TUGAS AKHIR – EC184801

DETEKSI PENGGUNAAN HELM PADA PENGENDARA BERMOTOR BERBASIS *DEEP LEARNING*

Yusuf Umar Hanafi
NRP 07211640000006

Dosen Pembimbing
Dr. Supeno Mardi Susiki Nugroho, ST., MT.
Reza Fuad Rachmadi, ST., MT., Ph.D

DEPARTEMEN TEKNIK KOMPUTER
Fakultas Teknologi Elektro dan Informatika Cerdas
Institut Teknologi Sepuluh Nopember
Surabaya 2020

TUGAS AKHIR – EC184801

DETEKSI PENGGUNAAN HELM PADA PENGENDARA BERMOTOR BERBASIS *DEEP LEARNING*

Yusuf Umar Hanafi
NRP 07211640000006

Dosen Pembimbing
Dr. Supeno Mardi Susiki Nugroho, ST., MT.
Reza Fuad Rachmadi, ST., MT., Ph.D

DEPARTEMEN TEKNIK KOMPUTER
Fakultas Teknologi Elektro dan Informatika Cerdas
Institut Teknologi Sepuluh Nopember
Surabaya 2020

[Halaman ini sengaja dikosongkan].

Institut
Teknologi
Sepuluh Nopember

FINAL PROJECT – EC184801

HELMET USE DETECTION FOR MOTORCYCLE RIDER BASED ON DEEP LEARNING

Yusuf Umar Hanafi
NRP 07211640000006

Advisor
Dr. Supeno Mardi Susiki Nugroho, ST., MT.
Reza Fuad Rachmadi, ST., MT., Ph.D

Departement of Computer Engineering
Faculty of Intelligent Electrical and Information Technology
Sepuluh Nopember Institute of Technology
Surabaya 2020

[Halaman ini sengaja dikosongkan].

PERNYATAAN KEASLIAN
TUGAS AKHIR

Dengan ini saya menyatakan bahwa isi sebagian maupun keseluruhan Tugas Akhir saya dengan judul **"Deteksi Penggunaan Helm pada Pengendara Bermotor berbasis *Deep Learning*"** adalah benar-benar hasil karya intelektual sendiri, diselesaikan tanpa menggunakan bahan-bahan yang tidak diizinkan dan bukan merupakan karya orang lain yang saya akui sebagai karya sendiri.

Semua referensi yang dikutip maupun dirujuk telah ditulis secara lengkap pada daftar pustaka.

Apabila ternyata pernyataan ini tidak benar, saya bersedia menerima sanksi sesuai dengan peraturan yang berlaku.

Surabaya, Juli 2020

Yusuf Umar Hanafi
NRP. 07211640000006

LEMBAR PENGESAHAN

Deteksi Penggunaan Helm Pada Pengendara Bermotor Berbasis *Deep Learning*

Tugas Akhir ini disusun untuk memenuhi salah satu syarat memperoleh gelar Sarjana Teknik di Institut Teknologi Sepuluh Nopember Surabaya

Oleh: Yusuf Umar Hanafi (NRP: 07211640000006)

Tanggal Ujian : 06 Juli 2020 Periode Wisuda : September 2020

Disetujui oleh:

Dr. Supeno Mardi Susiki N, ST., MT. (Pembimbing I)
NIP: 197003131995121001

Reza Fuad Rachmadi, ST., MT., Ph.D (Pembimbing II)
NIP: 198504032012121001

Dr. Surya Sumpeno, ST., M.Sc. (Penguji I)
NIP: 196906131997021003

Dr. Adhi Dharma Wibawa, ST., MT. (Penguji II)
NIP. 197605052008121003

Susi Juniastuti, ST., M.Eng. (Penguji III)
NIP. 196506181999032001

Mengetahui
Kepala Departemen Teknik Komputer

Dr. Supeno Mardi Susiki Nugroho, ST., MT.
NIP. 197003131995121001

ABSTRAK

Nama Mahasiswa	:	Yusuf Umar Hanafi
Judul Tugas Akhir	:	Deteksi Penggunaan Helm pada Pengendara Bermotor berbasis *Deep Learning*
Pembimbing	:	1. Dr. Supeno Mardi Susiki Nugroho, S.T., M.T. 2. Reza Fuad Rachmadi, S.T., M.T., Ph.D

Bedasarkan data dari Badan Pusat Statistik (BPS), jumlah korban meninggal dalam kejadian kecelakaan di Indonesia sebanyak 30.568 jiwa pada tahun 2017. Pertumbuhan jumlah korban jiwa setiap tahun terus bertambah dengan persentase 3,72% [1]. Tingginya jumlah korban meninggal dunia diikuti dengan tingginya jenis pelanggaran yang sering dilanggar salah satunya tidak menggunakan helm [2]. Maka dari itu, deteksi penggunaan helm pada pengendara bermotor penting untuk mengurangi jumlah pelanggaran tidak menggunakan helm guna mengantisipasi adanya korban jiwa saat kecelakaan lalu lintas. Sistem ini memanfaatkan kamera IP sebagai alat deteksi. Hasil tangkapan kamera digunakan untuk mendeteksi penggunaan helm menggunakan *Deep Learning*, yang kemudian mendeteksi pengendara bermotor yang tidak menggunakan helm. Dengan adanya sistem pendeteksi penggunaan helm pada pengendara bermotor, diharapkan dapat mendeteksi adanya pelanggaran tidak menggunakan helm yang nantinya akan dikembangkan lebih lanjut menjadi sistem *e-Tilang*.

Kata Kunci: *You Only Look Once* (YOLO), Lalu Lintas, IP Camera

Halaman ini sengaja dikosongkan

ABSTRACT

Name	:	Yusuf Umar Hanafi
Title	:	*Helmet Use Detection for Motorcycle Rider based on Deep Learning*
Advisors	:	1. Dr. Supeno Mardi Susiki Nugroho, S.T., M.T. 2. Reza Fuad Rachmadi, S.T., M.T., Ph.D

Based on data from the Central Statistics Agency (BPS), the death rate in accidents in Indonesia was 30,568 in 2017. Growth in the number of deaths every year continues to increase with a percentage of 3.72% [1]. The high number of deaths was followed by high types of violations that were often violated, one of which was not wearing a helmet [2]. Therefore, detection of helmet use in motorized motorists is important to reduce the number of violations that do not use helmets to anticipate deaths during traffic accidents. This system uses IP cameras to capture. Camera footage is used to detect helmet use using Deep Learning, which then detects motorized motorists who don't use helmets. With the presence of a helmet detection system for motorists, it is expected to detect violations that do not use helmets which will later be further developed into e-tilang systems.

Keywords: You Only Look Once (YOLO), Traffic, IP Camera

Halaman ini sengaja dikosongkan

KATA PENGANTAR

Puji dan syukur kehadirat Allah Swt. atas segala limpahan berkah, rahmat, serta hidayah-Nya, penulis dapat menyelesaikan penelitian ini dengan judul **Deteksi Penggunaan Helm pada Pengendara Bermotor berbasis *Deep Learning***.

Penelitian ini disusun dalam rangka pemenuhan bidang riset di Departemen Teknik Komputer, serta digunakan sebagai persyaratan menyelesaikan pendidikan S1. Penelitian ini dapat terselesaikan tidak lepas dari bantuan berbagai pihak. Oleh karena itu, penulis mengucapkan terima kasih kepada:

1. Keluarga, Ibu, Bapak dan Kakak-Kakak tercinta yang telah memberikan dorongan spiritual dan material dalam penyelesaian buku penelitian ini.
2. Bapak Dr. Supeno Mardi Susiki Nugroho, ST., MT. selaku Kepala Departemen Teknik Komputer, Fakultas Teknologi Elektro dan Informatika Cerdas (FTEIC), Institut Teknologi Sepuluh Nopember.
3. Bapak Dr. Supeno Mardi Susiki Nugroho, ST., MT. selaku dosen pembimbing I dan Bapak Reza Fuad Rachmadi, S.T., M.T., Ph.D selaku dosen pembimbing II yang selalu memberikan arahan selama mengerjakan penelitian tugas akhir ini.
4. Bapak-ibu dosen pengajar Departemen Teknik Komputer, atas pengajaran, bimbingan, serta perhatian yang diberikan kepada penulis selama ini.
5. Secara khusus penulis mengucapkan terimakasih kepada Dinas Perhubungan Kota Surabaya khususnya *Surabaya Intelligent Transport System* (SITS) yang telah memberi izin dan bimbingan guna pengambilan data lalu lintas Kota Surabaya.
6. Seluruh teman-teman dari angkatan e56, Teknik Komputer, Laboratorium B401 Komputasi Multimedia, dan B201 Telematika Teknik Komputer ITS.
7. Aisyah Alifatul Zahidah Rohmah (Mahasiswa S1 Teknik Kimia ITS 2017) yang telah memberi dorongan dan motivasi selama di Surabaya.
8. Rekan seperjuangan IKAMALA ITS (Ikatan Mahasiswa Lamongan ITS) yang telah menemani perjuangan di ITS.

9. Rekan-rekan Badan Eksekutif Mahasiswa (BEM ITS) yang sudah berjuang bersama, membimbing dan berbagi ilmu manajerial selama di ITS.
10. Rekan-rekan Komunitas SurabayaDev, LamonganDev dan Dilo Surabaya yang menjadi tempat berbagi dan menambah ilmu.

Kesempurnaan hanya milik Allah SWT, untuk itu penulis memohon segenap kritik dan saran yang membangun. Semoga penelitian ini dapat memberikan manfaat bagi kita semua. Amin.

Surabaya, Juli 2020

Yusuf Umar Hanafi

DAFTAR ISI

Abstrak **i**

Abstract **iii**

KATA PENGANTAR **v**

DAFTAR ISI **vii**

DAFTAR GAMBAR **xi**

DAFTAR TABEL **xiii**

1 PENDAHULUAN **1**
1.1 Latar belakang 1
1.2 Permasalahan 2
1.3 Tujuan 2
1.4 Batasan masalah 2
1.5 Sistematika Penulisan 3

2 TINJAUAN PUSTAKA **5**
2.1 UU Lalu Lintas dan Angkutan Jalan 5
2.1.1 Sepeda Motor 5
2.1.2 Aturan Penggunaan Helm 5
2.2 Kamera IP 6
2.3 *Machine Learning* 6
2.3.1 *Supervised Learning* 8
2.3.2 *Unsupervised Learning* 8
2.3.3 *Reinforcement Learning* 9
2.4 *Deep Learning* 9
2.5 *Convolutional Neural Network* (CNN) 9
2.5.1 *Convolutional Layer* 10
2.5.2 *Subsampling Layer* 11
2.5.3 *Fully-Connected Layer* 12
2.6 Visi Komputer 12
2.7 Pengolahan Citra 13
2.7.1 Citra Digital 13

2.8 *You Only Look Once* (YOLO) 14
2.8.1 *Feature Extractor* 18
2.8.2 *Multilabel Classification* 19
2.9 Metode Pengujian 20
2.9.1 *Recall* 20
2.9.2 *Precision* 21
2.9.3 *Intersection over Union* (IoU) 21
2.9.4 *mean Average Precision* (mAP) 22
2.9.5 *Confusion Matrix* 23

3 DESAIN DAN IMPLEMENTASI SISTEM 25
3.1 Desain Sistem 26
3.2 Alur Kerja 26
3.3 Akuisisi Data 26
3.3.1 Lokasi Akusisi Data 27
3.3.2 Posisi Kamera IP 27
3.3.3 Rekaman pada Kamera IP 31
3.4 Pelabelan Objek 31
3.5 Proses Melatih Himpunan Data 33
3.6 Pengembangan Sistem Deteksi 36
3.6.1 Proses *Input* 36
3.6.2 *Feature Extraction* 37
3.6.3 *Bounding Box* 37
3.6.4 Deteksi Objek 37
3.6.5 Pengembangan Deteksi Pelanggaran 38
3.7 Analisa Peforma 40

4 PENGUJIAN DAN ANALISIS 43
4.1 Pengujian Peforma berdasarkan Lokasi 43
4.1.1 Pengujian pada Lokasi Jembatan Surabaya . 44
4.1.2 Pengujian pada Lokasi Kedung Cowek 46
4.1.3 Pengujian pada Lokasi Jalan Pandegiling . . 48
4.2 Pengujian Peforma berdasarkan Kondisi 49
4.2.1 Kondisi Kecerahan Tinggi 51
4.2.2 Kondisi Hujan 52
4.3 Pengujian Deteksi berdasarkan Objek Pelanggar . . 53
4.3.1 Pelanggar Kondisi Normal 54
4.3.2 Pelanggar Menggunakan Masker 54

4.3.3 Pelanggar Menggunakan *Hoodie* 55
4.3.4 Pelanggar Menggunakan Jilbab 57
4.3.5 Pelanggar Menggunakan Topi 59
4.3.6 Pelanggar Menggunakan Jas Hujan 60
4.3.7 Pelanggaran pada Penumpang 60
4.3.8 Pelanggaran pada Anak-Anak 61
4.4 Pengujian Sistem 62

5 PENUTUP **65**
5.1 Kesimpulan 65
5.2 Saran 66

DAFTAR PUSTAKA **67**

LAMPIRAN **69**

Biografi Penulis **71**

Halaman ini sengaja dikosongkan

DAFTAR GAMBAR

2.1 Jenis Kamera yang Digunakan 8
2.2 Deep Learning dengan 4 lapisan 10
2.3 Proses Konvolusi pada CNN 10
2.4 Ilustrasi Proses Konvolusi 11
2.5 Ilustrasi Operasi *Max Pooling* 12
2.8 Sistem Deteksi pada YOLO 14
2.6 Macam-macam citra digital beserta reprensentasi dalam bentuk matriks 15
2.7 Arsitektur YOLO 16
2.9 *Bounding Box* pada YOLOv3 17
2.10 Proses Deteksi pada YOLO 18
2.11 Susunan Lapisan pada *Darknet53* 19
2.12 Susunan Lapisan pada model YOLOv3-*tiny* 20
2.13 Ilustrasi Representasi *desire output* pada *Multilabel Classification* 20
2.14 Ilustrasi Predicted dan Ground-Truth Bounding Box pada *Intersection over Union* 22
2.15 Ilustrasi Persamaan *Intersection over Union* 23

