>AI Project Cycle</i> Metode <i>CNN</i>	28
Gambar 3. 6 Diagram Alir Data <i>Exploration</i>	30
Gambar 3. 7 Diagram Model	32
Gambar 3. 8 Desain Peletakan Kamera.....	33
Gambar 3. 9 Prinsip Kamera.....	34
Gambar 3.10 Gambar (a)Tampak Kendaraan Alat Berat (b)Tampak Manusia	
(c)PemodelanLingkunganUji	35
Gambar 4. 1 Implementasi Sistem	39
Gambar 4. 2 Grafik Kurva <i>Loss</i>	41
Gambar 4. 3 Grafik Kurva mAP50-90 per <i>Epoch</i>	42
Gambar 4. 4 Hasil Evaluasi <i>Faster RCNN</i>	43
Gambar 4. 5 Pengujian Model Terlatih Dengan 16 Gambar	44
Gambar 4. 6 Uji Coba Deteksi Variasi Objek.....	52
Gambar 4. 7 Grafik Analisis Waktu Inferensi	53
Gambar 4.8 Proses Penetapan <i>Ground Truth</i> Jarak di Lingkungan Simulator	
<i>UNITY</i>	54
Gambar 4. 9 Pengukuran Jarak di Bawah 6 Meter Hasil Deteksi Pada	
Kendaraan Alat Berat.....	56
Gambar 4. 10 Pengukuran Jarak di Atas 6 Meter Hasil Deteksi Pada	
Kendaraan Alat Berat	57
Gambar 4. 11 Pengukuran Jarak Hasil Deteksi Pada Kendaraat Alat Berat.....	59
Gambar 4. 12 Pengukuran Jarak Hasil Deteksi Pada Manusia	60
Gambar 4. 13 Grafik Regresi Linear Jarak Prediksi vs <i>Ground Truth</i>	62

DAFTAR TABEL

Tabel 2. 1 Penelitian Terdahulu	7
Tabel 2. 2 Parameter <i>Convolutional Layer</i>	11
Tabel 3. 1 Alat dan Bahan.....	23
Tabel 3. 2 Parameter Dimensi Objek	24
Tabel 3. 3 <i>Problem Scoping</i>	29
Tabel 3. 4 Jumlah Data Pelatihan yang Digunakan	30
Tabel 3. 5 Perincian Perbandingan Data Gambar untuk Dataset	31
Tabel 3. 6 Perincian <i>Preprocessing Data</i> dan <i>Augmentasi Data</i>	31
Tabel 3. 7 Pengujian Sistem.....	36
Tabel 4. 1 Hasil Akhir Performa Model Saat <i>Training</i>	43
Tabel 4. 2 Hasil Pengujian Akurasi Estimasi Jarak Objek Kendaraan Alat	
Berat dan Manusia Dalam Zona <i>Blind Spot</i>	58
Tabel 4. 3 Hasil Pengujian Akurasi Estimasi Jarak Objek Kendaraan Alat	
Berat dan Manusia	61

I. PENDAHULUAN

1.1 Latar Belakang

Perkembangan teknologi berbasis kecerdasan buatan (*Artificial Intelligence*) terus mengalami kemajuan yang pesat dalam beberapa dekade terakhir. Teknologi ini telah mengubah banyak aspek kehidupan manusia dan diterapkan di berbagai bidang, seperti transportasi, keamanan, pengawasan, kesehatan, hingga hiburan. Salah satu penerapan kecerdasan buatan yang paling signifikan adalah pada sistem pendeteksian objek menggunakan jaringan saraf tiruan (*Neural Network*), yang memungkinkan perangkat keras dan perangkat lunak untuk memproses, menganalisis, dan memahami data visual secara lebih mendalam. Salah satu jenis jaringan saraf tiruan yang sering digunakan adalah *Convolutional Neural Network (CNN)*, yang dirancang secara khusus untuk menangani data berbentuk gambar atau video. *CNN* mampu mendeteksi dan mengklasifikasikan berbagai objek, termasuk kendaraan berat seperti truk dan alat berat lainnya, serta manusia yang berada di sekitar area operasional kendaraan tersebut. Teknologi ini penting dalam sistem transportasi dan industri, terutama untuk meningkatkan keselamatan kerja di lingkungan dengan kendaraan besar yang memiliki titik buta yang luas. Hal ini membuat *CNN* menjadi komponen penting dalam pengembangan solusi berbasis *computer vision* yang dapat mendukung kebutuhan di dunia nyata.

Interaksi antara kendaraan alat berat dan manusia di lingkungan industri seperti pertambangan dan konstruksi menciptakan risiko keselamatan yang tinggi, terutama akibat adanya *blind spot* (titik buta) yang luas di bagian depan kendaraan seperti pada Gambar 1.1. Posisi operator yang tinggi membuat area di depan kendaraan menjadi zona tak terlihat yang sangat berbahaya, di mana pekerja bisa berada tanpa disadari sama sekali oleh operator. Oleh karena itu, sangat dibutuhkan sebuah sistem deteksi cerdas yang dapat secara otomatis dan andal

mengidentifikasi manusia di area *blind spot* depan, guna memberikan penglihatan tambahan kepada operator untuk mencegah kecelakaan fatal dan meningkatkan keselamatan kerja.



Gambar 1. 1 *Blind Spot* Bagian Depan Kendaraan Alat Berat

Untuk mengatasi tantangan yang ada dalam pendeteksian objek, teknologi berbasis *CNN* menawarkan solusi yang jauh lebih adaptif dan efisien. Sebagai salah satu jenis jaringan saraf tiruan dalam deep learning, dirancang dengan menggunakan prinsip *Multi Layer Perceptron* yang memungkinkan pengolahan data secara hierarkis [1]. Teknologi ini bekerja dengan memanfaatkan *dataset* gambar yang beragam untuk melatih jaringan agar mampu mengenali berbagai karakteristik objek, seperti bentuk, ukuran, dan warna. Dalam konteks pendeteksian kendaraan dan manusia, *CNN* mampu mendeteksi objek dari berbagai sudut pandang dan kondisi pencahayaan. Fitur ini membuatnya sangat cocok untuk diterapkan pada sistem pendeteksian di lingkungan nyata, yang sering kali tidak terstruktur dan penuh tantangan. Selain itu, *CNN* juga dapat diintegrasikan dengan algoritma lain, seperti perhitungan jarak berbasis data visual, untuk meningkatkan fungsionalitasnya lebih lanjut. Dengan kemampuan “melihat” dan “membedakan” objek melalui teknologi *image recognition*, *CNN* memungkinkan komputer untuk mendeteksi mobil dan manusia dengan tingkat akurasi yang tinggi [2].

Selain kemampuan mendeteksi objek, pengukuran jarak antar objek juga menjadi aspek penting dalam sistem pendeteksian modern. Perhitungan jarak berbasis data visual memanfaatkan hubungan geometris antar objek dalam gambar untuk menentukan estimasi jarak secara akurat. Pendekatan ini memungkinkan sistem untuk mengolah informasi visual secara langsung dari gambar atau video tanpa memerlukan sensor tambahan. Dengan teknologi ini, sistem pendeteksian

dapat diimplementasikan secara lebih kompak dan mudah diintegrasikan ke dalam berbagai aplikasi, seperti sistem keselamatan kerja, pemantauan lalu lintas, serta kendaraan otonom. Dalam penelitian ini, perhitungan jarak berbasis data visual diharapkan dapat meningkatkan efisiensi dan akurasi dalam mendeteksi serta menganalisis posisi objek, sehingga mendukung pengembangan sistem yang lebih adaptif dan cerdas di berbagai lingkungan operasional.

Penelitian ini bertujuan untuk mengembangkan sistem pendeteksian kendaraan alat berat dan manusia berbasis *CNN* dengan kemampuan estimasi jarak antar objek. Dengan memanfaatkan *CNN* dan algoritma visual, sistem ini dirancang untuk memberikan deteksi, yang dapat diterapkan dalam berbagai bidang seperti keselamatan kerja, pengawasan lalu lintas, dan kendaraan otonom. Untuk menguji dan memvalidasi kinerjanya, sistem ini akan diimplementasikan dalam simulator menggunakan *UNITY*, yang memungkinkan pengujian dalam berbagai kondisi lingkungan, seperti pencahayaan, sudut pandang kamera, serta interaksi antara kendaraan alat berat dan manusia. Simulator ini membantu mengidentifikasi kendala serta mengoptimalkan sistem sebelum diterapkan di dunia nyata, sehingga meningkatkan efisiensi pengembangan dan memastikan performa deteksi yang optimal di lingkungan industri yang kompleks.

1.2 Rumusan Masalah

Berdasarkan identifikasi masalah yang telah dijelaskan, rumusan masalah dalam penelitian ini adalah kurangnya sistem pendeteksian objek yang akurat pada kendaraan alat berat untuk mengenali keberadaan manusia di sekitarnya serta mendeteksi kendaraan berat lainnya dalam kondisi lingkungan yang kompleks. Tantangan utama dalam deteksi objek meliputi variasi ukuran, bentuk, serta kondisi operasional yang beragam, seperti pencahayaan yang berubah-ubah dan latar belakang yang dinamis. Selain itu, pengukuran jarak antar kendaraan dan manusia yang akurat menggunakan data visual juga menjadi aspek krusial dalam meningkatkan keselamatan dan efisiensi operasional, terutama di area kerja. Oleh karena itu, penelitian ini berfokus pada perancangan sistem pendeteksian berbasis *CNN* yang mampu mendeteksi manusia dan kendaraan alat berat seperti truk secara akurat serta menghitung jarak antar objek secara efisien tanpa memerlukan

perangkat keras tambahan, sehingga dapat meningkatkan keselamatan dan efektivitas operasional di berbagai lingkungan industri.

1.3 Tujuan Penelitian

Adapun tujuan dari penelitian ini adalah untuk merancang sebuah sistem deteksi cerdas berbasis *Convolutional Neural Network (CNN)*. Sistem ini ditujukan untuk memiliki kemampuan mendeteksi dua kelas objek utama, yaitu kendaraan alat berat dan manusia, serta mengintegrasikan fungsi untuk mengestimasi jarak ke objek yang terdeteksi menggunakan kamera.

1.4 Manfaat Penelitian

Dengan kemampuan ekstraksi fitur yang kuat, *CNN* mampu mengenali kendaraan alat berat dan manusia secara lebih akurat dibandingkan metode konvensional. Implementasi sistem ini bertujuan untuk meningkatkan keselamatan di area industri dan konstruksi dengan mendeteksi keberadaan pekerja di sekitar kendaraan alat berat serta memperkirakan jarak antar objek secara *real-time*. Dengan demikian, sistem ini dapat membantu mengurangi risiko kecelakaan akibat titik buta yang luas serta mendukung otomatisasi dan pengawasan berbasis teknologi di lingkungan kerja yang berisiko tinggi.

1.5 Batasan Masalah

Adapun batasan masalah yang ditetapkan dalam penelitian ini adalah sebagai berikut:

1. Sistem hanya dirancang untuk mendeteksi kendaraan alat berat dan manusia sebagai objek utama dalam lingkungan kerja industri atau konstruksi. Objek lain yang muncul dalam simulator tidak menjadi fokus deteksi.
2. Simulator dilakukan menggunakan *UNITY* dengan kondisi lingkungan yang telah ditentukan, termasuk pencahayaan standar dan skenario operasional kendaraan alat berat. Penelitian ini tidak mencakup pengujian dalam kondisi ekstrem seperti pencahayaan rendah atau gangguan visual yang kompleks.
3. Implementasi pada kendaraan alat berat nyata tidak termasuk dalam cakupan penelitian ini.

1.6 Sistematika Penulisan

Sistematika penulisan yang digunakan pada penelitian tugas akhir ini adalah sebagai berikut.