3.1 Bagan Umum Metodologi Sistem 25
3.2 Posisi IP Camera 27
3.3 Peta Lokasi Kamera yang Berada di Jembatan Surabaya 28
3.6 Kondisi Rekaman IP Camera Jembatan Surabaya . . 28
3.4 Peta Lokasi Kamera yang Berada di Pandegiling . . 29
3.7 Kondisi Rekaman IP Camera Jalan Pandegiling . . . 29
3.5 Peta Lokasi Kamera yang Berada di Kedung Cowek 30
3.8 Kondisi Rekaman IP Camera Jalan Kedung Cowek . 30
3.9 Data Video yang sudah diolah menjadi Gambar . . . 32
3.10 Skenario Pelabelan Objek 33
3.11 Susunan Lapisan pada *Darknet53* 35
3.12 Susunan Lapisan pada model YOLOv3-*tiny* 36
3.13 *Bounding Box* pada YOLOv3 38
3.14 Hasil Deteksi yang Ditunjukkan oleh *Bounding Box* 39

3.15 Flowchart Sistem Deteksi Pelanggaran Tidak Menggunakan Helm 40
3.16 Skema Peletakan Garis Deteksi 41
3.17 Proses Deteksi Pelanggaran tidak Menggunakan Helm 42
3.18 Hasil Pengambilan Gambar pada Pelanggar 42
3.19 Alur Analisa Peforma 42

4.1 Posisi Kamera di Lokasi Jembatan Surabaya 45
4.2 Posisi Kamera di Lokasi Kedung Cowek 47
4.3 Kondisi Rekaman IP Camera saat Cuaca Hujan . . . 50
4.4 Kondisi Rekaman IP Camera saat Cuaca Cerah . . . 51
4.5 Deteksi pada Pelanggar Kondisi Normal 55
4.6 Deteksi pada Pelanggar Menggunakan Masker 56
4.7 Deteksi pada Pelanggar Menggunakan *Hoodie* 57
4.8 Deteksi pada Pelanggar Menggunakan Jilbab 58
4.9 Deteksi pada Pelanggar Menggunakan Topi 59
4.10 Deteksi pada Pelanggar Menggunakan Jas Hujan . . 61
4.11 Deteksi Pelanggaran pada Penumpang Helm 62
4.12 Deteksi Pelanggaran pada Anak-Anak 63
4.13 Hasil Deteksi Pelanggaran yang Disimpan 64

1 Grafik *Average Loss* terhadap Iterasi pada Proses Training YOLOv3 69
2 Grafik *Average Loss* terhadap Iterasi pada Proses Training YOLOv3-*tiny* 70
3 Proses *Running Program* Deteksi Penggunaan Helm 70

DAFTAR TABEL

2.1 Spesifikasi Kamera IP 7
2.2 *Confusion Matrix* untuk Dua Kelas 23

3.1 Spesifikasi video pada masing-masing rekaman. . . . 31
3.2 Jumlah anotasi pada masing-masing kelas 32
3.3 Konfigurasi pada Darknet 34
3.4 Konfigurasi pada YOLOv3 dan YOLOv3-*tiny* 34

4.1 Spesifikasi PC yang Digunakan. 43
4.2 Jumlah Himpunan Data Uji pada Masing-Masing Lokasi 44
4.3 Pengujian Peforma pada Lokasi Jembatan Surabaya 46
4.4 Pengujian Peforma pada Lokasi Kedung Cowek . . . 48
4.5 Pengujian Peforma pada Lokasi Jalan Pandegiling . 49
4.6 Jumlah Himpunan Data Uji pada Masing-Masing Kondisi 50
4.7 Pengujian Peforma pada Kondisi Cerah 52
4.8 Pengujian Peforma pada Kondisi Hujan 53

Halaman ini sengaja dikosongkan

BAB 1
PENDAHULUAN

Penelitian ini di latar belakangi oleh berbagai kondisi yang menjadi acuan. Selain itu juga terdapat beberapa permasalahan yang akan dijawab sebagai luaran dari penelitian.

1.1 Latar belakang

Bedasarkan data dari Badan Pusat Statistik jumlah korban meninggal dalam kejadian kecelakaan di Indonesia sebanyak 30.568 jiwa pada tahun 2017. Pertumbuhan jumlah korban jiwa setiap tahun terus bertambah dengan persentase 3,72% [1]. Tingginya jumlah korban meninggal dunia diikuti dengan tingginya jenis pelanggaran yang sering dilanggar salah satunya tidak menggunakan helm [2]. Berbagai upaya telah dilakukan untuk menekan jumlah pelanggaran lalu lintas terutama pelanggaran tidak menggunakan helm seperti mengadakan sosialisasi dan razia pada jalan raya. Solusi tersebut mempunyai kelemahan yaitu keterbatasan Sumber Daya Manusia untuk disebar diseluruh titik jalan raya guna menindak pelanggaran lalu lintas.

Teknologi Visi Komputer merupakan bidang kecerdasan buatan atau Artificial Intelligence yang melatih komputer untuk menafsirkan dan memahami sesuatu secara visual. Visi Komputer memanfaatkan gambar digital melalui kamera sebagai masukan serta jaringan neural untuk proses mengidentifikasi dan mengklasifikasikan objek dan kemudian mendeteksi apa yang dilihat oleh komputer. Saat ini, Visi Komputer telah banyak diimplementasikan dalam berbagai bidang seperti kesehatan, sosial, administrasi sampai transportasi.

Dengan adanya penelitian ini, diharapkan hasil deteksi pelanggaran tidak menggunakan helm ini dapat diolah dan dikembangkan untuk mengatasi pelanggaran lalu lintas menggunakan sistem tilang elektronik.

1.2 Permasalahan

Dari permasalahan tersebut, maka masalah yang dapat diambil adalah deteksi pelanggaran tidak menggunakan helm yang tidak terdeteksi secara menyeluruh. Oleh karena itu, diperlukan sistem untuk mendeteksi pengendara yang tidak menggunakan helm pada pengendara bermotor untuk dikembangkan lagi dalam upaya mendeteksi pelanggar yang tidak menggunakan helm.

1.3 Tujuan

Tujuan dari tugas akhir ini adalah sebagai berikut:

1. Membuat sebuah sistem pendeteksi penggunaan helm pada pengendara bermotor.
2. Mengumpulkan data pengendara yang menggunakan helm dan pengendara yang tidak menggunakan helm berupa gambar capture.

1.4 Batasan masalah

Untuk memfokuskan permasalahan yang diangkat maka dilakukan pembatasan masalah. Batasan-batasan masalah tersebut di antaranya adalah:

1. Metode yang digunakan untuk proses deteksi penggunaan helm adalah *You Only Look Once* (YOLO) yang terdiri dari YOLOv3 dan YOLOv3-*tiny*.
2. Data input yang digunakan adalah data video rekaman Kamera IP lalu lintas milik Dinas Perhubungan Kota Surabaya pada pagi, siang dan sore.
3. Jenis kendaraan yang dideteksi penggunaan helm hanya sepeda motor.
4. Hasil deteksi berupa sistem perhitungan dan pengambilan gambar pengendara bermotor yang tidak menggunakan helm.
5. Sistem hanya mendeteksi pengendara bermotor yang menggunakan helm atau tidak menggunakan helm.
6. Pengujian peforma dengan cara mengukur nilai *mean Average Precision* dan deteksi pada objek pelanggar yang berbeda-beda

1.5 Sistematika Penulisan

Laporan penelitian tugas akhir ini tersusun dalam sistematika dan terstruktur sehingga mudah dipahami dan dipelajari oleh pembaca maupun seseorang yang ingin melanjutkan penelitian ini. Alur sistematika penulisan laporan penelitian ini yaitu:

1. BAB I Pendahuluan
   Bab ini berisi uraian tentang latar belakang permasalahan, penegasan dan alasan pemilihan judul, sistematika laporan, tujuan, dan metodologi penelitian.
2. BAB II Tinjauan Pustaka
   Bab ini berisi tentang uraian secara sistematis teori-teori yang berhubungan dengan permasalahan yang dibahas pada penelitian ini. Teori-teori ini digunakan sebagai dasar dalam penelitian, yaitu informasi terkait *Deep Learning, Convolutional Neural Network* (CNN), YOLOv3 dan teori-teori penunjang lainya.
3. BAB III Desain dan Implementasi Sistem
   Bab ini berisi tentang penjelasan-penjelasan terkait eksperimen yang akan dilakukan, langkah-langkah pengambilan data video dan proses deteksi pelanggaran tidak menggunakan helm, serta analisis performa dari sistem. Guna mendukung itu digunakanlah blok diagram atau *workflow* agar sistem yang akan dibuat dapat terlihat dan mudah dibaca untuk implementasi pada pelaksanaan tugas akhir.
4. BAB IV Pengujian dan Analisis
   Bab ini menjelaskan tentang hasil serta analisis yang didapatkan dari pengujian yang dilakukan mulai dari hasil pengujian mean Average Precision (mAP), Intersection over Union (IoU), Confusion Matrix serta rekomendasi penerapan sistem.
5. BAB V Penutup
   Bab ini merupakan penutup yang berisi kesimpulan yang diambil dari penelitian dan pengujian yang telah dilakukan. Saran dan kritik yang membangun untuk pengembangan lebih lanjut juga dituliskan pada bab ini.

Halaman ini sengaja dikosongkan

BAB 2
TINJAUAN PUSTAKA

Demi mendukung penelitian ini, dibutuhkan beberapa teori penunjang sebagai bahan acuan dan referensi. Dengan demikian penelitian ini menjadi lebih terarah.

2.1 UU Lalu Lintas dan Angkutan Jalan

UU Lalu Lintas dan Angkutan Jalan merupakan undang-undang yang mengatur tentang penyelenggara lalu lintas dan Angkutan Jalan seperti perencanaan, pengaturan, pengendalian dan pengawasan. UU tersebut diterbitkan pada tahun 2009 oleh Dewan Perwakilan Rakyat (DPR) yang mempunyai tujuan seperti yang tertuang pada pasal 3 UU. No. 22/2009 Tentang Lalu Lintas dan Angkutan Umum sebagai berikut: [3]

1. *Terwujudnya pelayanan Lalu Lintas dan Angkutan Jalan yang aman, selamat, tertib, lancar, dan terpadu dengan moda angkutan lain untuk mendorong perekonomian nasional, memajukan kesejahteraan umum, memperkukuh persatuan dan kesatuan bangsa, serta mampu menjunjung tinggi martabat bangsa.*
2. *Terwujudnya etika berlalu lintas dan budaya bangsa.*
3. *Terwujudnya penegakan hukum dan kepastian hukum bagi masyarakat.*

2.1.1 Sepeda Motor

Definisi sepeda motor tertuang pada pasal 1 ayat (20) UU. No. 22/2009 Tentang Lalu Lintas dan Angkutan Umum yang berbunyi: *"Sepeda Motor adalah Kendaraan Bermotor beroda dua dengan atau tanpa rumah-rumah dan dengan atau tanpa kereta samping atau Kendaraan Bermotor beroda tiga tanpa rumah-rumah."*

2.1.2 Aturan Penggunaan Helm

Peraturan penggunaan helm pada pengendara sepeda motor diatur pada pasal 57 ayat (1) dan ayat (2) UU. No. 22/2009 Tentang Lalu Lintas dan Angkutan Umum yang berbunyi:

1. *Setiap Kendaraan Bermotor yang dioperasikan di Jalan wajib*

dilengkapi dengan perlengkapan Kendaraan Bermotor.

2. *Perlengkapan sebagaimana dimaksud pada ayat (1) bagi Sepeda Motor berupa helm standar nasional Indonesia.*

Selain itu, aturan penggunaan helm juga diatur pada pasal 106 ayat (8) UU. No. 22/2009 Tentang Lalu Lintas dan Angkutan Umum yang berbunyi:*"Setiap orang yang mengemudikan Sepeda Motor dan Penumpang Sepeda Motor wajib mengenakan helm yang memenuhi standar nasional Indonesia."*. Dari pasal-pasal tersebut, penggunaan helm berlaku pada pengendara dan penumpang yang mengendarai sepeda motor. Apabila melanggar, terdapat hukuman atas pelanggaran tersebut yang diatur dalam pasal 291 ayat(1) dan ayat(2) UU. No. 22/2009 Tentang Lalu Lintas dan Angkutan Umum yang berbunyi:

1. *Setiap orang yang mengemudikan Sepeda Motor tidak mengenakan helm standar nasional Indonesia sebagaimana dimaksud dalam Pasal 106 ayat (8) dipidana dengan pidana kurungan paling lama 1 (satu) bulan atau denda paling banyak Rp250.000,00 (dua ratus lima puluh ribu rupiah).*
2. *Setiap orang yang mengemudikan Sepeda Motor yang membiarkan penumpangnya tidak mengenakan helm sebagaimana dimaksud dalam Pasal 106 ayat (8) dipidana dengan pidana kurungan paling lama 1 (satu) bulan atau denda paling banyak Rp250.000,00 (dua ratus lima puluh ribu rupiah).*

2.2 Kamera IP

Dalam penelitian ini terdapat Kamera IP yang digunakan Hikvision dengan spesifikasi dan bentuk seperti yang terdapat pada Tabel 2.1 dan Gambar 2.1.