BAB I PENDAHULUAN

Bab ini memaparkan latar belakang yang menjelaskan pentingnya sistem pendeteksian mobil dan orang dalam mendukung keselamatan, efisiensi, dan pengawasan di berbagai bidang, serta peran teknologi *Convolutional Neural Network (CNN)* dalam mengatasi tantangan tersebut. Selanjutnya, rumusan masalah akan mengidentifikasi permasalahan spesifik yang hendak diselesaikan melalui penelitian ini. Tujuan penelitian akan dijelaskan untuk merinci hasil yang diharapkan, diikuti oleh batasan masalah yang menetapkan ruang lingkup penelitian agar lebih terarah. Manfaat penelitian juga diuraikan untuk menjelaskan dampak dari penelitian ini, baik dari segi teknis maupun praktis.

BAB II TINJAUAN PUSTAKA

Bab ini menguraikan dasar-dasar teori yang menjadi landasan penelitian. Pembahasan mencakup konsep dasar pendeteksian objek, pengenalan *CNN*, dan penerapannya dalam berbagai studi terdahulu. Selanjutnya, bab ini juga menjelaskan berbagai metode perhitungan jarak berbasis visual dan algoritma yang relevan untuk mendukung penelitian. Literatur terkait pengembangan sistem pendeteksian berbasis *CNN* juga akan dipaparkan untuk memperkuat kajian teoritis.

BAB III METODOLOGI PENELITIAN

Bab ini menjelaskan langkah-langkah yang dilakukan dalam penelitian, mulai dari desain penelitian, pengumpulan *dataset* gambar mobil dan manusia, hingga proses pelabelan data menggunakan *bounding box*. Bab ini juga mencakup tahapan pengolahan data, seperti *preprocessing* gambar, pelatihan model *CNN*, dan algoritma perhitungan jarak. Selain itu, perangkat keras dan perangkat lunak yang digunakan dalam penelitian, seperti *Python*, *TensorFlow*, dan *library* pendukung lainnya, akan dijelaskan secara rinci.

BAB IV HASIL DAN PEMBAHASAN

Bab ini menyajikan hasil penelitian yang meliputi performa model *CNN* dalam mendeteksi mobil dan manusia, serta tingkat akurasi sistem dalam berbagai

kondisi lingkungan. Analisis hasil dilakukan dengan membandingkan hasil pendeteksian dengan *ground truth* dari dataset yang digunakan. Selain itu, implementasi perhitungan jarak antar objek akan dijelaskan, termasuk efektivitasnya dalam berbagai skenario. Hasil penelitian akan dikaitkan dengan rumusan masalah, tujuan, dan batasan masalah yang telah ditentukan sebelumnya untuk memberikan pemahaman yang komprehensif.

BAB V KESIMPULAN DAN SARAN

Bab terakhir ini berisi kesimpulan yang merangkum hasil utama penelitian dan menjawab rumusan masalah berdasarkan temuan penelitian. Selain itu, saran diberikan untuk pengembangan lebih lanjut, baik dalam pengembangan sistem pendeteksian berbasis *CNN* maupun perhitungan jarak yang lebih kompleks. Saran juga mencakup rekomendasi untuk implementasi di dunia nyata dan penelitian lanjutan untuk menyempurnakan sistem yang dikembangkan.

DAFTAR PUSTAKA

LAMPIRAN

II . TINJAUAN PUSTAKA

2.1 Penelitian Terkait

Selama beberapa tahun terakhir, sejumlah penelitian telah membahas pengolahan citra dan pelacakan objek untuk pendeteksian jarak antar objek, terutama dalam periode tahun 2015 hingga 2024. Beberapa artikel yang tercantum dalam Tabel 2.1 digunakan sebagai referensi dalam penelitian ini.

Tabel 2. 1 Penelitian Terdahulu

No	Judul	Nama Penulis	Kelebihan/kekurangan
1.	<i>Real-Time Vehicle Detection Based on Improved YOLO v5</i> [3].	Yu Zhang, Zhongyin Guo, dkk. 2022	• Menggunakan berbagai skenario jalan raya dari sudut pandang yang berbeda, meningkatkan kemampuan generalisasi model. • Kondisi pencahayaan yang buruk dan noise dari kamera dapat menurunkan akurasi deteksi.
2.	<i>Object Detection in Autonomous Driving Scenarios Based on an Improved Faster-RCNN</i> [4].	Yan Zhou ¹ , Sijie Wen, dkk. 2022	• Meningkatkan akurasi dengan mengurangi efek negatif dari bounding box yang berdekatan. • Penambahan <i>deformable convolution</i> dan <i>feature pyramid structure</i> meningkatkan kebutuhan komputasi, sehingga inferensi menjadi lebih lambat dibanding <i>Faster-RCNN</i> standar.
3.	<i>Improved Vision-Based Vehicle Detection and Classification by Optimized YOLOv4</i> [5].	JINGYI ZHAO, SHENGNAN HAO ¹ , dkk. 2022	• Penambahan CBAM dan modifikasi FPN meningkatkan beban komputasi model • Model YOLOv4_AF menunjukkan peningkatan signifikan dalam deteksi kendaraan
4.	Sistem Deteksi dan Tracking Keretakan Bangunan dengan <i>Unmanned Aerial Vehicle</i> Menggunakan Algoritma <i>CNN</i> [6].	Muhammad Aqil Rayhan Majid, Mohamad Abdul Hady, Mochammad Sahal, Yurid Eka Nugraha. 2021	• Deteksi keretakan akurat secara <i>real-time</i>. • Integrasi <i>vision feedback</i> dengan kontrol PID. • Terbatas pada lingkungan dalam ruangan.

No	Judul	Nama Penulis	Kelebihan/kekurangan
5.	<i>Mask Detection and Social Distance Identification Using Internet of Things and Faster R-CNN Algorithm</i> [7]	S. Meivel, Nidhi Sindhwani, Rohit Anand, dkk. 2022	• Menggabungkan <i>Faster R-CNN</i> dan YOLOv3 untuk akurasi tinggi. • Kompleksitas tinggi pada <i>Faster R-CNN</i>. • Ketergantungan pada dataset terbatas.

2.2 Pengolahan Citra

Pengolahan citra digital adalah proses transformasi data visual berupa gambar atau video menjadi informasi yang dapat dimanfaatkan untuk berbagai aplikasi. Secara umum, tahapan pengolahan citra mencakup akuisisi, peningkatan kualitas, segmentasi, representasi, dan interpretasi. Proses ini memanfaatkan prinsip matematika dan komputerisasi untuk menghasilkan data visual yang lebih informatif atau sesuai kebutuhan, seperti deteksi objek, pengenalan pola, atau analisis citra. Tahapan ini melibatkan manipulasi piksel pada citra menggunakan operasi intensitas, histogram, transformasi spasial, serta teknik penapisan untuk meningkatkan kejelasan informasi atau mempersiapkan data untuk analisis lanjutan. Nilai piksel biasanya mewakili skala abu-abu, warna, tinggi, kekeruhan, dan sebagainya. Format gambar yang umum digunakan adalah 1 *sample per point* (*B&W or Grayscale*), 3 *samples per point* (Red, Green, and Blue), 4 *samples per point* (Red, Green, Blue, and “Alpha”).

Dalam penerapan modern, pengolahan citra digital memegang peranan penting dalam sistem berbasis visi komputer, seperti pendeteksian objek menggunakan deep learning. Misalnya, algoritma seperti *Faster R-CNN* mampu mendeteksi objek dengan presisi tinggi melalui kombinasi ekstraksi fitur dengan *Convolutional Neural Networks (CNN)* dan klasifikasi *bounding box*. Keunggulan metode ini terletak pada efisiensi dan kemampuannya untuk digunakan dalam aplikasi waktu nyata. Teknologi ini banyak digunakan dalam aplikasi seperti sistem keamanan, analisis medis, navigasi kendaraan otonom, dan pengukuran jarak berbasis visual, yang memperlihatkan integrasi erat antara pengolahan citra dengan pengembangan teknologi cerdas [8].

2.3 Arsitektur *Neural Network*

Arsitektur *neural network* adalah kerangka kerja matematis yang dirancang untuk meniru cara kerja otak manusia dalam memproses informasi. Arsitektur ini terdiri dari lapisan-lapisan yang saling terhubung, yaitu lapisan input, lapisan tersembunyi (*hidden layer*), dan lapisan *output*. Setiap lapisan tersusun atas neuron yang memproses dan mengirimkan informasi melalui bobot (*weights*) yang dapat disesuaikan selama proses pelatihan. *Neural network* modern, seperti *CNN*, menggunakan lapisan konvolusi untuk mengekstrak fitur dari data input, seperti pola visual dari gambar, diikuti oleh lapisan *pooling* untuk mengurangi dimensi data. Lapisan *fully connected* mengintegrasikan semua informasi untuk menghasilkan prediksi akhir. Penggunaan fungsi aktivasi, seperti ReLU, dan teknik regularisasi, seperti *dropout*, membantu meningkatkan efisiensi dan mencegah *overfitting*. Arsitektur ini sangat kuat untuk aplikasi seperti pengenalan gambar, pengolahan bahasa alami, dan deteksi objek, yang memanfaatkan hierarki fitur yang dihasilkan dari lapisan-lapisan dalam model tersebut [9].

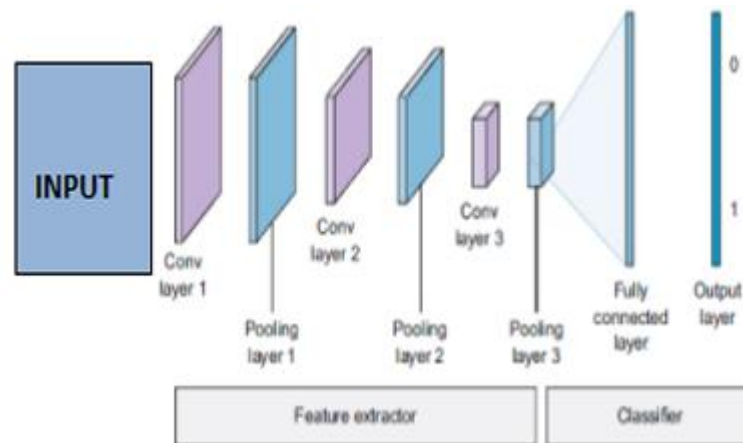
Arsitektur *neural network* telah mengalami perkembangan signifikan, dengan berbagai desain yang disesuaikan untuk menangani tugas-tugas kompleks secara efisien. Salah satu pendekatan utama adalah penggunaan *CNN*, yang dirancang khusus untuk menangkap pola dan fitur pada data visual seperti gambar. *CNN* terdiri dari lapisan konvolusi yang bertugas mengekstraksi fitur lokal, lapisan *pooling* untuk mengurangi dimensi data, dan lapisan *fully connected* untuk melakukan klasifikasi. Arsitektur seperti *AlexNet*, *VGGNet*, dan *ResNet* menunjukkan evolusi dalam kedalaman jaringan, dari 8 hingga lebih dari 150 lapisan, untuk meningkatkan akurasi dalam tantangan seperti *ImageNet*. Namun, peningkatan kedalaman ini sering kali disertai dengan tantangan seperti waktu eksekusi dan kebutuhan daya yang tinggi. Oleh karena itu, berbagai optimasi, termasuk metode kompresi model seperti *Conditional Deep Learning Networks* (CDLN), telah diterapkan untuk mengurangi parameter dan mempercepat inferensi tanpa kehilangan akurasi yang signifikan, sehingga memungkinkan penerapan pada perangkat dengan sumber daya terbatas, seperti *smartphone* [10].