2.3 *Machine Learning*

Machine Learning (ML) atau Pembelajaran Mesin merupakan bagian dari *Artificial Intelligence* (AI) yang bertujuan untuk memberi optimalisasi dalam kriteria dengan cara menganalisa sampel data yang terdahulu yang sudah disimpan atau direkam untuk menghasilkan sebuah prediksi. Sehingga manusia tidak perlu mengindentifikasi sebuah proses sepenuhnya, karena dengan Machine Learning, komputer mampu membuat pola untuk membuat keputus-

Tabel 2.1: Spesifikasi Kamera IP

Merk Kamera	Hikvision DS-2CD1021-I	Hikvision DS-2CD1653G0-I(Z)
Image Sensor	1/2.8" progressive scan CMOS	1/2.7" Progressive Scan CMOS
Min. Illumination	Color: 0.01 Lux @(F1.2, AGC ON), 0 Lux with IR	Color: 0.01 Lux @(F1.2; AGC ON), 0.02 Lux @ (F1.6, AGC ON)
Shutter Speed	1/3 s to 1/100, 000 s, support slow shutter	1/3 s to 1/100,000 s
Image Setting	Brightness, saturation, contrast, sharpness are adjustable via web browser or client software	Saturation, brightness, contrast and sharpness are adjustable via client software and web browser
Max. Resolution	1920 × 1080	2560 × 1920
Video Compression	Main stream: H.264+/H.264. Sub stream: H.264/MJPEG	Main stream: H.265/H.264 Sub-stream: H.265/H.264/MJPEG
Video Bit Rate	32 Kbps ∼ 8 Mbps	32 Kbps ∼ 8 Mbps
Main Stream	50Hz: 25fps @(1920 × 1080, 1280 × 960, 1280 × 720) 60Hz: 30fps @(1920 × 1080, 1280 × 960, 1280 × 720)	50Hz:12.5fps (2560 × 1920), 20fps(2560 × 1440), 25fps (2304 × 1296, 1920 × 1080, 1280 × 720) 60Hz: 15fps (2560 × 1920), 20fps(2560 × 1440), 30fps (2304 × 1296, 1920 × 1080, 1280 × 720)
Sub Stream	50Hz: 25fps @(704 × 576, 640 × 480, 352 × 288, 320 × 240) 60Hz: 30fps @(704 × 480, 640 × 480, 352 × 240, 320 × 240)	50Hz: 25fps (640 × 480, 640 × 360, 320 × 240) 60Hz: 30fps (640 × 480, 640 × 360, 320 × 240)
Comm. Interface	1 RJ45 10M/100M Ethernet port	1 RJ45 10M/100M self-adaptive Ethernet port
Penggunaan	Jalan Kedung Cowek	Jalan Pandegiling, Jembatan Surabaya, Jalan Tunjungan

(a) Hikvision DS-2CD4A26FWD-IZ(H)(S).

(b) Hikvision DS-2CD2621G0-I(Z)(S).

Gambar 2.1: Jenis Kamera yang Digunakan.

an.

Machine Learning melakukan pelatihan yang merupakan proses pembelajaran terhadap model data yang sudah terdefinisikan ke beberapa parameter (data pelatihan) yang menghasilkan beberapa pola sehingga komputer dapat melakukan proses klasifikasi berdasarkan pola atau ciri-ciri yang sudah didapatkan dalam proses pelatihan. Kemudian komputer dapat memberikan sebuah prediksi pada data baru selanjutnya berdasarkan hasil pelatihan. *Machine Learning* dapat memberi solusi dalam berbagai permasalahan seperti *Vision* (Visi Komputer), *Speech Recognition* (Pengenalan Suara) dan *Robotics* (Robotika).[4]

2.3.1 *Supervised Learning*

Supervised Learning merupakan metode pembelajaran himpunan data yang akan dilatih untuk diekstrasi polanya sudah memiliki label atau kelas yang menunjukkan klasifikasi[5].

2.3.2 *Unsupervised Learning*

Unsupervised Learning merupakan metode pembelajaran himpunan data yang akan dilatih untuk diekstrasi polanya yang himpunan data tersebut tidak memiliki label atau kelas. Sehingga komputer harus melakukan analisa pola yang dimiliki data tersebut un-

tuk ekstrasi atribut atau sifat-sifat dari data yang ada.

2.3.3 *Reinforcement Learning*

Reinforcement Learning merupakan metode pembelajaran komputer jika memberikan nilai positif maka mendapatkan *reward* dan sebaliknya jika komputer mendapatkan nilai negatif maka mendapatkan *punishment* sehingga komputer diharapkan belajar secara mandiri terus-menerus.

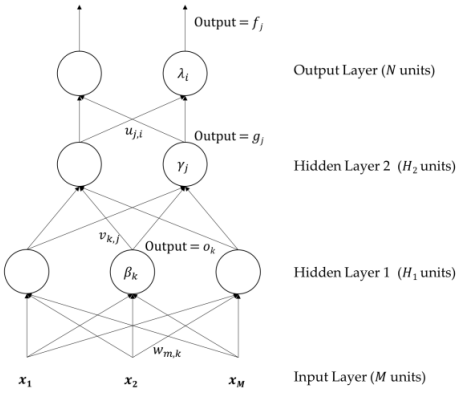
2.4 *Deep Learning*

Deep Learning merupakan *artificial neural network* yang memiliki banyak lapisan dan *synapse weight*. *Deep learning* dapat menemukan relasi tersembunyi atau pola yang rumit antara masukan dan keluaran, yang tidak dapat diselesaikan menggunakan *multilayer perceptron* (3 lapisan). Keuntungan utama *deep learning* yaitu mampu merubah data dari *non-linearly separable* menjadi *linearly separable* melalui serangkaian transformasi (lapisan tersembunyi). Selain itu, *deep learning* juga mampu mencari *decision boundary* yang berbentuk non-linier, serta mengsimulasikan interaksi non-linier antar fitur. Jadi, input ditransformasikan secara non-linier sampai akhirnya pada output, berbentuk distribusi *class-assignment*. Pada pelatihan, *deep learning* menggunakan *back propagation*. Berikut contoh *Deep Learning* dengan 4 lapisan terdapat pada Gambar 2.2[6].

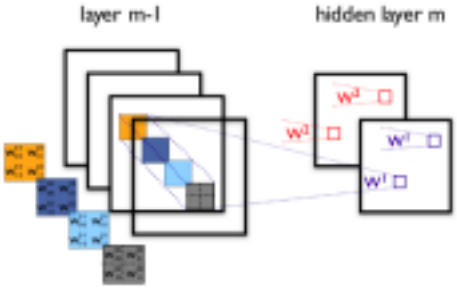
2.5 *Convolutional Neural Network* (CNN)

Convolutional Neural Network (CNN) merupakan cabang dari *Multilayer Perceptron* (MLP) yang digunakan untuk mengolah data dua dimensi. CNN memiliki kedalaman jaringan yang tinggi sehingga CNN termasuk dalam jenis Deep Neural Network. Perbedaan CNN dengan MLP terdapat pada neuron. Pada MLP setiap neuron hanya berukuran satu dimensi, sedangkan CNN setiap neuronnya berukuran dua dimensi. Pada CNN, operasi linier menggunakan operasi konvolusi. Bobot pada CNN berbentuk empat dimensi seperti pada Gambar 2.3. Persamaan 2.1 untuk dimensi bobot pada CNN[7].

$$neuron\ input \times neuron\ output \times tinggi \times lebar \tag{2.1}$$



Gambar 2.2: *Deep Learning* dengan 4 lapisan[6]

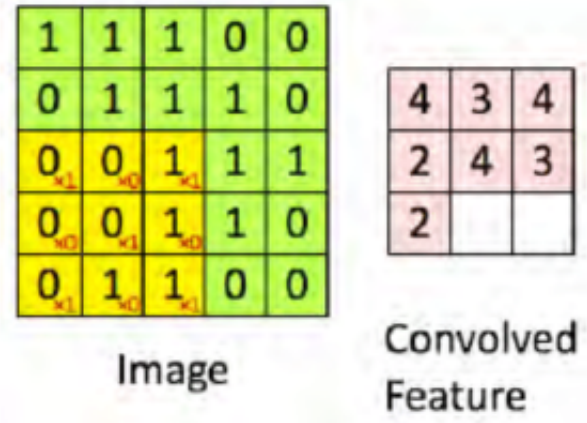


Gambar 2.3: Proses Konvolusi pada CNN[7]

2.5.1 *Convolutional Layer*

Convolutional Layer merupakan proses utama dalam CNN yang bertugas melakukan operasi konvolusi pada keluaran dari lapisan sebelumnya. Konvolusi yaitu mengaplikasikan sebuah fungsi pada keluaran fungsi lain secara berulang. Seperti pada Gambar 2.4, konvolusi mengaplikasikan kernel yang berwarna kuning pada citra disemua offset. Kotak hijau secara keseluruhan adalah citra yang

akan dikonvolusi. Kernel bergerak dari sudut kiri atas ke kanan bawah. Sehingga hasil konvolusi dari citra tersebut dapat dilihat pada gambar bagian kanan. Tujuan dilakukannya konvolusi pada



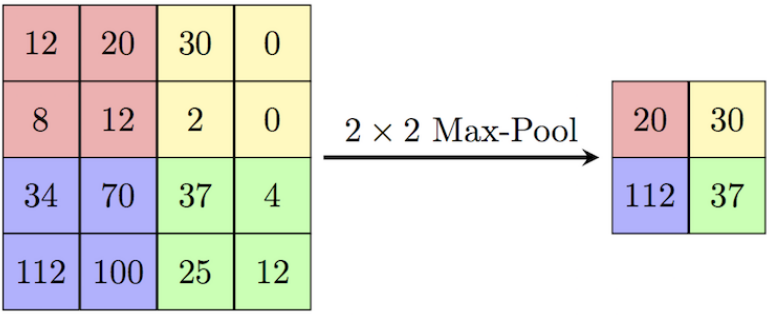
Gambar 2.4: Ilustrasi Proses Konvolusi[7]

data citra adalah untuk mengekstraksi fitur dari citra input. Konvolusi akan menghasilkan transformasi linier dari data input sesuai informasi spasial pada data. Bobot pada layer tersebut menspesifikasikan kernel konvolusi yang digunakan, sehingga kernel konvolusi dapat dilatih berdasarkan input pada CNN.

2.5.2 *Subsampling Layer*

Subsampling Layer adalah proses mereduksi ukuran sebuah data citra. Tujuannya untuk meningkatkan invariansi posisi dari fitur. Dalam CNN, metode *subsampling* yang digunakan adalah *max pooling*. Pada ilustrasi pada Gambar 2.5, *Max pooling* membagi keluaran dari *convolution layer* menjadi beberapa *grid* kecil lalu mengambil nilai maksimal dari setiap *grid* untuk menyusun matriks citra yang telah direduksi. *Grid* yang berwarna merah, hijau, kuning dan biru merupakan kelompok grid yang akan dipilih nilai maksimumnya. Sehingga hasil dari proses tersebut dapat dilihat pada kumpulan grid disebelah kanannya. Proses tersebut memastikan

fitur yang didapatkan akan sama meskipun objek citra mengalami translasi (pergeseran). Penggunaan *pooling layer* pada CNN hanya bertujuan untuk mereduksi ukuran citra sehingga dapat dengan mudah digantikan dengan sebuah *convolution layer* dengan *stride* yang sama dengan pooling layer yang bersangkutan.



Gambar 2.5: Ilustrasi Operasi *Max Pooling*[7]

2.5.3 *Fully-Connected Layer*

Fully-Connected Layer bertujuan untuk melakukan transformasi pada dimensi data agar data dapat diklasifikasikan secara linier. Setiap neuron pada convolution layer perlu ditransformasi menjadi data satu dimensi terlebih dahulu sebelum dapat dimasukkan ke dalam sebuah *fully-connected layer*. Karena hal tersebut menyebabkan data kehilangan informasi spasialnya dan tidak reversibel, *fully-connected layer* hanya dapat diimplementasikan diakhir jaringan. Convolution layer dengan ukuran kernel 1×1 melakukan fungsi yang sama dengan sebuah *fully-connected layer* namun dengan tetap mempertahankan karakter spasial dari data.

2.6 Visi Komputer

Visi Komputer adalah cabang Artificial Intelligent (AI) yang mencakup proses analisa citra dan video. Visi komputer mengimplementasikan beberapa kemampuan visual manusia yang diteruskan menuju otak seperti deteksi benda, pengenalan wajah dan mengenali bahaya.

Pada visi komputer, *Deep Learning* sering digunakan untuk pengenalan dan deteksi objek. Proses *Deep Learning* pada visi komputer memanfaatkan piksel pada citra untuk ekstrasi pola atau atribut dari citra yang ingin dideteksi. Akan tetapi, hal tersebut mengakibatkan sistem komputasi menjadi lama karena pada suatu citra mengandung ribuan piksel. Sehingga banyak arsitektur visi komputer membuat standar ukuran, jadi citra tersebut harus dipotong atau diperkecil untuk mempercepat proses komputasi.[5]

2.7 Pengolahan Citra

Pengolahan Citra merupakan teknik dalam pemrosesan gambar dengan input berupa citra dua dimensi yang bertujuan untuk menyempurnakan citra atau mendapatkan informasi yang berguna untuk diolah menjadi beberapa keputusan. Dalam operasi pemrosesan citra, operasi yang sering dilakukan dalam gambar *grayscale*. Gambar *grayscale* didapatkan dari pemrosesan gambar berwarna yang didekomposisi menjadi komponen merah (R), hijau (G) dan biru (B) yang diproses secara independen sebagai gambar *grayscale*. Pengolahan citra terbagi menjadi dalam 3 tingkatan[8]:

1. *Low-Level Image Processing*
   Low-Level Image Processing merupakan operasi sederhana dalam pengolahan gambar masukan dan keluaran berupa gambar. Contoh: *contrast enchancement dan noise reduction.*
2. *Mid-Level Image Processing*
   Mid-Level Image Processing merupakan operasi pengolahan gambar yang melibatkan ekstrasi atribut dari gambar masukan. Contoh: *edges, contours dan regions.*
3. *High-Level Image Processing*
   High-Level Image Processing merupakan merupakan kategori yang melibatkan pemrosesan gambar kompleks yang terkait dengan analisis dan interpretasi konten dalam sebuah keadaan untuk pengambilan keputusan.