Setelah sebuah arsitektur *neural network* ditentukan, tahap selanjutnya adalah melatih model tersebut menggunakan data. Namun, dalam proses pelatihan

ini, terdapat dua tantangan utama yang harus dihindari untuk menghasilkan model yang andal, yaitu *overfitting* dan *underfitting*. Fenomena *overfitting* terjadi ketika model menjadi terlalu "pintar" dalam mengenali data latihnya, hingga ia menghafal detail dan *noise* yang tidak relevan, mirip seperti seorang siswa yang hanya menghafal soal tanpa memahami konsep. Akibatnya, model menunjukkan performa yang sangat tinggi pada data latih, namun gagal melakukan generalisasi dan berkinerja buruk saat dihadapkan pada data baru yang belum pernah dilihat sebelumnya. Sebaliknya, *underfitting* adalah kondisi di mana model terlalu sederhana atau belum cukup dilatih sehingga gagal menangkap pola yang mendasari data, yang menyebabkannya berkinerja buruk baik pada data latih maupun data baru. Oleh karena itu, tujuan utama dari proses pelatihan adalah mencapai model yang seimbang (*good fit*), yang mampu mengenali pola secara general dan memiliki performa yang baik pada data baru, bukan sekadar menghafal data latih.

2.4 Convolutional Neural Network

CNN merupakan salah satu metode *deep learning* yang dirancang untuk mengolah data dua dimensi, terutama citra.



Gambar 2. 1 Arsitektur *CNN* [11]

Gambar 2.1 adalah jaringan arsitektur *CNN* yang bekerja dengan meniru sistem pengenalan visual manusia melalui arsitektur yang terdiri dari beberapa lapisan utama, yaitu *Convolutional Layer*, *Pooling Layer*, dan *Fully Connected Layer*. Metode ini memiliki kemampuan dalam mengekstraksi fitur dari citra dan mengelompokkan data berdasarkan fitur tersebut. Dalam implementasinya, *CNN*

menggunakan operasi konvolusi untuk memproses data, yang memungkinkan model mengenali pola spasial pada gambar [11].

2.4.1 *Convolutional Layer*

Convolutional layer adalah elemen fundamental dalam *CNN* yang berfungsi untuk mengekstraksi fitur dari data masukan, seperti citra. Proses ini dilakukan melalui filter konvolusi atau kernel yang bergerak melintasi tensor masukan secara sistematis, menghitung hasil perkalian elemen-elemen di setiap wilayah masukan dengan kernel tersebut. *Output* dari lapisan ini berupa tensor baru yang mencerminkan pola-pola tertentu dari data awal. Lapisan konvolusi memainkan peran penting dalam mengenali fitur spasial seperti tepi, tekstur, atau objek tertentu dalam citra, yang kemudian diteruskan ke lapisan berikutnya untuk proses lebih lanjut. Lapisan ini memungkinkan *CNN* untuk mempelajari representasi fitur yang kompleks secara hierarkis, mulai dari pola sederhana hingga struktur yang lebih kompleks, sesuai dengan kedalaman jaringan [12]. Adapun parameter dalam menentukan jumlah dan ukuran pada ekstraksi pada layer konvolusi yaitu pada Tabel 2.2.

Tabel 2. 2 Parameter *Convolutional Layer*

Parameter	Keterangan
<i>Depth</i>	Kedalaman layer atau jumlah layer konvolusi
<i>Stride</i>	Jumlah pergeseran filter pada layer konvolusi
<i>Zero Padding</i>	Jenis penambahan nol di daerah sekitar input gambar

Convolutional layer adalah komponen utama dalam *CNN* yang bertugas mengekstraksi fitur penting dari data *input*. Proses ini menggunakan kernel atau filter, yaitu matriks kecil (misalnya 3×3 atau 5×5) yang bergerak melintasi *input* data untuk mendeteksi pola, tepi, atau tekstur. Selama proses ini, kernel melakukan operasi konvolusi pada setiap bagian input dengan menghitung perkalian elemen matriks kernel dan piksel *input*, lalu menjumlahkan hasilnya. Rumus operasi konvolusi dapat dinyatakan sebagai Persamaan 2.1 [13].

$$F(i, j) = \sum_{m=1}^K \sum_{n=1}^K (I(i + m - 1, j + n - 1) \cdot W(m, n)) + b \quad (2.1)$$

Dimana $F(i, j)$ adalah nilai pada *feature map* di posisi (i, j) , I adalah *input*, W adalah *kernel*, K adalah ukuran *kernel*, dan b adalah *bias*. *Stride* menentukan langkah pergerakan kernel pada *input*, sehingga ukuran *output (feature map)* dapat dihitung dengan Persamaan 2.2 [14].

$$O = \frac{(I-K+2P)}{S} \quad (2.2)$$

Dimana O adalah ukuran *output*, I adalah ukuran *input*, K adalah ukuran *kernel*, P adalah *padding*, dan S adalah *stride*. *Padding* digunakan untuk menambahkan piksel di tepi *input*, terutama dalam *same padding*, agar ukuran output tetap sama dengan input.

Hasil dari operasi konvolusi adalah *feature map*, yang merepresentasikan fitur penting dari input data. *Feature map* ini memainkan peran penting dalam menganalisis data visual, karena memungkinkan jaringan untuk mendeteksi pola-pola relevan, seperti tepi, tekstur, dan objek, di berbagai lokasi input dengan cara yang lebih efisien dan terstruktur.

2.4.2 Pooling Layer

Pooling layer adalah salah satu lapisan dalam *CNN* yang berfungsi untuk mengurangi dimensi *feature map* sambil mempertahankan informasi penting. *Pooling layer* bertugas mengurangi jumlah parameter dan komputasi, serta membantu mencegah *overfitting*. *Pooling* dilakukan dengan cara membagi *feature map* menjadi beberapa wilayah kecil dan menerapkan fungsi tertentu, seperti *max pooling* atau *average pooling*, pada setiap wilayah:

- *Max pooling* memilih nilai maksimum dalam setiap wilayah, menyoroti fitur paling dominan.
- *Average pooling* menghitung rata-rata nilai dalam setiap wilayah, memberikan representasi yang lebih halus.

Hasil dari *pooling layer* adalah *pooled feature map* yang lebih kecil dari *feature map* asli.

2.4.3 Activation Function

Activation function adalah sebuah *node* yang dimasukan pada akhir keluaran dari setiap jaringan syaraf. Pada arsitektur *CNN*, fungsi aktivasi dapat digunakan pada perhitungan akhir keluaran *feature map* atau sehabis proses konvolusi maupun *pooling layer*. Terdapat banyak fungsi aktivasi yang banyak digunakan dalam *neural network* yaitu *sigmoid*, *tanh*, *ReLU* (*Rectified Linear*), parameter *ReLU*, dan *leaky ReLU*.

ReLU layer atau lapisan *Rectified Linear Units* ini menerapkan fungsi aktivasi $f(x)=\max(0,x)$. Ini meningkat sifat *non-linear* fungsi keputusan dan jaringan secara keseluruhan tanpa mempengaruhi bidang-bidang reseptif pada *convolution layer*.

2.4.4 Fully Connected Layer

Fully Connected Layer adalah lapisan dalam *CNN* yang berfungsi untuk menghubungkan setiap neuron pada lapisan sebelumnya dengan semua neuron di dalamnya. Lapisan ini biasanya terletak di bagian akhir jaringan dan berperan dalam menggabungkan fitur-fitur yang telah diekstraksi oleh lapisan sebelumnya, seperti *convolutional layer* dan *pooling layer*, menjadi representasi akhir yang dapat digunakan untuk membuat keputusan. Setelah fitur diekstraksi melalui berbagai lapisan konvolusi dan *pooling*, *FC Layer* mengubahnya menjadi vektor satu dimensi yang dapat diproses oleh algoritma klasifikasi. Setiap neuron dalam lapisan ini memiliki bobot dan bias yang akan diperbarui selama proses pelatihan menggunakan metode *backpropagation* dan optimasi seperti *Stochastic Gradient Descent* atau *Adam*.

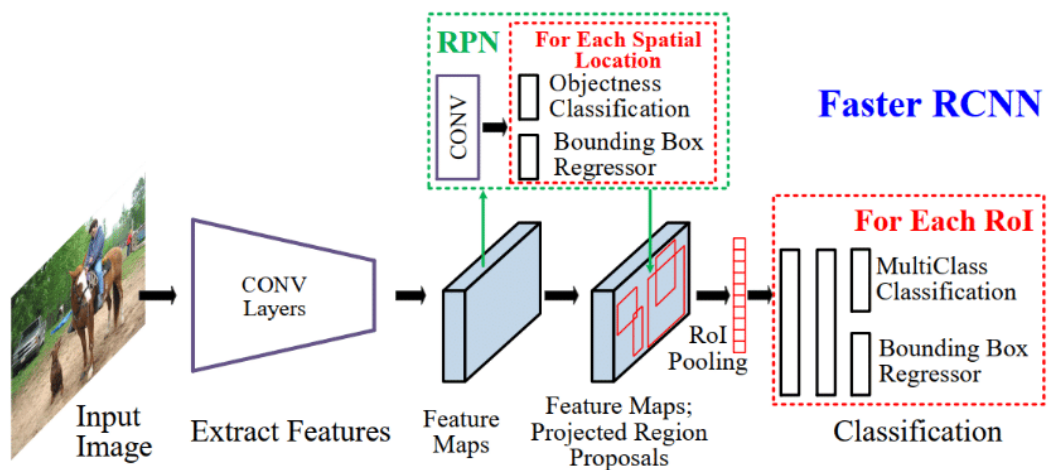
2.5 Deteksi Objek dengan CNN

Klasifikasi gambar dan deteksi objek adalah dua tugas utama dalam visi komputer yang menggunakan *CNN*. Klasifikasi gambar bertujuan untuk mengidentifikasi kategori utama yang terdapat dalam suatu gambar tanpa memperhatikan lokasi spesifik objek tersebut. Sebaliknya, deteksi objek tidak hanya mengidentifikasi jenis objek yang ada dalam gambar, tetapi juga menentukan

posisi setiap objek dengan memberikan kotak pembatas (bounding box) di sekitarnya. Misalnya, dalam klasifikasi gambar, model dapat menentukan bahwa sebuah gambar mengandung seekor anjing, sementara dalam deteksi objek, model akan mengidentifikasi keberadaan anjing tersebut dan menunjukkan lokasinya dalam gambar.

2.6 *Faster R-CNN*

Faster R-CNN adalah model deteksi objek yang menggabungkan jaringan proposal wilayah (*Region Proposal Network* atau RPN) dengan jaringan deteksi, memungkinkan pembuatan proposal wilayah hampir tanpa biaya tambahan. RPN adalah jaringan konvolusional penuh yang secara simultan memprediksi batas objek dan skor keberadaan objek di setiap posisi. Dengan berbagi fitur konvolusional gambar penuh antara RPN dan jaringan deteksi, *Faster R-CNN* mencapai deteksi objek yang efisien dan akurat



Gambar 2. 2 Arsitektur *Faster RCNN* [15]

Gambar 2.2 menunjukkan arsitektur *Faster R-CNN*, sebuah model deteksi objek yang dikembangkan untuk meningkatkan efisiensi dan akurasi dibandingkan metode pendahulunya. *Faster R-CNN* bekerja dengan memanfaatkan dua tahap utama: pertama, *Region Proposal Network (RPN)* untuk menghasilkan kandidat *bounding box* secara langsung dari *feature maps*, dan kedua, *Fast R-CNN* untuk mengklasifikasikan objek dan menyempurnakan koordinat *bounding box*.

2.6.1 *Feature Extraction*

Tahap ekstraksi fitur merupakan langkah awal dalam proses deteksi objek menggunakan *Faster R-CNN*. Pada tahap ini, gambar masukan akan diproses melalui *CNN* yang berfungsi sebagai *feature extractor*, seperti *ResNet*, *VGG*, atau *MobileNet*. *CNN* digunakan untuk menangkap informasi penting dari gambar, seperti tepi, tekstur, bentuk, dan pola kompleks yang merepresentasikan karakteristik objek dalam gambar.