2.7.1 Citra Digital

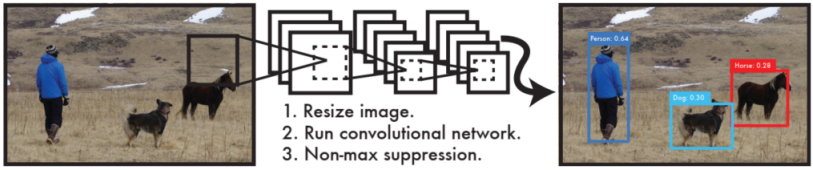
Citra Digital merupakan fungsi dua dimensi *f(x,y)* yang merupakan proyeksi dari bentuk tiga dimensi kedalam bentuk dua dimensi. x dan y merupakan lokasi elemen gambar atau piksel yang beri-

sikan nilai. Ketika nilai *x,y* dan intensitasnya berupa diskrit, maka gambar tersebut dapat dikategorikan sebagai citra digital. Secara matematis, digital image adalah representasi matriks dari gambar dua dimensi menggunakan piksel. Setiap piksel diwakili oleh nilai numerik. Untuk gambar *grayscale*, hanya memiliki satu nilai dengan kisaran antara 0-255.Pada Gambar 2.6, untuk gambar yang berwarna, memiliki tiga nilai yang mewakili merah (R), hijau (G) dan biru (B) yang masing-masing memiliki kisaran nilai yang sama antara 0-255. Jika suatu gambar hanya memiliki dua intensitas, gambar tersebut dikenal sebagai citra biner[8].

2.8 *You Only Look Once* (YOLO)

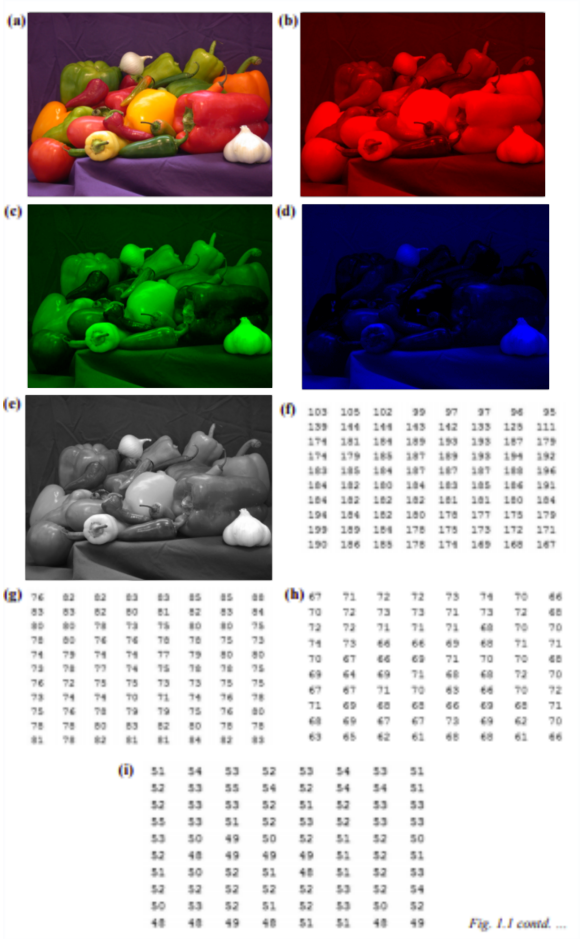
You Only Look Once (YOLO) merupakan sistem deteksi yang berbasis *Convolutional Neural Network*. Pada Gambar 2.7, Dalam arsitektur YOLO terdiri dari 24 convolutional layer yang berfungsi sebagai mendapatkan fitur dari citra. Kemudian diikuti 2 connected layer yang berfungsi sebagai memprediksi probabilitas dan koordinat[9]. Seperti pada Gambar 2.8, terdapat tiga langkah deteksi objek menggunakan YOLO seperti berikut:

1. Mengubah ukuran dimensi masukan citra menjadi 448 × 448.
2. Menjalankan *single convolutional network* pada citra.
3. Melakukan threshold pada hasil deteksi berdasarkan nilai *confidence* yang didapatkan oleh model

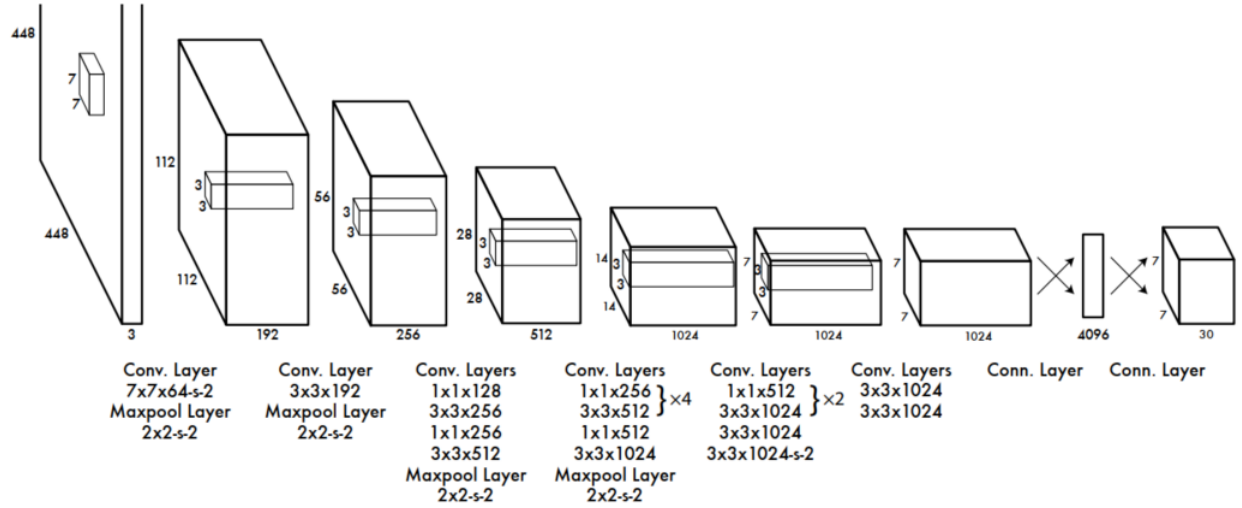


Gambar 2.8: Sistem Deteksi pada YOLO[9]

Pada proses pertama, YOLO mendeteksi objek menggunakan *unified detection* yang menyatukan antara komponen deteksi objek kedalam jaringan syarat tunggal. Desain YOLO memungkinkan pelatihan end-to-end dan real-time speed dengan mempertahankan

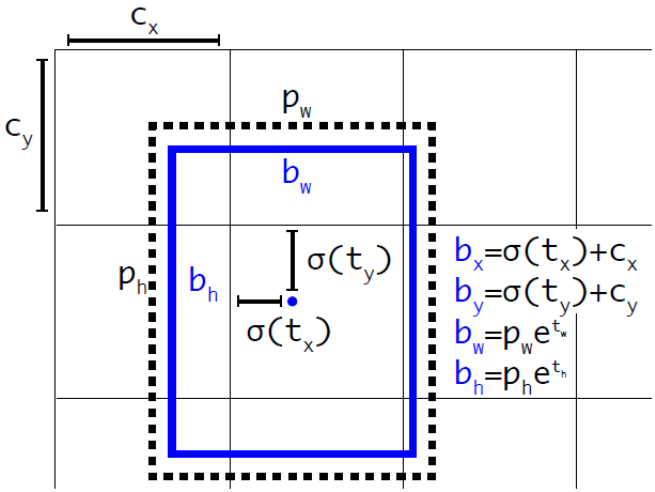


Gambar 2.6: (a) Gambar berwarna, (b) Komponen merah dalam gambar berwarna, (c) Komponen hijau dalam gambar bewarna, (d) Komponen biru dalam gambar bewarna, (e) Gambar bewarna dikonversi dalam *8-bit grayscale*, (f) Matriks dari gambar (b), (g) Matriks dari gambar (c), (h) Matriks dari gambar (d), (i) Matriks dari gambar (e)[8].



Gambar 2.7: Arsitektur YOLO[9].

rata-rata presisi yang tinggi. Sistem pada YOLO membagi gambar masukan kedalam *grid* $S \times S$. Jika titik tengah dari sebuah objek terdapat didalam salah satu sel, maka sel *grid* itu bertanggung jawab untuk mendeteksi objek tersebut. Setiap sel kota memprediksi *bounding box* B dan nilai *confidence* untuk setiap kotak. Nilai *confidence* merepresentasikan keakuratan model bahwa terdapat objek dalam *bounding box* tersebut. Setiap *bounding box* memiliki 5 parameter prediksi yaitu x, y, w, h, seperti pada Gambar 2.9 dan *confidence*. Koordinat (x,y) merupakan pusat dari kotak rela-



Gambar 2.9: *Bounding Box* pada YOLOv3[10]

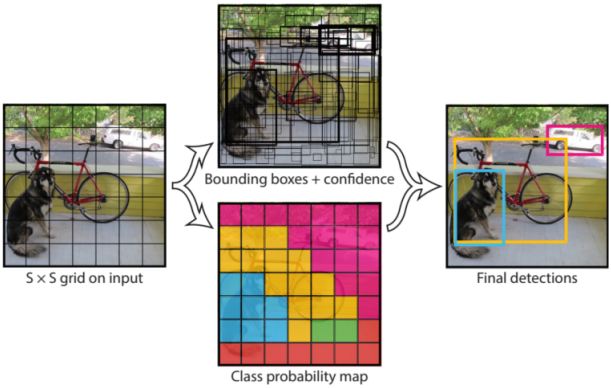
tif ke gambar dan *confidence* merupakan *Intersection over Union* (IoU) antara *predicted box* dengan *ground-truth box*. Setiap sel grid memprediksi probabilitas kelas C. Setiap sel *grid* memprediksi nilai probabilitas pada kelas C. Probabilitas tersebut dikondisikan berdasarkan sel *grid* yang memuat objek. Sehingga hanya terdapat satu kelas probabilitas yang terdeteksi disetiap sel *grid* tanpa memperhitungkan jumlah *bounding box* B. Saat deteksi, probabilitas kelas dikalikan nilai confidence sesuai persamaan 2.2 atau disederhanakan

menjadi seperti persamaan 2.3.

$$Pr(Class_i|Object) \times Pr(Object) \times_{truth} IOU_{pred} \tag{2.2}$$

$$Pr(Class_i) \times_{truth} IOU_{pred} \tag{2.3}$$

Dari persamaan tersebut, didapatkan nilai *confidence* dari kelas spesifik. Nilai ini merepresentasikan probabilitas kelas yang muncul didalam kotak dan seberapa baik kotak yang diprediksi akurat dengan objek. Seperti pada ilustrasi pada Gambar 2.10, YOLO mendeteksi model sebagai regresi. Hal ini membagi gambar menjadi grid dan secara bersamaan memprediksi bounding box dan confidence pada bounding box tersebut dan kelas probabilitas.



Gambar 2.10: Proses Deteksi pada YOLO[9]

2.8.1 *Feature Extractor*

Pada YOLOv2 menggunakan Darknet19 sebagai *feature extractor*. Pada penelitian ini menggunakan YOLOv3 sehingga feature extractor yang digunakan yaitu Darknet53 yang menggunakan 53 lapisan seperti ilustrasi pada gambar 2.11[10]. Untuk model yang kedua yaitu YOLOv3-*tiny* menggunakan feature extractor yang menggunakan susunan lapisan seperti yang diilustrasikan pada gambar

	Type	Filters	Size	Output
	Convolutional	32	3 × 3	256 × 256
	Convolutional	64	3 × 3 / 2	128 × 128
1×	Convolutional	32	1 × 1	
	Convolutional	64	3 × 3	
	Residual			128 × 128
	Convolutional	128	3 × 3 / 2	64 × 64
2×	Convolutional	64	1 × 1	
	Convolutional	128	3 × 3	
	Residual			64 × 64
	Convolutional	256	3 × 3 / 2	32 × 32
8×	Convolutional	128	1 × 1	
	Convolutional	256	3 × 3	
	Residual			32 × 32
	Convolutional	512	3 × 3 / 2	16 × 16
8×	Convolutional	256	1 × 1	
	Convolutional	512	3 × 3	
	Residual			16 × 16
	Convolutional	1024	3 × 3 / 2	8 × 8
4×	Convolutional	512	1 × 1	
	Convolutional	1024	3 × 3	
	Residual			8 × 8
	Avgpool		Global	
	Connected		1000	
	Softmax			

Gambar 2.11: Susunan Lapisan pada *Darknet53*[10]

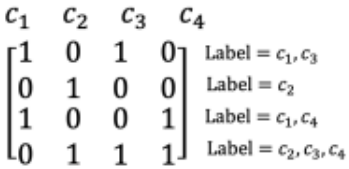
2.12.

2.8.2 *Multilabel Classification*

Multilabel Classification merupakan klasifikasi yang menggunakan sejumlah C neuron untuk merepresentasikan C kelas. Berbeda dengan multiclass classification, pada multilabel classification, layer terakhir diaktivasi dengan fungsi sigmoid. Setiap neuron n_i merepresentasikan probabilitas yang diklasifikasikan sebagai kelas c_i atau tidak seperti pada Gambar 2.13[6].

Layer	Type	Filters	Size/Stride	Input	Output
0	Convolutional	16	3 × 3/1	416 × 416 × 3	416 × 416 × 16
1	Maxpool		2 × 2/2	416 × 416 × 16	208 × 208 × 16
2	Convolutional	32	3 × 3/1	208 × 208 × 16	208 × 208 × 32
3	Maxpool		2 × 2/2	208 × 208 × 32	104 × 104 × 32
4	Convolutional	64	3 × 3/1	104 × 104 × 32	104 × 104 × 64
5	Maxpool		2 × 2/2	104 × 104 × 64	52 × 52 × 64
6	Convolutional	128	3 × 3/1	52 × 52 × 64	52 × 52 × 128
7	Maxpool		2 × 2/2	52 × 52 × 128	26 × 26 × 128
8	Convolutional	256	3 × 3/1	26 × 26 × 128	26 × 26 × 256
9	Maxpool		2 × 2/2	26 × 26 × 256	13 × 13 × 256
10	Convolutional	512	3 × 3/1	13 × 13 × 256	13 × 13 × 512
11	Maxpool		2 × 2/1	13 × 13 × 512	13 × 13 × 512
12	Convolutional	1024	3 × 3/1	13 × 13 × 512	13 × 13 × 1024
13	Convolutional	256	1 × 1/1	13 × 13 × 1024	13 × 13 × 256
14	Convolutional	512	3 × 3/1	13 × 13 × 256	13 × 13 × 512
15	Convolutional	255	1 × 1/1	13 × 13 × 512	13 × 13 × 255
16	YOLO				
17	**Route 13**				
18	Convolutional	128	1 × 1/1	13 × 13 × 256	13 × 13 × 128
19	Up-sampling		2 × 2/1	13 × 13 × 128	26 × 26 × 128
20	**Route 19 8**				
21	Convolutional	256	3 × 3/1	13 × 13 × 384	13 × 13 × 256
22	Convolutional	255	1 × 1/1	13 × 13 × 256	13 × 13 × 256
23	YOLO				

Gambar 2.12: Susunan Lapisan pada model YOLOv3-*tiny*[11].



Gambar 2.13: Ilustrasi Representasi *desire output* pada *Multilabel Classification*[6].

2.9 Metode Pengujian

2.9.1 *Recall*

Recall didefinisikan sebagai rasio dari jumlah total contoh positif yang diklasifikasikan dengan benar dibagi dengan jumlah total contoh positif. High Recall menunjukkan bahwa kelas dikenali de-

ngan benar (FN sedikit). *Recall* dapat dihitung dengan persamaan 2.4[12].

$$Recall = \frac{TP}{TP + FN} \tag{2.4}$$

2.9.2 *Precision*

Nilai presisi didapatkan dengan cara membagi jumlah total contoh positif yang diklasifikasikan dengan benar dengan jumlah total contoh positif yang diprediksi seperti pada persamaan 2.5. Presisi Tinggi menunjukkan contoh berlabel positif memang positif (FP sedikit)[12].

$$Precision = \frac{TP}{TP + FP} \tag{2.5}$$

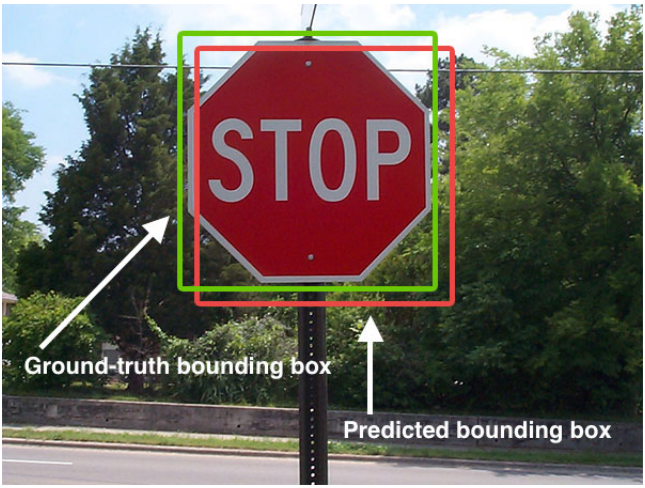
1. *High Recall, Low Precision*: artinya sebagian besar contoh positif dikenali dengan benar (FN rendah) tetapi ada banyak positif palsu.
2. *Low Recall, High Precision*: artinya kehilangan banyak contoh positif (FN tinggi) tetapi yang kami prediksi positif benar-benar positif (FP rendah).

2.9.3 *Intersection over Union* (IoU)

Intersection over Union (IoU) merupakan metrik evaluasi untuk mengukur keakuratan detektor objek pada himpunan data tertentu. IoU dapat digunakan dengan ketentuan sebagai berikut[13]:

1. Memiliki *ground-truth bounding box* pada himpunan data objek
2. Prediksi *bounding box* pada himpunan data objek

Ilustrasi perbandingan *ground-truth bounding box* dan predicted bounding box dari model seperti pada Gambar 2.14. *Intersection over Union* (IoU) merupakan perbandingan antara ground-truth bounding box dengan predicted bounding box pada model sesuai dengan persamaan 2.6. Persamaan tersebut jika diilustrasikan seperti pada



Gambar 2.14: Ilustrasi Predicted dan Ground-Truth Bounding Box pada *Intersection over Union*. *Bounding box* merah merupakan *ground-truth* serta *bounding box* hijau merupakan *predicted model*[13].

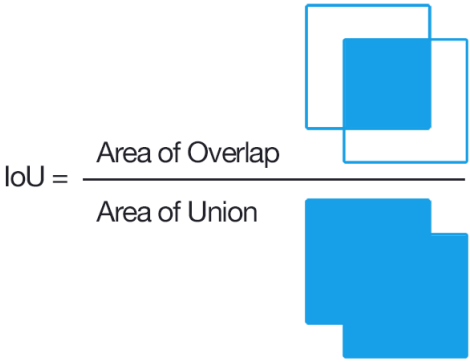
Gambar 2.15.

$$IoU = \frac{A \cap B}{A \cup B} \tag{2.6}$$

2.9.4 *mean Average Precision* (mAP)

mean Average Precision (mAP) merupakan nilai rata-rata dari *Average Precision* (AP) yang membentuk metrik evaluasi untuk mengukur kinerja dari sebuah deteksi objek. Nilai AP didapatkan dari perhitungan *precision* pada persamaan 2.5 dan perhitungan *recall* pada persamaan 2.4 yang selanjutnya dilakukan perhitungan seperti pada persamaan 2.7 dan 2.8[14].

$$AP = \sum (recall_{n+1} - recall_n) \times precision_{interp} \times (recall_{n+1}) \tag{2.7}$$



Gambar 2.15: Ilustrasi Persamaan *Intersection over Union*[13].

$$p_{interp}(r_{n+1}) = \max_{\widetilde{r} \geq r_{n+1}} p(\widetilde{r}) \tag{2.8}$$

2.9.5 *Confusion Matrix*

Confusion Matrix atau *error matrix* adalah tabel yang sering digunakan untuk mendeskripsikan kinerja model klasifikasi (*classifier*) pada beberapa data uji yang sudah diketahui nilainya, sehingga menciptakan visualisasi bagaimana kinerja suatu algoritma dan mengidentifikasi *confusion* antar kelas seperti kesalahan pemberian label. Pada Tabel 2.2 merupakan tabel *confusion matrix*.

Tabel 2.2: *Confusion Matrix* untuk Dua Kelas.

		Aktual	
		Positif	Negatif
Prediksi	Positif	TP	FP
	Negatif	FN	TN

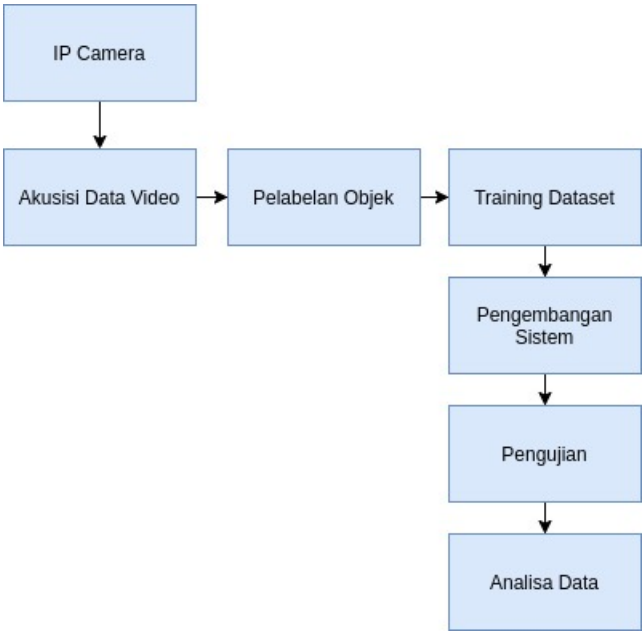
Confusion Matrix merupakan ringkasan hasil prediksi pada permasalahan klasifikasi. Jumlah prediksi yang benar dan salah dikumpulkan dengan nilai-nilai hitung dan dipecah oleh masing-

masing kelas. Sehingga dengan *Confusion Matrix* tidak hanya memberikan informasi kesalahan yang dibuat *classifier* tetapi juga jenis kesalahannya. Dalam *confusion matrix* terdapat beberapa ketentuan sebagai berikut[12]:

1. *Positive* (P): Aktual bernilai positif
2. *Negative* (N): Aktual bernilai negatif
3. *True Positive* (TP): Aktual bernilai positif, dan diprediksi positif juga
4. *True Negative* (TN): Aktual bernilai positif, tetapi diprediksi negatif
5. *False Positive* (FP): Aktual bernilai negatif, tetapi diprediksi negatif
6. *False Negative* (FN): Aktual bernilai negatif, dan diprediksi negatif juga

BAB 3
DESAIN DAN IMPLEMENTASI SISTEM

Penelitian ini dilaksanakan sesuai dengan desain sistem berikut dengan implementasinya. Desain sistem merupakan konsep dari pembuatan dan perancangan infrastruktur dan kemudian diwujudkan dalam bentuk blok-blok alur yang harus dikerjakan. Pada bagian implementasi merupakan pelaksanaan teknis untuk setiap blok pada desain sistem. Gambar 3.1 menunjukkan bagan umum metodologi sistem.



Gambar 3.1: Bagan Umum Metodologi Sistem

3.1 Desain Sistem

Tugas akhir ini merupakan penelitian dalam bidang visi komputer yang bertujuan untuk mendeteksi pelanggaran pengendara bermotor yang tidak menggunakan helm berbasis *Deep Learning*. Sistem deteksi ini memanfaatkan data pelatihan yang diambil dari video rekaman Kamera IP yang terpasang diseluruh jalan Kota Surabaya oleh Dinas Perhubungan Kota Surabaya.

3.2 Alur Kerja

Alur implementasi dalam pengerjaan penelitian ini dibagi menjadi beberapa tahapan berdasarkan metodologi penelitian, yaitu:

1. Akuisisi Data.
2. Pelabelan Objek.
3. Proses Melatih Himpunan Data
4. Pengembangan Sistem
5. Analisa Peforma.

Berdasarkan diagram alur pada Gambar 3.1, proses pertama yaitu akusisi data video. Data video tersebut diambil dari Kamera IP milik Dinas Perhubungan Kota Surabaya yang terpasang diseluruh lalu lintas Kota Surabaya. Tujuan dari akusisi data video ini adalah untuk mengumpulkan himpunan data yang dibutuhkan dan sebagai uji coba sistem deteksi yang telah dibuat. Dalam sistem ini hanya ada 3 objek yang dideteksi yaitu pengendara bermotor, kepala menggunakan helm dan kepala tidak menggunakan helm. Kemudian objek tersebut diberikan label atau anotasi sesuai kelas yang ingin ditambahkan berupa bounding box yang menunjukkan posisi objek kelas tersebut. Setelah didapatkan ratusan himpunan data gambar dilatih menggunakan metode deteksi objek berbasis CNN yaitu *You Only Look Once* (YOLO) menggunakan Darknet sebagai *neural network framework*

3.3 Akuisisi Data

Pada tahap akuisisi data video, data diambil dari IP Camera milik Dinas Perhubungan Kota Surabaya yang terpasang diseluruh lalu lintas Kota Surabaya. Pengambilan data video dilakukan di *Surabaya Intelligent Transport System* (SITS) yang merupakan kantor Dinas Perhubungan khusus untuk pengendalian seluruh Kamera IP

lalu lintas yang tersebar di Kota Surabaya. Kamera IP dipasang disamping lampu lalu lintas pada ketinggian 6 meter menyesuaikan aturan tinggi tiang lampu lalu lintas seperti pada Gambar 3.2.

Gambar 3.2: Posisi Kamera IP.

3.3.1 Lokasi Akusisi Data

Data akusisi video diambil berdasarkan lokasi sering terjadi pelanggaran pengendara bermotor yang tidak menggunakan helm. Setelah survei lapangan, didapatkan beberapa lokasi yang sering terjadi pelanggaran tidak menggunakan helm. Lokasi pertama berada di Jembatan Surabaya yang ditunjukkan penanda biru peta pada Gambar 3.3. Lokasi kedua berada di Jalan Pandegiling seperti yang ditunjukkan penanda biru pada peta pada Gambar 3.4. Lokasi ketiga berada di Kedung Cowek disekitar area SMP Negeri 19 Surabaya seperti yang ditunjukkan penanda biru peta pada Gambar 3.5.

3.3.2 Posisi Kamera IP

Untuk menguji deteksi objek, Kamera IP yang terletak di masing-masing tempat untuk akuisisi data memiliki posisi dan sudut yang berbeda-beda karena dalam kondisi di lapangan. Hal ini dilakukan karena setiap Kamera IP yang tersebar diseluruh lalu lintas Kota Surabaya memiliki kondisi yang berbeda-beda. Sehingga diharapkan sistem yang dibuat mudah implementasikan diberbagai kondisi. Kamera IP yang terletak di Jembatan Surabaya berada

Gambar 3.3: Peta Lokasi Kamera yang Berada di Jembatan Surabaya.

disamping jalan, sehingga gambaran kondisi rekaman Kamera IP seperti Gambar 3.6.

Gambar 3.6: Kondisi Rekaman IP Camera Jembatan Surabaya.

Kemudian untuk Kamera IP yang terletak di Jalan Pandegiling terletak ditengah jalan dengan sudut 60 derajat, sehingga gambaran kondisi rekaman Kamera IP seperti Gambar 3.7.

Gambar 3.4: Peta Lokasi Kamera yang Berada di Pandegiling.

Gambar 3.7: Kondisi Rekaman IP Camera Jalan Pandegiling.

Selanjutnya untuk Kamera IP yang terletak di Jalan Kedung Cowek memiliki sudut 45 derajat, sehingga gambaran kondisi rekaman Kamera IP seperti Gambar 3.8.

Gambar 3.5: Peta Lokasi Kamera yang Berada di Kedung Cowek.

Gambar 3.8: Kondisi Rekaman IP Camera Jalan Kedung Cowek.

3.3.3 Rekaman pada Kamera IP

Rekaman yang diambil pada masing-masing kamera IP yang terdapat pada tiga tempat yang berbeda, memiliki spesifikasi yang berbeda-beda seperti pada tabel 3.1. Hal ini bertujuan untuk menguji seberapa baik sistem deteksi dalam melakukan pendeteksian diberbagai kondisi.