Proses ini menghasilkan *feature maps* atau peta fitur, yang merupakan representasi spasial dari gambar asli dalam bentuk yang lebih abstrak dan kompak. *Feature maps* ini mengandung informasi penting yang akan digunakan dalam tahap selanjutnya, seperti *Region Proposal Network* (RPN) untuk menghasilkan kandidat *bounding box*, serta *Fast R-CNN* untuk klasifikasi dan regresi koordinat *bounding box* secara lebih akurat. Dengan adanya ekstraksi fitur, sistem dapat memahami struktur objek dalam gambar dan meningkatkan performa deteksi objek dengan lebih efisien.

2.6.2 *Region Proposal Network (RPN)*

Region Proposal Network (RPN) merupakan salah satu komponen utama dalam *Faster R-CNN* yang bertugas menghasilkan *region proposals*, yaitu daerah yang kemungkinan besar mengandung objek. RPN membuat metode ini lebih efisien karena dapat secara langsung menghasilkan proposal *bounding box* dari *feature maps* tanpa perlu menggunakan algoritma seleksi eksternal seperti *Selective Search*.

2.6.2.1 Pemindaian Lokasi Spasial

RPN bekerja dengan memindai setiap lokasi pada *feature maps* yang telah diekstraksi oleh *backbone CNN* (misalnya *ResNet* atau *MobileNet*). Untuk setiap lokasi, RPN menggunakan sejumlah *anchor boxes*, yaitu kotak referensi dengan berbagai ukuran dan rasio aspek yang dirancang untuk mencakup kemungkinan variasi bentuk dan ukuran objek dalam gambar. Misalnya, jika terdapat 9 *anchor boxes* per lokasi, maka RPN akan memeriksa 9 kemungkinan daerah pada setiap

titik di *feature maps*. *Anchor boxes* ini didefinisikan dengan berbagai ukuran tinggi-lebar, yang umumnya meliputi aspek persegi, vertikal, dan horizontal.

2.6.2.2 Objectness Classification

Setelah menetapkan *anchor boxes*, RPN melakukan klasifikasi biner pada setiap *anchor box* untuk menentukan apakah daerah tersebut mengandung objek (*foreground*) atau hanya merupakan latar belakang (*background*).

Untuk melakukan klasifikasi ini, digunakan fungsi *sigmoid* pada Persamaan 2.3 [16].

$$\text{Prob} = \frac{1}{1+e^{-z}} \quad (2.3)$$

Di sini z adalah *output linear* (logit) dari *fully connected layer* di RPN. Hasil dari tahap ini adalah *Objectness Score*, yaitu skor kepercayaan yang menunjukkan seberapa besar kemungkinan suatu *anchor box* mengandung objek.

2.6.2.3 Bounding Box Regression

Selain klasifikasi, RPN juga melakukan regresi untuk menyempurnakan posisi, ukuran, dan bentuk *anchor boxes* agar lebih sesuai dengan objek yang terdeteksi. Proses ini dilakukan dengan menghitung perubahan skala dan pergeseran posisi *anchor box* awal menggunakan Persamaan 2.4 [15].

$$t_x = \frac{x - x_a}{w_a}, \quad t_y = \frac{y - y_a}{h_a}, \quad t_w = \log \frac{w}{w_a}, \quad t_h = \log \frac{h}{h_a} \quad (2.4)$$

di mana:

- (x_a, y_a, w_a, h_a) adalah koordinat *anchor box* awal,
- (x, y, w, h) adalah koordinat *bounding box* yang telah disempurnakan,
- (t_x, t_y, t_w, t_h) adalah parameter transformasi yang diprediksi oleh RPN.

RPN kemudian mengoptimalkan nilai-nilai ini untuk menyesuaikan *anchor boxes* dengan objek yang sesungguhnya dalam gambar.

2.6.2.4 Seleksi *Region Proposals*

Setelah RPN menghasilkan sejumlah *region proposals*, dilakukan proses penyaringan untuk mengurangi redundansi dan mempertahankan proposal yang paling relevan. Salah satu metode yang umum digunakan adalah *Non-Maximum Suppression* (NMS), yang bertujuan untuk menghilangkan *bounding box* yang memiliki tumpang tindih tinggi dengan skor kepercayaan lebih rendah. Metode NMS bekerja dengan menghitung *Intersection over Union* (IoU) antara *bounding box* dengan skor tertinggi dan *bounding box* lainnya.

2.6.3 *Region of Interest (RoI) Pooling*

Region of Interest (RoI) Pooling adalah proses dalam *Faster R-CNN* yang bertujuan untuk menyesuaikan ukuran *region proposals* yang dihasilkan oleh RPN menjadi ukuran tetap sebelum dikirim ke tahap klasifikasi. Karena setiap *bounding box* yang dihasilkan oleh RPN memiliki ukuran yang berbeda-beda, *RoI Pooling* memastikan bahwa setiap fitur dari berbagai objek dalam gambar dapat diproses dengan ukuran yang seragam, sehingga jaringan dapat menangani data secara efisien.

2.6.3.1 *Input Data*

Proses *RoI Pooling* membutuhkan dua *input* utama:

1. *Feature maps* hasil ekstraksi dari *backbone CNN*. *Feature maps* ini adalah representasi spasial dari gambar yang sudah melalui beberapa lapisan konvolusi dan *pooling*.
2. *Bounding box proposals* yang diperoleh dari RPN. *Bounding box* ini adalah wilayah-wilayah potensial yang kemungkinan besar mengandung objek yang ingin dideteksi.

2.6.3.2 *Mapping Bounding Box*

Feature maps memiliki resolusi yang lebih kecil dibandingkan gambar asli karena efek dari operasi konvolusi dan *pooling*. Oleh karena itu, *bounding box* yang

dihasilkan oleh RPN harus dipetakan ulang ke skala *feature maps* sebelum dilakukan *RoI Pooling*.

Misalnya:

- Jika gambar asli berukuran 224×224 dan setelah ekstraksi fitur menghasilkan *feature map* 14×14 , maka koordinat *bounding box* harus disesuaikan dengan skala $224 \rightarrow 14$.
- Jika *bounding box* di gambar asli berada di koordinat (50, 60, 150, 200), maka koordinat pada *feature map* akan berubah menjadi (3, 4, 10, 12) (dengan skala sekitar $1/16$ dari ukuran asli).

2.6.3.3 Membagi *Bounding Box* ke dalam *Grid* Tetap

Setelah *bounding box* dipetakan ke *feature map*, langkah selanjutnya adalah membagi setiap *region proposal* menjadi *grid* tetap. Misalnya, jika ukuran *output* yang diinginkan adalah 7×7 , maka setiap *region proposal* dibagi menjadi 7×7 sel dengan ukuran yang disesuaikan berdasarkan skala *bounding box*. Misalkan *bounding box* dalam *feature map* berukuran (14×14 piksel) dan akan diubah menjadi 7×7 *grid*, maka setiap sel dalam *grid* akan mencakup area 2×2 piksel.

2.6.3.4 Melakukan *Max Pooling* pada Setiap Sel

Setiap sel dalam *grid* akan menjalankan operasi *max pooling*, yaitu memilih nilai maksimum dari setiap wilayah kecil untuk dijadikan representasi fitur dalam sel tersebut. Misalnya, jika *bounding box* terbagi menjadi 7×7 sel, dan setiap sel memiliki ukuran 2×2 piksel, maka nilai maksimum dari keempat piksel di dalam sel akan dipilih sebagai nilai representatif.

Jika pada satu sel terdapat nilai fitur berikut:

$$\begin{bmatrix} 0.8 & 1.2 \\ 0.5 & 0.9 \end{bmatrix}$$

Maka hasil *max pooling* untuk sel tersebut adalah 1.2.

Dengan melakukan operasi ini, ukuran *bounding box* akan berubah menjadi ukuran yang tetap, misalnya $7 \times 7 \times C$, di mana C adalah jumlah *channel* dari *feature maps*. Setelah melalui *RoI Pooling*, setiap *region proposal* kini memiliki ukuran tetap. Ukuran ini memungkinkan semua data dikirim ke *Fully Connected Layer* untuk tahap klasifikasi dan regresi *bounding box* lebih lanjut.

2.6.3.5 Classification & Regression

Setelah fitur dari *RoI Pooling* diperoleh, tahap akhir dalam *Faster R-CNN* adalah klasifikasi dan regresi *bounding box*. Pada tahap ini, model melakukan dua tugas utama, yaitu klasifikasi untuk menentukan apakah suatu RoI mengandung objek serta mengidentifikasi kelas objek tersebut dengan menggunakan fungsi *softmax*, dan *bounding box regression* untuk menyesuaikan posisi, ukuran, dan bentuk *bounding box* agar lebih presisi dalam mencakup objek yang terdeteksi dengan menggunakan regresi linear pada Persamaan 2.4.

2.7 Metrik Evaluasi Deteksi Objek

Untuk mengukur performa model deteksi objek secara kuantitatif, kinerjanya dievaluasi berdasarkan dua aspek utama: ketepatan klasifikasi dan presisi lokalisasi *bounding box*. *Loss*, Metrik *Intersection over Union* (IoU), dan *mean Average Precision* (mAP) digunakan untuk melakukan pengukuran ini dan akan diuraikan lebih lanjut pada bagian berikut.

2.7.1 Loss

Dalam model deteksi objek multi-tahap seperti *Faster R-CNN*, metrik *Loss* tidak dihitung sebagai satu nilai tunggal, melainkan merupakan gabungan dari beberapa fungsi kerugian yang berjalan secara simultan. Hal ini karena arsitekturnya melakukan beberapa tugas sekaligus. Secara garis besar, total *loss* dihitung dari dua komponen utama: *Region Proposal Network* (RPN) dan kepala detektor akhir (*Final Detector Head*), di mana masing-masing komponen memiliki fungsi *loss* untuk tugas klasifikasi dan regresi.

Komponen RPN bertanggung jawab atas dua jenis *loss*: pertama, *Classification Loss* untuk membedakan antara latar depan (objek) dan latar belakang, dan kedua, *Regression Loss* untuk memperbaiki koordinat awal dari

kotak proposal. Selanjutnya, kepala detektor akhir juga menghitung dua *loss* serupa: *Classification Loss* untuk menentukan kelas spesifik dari objek (misalnya 'kendaraan alat berat' atau 'manusia') dan *Regression Loss* untuk penyempurnaan akhir koordinat *bounding box* agar lebih presisi. Keempat nilai *loss* ini kemudian dijumlahkan untuk membentuk nilai Total *Loss*, yang menjadi target utama untuk diminimalkan oleh algoritma optimisasi selama proses pelatihan.

2.7.2 *Intersection over Union*

Metrik fundamental dalam deteksi objek adalah *Intersection over Union* (IoU). IoU berfungsi sebagai nilai untuk mengukur seberapa besar tumpang tindih (*overlap*) antara *bounding box* hasil prediksi model dengan *bounding box* yang sebenarnya atau *ground truth*. Secara matematis, nilai IoU yang berkisar dari 0 (tidak ada tumpang tindih) hingga 1 (tumpang tindih sempurna) dihitung dengan membagi luas area irisan (*intersection*) kedua kotak dengan luas area gabungannya (*union*) menggunakan Persamaan 2.5 [15].

$$IoU = \frac{\text{Area of Overlap}}{\text{Area of Union}} \quad (2.5)$$

Peran utama IoU adalah menjadi ambang batas (*threshold*) untuk menentukan apakah sebuah deteksi dianggap benar atau salah. Dalam praktiknya, sebuah nilai ambang batas IoU (umumnya 0.5 atau 50%) ditetapkan. Jika IoU antara kotak prediksi dan *ground truth* melebihi ambang batas ini, deteksi tersebut diklasifikasikan sebagai *True Positive* (TP), yang berarti model berhasil mendeteksi objek dengan lokasi yang akurat. Sebaliknya, jika nilai IoU di bawah ambang batas, deteksi tersebut dianggap sebagai *False Positive* (FP), yang berarti lokasi prediksinya tidak cukup akurat meskipun mungkin kelasnya benar.

2.7.3 *Precision dan Recall*

Setelah deteksi diklasifikasikan sebagai *True Positive* (TP) atau *False Positive* (FP) menggunakan ambang batas IoU, metrik *Precision* dan *Recall* digunakan untuk meringkas performa model dari dua sudut pandang yang berbeda. *Precision* menjawab pertanyaan: "Dari semua objek yang dideteksi oleh model sebagai target (misalnya, sebagai 'kendaraan alat berat'), berapa persen yang benar-

benar merupakan objek 'kendaraan alat berat '?" Metrik ini mengukur tingkat keakuratan dari prediksi positif yang dibuat oleh model, atau seberapa rendah tingkat *false alarm* (alarm palsu) dari sistem. *Recall*, di sisi lain, menjawab pertanyaan: "Dari semua objek 'kendaraan alat berat ' yang sebenarnya ada di dalam gambar, berapa persen yang berhasil ditemukan oleh model?" Metrik ini juga dikenal sebagai sensitivitas, yang mengukur kemampuan model untuk menemukan semua objek relevan yang ada.

2.7.3 *Average Precision dan mean Average Precision*

Dalam praktiknya, sering terjadi pertukaran antara *Precision* dan *Recall*. Untuk mendapatkan gambaran performa yang lebih utuh, digunakan metrik *Average Precision* (AP). AP merangkum kurva *precision-recall* menjadi satu angka tunggal yang merepresentasikan kualitas deteksi untuk satu kelas objek tertentu. Sebagai contoh, dalam penelitian ini akan ada nilai AP untuk kelas 'kendaraan alat berat' dan nilai AP yang terpisah untuk kelas 'manusia'.

Untuk mendapatkan satu metrik tunggal yang dapat menilai performa model secara keseluruhan di semua kelas, digunakan *mean Average Precision* (mAP). Metrik mAP adalah nilai rata-rata dari AP di semua kelas objek yang dievaluasi. Dalam konteks penelitian ini, nilai mAP dihitung dari rata-rata AP kelas 'kendaraan alat berat' dan AP kelas 'manusia', sehingga memberikan skor performa akhir untuk keseluruhan sistem deteksi., mAP sering kali dihitung pada berbagai ambang batas IoU. Notasi yang umum digunakan adalah $mAP=0,5$, yang berarti mAP dihitung dengan ambang batas IoU 0,5, dan $mAP=0,5:0,95$ (standar COCO), yang menghitung rata-rata mAP pada sepuluh ambang batas IoU berbeda dari 0,5 hingga 0,95. Penggunaan rentang IoU ini memberikan penilaian yang lebih ketat dan menyeluruh terhadap kemampuan lokalisasi model.

2.8 *Konfigurasi Faster RCNN*

Keberhasilan pelatihan model deteksi objek seperti *Faster R-CNN* bergantung pada serangkaian konfigurasi yang mengatur bagaimana model tersebut belajar dari data. Untuk melatih model, digunakan algoritma optimisasi *Stochastic Gradient Descent* (SGD), yang berfungsi sebagai mesin untuk menyesuaikan parameter internal model secara berulang agar tingkat kesalahannya (*loss*) semakin

minimum. Setiap penyesuaian tersebut dikontrol oleh *learning rate*, sebuah parameter yang diatur ke nilai yang kecil untuk memastikan proses belajar berjalan stabil dan tidak melompati solusi optimal. Proses pelatihan ini dijalankan secara berulang untuk beberapa *epoch*, di mana setiap *epoch* menandakan model telah melihat keseluruhan data latih. Agar efisien, data disajikan dalam kelompok-kelompok kecil atau *batch* berisi beberapa gambar, di mana setelah setiap *batch* diproses, model akan melakukan pembaruan bobot. Untuk mengamankan progres dan mengantisipasi interupsi, kondisi model beserta bobot yang telah dipelajarinya disimpan sebagai *checkpoint* pada setiap akhir *epoch*, memungkinkan pelatihan untuk dilanjutkan kembali dari titik terakhir yang tersimpan.

III. METODOLOGI PENELITIAN

3.1 Waktu dan Tempat Penelitian

Pelaksanaan penelitian ini dilakukan di Laboratorium Teknik Kendali dan Laboratorium Pengukuran Besaran Listrik untuk menyelesaikan pengembangan sistem pendeteksian kendaraan alat berat dan orang menggunakan *CNN*. Adapun pelaksanaannya penelitian ini mulai Februari – Juli 2025.

3.2 Alat dan Bahan

Adapun alat dan bahan yang akan digunakan yaitu pada Tabel 3.1.

Tabel 3. 1 Alat dan Bahan

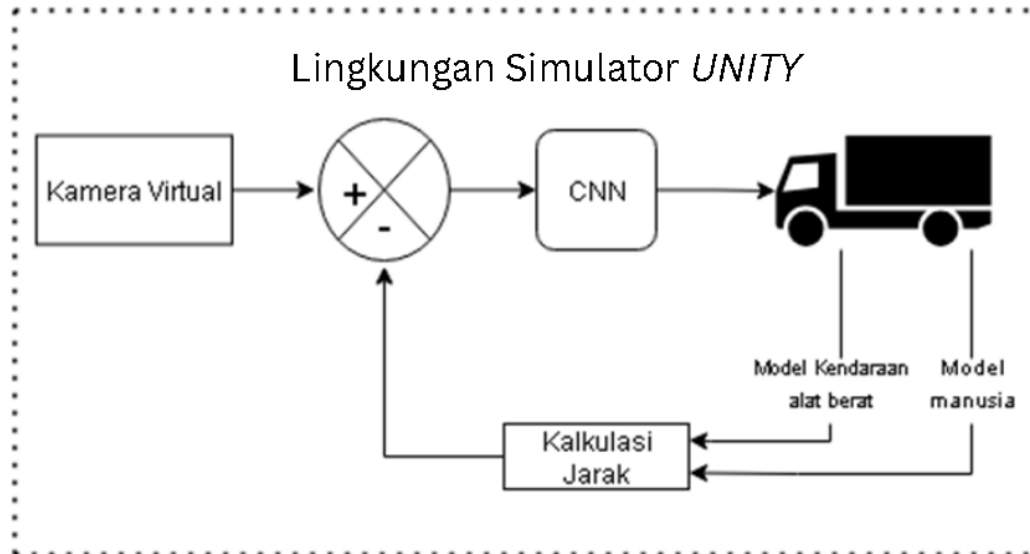
No	Alat dan Bahan	Keterangan Penggunaan
1.	Foscam Webcam	Foscam Webcam adalah webcam dengan resolusi 1080P yang digunakan untuk melatih dari model <i>CNN</i>
2.	PC PBL	PC PBL dengan Prosessor Core I5 1135G7 Intel® Iris® Xe Graphics RAM 8 GB. PC digunakan sebagai media untuk menjalankan program dan menampilkan hasil yang dilakukan.
3.	Perangkat Lunak Google Colab	Perangkat Lunak Google Colab digunakan untuk memperoleh data gambar yang dibutuhkan untuk dilatih menjadi model <i>machine learning</i> .
4.	<i>CNN</i> dan <i>Faster RCNN</i>	<i>CNN</i> dan <i>Faster RCNN</i> digunakan untuk mendeteksi letak kendaraan alat berat dan manusia.

3.3 Metode Penelitian

Adapun metode penelitian dalam pengerjaan simulator ini adalah sebagai berikut:

3.3.1 Konsep Sistem

Pada penelitian ini dilakukan konsep sistem yang dibangun dapat dilihat pada Gambar 3.1.



Gambar 3. 1 Konsep Perancangan Sistem Deteksi Kendaraan Alat Berat dan Manusia

Seperti yang ditunjukkan pada Gambar 3.1, konsep sistem ini diimplementasikan sepenuhnya di dalam sebuah Lingkungan Simulator *UNITY* untuk mengawasi area *blind spot* di bagian depan model kendaraan alat berat. Cara kerjanya dimulai dengan sebuah kamera virtual yang ditempatkan pada model truk untuk menangkap gambar dari *scene* simulator. Gambar virtual tersebut kemudian diolah oleh model kecerdasan buatan (*CNN*) yang telah dilatih untuk menemukan dan mengenali objek. Setelah sebuah objek terdeteksi, sistem akan langsung menghitung perkiraan jaraknya untuk menentukan tingkat bahaya. Proses ini penting untuk membedakan apakah objek di depan adalah model virtual manusia atau kendaraan lain dalam simulator, sehingga dapat memberikan penglihatan tambahan yang akurat kepada operator untuk mencegah kecelakaan dalam skenario uji.

Tabel 3. 2 Parameter Dimensi Objek

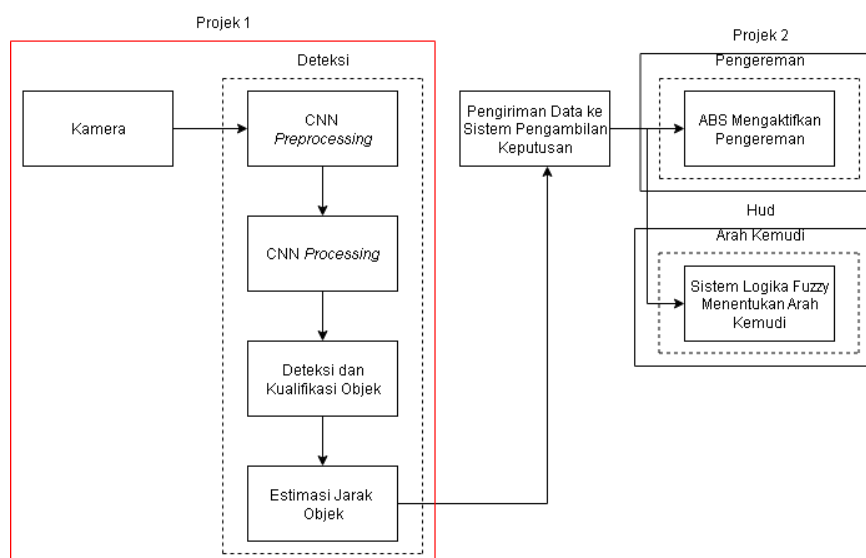
Objek	Deskripsi Dimensi
Kendaraan Alat Berat	Panjang : 10—22 m Lebar : 2,5—3 m Tinggi : 3,5—4,2 m

Objek	Deskripsi Dimensi
Manusia	Tinggi : 1,5—1,9 m Lebar (bahu): 0,4—0,6 m

Tabel 3.2 di atas menyajikan parameter dimensi fisik yang digunakan sebagai acuan untuk sistem deteksi dalam mengenali dua kelas objek, yaitu Kendaraan Alat Berat dan Manusia. Untuk Kendaraan Alat Berat, sistem menggunakan rentang ukuran yang besar mencakup panjang (10—22 m), lebar (2,5—3 m), dan tinggi (3,5—4,2 m) sebagai ciri khasnya. Sebaliknya, untuk mengidentifikasi Manusia, sistem mengacu pada parameter dimensi yang jauh lebih kecil, yaitu tinggi badan (1,5—1,9 m) dan lebar bahu (0,4—0,6 m). Penetapan rentang ukuran yang kontras ini memungkinkan sistem untuk secara efektif membedakan dan mengklasifikasikan kedua objek tersebut berdasarkan karakteristik fisiknya di dunia nyata.