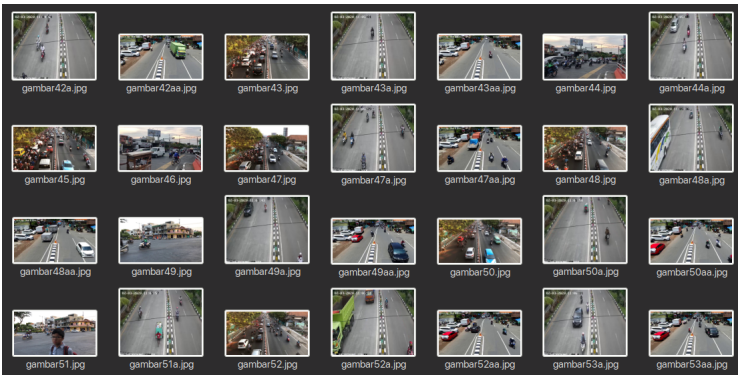
Tabel 3.1: Spesifikasi video pada masing-masing rekaman.

	Jembatan Surabaya	Jalan Pandegiling	Kedung Cowek
Resolusi (pixels)	1280 x 720	1280 x 720	704 x 576
Frame rate (FPS)	25	50	25
Rasio aspek	16:9	16:9	4:3
Bit rate (kb/s)	726	938	591
Posisi Kamera	Samping Jalan	Tengah Jalan	Tengah Jalan
Sudut Pandang Kamera (derajat)	60	60	45

3.4 Pelabelan Objek

Pelabelan atau anotasi bertujuan untuk memberikan informasi berupa nama kelas dan posisi yang berupa *bounding box* dari objek yang ingin dideteksi. Tahap pelabelan Objek dilakukan ketika data akuisisi yang berupa video rekaman Kamera IP milik Dinas Perhubungan Kota Surabaya telah diambil. Data berupa video tersebut diolah menjadi gambar yang ditangkap setiap 6 detik pada masing-masing video selama video diputar seperti pada Gambar 3.9.

Proses pelabelan dibagi menjadi tiga kelas, yaitu pengendara, dengan helm dan tanpa helm. Objek yang diberi label hanya



Gambar 3.9: Data Video yang sudah diolah menjadi Gambar

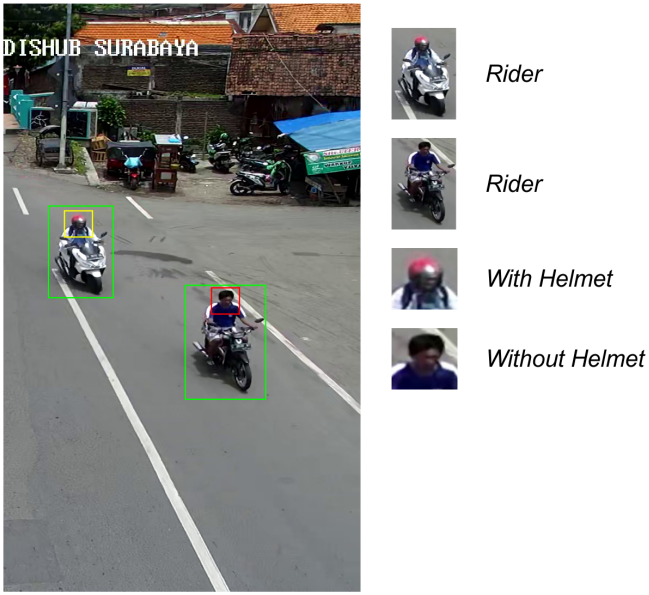
tampak depan. Skenario pelabelan pengendara dimulai dari kepala sampai roda motor, kemudian dalam pelabelan kelas dengan helm dimulai dari helm sampai dada pengendara, dan sama seperti kelas dengan helm, pelabelan kelas tanpa helm juga dimulai dari kepala sampai dada pengendara. Sehingga skenario secara visual untuk mendapatkan ketiga kelas tersebut seperti pada Gambar 3.10.

Proses pelabelan menggunakan software berbasis *GUI* yaitu LabelImg oleh Tzutalin. Setelah pelabelan, hasil anotasi disimpan dalam format Anotasi YOLO (*You Only Look Once*).

Hasil proses pelabelan dari 519 gambar, didapatkan jumlah anotasi seperti pada Tabel 3.2.

Tabel 3.2: Jumlah anotasi pada masing-masing kelas

Kelas	Jumlah Anotasi
Pengendara	1246
Dengan Helm	1082
Tanpa Helm	224
Total	2552



Gambar 3.10: Skenario Pelabelan Objek

3.5 Proses Melatih Himpunan Data

Setelah didapatkan anotasi dari kelas-kelas objek yang dideteksi, hasil anotasi tersebut diteruskan menuju proses pelatihan. Proses ini bertujuan untuk melatih komputer dengan cara mengolah gambar dan anotasi yang telah dibuat sehingga terbentuk pola atau karakteristik dari masing-masing kelas yang akan menjadi bahan pertimbangan komputer dalam mencapai sebuah keputusan atau prediksi. Proses training dilakukan menggunakan *framework neural network Darknet* dengan konfigurasi seperti pada Tabel 3.3.

Proses pelatihan menggunakan *darknet-53* sebagai *load model* dengan susunan lapisan seperti pada Gambar 3.11 dan YOLOv3 sebagai *load weight* dengan konfigurasi seperti Tabel 3.4. Untuk model yang kedua yaitu YOLOv3-*tiny* menggunakan feature extractor yang menggunakan susunan lapisan seperti yang diilustrasikan pada gambar 3.12. Jumlah *batch* menentukan jumlah gambar

Tabel 3.3: Konfigurasi pada Darknet

Jenis Konfigurasi	Keterangan
Load Model	*darknet-53*
Load Weight	YOLOv3
GPU	Yes
CUDNN	Yes
CUDNN_HALF	No
OPENCV	Yes
DEBUG	Yes
OPENMP	No
LIBSO	Yes
ZED_CAMERA	No

Tabel 3.4: Konfigurasi pada YOLOv3 dan YOLOv3-*tiny*.

Jenis Konfigurasi	Keterangan
batch	64
subdivisions	64
max_batches	6000
steps	4800,5400
width	416
height	416
classes	3
filters	24

yang akan diproses sebelum *network weight* mengalami pembaharuan. Selanjutnya jumlah subdivision bertujuan untuk memproses sebagian kecil *batch size* sekaligus pada GPU. *max_batches* merupakan batas iterasi dalam melakukan pelatihan yang didapatkan dengan persamaan. Ketika iterasi sudah mencapai 6000, maka secara otomatis proses pelatihan akan berhenti. Nilai *max_batches* ditentukan dengan persamaan 3.1.

$$max_batches = jumlahclass \times 2000 \tag{3.1}$$

	Type	Filters	Size	Output
	Convolutional	32	3 × 3	256 × 256
	Convolutional	64	3 × 3 / 2	128 × 128
1×	Convolutional	32	1 × 1	
	Convolutional	64	3 × 3	
	Residual			128 × 128
	Convolutional	128	3 × 3 / 2	64 × 64
2×	Convolutional	64	1 × 1	
	Convolutional	128	3 × 3	
	Residual			64 × 64
	Convolutional	256	3 × 3 / 2	32 × 32
8×	Convolutional	128	1 × 1	
	Convolutional	256	3 × 3	
	Residual			32 × 32
	Convolutional	512	3 × 3 / 2	16 × 16
8×	Convolutional	256	1 × 1	
	Convolutional	512	3 × 3	
	Residual			16 × 16
	Convolutional	1024	3 × 3 / 2	8 × 8
4×	Convolutional	512	1 × 1	
	Convolutional	1024	3 × 3	
	Residual			8 × 8
	Avgpool		Global	
	Connected		1000	
	Softmax			

Gambar 3.11: Susunan Lapisan pada *Darknet53*[10]

Nilai *steps* didapatkan dari persamaan 3.1.

$$steps = (80\%max_batches), (90\%max_batches) \tag{3.2}$$

Height dan *Weight* merupakan dimensi gambar masukan yang akan dilatih. Classes merupakan jumlah kelas yang ingin dideteksi. Kemudian nilai *Filters* didapatkan dengan persamaan 3.3.

$$filters = (jumlahkelas + 5) \times 3 \tag{3.3}$$

Layer	Type	Filters	Size/Stride	Input	Output
0	Convolutional	16	3 × 3/1	416 × 416 × 3	416 × 416 × 16
1	Maxpool		2 × 2/2	416 × 416 × 16	208 × 208 × 16
2	Convolutional	32	3 × 3/1	208 × 208 × 16	208 × 208 × 32
3	Maxpool		2 × 2/2	208 × 208 × 32	104 × 104 × 32
4	Convolutional	64	3 × 3/1	104 × 104 × 32	104 × 104 × 64
5	Maxpool		2 × 2/2	104 × 104 × 64	52 × 52 × 64
6	Convolutional	128	3 × 3/1	52 × 52 × 64	52 × 52 × 128
7	Maxpool		2 × 2/2	52 × 52 × 128	26 × 26 × 128
8	Convolutional	256	3 × 3/1	26 × 26 × 128	26 × 26 × 256
9	Maxpool		2 × 2/2	26 × 26 × 256	13 × 13 × 256
10	Convolutional	512	3 × 3/1	13 × 13 × 256	13 × 13 × 512
11	Maxpool		2 × 2/1	13 × 13 × 512	13 × 13 × 512
12	Convolutional	1024	3 × 3/1	13 × 13 × 512	13 × 13 × 1024
13	Convolutional	256	1 × 1/1	13 × 13 × 1024	13 × 13 × 256
14	Convolutional	512	3 × 3/1	13 × 13 × 256	13 × 13 × 512
15	Convolutional	255	1 × 1/1	13 × 13 × 512	13 × 13 × 255
16	YOLO				
17	**Route 13**				
18	Convolutional	128	1 × 1/1	13 × 13 × 256	13 × 13 × 128
19	Up-sampling		2 × 2/1	13 × 13 × 128	26 × 26 × 128
20	**Route 19 8**				
21	Convolutional	256	3 × 3/1	13 × 13 × 384	13 × 13 × 256
22	Convolutional	255	1 × 1/1	13 × 13 × 256	13 × 13 × 256
23	YOLO				

Gambar 3.12: Susunan Lapisan pada model YOLOv3-*tiny*[11]

3.6 Pengembangan Sistem Deteksi

Proses deteksi penggunaan helm pada pengendara bermotor menggunakan program yang telah dibuat sesuai dengan flowchart yang telah dibuat serta penambahan beberapa program untuk kebutuhan analisis. Proses pengenalan objek dilakukan menggunakan metode *Convolutional Neural Network* (CNN) dan proses deteksi menggunakan YOLOv3 (*You Look Only Once*).

3.6.1 Proses *Input*

Dalam proses deteksi, YOLO menggunakan gambar masukan dengan *network size* atau ukuran gambar masukan dengan nilai kelipatan 32. Semakin besar *network size* yang diberikan, maka semakin akurat komputer dalam memprediksi deteksi yang didapatkan, akan tetapi jika *network size* yang berikan semakin besar, maka proses komputasi pada komputer akan menjadi lebih lambat.

Sebaliknya jika *network size* yang diberikan semakin kecil maka proses komputasi akan menjadi lebih cepat namun dengan kosekuensi akurasi dalam deteksi menjadi kurang baik. Dalam penelitian ini, menggunakan network size sebesar 416 $\times$ 416.

3.6.2 *Feature Extraction*

Feature Extraction yang digunakan oleh YOLOv3 yaitu Darknet53. Darknet53 memiliki 53 *convolutional layers* yang mempunyai susunan lapisan seperti ilustrasi pada gambar 2.11 yang telah diuji coba pada Imagenet dan untuk model yang kedua yaitu YOLOv3-*tiny* menggunakan feature extractor yang mempunyai susunan lapisan seperti ilustrasi pada gambar 3.12. Pada proses ekstrasi fitur, semua lapisan *convolutional* pada Darknet53 menggunakan fungsi aktivasi *Leaky Relu* terkecuali pada layer terakhir yang menggunakan fungsi aktivasi Linier.

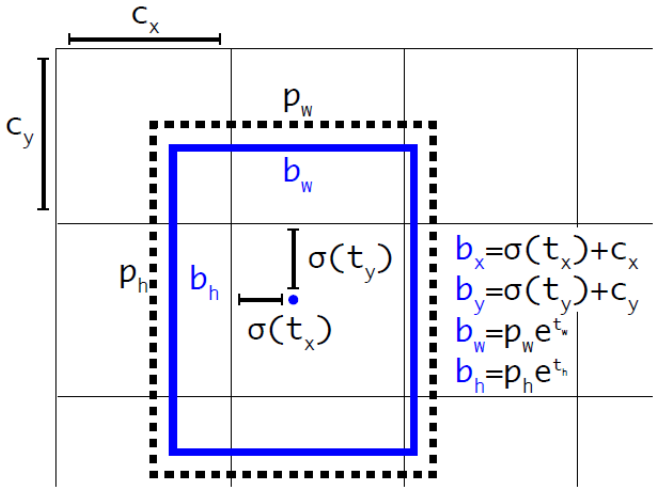
3.6.3 *Bounding Box*

YOLOv3 memberikan hasil prediksi berupa 6 nilai yang terdiri dari titik koordinat *bounding box* (t_x, t_y, t_w, t_h), *confidence* dan *class probability* (Probabilitas Kelas) kemudian diolah seperti ilustrasi pada Gambar 3.13.

Dalam memprediksi kelas *bounding box*, YOLOv3 menggunakan *multilabel classification*. YOLOv3 memprediksi skor kelas setiap *bounding box* menggunakan *logistic regression* untuk menghindari *bounding box* yang tumpang tindih dengan *bounding box* yang lain. Selain itu, terdapat *threshold* yang bertugas untuk memprediksi kelas pada suatu objek yang terdeteksi. Jika terdapat suatu skor kelas yang mempunyai nilai tinggi melebihi *threshold* maka kelas tersebut diberikan pada objek yang terdeteksi. Jika terdapat bounding box sebelumnya yang bertumpang tindih terhadap *ground-truth object* dan memenuhi nilai *threshold* maka deteksi tersebut akan diabaikan.

3.6.4 Deteksi Objek

YOLOv3 memberi prediksi bounding box dengan 3 skala yang berbeda. Masing-masing skala memiliki 3 *anchor* sehingga total memiliki 9 *anchor*. Dalam proses ekstrasi fitur dari skala-skala tersebut, dibutuhkan konsep yang mirip dengan *feature pyramid ne-*



Gambar 3.13: *Bounding Box* pada YOLOv3[10].

tworks. Sehingga dari dasar *feature extractor* ditambahkan beberapa lapisan *convolutional*.