3.3.2 Perancangan Sistem

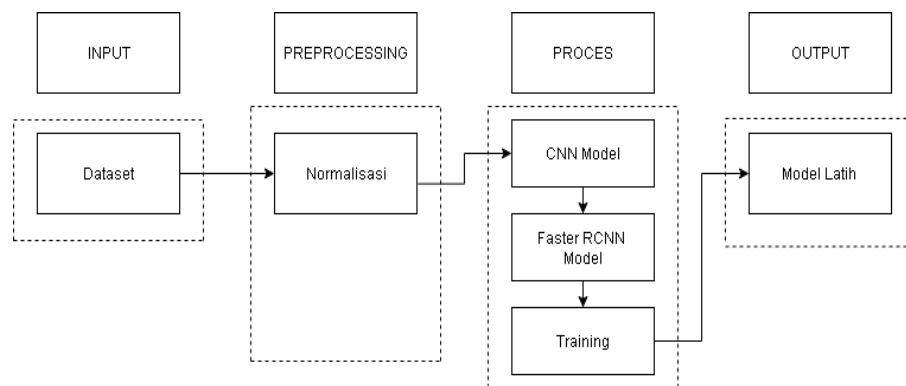
Secara keseluruhan, sistem ini terdiri dari tiga bagian utama yang saling terintegrasi, yaitu sistem deteksi objek menggunakan kamera, sistem pengereman otomatis berbasis ABS, dan sistem pengambilan keputusan pada kemudi menggunakan logika fuzzy. Pada Gambar 3.2 ditampilkan diagram blok sistem secara keseluruhan, yang menunjukkan bagaimana setiap komponen saling berhubungan dalam proses kerja sistem.



Gambar 3. 2 Blok Diagram Sistem

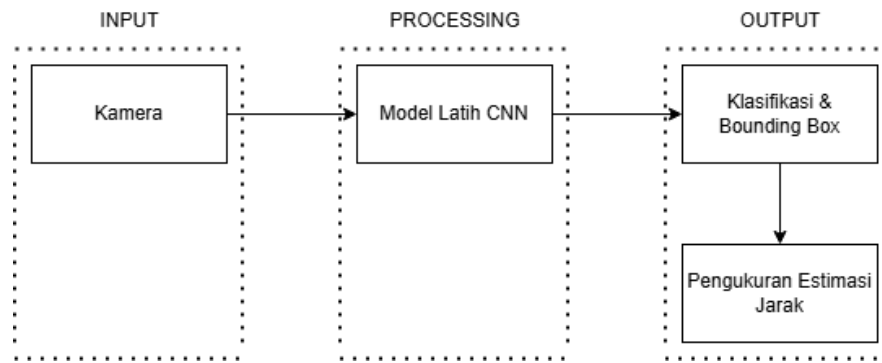
Proses dimulai dengan kamera yang dipasang pada *bumper* depan kendaraan untuk menangkap gambar lingkungan sekitar secara *real-time*. Data visual ini kemudian diproses oleh sistem deteksi berbasis *CNN* untuk mengidentifikasi kendaraan alat berat dan orang yang berada di sekitar. Informasi hasil deteksi, seperti posisi dan jarak objek, diteruskan ke sistem pengambilan keputusan. Jika jarak dengan objek terlalu dekat, sistem akan mengirimkan perintah ke sistem pengereman ABS untuk melakukan pengereman otomatis guna menghindari tabrakan. Selain itu, sistem pengambilan keputusan berbasis logika fuzzy akan menentukan arah kemudi yang optimal jika diperlukan manuver penghindaran. Dengan integrasi ketiga sistem ini, kendaraan dapat beroperasi dengan lebih aman dan responsif terhadap kondisi lingkungan sekitarnya.

Perancangan sistem deteksi objek dimulai dari menentukan komponen yang akan digunakan sebagai kebutuhan dalam penelitian, setelah itu membuat *source code* pada *software PyCharm* untuk pelatihan model latih dan menjalankan sistem pendeteksian kendaraan alat berat dan orang. Gambar 3.3 merupakan diagram blok dari sistem pendeteksian kendaraan alat berat dan manusia.



Gambar 3. 3 Blok Diagram *Training Model*

Pada bagian Gambar 3.3, sistem dimulai dengan tahap *input*, yaitu dataset gambar yang akan digunakan untuk melatih model *CNN*. Dataset ini melalui tahap *preprocessing*, yang mencakup normalisasi untuk menyelaraskan nilai piksel dalam rentang tertentu, dan *augmentasi* untuk meningkatkan keragaman data guna menghindari *overfitting*. Setelah *preprocessing*, gambar-gambar ini diproses oleh *CNN Model*, di mana proses *training* dilakukan untuk menghasilkan model latih *CNN* sebagai *output*.

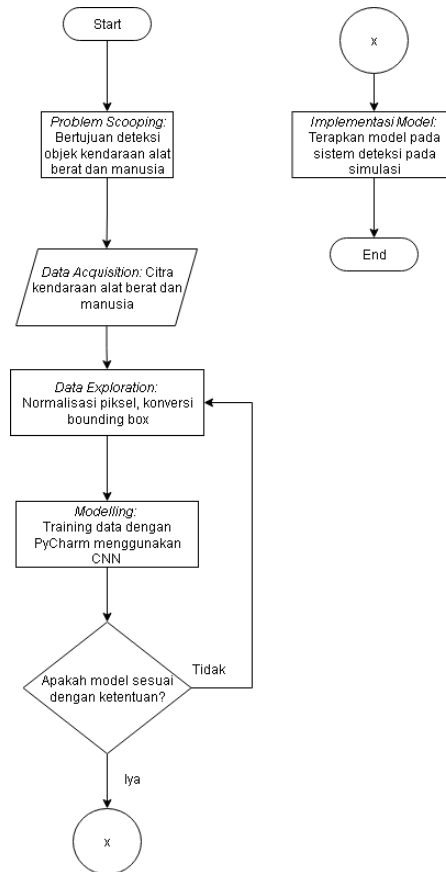


Gambar 3. 4 Diagram Blok Sistem Deteksi Objek

Pada bagian Gambar 3.4, diagram blok menunjukkan bagaimana model yang telah dilatih digunakan dalam implementasi sistem deteksi objek secara langsung. Kamera berfungsi sebagai *input*, menangkap gambar secara *real-time* yang kemudian diproses menggunakan model latih *CNN*. Model ini akan mengklasifikasikan objek yang terdeteksi serta menambahkan *bounding box* untuk menandai posisi objek dalam gambar. Selain itu, sistem juga melakukan pengukuran estimasi jarak. Keseluruhan sistem ini bertujuan untuk mendeteksi dan mengklasifikasikan kendaraan alat berat serta orang, sekaligus memberikan informasi mengenai jarak objek untuk keperluan lebih lanjut dalam pengambilan keputusan.

3.4 Perancangan Metode *CNN*

Alur kerja dan metode dalam perancangan program menggunakan metode *CNN* mengikuti tahapan *AI Project Cycle*, yang merupakan pendekatan sistematis untuk mengembangkan sebuah proyek berbasis kecerdasan buatan. *AI Project Cycle* memastikan setiap langkah dalam proses pengembangan dilakukan secara terstruktur, mulai dari identifikasi masalah, pengumpulan dan pengolahan data, perancangan model, pelatihan, hingga evaluasi dan implementasi. Dengan mengikuti siklus ini, pengembangan model *CNN* menjadi lebih terarah, efisien, dan mampu menghasilkan performa yang optimal dalam mendeteksi objek, khususnya manusia dan kendaraan dalam penelitian ini.



Gambar 3. 5 Diagram Alir AI Project Cycle Metode CNN

Dalam penelitian ini, metode *CNN* digunakan untuk melakukan deteksi objek dengan memanfaatkan arsitektur berbasis *transfer learning* guna meningkatkan akurasi dan efisiensi model. Setiap tahapan dalam *AI Project Cycle* saling terhubung dan memungkinkan dilakukan iterasi untuk perbaikan model yang berkelanjutan. Proses ini memastikan bahwa model yang dihasilkan mampu beradaptasi dengan kebutuhan data yang beragam dan kompleks. Alur kerja dari *AI Project Cycle* yang diterapkan dalam perancangan program ini digambarkan secara jelas dalam diagram alir pada Gambar 3.5 berikut.

Berdasarkan Gambar 3.5, tahapan perancangan program metode *CNN* menggunakan *AI Project Cycle* pada penelitian ini adalah sebagai berikut:

3.4.1 Problem Scoping

Langkah ini bertujuan untuk merinci masalah yang dihadapi dan merumuskan solusi yang melibatkan kecerdasan buatan seperti yang ditunjukkan pada Tabel 3.3.

Tabel 3. 3 *Problem Scoping*

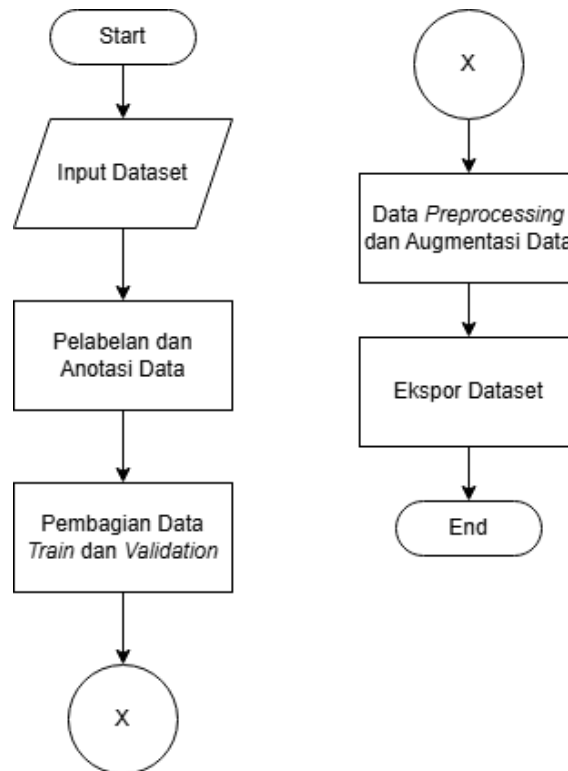
Who?	Saya adalah peneliti yang terlibat dalam penelitian ini. Saya bekerja secara mandiri dalam pengembangan sistem pendeteksian objek untuk meningkatkan akurasi untuk mendeteksi manusia dan kendaraan.
What?	Penelitian ini bertujuan untuk mengembangkan program menggunakan metode <i>CNN</i> untuk mendeteksi objek dan menentukan koordinat <i>bounding box</i> secara akurat.
Where?	Laboratorium Teknik Kendali dan Laboratorium Pengukuran Besaran Listrik.
Why?	Sistem pendeteksian objek konvensional memiliki keterbatasan dalam akurasi dan efisiensi. Penggunaan model <i>CNN</i> yang ringan namun tetap akurat diharapkan dapat meningkatkan efisiensi pendeteksian, khususnya dalam pemantauan real-time.

3.4.2 *Data Acquisition*

Data pelatihan yang digunakan dalam penelitian ini merupakan data citra yang berisi gambar objek manusia dan kendaraan yang diambil dari *Dataset COCO* (*Common Objects in Context*), sebuah dataset besar yang menyediakan gambar dengan anotasi bounding box untuk berbagai kategori objek. Dengan ini, memungkinkan model *CNN* dilatih dengan data yang memiliki variasi tinggi dalam hal posisi, sudut pandang, latar belakang, dan kondisi pencahayaan, sehingga meningkatkan kemampuan generalisasi model dalam mendeteksi objek secara akurat di berbagai skenario.

3.4.3 *Data Exploration*

Proses ini dilakukan untuk mengolah dataset yang telah dikumpulkan dari *Dataset COCO* agar sesuai dengan kebutuhan model *CNN* yang dikembangkan. Pengolahan data dilakukan melalui tahap praproses menggunakan *Python* dengan beberapa *library* untuk memastikan format gambar dan anotasi sesuai dengan prasyarat model. Alur kerja pada *data exploration* dapat dilihat pada Gambar 3.6.



Gambar 3. 6 Diagram Alir Data *Exploration*

Pada tahap *data exploration*, proses pelabelan dan anotasi data memanfaatkan format standar dari *Dataset COCO*, yang menggunakan sebuah file JSON untuk menyimpan seluruh informasi anotasi. File JSON ini memiliki struktur yang terorganisir, berisi daftar gambar, daftar kategori objek (seperti 'manusia' dan 'kendaraan alat berat'), serta daftar anotasi itu sendiri. Setiap anotasi menandai objek di dalam gambar menggunakan *bounding box* dengan format $[x, y, \text{width}, \text{height}]$, di mana (x, y) adalah koordinat pojok kiri atas, diikuti oleh lebar dan tinggi kotak.

Tabel 3. 4 Jumlah Data Pelatihan yang Digunakan

Nama Kelas	Total Gambar
Kendaraan Alat Berat	2000
Manusia	2000

Setelah proses anotasi selesai, dataset dibagi menjadi dua bagian, yaitu 2.000 gambar untuk data kendaraan alat berat dan 2.000 gambar untuk data manusia. Dari total 4.000 gambar, proporsi data training adalah 80%, sedangkan data test adalah 20%. Pembagian ini dilakukan untuk memastikan model dapat

belajar secara optimal melalui data training dan dievaluasi performanya menggunakan data test.

Tabel 3. 5 Perincian Perbandingan Data Gambar untuk Dataset

Perbandingan Data	Gambar
80% data <i>train</i>	3547
20% data <i>test</i>	887
Jumlah Gambar	4434

Tahapan selanjutnya pada data *exploration* adalah tahap *preprocessing* data dan augmentasi data menggunakan Python dengan bantuan *library* seperti *OpenCV* dan *TensorFlow*. Detail *preprocessing data* dan *augmentasi data* yang digunakan dapat dilihat pada Tabel 3.6.

3.4.4 Preprocessing

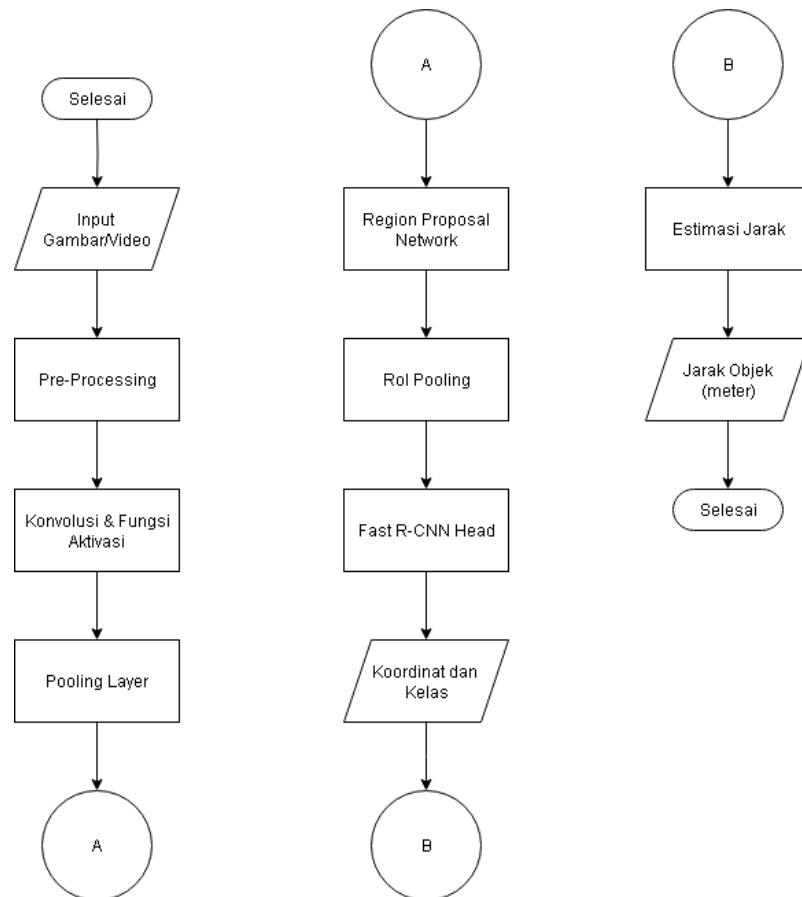
Tahap pra-pemrosesan data dilakukan untuk memastikan konsistensi selama proses pelatihan dan inferensi model *CNN*. Proses ini mencakup validasi data untuk memastikan gambar yang digunakan memiliki anotasi lengkap dan tidak ada *file* yang hilang. Gambar-gambar yang tidak memiliki objek atau mengalami kesalahan saat pemrosesan dihapus dari dataset untuk menghindari bias dalam pelatihan model. Selanjutnya, dilakukan normalisasi nilai piksel dengan mengubah skala warna dari 0—255 menjadi 0—1 saat gambar dikonversi menjadi tensor PyTorch. Normalisasi ini bertujuan untuk mempercepat konvergensi model selama pelatihan. Dataset yang digunakan untuk pelatihan dan pengujian adalah hasil setelah pra-pemrosesan ini.

Tabel 3. 6 Perincian Preprocessing Data dan Augmentasi Data

Preprocessing	Normalisasi: Skala piksel 0—1 Validasi Data: Menghapus gambar tanpa objek atau hilang
Augmentation	Tidak Dilakukan

3.4.5 Modelling

Pada tahap *modelling*, dataset yang telah diproses digunakan untuk melatih model *CNN* dengan memanfaatkan arsitektur *CNN* sebagai *pretrained model* yang bertindak sebagai fitur ekstraktor utama. Penggunaan *pretrained model* ini bertujuan untuk mempercepat proses pelatihan dan meningkatkan performa model dalam mendeteksi objek, khususnya manusia dan kendaraan.



Gambar 3. 7 Diagram Model

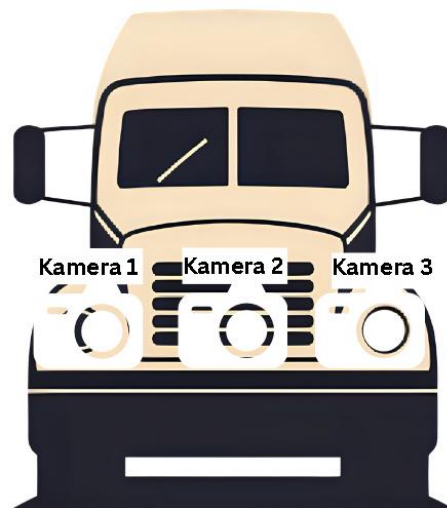
Seperti yang diilustrasikan dalam Gambar 3.7, alur kerja sistem dimulai setelah tahap pra-pemrosesan, di mana gambar input masuk ke arsitektur *Backbone CNN*. Di sini, lapisan-lapisan Konvolusi, Fungsi Aktivasi, dan *Pooling* bekerja secara hierarkis untuk mengekstraksi fitur-fitur visual penting, mulai dari tepi dan tekstur hingga pola yang lebih kompleks dan mengubahnya menjadi peta fitur. Peta fitur ini kemudian diteruskan ke kepala *Faster R-CNN*, yang bertugas mengidentifikasi lokasi objek, menentukan kelasnya ('kendaraan alat berat' atau 'manusia'), dan menyempurnakan koordinat *bounding box* agar pas mengelilingi objek tersebut. Selanjutnya, informasi *bounding box* yang akurat tersebut

digunakan sebagai input untuk mengestimasi jarak ke objek berdasarkan prinsip kamera monokular.

Keseluruhan kemampuan model ini dicapai melalui proses pelatihan intensif. Pelatihan ini dijalankan menggunakan GPU untuk akselerasi komputasi, dengan konfigurasi seperti jumlah *epoch* sebanyak 30 dan *batch size* sebesar 2. Penentuan parameter tersebut merupakan hasil dari serangkaian eksperimen yang dilakukan untuk mencapai keseimbangan terbaik antara kecepatan dan akurasi model yang optimal.

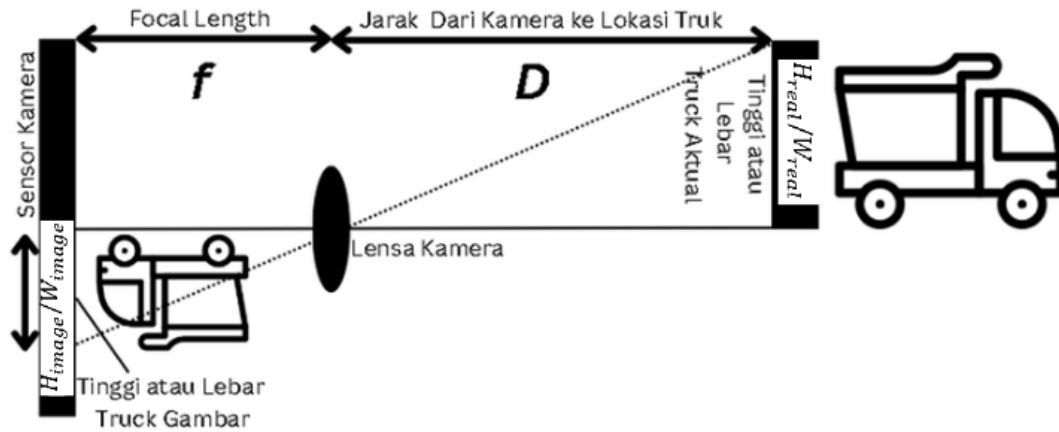
3.5 Denah Peletakan Kamera

Seperti pada Gambar 3.8 menunjukkan desain atau tata letak kamera yang digunakan pada kendaraan alat berat. Tiga kamera akan dipasang pada bagian depan kendaraan alat berat untuk memastikan cakupan visual yang optimal dalam mendeteksi kendaraan dan pejalan kaki. Kamera utama ditempatkan di tengah bumper depan, memberikan sudut pandang luas terhadap jalan di depan tanpa menghalangi visibilitas pengemudi, serta memungkinkan deteksi objek dengan perspektif terbaik. Dua kamera tambahan dipasang di sisi kanan dan kiri bumper depan untuk memperluas jangkauan pandangan terhadap area samping kendaraan alat berat, membantu dalam mengurangi *blind spot* dan meningkatkan keselamatan saat bermanuver. Kamera-kamera ini akan mengirimkan gambar ke sistem berbasis *CNN*, di mana gambar akan diproses untuk mendeteksi objek dan menampilkan *bounding box*.



Gambar 3. 8 Desain Peletakan Kamera

3.6 Mengukur Jarak dengan Satu Kamera



Gambar 3. 9 Prinsip Kamera

Pada gambar 3.8 menunjukkan prinsip dasar kerja kamera monokular dalam melakukan estimasi jarak menggunakan model kamera pinhole. Ketika sebuah model *CNN* berhasil mengidentifikasi sebuah objek, misalnya manusia atau mobil, model tersebut akan menghasilkan sebuah *bounding box* di sekeliling objek pada gambar. *Bounding box* inilah yang menjadi kunci untuk perhitungan jarak. Secara geometris, terbentuk dua segitiga yang sebangun antara objek asli dan proyeksi bayangannya pada sensor kamera[17]. Segitiga pertama dibentuk oleh tinggi objek aktual (*Object actual dimension*) dan jarak dari kamera ke objek (*Distance from camera to measurement location*). Segitiga kedua Dibentuk oleh tinggi *bounding box* dalam piksel dan panjang *Focal length* dalam piksel, yang merupakan nilai intrinsik dari kamera. Dari hubungan segitiga sebangun ini, diperoleh hubungan matematis pada persamaan 3.1 dan 3.2

$$D = \frac{f \times H_{real}}{H_{image}} \quad (3.1)$$

$$D = \frac{f \times W_{real}}{W_{image}} \quad (3.2)$$

dengan:

- D : Jarak ke objek dalam meter
- f : Panjang fokus kamera dalam piksel
- H_{real} : Tinggi sebenarnya dari objek
- H_{image} : Tinggi objek dalam gambar
- W_{real} : Lebar sebenarnya dari objek
- W_{image} : Lebar objek dalam gambar