Pada Gambar 3.14 merupakan objek dari 3 kelas yang terdeteksi menggunakan hasil pelatihan oleh YOLOv3 dengan bantuan library OpenCV yang kemudian diberi *bounding box*

3.6.5 Pengembangan Deteksi Pelanggaran

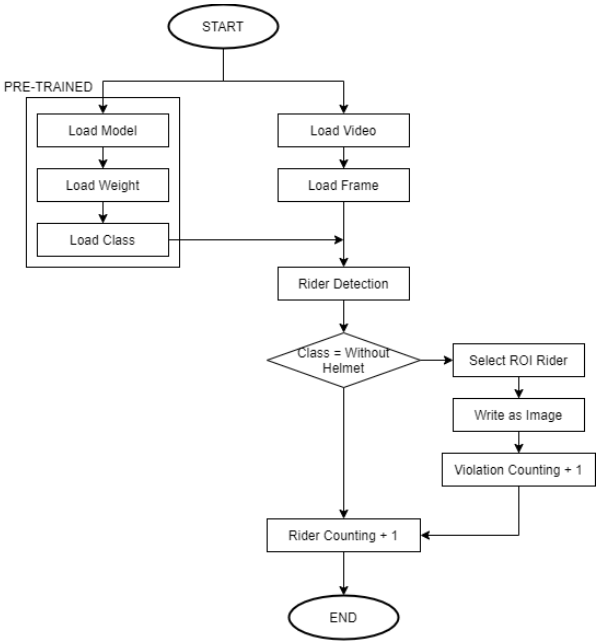
Pada penelitian ini, diharapkan sistem dapat mendeteksi pelanggaran lalu lintas berupa pengendara bermotor yang tidak menggunakan helm. Hasil yang diharapkan yaitu berupa tangkapan gambar sesuai titik koordinat pengendara motor yang melakukan pelanggaran dan perhitungan jumlah pelanggaran tidak menggunakan helm. Dalam sistem deteksi, komputer harus mampu mendeteksi pelanggaran tidak menggunakan helm baik itu pada pengendara atau penumpang dan dapat mendeteksi berbagai jenis kendaraan roda dua bermotor selain sepeda motor seperti gerobak motor. Selain itu, sistem juga harus bisa mendeteksi bahwa salah satu atau semua orang baik itu pengendara atau penumpang terdeteksi sebagai pelanggaran tidak menggunakan helm. Dari permasalahan

Gambar 3.14: Hasil Deteksi yang Ditunjukkan oleh *Bounding Box*.

tersebut, didapatkan skenario seperti pada Gambar 3.15 Adanya garis deteksi bertujuan untuk mendeteksi adanya pelanggaran atau tidak ada pelanggaran saat pengendara sepeda motor pada video melewati garis. Penempatan garis deteksi harus memenuhi syarat seperti berikut:

1. Posisi garis merupakan area terdekat dengan kamera.
2. Peletakkan garis berjarak sekitar 1,5 meter dari batas bawah video. Hal ini bertujuan untuk memudahkan sistem dalam mendeteksi adanya objek pengendara sepeda motor.
3. Peletakkan garis harus mencakup seluruh lebar jalan agar garis deteksi dapat mendeteksi pengendaran dari segala jalur yang didalam badan jalan.

Sehingga skema gambar peletakan garis deteksi dapat dilihat pada gambar 3.16. Sistem yang dibuat diharapkan dapat mendeteksi pelanggaran tidak menggunakan helm pada pengendara bermotor. Sistem yang telah dibuat akan menghitung jumlah pelanggaran serta mengambil gambar yang ada didalam *bounding box* pengendara yang melakukan pelanggaran berupa tidak menggunakan helm. Untuk mendeteksi pelanggaran, diberikan sebuah garis yang berfungsi sebagai deteksi. Jika garis *intersect* dengan titik tengah dari ke-

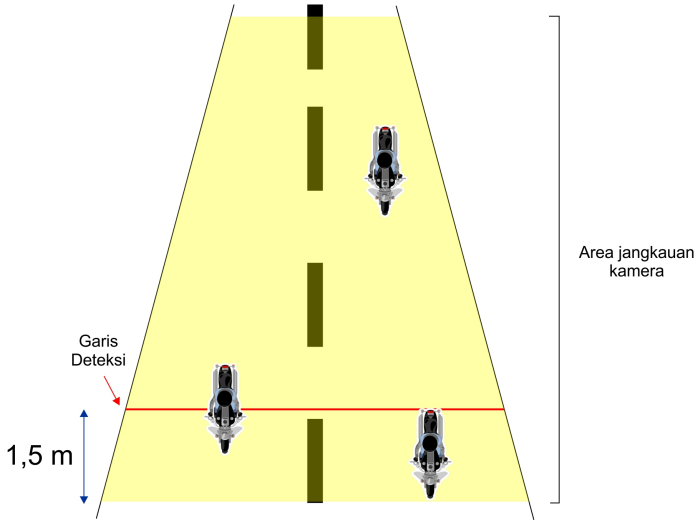


Gambar 3.15: Flowchart Sistem Deteksi Pelanggaran Tidak Menggunakan Helm.

las pengendara yang didalamnya terdapat kelas tanpa helm seperti pada gambar 3.17. Setelah terdeteksi pelanggaran, maka sistem akan mengambil ROI dari koordinat pengendara yang melakukan pelanggaran dan kemudian diambil dan disimpan menjadi gambar potongan seperti pada gambar 3.18.

3.7 Analisa Peforma

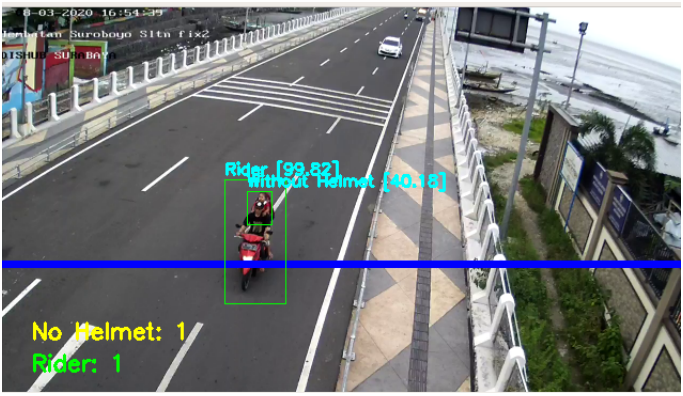
Analisa peforma bertujuan untuk mengukur validasi sistem yang telah dibuat dalam mendeteksi objek. Proses analisa diilustrasikan seperti pada gambar 3.19. Sistem deteksi akan memberikan hasil deteksi berupa *predicted bounding box*. Hasil deteksi tersebut kemudian dibandingkan dengan *ground-truth bounding box* yang telah diberikan sebelumnya pada tahap pelabelan objek. Perbandingan antara dua *bounding box* dilakukan dengan cara menghitung



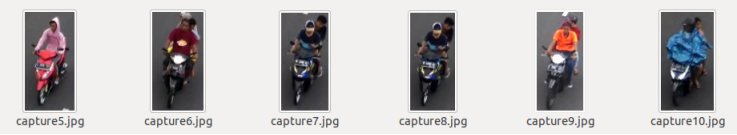
Gambar 3.16: Skema Peletakan Garis Deteksi.

nilai *Intersection over Union* (IoU) atau nilai *overlap* antara *predicted bounding box* dengan *ground-truth bounding box*. Jika nilai IoU yang didapatkan lebih dari nilai *threshold* yang sudah ditentukan yaitu sebesar 0,5, maka hasil deteksi tersebut dianggap benar. Sebaliknya, jika nilai IoU yang didapatkan kurang dari nilai *threshold*, maka hasil deteksi tersebut dianggap salah.

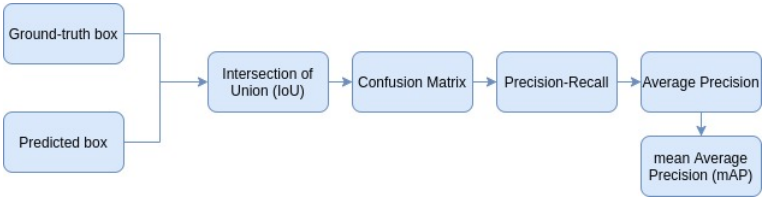
Kemudian hasil penghitungan IoU dikelompokkan dalam *Confusion Matrix* berupa nilai *True Positive*, *False Positive* atau *False Negative*. Hasil pengelompokkan tersebut digunakan untuk mencari nilai *precision* dan *recall* pada sistem deteksi yang akan menghasilkan sebuah kurva yang dinamakan *precision-recall curve*. Nilai *Average Precision* (AP) pada masing-masing kelas didapatkan dari hasil penghitungan area dibawah kurva pada setiap kelas yang terdeteksi oleh sistem deteksi. Kemudian nilai *Average Precision* (AP) dari tiga kelas tersebut dihitung rata-ratanya untuk mendapatkan nilai *mean Average Precision* (mAP).



Gambar 3.17: Proses Deteksi Pelanggaran tidak Menggunakan Helm.



Gambar 3.18: Hasil Pengambilan Gambar pada Pelanggar.



Gambar 3.19: Alur Analisa Peforma.

BAB 4
PENGUJIAN DAN ANALISIS

Pada penelitian ini, dipaparkan hasil pengujian serta analisis dari desain sistem dan implementasi. Data yang digunakan dalam pengujian data diambil dari IP Camera milik Dinas Perhubungan Kota Surabaya yang terpasang diseluruh lalu lintas Kota Surabaya. Pengambilan data video dilakukan di *Surabaya Intelligent Transport System* (SITS) yang merupakan kantor Dinas Perhubungan khusus untuk pengendalian seluruh kamera IP lalu lintas yang tersebar di Kota Surabaya. Pengujian dilakukan dalam beberapa bagian sebagai berikut:

1. Pengujian Peforma berdasarkan Lokasi.
2. Pengujian Peforma berdasarkan Kondisi.
3. Pengujian Deteksi berdasarkan Objek Pelanggar.
4. Pengujian Sistem

Pada pengujian, masing-masing model menggunakan *Personal Computer* (PC) dengan spesifikasi perangkat keras seperti pada tabel 4.1.

Tabel 4.1: Spesifikasi PC yang Digunakan.

Prosesor	Intel(R) Core(TM) i5-4590 CPU @ 3.30GHz
RAM	4 GB DIMM DDR3 Synchronous 1600 MHz x 2
Penyimpanan	SSD 2.5 inch 120 GB
Kartu Grafis	Nvidia GeForce GTX 1060 6GB
Sistem Operasi	Ubuntu 18.04 LTS 64-bit

4.1 Pengujian Peforma berdasarkan Lokasi

Pengujian peforma berdasarkan lokasi bertujuan untuk mengetahui tingkat keakurasian pada YOLOv3 dan YOLOv3-*tiny* terhadap lokasi yang memiliki karakteristik yang berbeda-beda. Pemilihan lokasi didasarkan kondisi seperti sudut pandang kamera dan posisi kamera yang hampir dimiliki oleh semua kamera yang tersebar

di Kota Surabaya. Sehingga dalam pengujian peforma dilakukan pada tiga lokasi antara lain:

1. Jembatan Surabaya.
2. Kedung Cowek (SMAN 19 Surabaya).
3. Jalan Pandegiling - Imam Bonjol.

Dalam melakukan uji peforma, dibutuhkan himpunan data tes yang telah diberi anotasi. Himpunan data tes merupakan himpunan data yang belum pernah digunakan sebagai himpunan data pelatihan. Himpunan data yang benar dibutuhkan sekitar 10% sampai 20% dari jumlah himpunan data pelatihan. Dalam tahap pelatihan terdapat 332 gambar yang telah dilakukan pelabelan dengan rincian seperti pada Tabel 4.2. Himpunan data yang benar yang sudah di-

Tabel 4.2: Jumlah Himpunan Data Uji pada Masing-Masing Lokasi

Pengujian	Jumlah Himpunan Data Gambar	Jumlah Aktual		
		Pengendara	Dengan Helm	Tanpa Helm
Jembatan Surabaya	80	397	312	114
Jalan Pandegiling	100	381	269	147
Jalan Kedung Cowek	152	119	62	79

beri anotasi dan kelas tersebut diolah sebagai *ground-truth box* dibandingkan dan dengan *predicted box* yang kemudian menghasilkan *Confusion Matrix*, dan kemudian dikalkulasi untuk mendapatkan nilai *Precision, Recall, Average Precision (AP), F1-score, Intersection over Union (IoU)* dan *mean Average Precision* sesuai rumus yang telah diberikan pada tinjauan pustaka.

4.1.1 Pengujian pada Lokasi Jembatan Surabaya

Pada lokasi Jembatan Surabaya memiliki beberapa karakteristik diantara lain kondisi pada lokasi terjadi pada saat cerah berawan (normal). posisi kamera terletak pada tiang lampu lalu lintas dengan sudut pandang kamera menyamping seperti pada gambar 4.1 dan memiliki kualitas gambar yang baik. Pada lokasi ini, ditemukan banyak pelanggaran tidak menggunakan helm dikarenakan Jembatan Surabaya bukan merupakan jalan umum. Pada himpunan data lokasi ini belum pernah dijadikan sebagai bahan untuk himpunan data pelatihan. Dari hasil pengujian pada masing-masing model,

didapatkan hasil seperti pada tabel 4.3. Pada pengujian di lokasi

(a) Tampak depan.

(b) Tampak belakang.