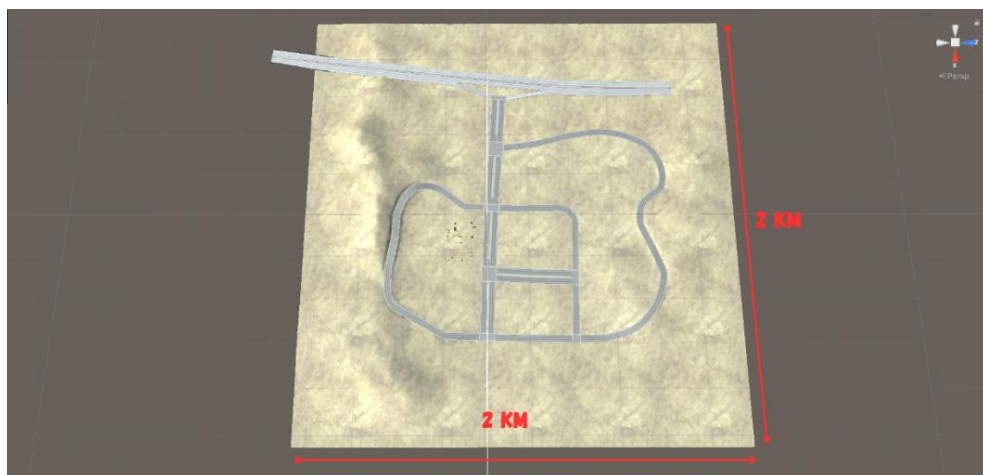
3.7 Perancangan Sketsa Simulator



(a)



(b)



(c)

Gambar 3. 10 Gambar (a) Tampak Kendaraan Alat Berat (b) Tampak Manusia
(c) Pemodelan Lingkungan Uji

Gambar 3.10 menampilkan komponen visual utama yang digunakan dalam perancangan lingkungan simulator untuk pengujian sistem. Gambar bagian (a) dan (b) menampilkan lingkungan simulator yang menyerupai area industri atau konstruksi, lengkap dengan aset-aset yang relevan. Ini mencakup model berbagai jenis kendaraan alat berat sebagai target deteksi utama, yang secara spesifik dalam penelitian ini didefinisikan memiliki dimensi lebar 3,5 meter dan tinggi 4 meter, dan model manusia yang berperan sebagai pejalan kaki di sekitar area operasional. Sementara itu, gambar (c) bawah menyajikan denah atau tata letak keseluruhan dari arena pengujian. Peta ini menunjukkan desain jalur jalan dan skala area yang luas (sekitar 2 km), yang memungkinkan dilakukannya pengujian deteksi dan estimasi jarak dalam berbagai skenario.

3.8 Pengujian Sistem

Pada tahap pengujian sistem dilakukan beberapa pengujian dan lokasi pengujian sistem yang dilakukan di Laboratorium Teknik Kendali Universitas Lampung. Adapun tabel pengujian dapat dilihat pada Tabel 3.2 adalah sebagai berikut:

Tabel 3. 7 Pengujian Sistem

No	Jenis Pengujian	Indikator Keberhasilan	Keterangan
1	Uji Akurasi Deteksi Objek	Nilai $mAP[0,5:0,95] > 0,400$	Menguji pengukuran dengan kamera system.
2	Validasi Jarak Deteksi	$Error\ margin \leq 5\%$	Menguji terhadap berbagai jarak
3	Uji Kinerja Model <i>CNN</i>	Waktu inferensi < 1 detik	Mengukur kecepatan sistem dalam mendeteksi objek secara real-time di dalam simulator <i>UNITY</i> .

Pada Tabel 3.2 merupakan tabel pengujian yang akan dilakukan untuk mengetahui keberhasilan dari sistem.

3.9 Pengujian Validasi Jarak Deteksi

Pengujian validasi jarak bertujuan untuk mengevaluasi akurasi sistem dalam mengestimasi jarak objek secara kuantitatif pada berbagai rentang. Proses ini

dilakukan dengan membandingkan Jarak Prediksi yang dihasilkan oleh sistem terhadap Jarak *Ground Truth* yang diukur secara presisi di dalam lingkungan simulator. Untuk mendapatkan analisis yang komprehensif, pengujian difokuskan pada dua skenario utama yang relevan dengan keselamatan operasional: validasi kinerja pada zona *blind spot* kritis (1—6 meter) yang merupakan area paling berbahaya, serta pengujian pada jarak operasional umum (di atas 6 meter) untuk mengevaluasi konsistensi performa sistem. Akurasi pada setiap skenario diukur menggunakan metrik *Percentage Error*, dengan sistem dianggap berhasil jika mampu mempertahankan margin kesalahan tidak lebih dari 5%

3.10 Pengujian Akurasi Deteksi Objek

Pengujian akurasi deteksi objek bertujuan untuk mengukur performa model *CNN* secara kuantitatif dalam mengenali dan melokalisasi objek (manusia dan mobil) secara akurat. Metrik yang digunakan adalah *mean Average Precision* (mAP) dengan rentang IoU standar COCO, yang sangat baik untuk mengevaluasi ketepatan klasifikasi dan presisi lokasi *bounding box* secara komprehensif. Indikator keberhasilan untuk pengujian ini adalah pencapaian nilai mAP[0,5:0,95] di atas 40%. Notasi [0,5:0,95] menandakan bahwa metrik ini menghitung rata-rata akurasi pada berbagai tingkat presisi lokasi, mulai dari ambang batas *Intersection over Union* (IoU) 0,5 hingga 0,95. Hasil ini memvalidasi kelayakan model untuk diintegrasikan pada tahap pengembangan sistem selanjutnya.

3.11 Pengujian Kinerja Model *CNN*

Pengujian kinerja model *CNN* dilakukan untuk mengukur seberapa cepat sistem dapat mendeteksi objek dalam kondisi *real-time*. Uji ini berfokus pada penghitungan waktu inferensi ($T_{inferensi}$), yaitu total waktu yang dibutuhkan oleh model untuk memproses satu citra hingga mengeluarkan hasil deteksi. Secara matematis, waktu inferensi dihitung menggunakan Persamaan 3.1.

$$T_{inferensi} = (t_{selesai} - t_{mulai}) \times 1000 \quad (3.1)$$

Di mana t_{mulai} adalah waktu sebelum proses deteksi dan $t_{selesai}$ adalah waktu setelahnya. Pengujian dilakukan dalam lingkungan simulator *UNITY* untuk

memastikan model dapat bekerja secara efisien. Sistem dianggap memiliki kinerja yang memadai jika waktu inferensi rata-rata yang diperoleh kurang dari 1000 ms (1 detik), sehingga memungkinkan deteksi objek yang responsif dalam situasi nyata

V. KESIMPULAN DAN SARAN

5.1 Kesimpulan

Adapun kesimpulan yang diperoleh adalah penelitian ini berhasil merancang dan mengimplementasikan sistem deteksi objek berbasis *CNN* dalam simulator *UNITY* yang mampu mengenali objek kendaraan alat berat dan manusia dengan kinerja deteksi model dengan nilai *mean Average Precision* (mAP[0,5:0,95]) sebesar 0,422. Sistem ini juga menunjukkan presisi tinggi dalam estimasi jarak, dengan rata-rata *Percentage Error* hanya 1,06% untuk kendaraan alat berat dan 2,7% untuk manusia, sekalipun akurasi pada objek manusia menurun pada jarak terjauh hingga 8,76%. Sementara dari sisi kecepatan, waktu inferensi telah memenuhi target di bawah 1 detik, berkisar antara 537—832 ms.

5.2 Saran

Adapun saran untuk penelitian ini adalah :

1. Melakukan optimasi lebih lanjut untuk meningkatkan performa deteksi objek berukuran kecil, khususnya manusia pada jarak jauh. Hal ini dapat dicapai melalui strategi data *augmentation* yang lebih spesifik untuk objek kecil, penyesuaian *anchor box* pada RPN, atau eksplorasi arsitektur model yang dirancang khusus untuk deteksi multi-skala atau objek kecil.
2. Untuk optimasi kecepatan menuju performa *real-time* (>20 FPS), disarankan dua pendekatan utama: pertama, akselerasi perangkat keras dengan menggunakan GPU diskrit yang lebih kuat. Kedua, eksplorasi arsitektur model yang lebih ringan dan cepat, seperti YOLO, yang dapat meningkatkan kecepatan inferensi secara signifikan.

DAFTAR PUSTAKA

- [1] S. Jaiswal, "Multilayer Perceptrons in Machine Learning: A Comprehensive Guide," 2024.
- [2] I. Nihayatul Husna, M. Ulum, A. Kurniawan Saputro, D. Tri Laksono, and D. Neipa Purnamasari, "Rancang Bangun Sistem Deteksi Dan Perhitungan Jumlah Orang Menggunakan Metode Convolutional Neural Network (CNN)," *Seminar Nasional Fortei Regional*, vol. 7.
- [3] Y. Zhang, Z. Guo, J. Wu, Y. Tian, H. Tang, and X. Guo, "Real-Time Vehicle Detection Based on Improved YOLO v5," *Sustainability (Switzerland)*, vol. 14, no. 19, Oct. 2022, doi: 10.3390/su141912274.
- [4] Y. Zhou, S. Wen, D. Wang, J. Mu, and I. Richard, "Object detection in autonomous driving scenarios based on an improved faster-RCNN," *Applied Sciences (Switzerland)*, vol. 11, no. 24, Dec. 2021, doi: 10.3390/app112411630.
- [5] J. Zhao *et al.*, "Improved Vision-Based Vehicle Detection and Classification by Optimized YOLOv4," *IEEE Access*, vol. 10, pp. 8590–8603, 2022, doi: 10.1109/ACCESS.2022.3143365.
- [6] M. A. R. Majid, M. A. Hady, M. Sahal, and Y. E. Nugraha, "Sistem Deteksi dan Tracking Keretakan Bangunan Dengan Unmanned Aerial Vehicle Menggunakan Algoritma CNN," Departemen Teknik Elektro, Institut Teknologi Sepuluh Nopember (ITS), Surabaya, 2023.
- [7] S. Meivel *et al.*, "Mask Detection and Social Distance Identification Using Internet of Things and Faster R-CNN Algorithm," *Comput Intell Neurosci*, vol. 2022, 2022, doi: 10.1155/2022/2103975.
- [8] S. R. Sulistiyanti, F. A. Setyawan, and M. Komarudin, "PENGOLAHAN CITRA; DASAR DAN CONTOH PENERAPANNYA," Yogyakarta, 2016.
- [9] L. Li, Y. Zhang, and P. Wang, "Compression of Conditional Deep Learning Network for Fast and Low Power Mobile Applications," 2017.

- [10] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Commun ACM*, vol. 60, no. 6, pp. 84–90, Jun. 2017, doi: 10.1145/3065386.
- [11] F. Nurona Cahya *et al.*, "SISTEMASI: Jurnal Sistem Informasi Klasifikasi Penyakit Mata Menggunakan Convolutional Neural Network (CNN)." [Online]. Available: <http://sistemasi.ftik.unisi.ac.id>
- [12] M. Bai and M. Li, "A Presentation of Structures and Applications of Convolutional Neural Networks," 2023.
- [13] L. Alzubaidi *et al.*, "Review of deep learning: concepts, CNN architectures, challenges, applications, future directions," *J Big Data*, vol. 8, no. 1, Dec. 2021, doi: 10.1186/s40537-021-00444-8.
- [14] V. Dumoulin and F. Visin, "A guide to convolution arithmetic for deep learning," Jan. 2018, [Online]. Available: <http://arxiv.org/abs/1603.07285>
- [15] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," Jan. 2016, [Online]. Available: <http://arxiv.org/abs/1506.01497>
- [16] A. Menon, K. Mehrotra, C. K. Mohan, and S. RANKA SyracuseUniversity, "Characterization of a Class of Sigmoid Functions with Applications to Neural Networks," 1996.
- [17] R. K. Megalingam, V. Shriram, B. Likhith, G. Rajesh, and S. Ghanta, "Monocular distance estimation using pinhole camera approximation to avoid vehicle crash and back-over accidents," in *Proceedings of the 10th International Conference on Intelligent Systems and Control, ISCO 2016*, Institute of Electrical and Electronics Engineers Inc., Oct. 2016. doi: 10.1109/ISCO.2016.7727017.