Gambar 4.1: Posisi Kamera di Lokasi Jembatan Surabaya

Jembatan Surabaya, seperti pada tabel 4.3 didapatkan *mean Average Precision* (mAP) tertinggi pada YOLOv3 dengan nilai 76,77% dibandingkan dengan YOLOv3-*tiny* yang mendapatkan nilai 57,06%. Dalam waktu pemrosesan himpunan data percobaan, YOLOv3-*tiny* memiliki keunggulan waktu proses untuk mendeteksi 80 himpunan data, dibutuhkan waktu 1 detik, sementara pada YOLOv3, dibutuhkan waktu deteksi selama 3 detik untuk seluruh gambar. Pada deteksi kelas kelas tanpa helm, pada YOLOv3 didapatkan hasil *True Positive* (TP) lebih besar daripada *False Positive* (FN) yaitu sebesar 41 objek yang terdeteksi sehingga nilai *Average Precision* yang

Tabel 4.3: Pengujian Peforma pada Lokasi Jembatan Surabaya

Load Model		YOLOv3	YOLOv3 tiny
Pengendara	**AP**	92,70%	88,92%
	TP	107	94
	FP	12	5
Dengan Helm	**AP**	80,01%	59,02%
	TP	54	34
	FP	33	36
Tanpa Helm	**AP**	57,61%	23,23%
	TP	41	24
	FP	11	53
FN		58	108
Waktu Pemrosesan (s)		3	1
Precision		0,78	0,62
Recall		0,78	0,58
f1-score		0,78	0,6
IoU		58,57%	45,60%
mAP@0,5		**76,77%**	**57,06%**

didapatkan sebesar 57,61%. Sebaliknya, pada YOLOv3-*tiny* mendapatkan hasil *False Positive* (FP) lebih besar daripada *True Positive* (TP) dengan jumlah 53 yang terdeteksi sebagai FP, sehingga menurunkan nilai *average precision* menjadi 23,23%.

4.1.2 Pengujian pada Lokasi Kedung Cowek

Pada lokasi Kedung Cowek, mempunyai karakteristik dengan kualitas gambar kurang baik daripada 2 lokasi lainnya yang menjadi titik pengujian. Posisi dari kamera terletak ditengah jalan pada tiang lampu seperti pada gambar 4.2. Pada uji peforma, didapatkan hasil seperti pada tabel 4.4. Pada masing-masing model setelah melalui pengujian, didapatkan nilai *mean Average Precision* (mAP) yang cukup tinggi. Pada model YOLOv3 memiliki nilai mAP tertinggi sebesar 81,82% dibandingkan dengan YOLOv3-*tiny* yang mempunyai nilai mAP sebesar 72,42%. Hal ini didukung juga

(a) Tampak depan.

(b) Tampak belakang.

Gambar 4.2: Posisi Kamera di Lokasi Kedung Cowek

dengan deteksi kelas kelas tanpa helm yang memiliki nilai *Average Precision* (AP) pada masing-masing model yang cukup baik. Pada model YOLOv3 dapat mendeteksi kelas kelas tanpa helm dengan benar sebanyak 97 objek dari 152 gambar sehingga mendapatkan nilai *Average Precision* (AP) sebesar 67,69%. Sedangkan pada model YOLOv3-*tiny* dapat mendeteksi kelas kelas tanpa helm dengan benar sebanyak 71 objek sehingga mendapatkan nilai *Average Precision* (AP) sebesar 51,05%. Pada waktu yang dibutuhkan untuk memproses 152 gambar, pada model YOLOv3 membutuhkan waktu sebanyak 5 detik, sedangkan pada model YOLOv3-*tiny* hanya membutuhkan waktu 1 detik sehingga pada model YOLOv3-*tiny* mempunyai waktu pemrosesan yang lebih cepat.

Tabel 4.4: Pengujian Peforma pada Lokasi Kedung Cowek

Load Model		YOLOv3	YOLOv3 tiny
Pengendara	**AP**	95,97%	96,62%
	TP	358	367
	FP	21	49
Dengan Helm	**AP**	81,81%	69,59%
	TP	222	209
	FP	51	133
Tanpa Helm	**AP**	67,69%	51,05%
	TP	97	71
	FP	34	46
FN		115	145
Waktu Pemrosesan (s)		5	1
Precision		0,86	0,74
Recall		0,85	0,82
f1-score		0,86	0,78
IoU		67,47%	55,28%
mAP@0,5		**81,82%**	**72,42%**

4.1.3 Pengujian pada Lokasi Jalan Pandegiling

Pada lokasi Jalan Pandegiling, posisi kamera terdapat di tiang lampu yang terletak pada tengah jalan. Kamera pada lokasi ini menghasilkan kualitas gambar yang baik. Berbeda dengan kamera pada lokasi Kedung Cowek, kamera yang terpasang pada lokasi Pandegiling memiliki sudut kamera sebesar 60 derajat. Pada uji peforma, didapatkan dengan hasil seperti pada tabel 4.5. Seperti pada tabel 4.5, nilai *mean Average Precision* (mAP) antara model YOLOv3 dengan YOLOv3-*tiny* mempunyai selisih mAP yang cukup jauh. Pada model YOLOv3 memiliki nilai mAP lebih tinggi yaitu sebesar 80,43% daripada nilai mAP pada model YOLOv3*tiny* sebesar 56,04% dari 100 gambar yang telah diuji. Begitu juga dengan banyaknya kelas kelas tanpa helm yang dikenali. Pada model YOLOv3 dapat mengenali 94 objek kelas kelas tanpa helm (*True Positive*) yang mempunyai selisih banyak dengan jumlah *False Po-*

Tabel 4.5: Pengujian Peforma pada Lokasi Jalan Pandegiling

Load Model		YOLOv3	YOLOv3 tiny
Pengendara	**AP**	81,00%	84,55%
	TP	316	313
	FP	23	49
Dengan Helm	**AP**	74,94%	47,55%
	TP	230	177
	FP	32	145
Tanpa Helm	**AP**	85,35%	36,01%
	TP	94	48
	FP	7	56
FN		183	285
Waktu Pemrosesan (s)		3	2
Precision		0,91	0,68
Recall		0,78	0,65
f1-score		0,84	0,67
IoU		71,40%	48,91%
mAP@0,5		**80,43%**	**56,04%**

sitive sebanyak 7 objek, sehingga didapatkan nilai *Average Precision* (AP) sebesar 85,35% pada model YOLOv3. Sebaliknya, pada model YOLOv3-*tiny*, obyek yang dapat dikenali (*True Positive*) memiliki jumlah yang lebih sedikit sebesar 48 objek yang dikenali dibandingkan dengan jumlah objek yang tidak dapat dikenali *False Positive* sebanyak 56 objek. Sehingga *Average Precision* (AP) kelas kelas tanpa helm pada model YOLOv3-*tiny* sebesar 36,01%. Pada waktu pemrosesan terhadap 100 himpunan data yang diujikan, model YOLOv3-*tiny* membutuhkan waktu sebanyak 2 detik lebih cepat dibandingkan dengan waktu pemrosesan pada model YOLOv3 sebanyak 3 detik.

4.2 Pengujian Peforma berdasarkan Kondisi

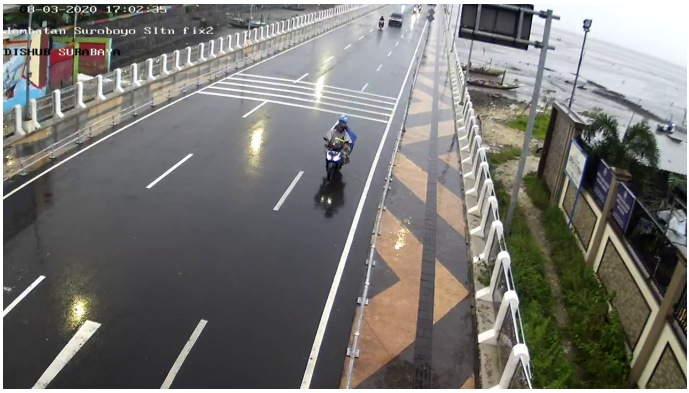
Pengujian peforma berdasarkan kondisi bertujuan untuk mengetahui keakuratan deteksi ketika dihadapkan dengan bermacam-

macam kondisi seperti cuaca. Dalam pengujian ini, terdapat dua kondisi yang diberikan antara lain:

1. Kondisi Kecerahan Tinggi
2. Kondisi Hujan

Himpunan data percobaan pada kondisi hujan diambil pada lokasi Jembatan Surabaya yang mengalami cuaca hujan pada sore hari seperti pada gambar 4.3. Pada kondisi kecerahan tinggi diambil pada lokasi yang sama yaitu pada Jembatan Surabaya yang diambil pada sore hari mendekati matahari terbenam seperti pada gambar 4.4.

Pada pengujian peforma berdasarkan kondisi menggunakan 224

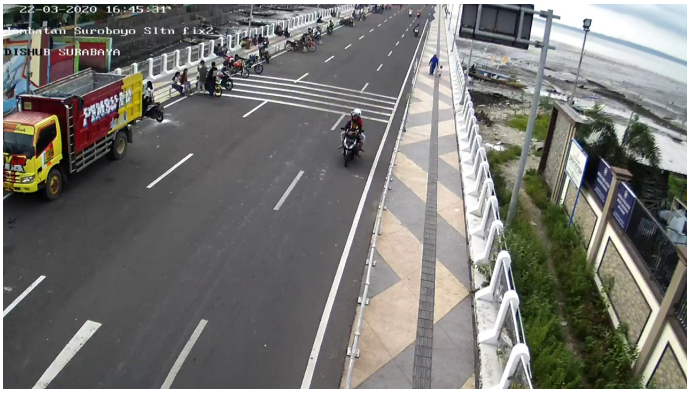


Gambar 4.3: Kondisi Rekaman IP Camera saat Cuaca Hujan.

gambar yang telah diberi anotasi dan kelas. Dari gambar tersebut terbagi seperti rincian pada tabel 4.6.

Tabel 4.6: Jumlah Himpunan Data Uji pada Masing-Masing Kondisi

Pengujian	Jumlah Himpunan Data Gambar	Jumlah Aktual		
		Pengendara	Dengan Helm	Tanpa Helm
Kondisi Hujan	135	102	44	114
Kondisi Cerah	233	87	183	147



Gambar 4.4: Kondisi Rekaman IP Camera saat Cuaca Cerah.

4.2.1 Kondisi Kecerahan Tinggi

Pada kondisi kecerahan tinggi, memiliki tingkat kecerahan lebih tinggi dibandingkan data rekaman yang telah diuji pada pengujian lokasi. Pada himpunan data yang memiliki kecerahan tinggi belum pernah dijadikan sebagai bahan untuk himpunan data pelatihan. Dari pengujian peforma dari 137 gambar dengan kondisi cerah, didapatkan hasil seperti pada tabel 4.7. Dari hasil pengujian seperti pada tabel 4.7, hasil *mean Average Precision* (mAP) mengalami penurunan dibandingkan hasil pengujian lainnya. Pada model YOLOv3 didapatkan mAP sebesar 53,63% lebih tinggi daripada hasil mAP pada model YOLOv3-*tiny* sebesar 47,44%. Salah satu penyebab penurunan mAP pada pengujian kondisi ini terdapat pada hasil deteksi kelas kelas tanpa helm. Pada model YOLOv3 dapat mengenali dengan benar atau *True Positive* (TP) pada kelas kelas tanpa helm sebanyak 45 objek, hasil ini lebih tinggi daripada hasil *False Positive* sebanyak 18 objek. Pada model YOLOv3-*tiny* dapat mengenali dengan benar atau *True Positive* (TP) sebanyak 46 objek, angka tersebut lebih kecil dibandingkan *False Positive* sebanyak 58 objek. Dalam waktu pemrosesan, YOLOv3-*tiny* dapat mendeteksi 137 gambar dalam 2 detik, lebih cepat daripada menggunakan model YOLOv3 yang dapat mendeteksi seluruh gambar selama 4 detik.

Tabel 4.7: Pengujian Peforma pada Kondisi Cerah

Load Model		YOLOv3	YOLOv3 tiny
Pengendara	**AP**	87,10%	83,46%
	TP	181	143
	FP	22	0
Dengan Helm	**AP**	45,62%	36,98%
	TP	47	34
	FP	73	51
Tanpa Helm	**AP**	28,16%	21,89%
	TP	45	46
	FP	18	58
FN		230	280
Waktu Pemrosesan (s)		4	2
Precision		0,71	0,67
Recall		0,54	0,44
f1-score		0,61	0,53
IoU		52,94%	49,20%
mAP@0,5		**53,63%**	**47,44%**

4.2.2 Kondisi Hujan

Karakteristik pengendara pada kondisi hujan sebagian besar menggunakan jas hujan. Kondisi lingkungan pada saat hujan mempunyai kondisi rekaman yang lebih gelap daripada 2 kondisi sebelumnya pada lokasi yang sama. Data rekaman pada kondisi hujan tidak termasuk dalam himpunan data pelatihan. Dari hasil pengujian peforma pada kondisi hujan didapatkan hasil seperti pada tabel 4.8. Dari hasil pengujian pada kondisi hujan seperti pada tabel 4.8, didapatkan nilai *mean Average Precision* (mAP) yang rendah pada kedua model. Pada model YOLOv3 mendapatkan nilai mAP sebesar 51,19% lebih tinggi jika dibandingkan dengan nilai mAP pada model YOLOv3-*tiny* sebesar 47,44%. Hal tersebut disebabkan rendahnya nilai *Average Precision* (AP) pada kelas kelas tanpa helm pada masing-masing model. Pada model YOLOv3 didapatkan jumlah *False Positive* (TP) sebanyak 17 lebih tinggi dibandingkan

Tabel 4.8: Pengujian Peforma pada Kondisi Hujan

Load Model		YOLOv3	YOLOv3 tiny
Pengendara	**AP**	72,15%	90,48%
	TP	92	93
	FP	14	4
Dengan Helm	**AP**	51,47%	35,58%
	TP	53	30
	FP	17	26
Tanpa Helm	**AP**	29,95%	16,26%
	TP	15	17
	FP	17	67
FN		121	141
Waktu Pemrosesan (s)		3	1
Precision		0,77	0,59
Recall		0,57	0,5
f1-score		0,65	0,54
IoU		50,52%	40,20%
mAP@0,5		**51,19%**	**47,44%**

dengan jumlah objek yang dikenali atau *True Positive* (TP) yang berjumlah 15, sehingga AP yang didapatkan hanya sebesar 29,95%. Sedangkan pada model YOLOv3-*tiny*, pada kelas kelas tanpa helm didapatkan jumlah dikenali dengan benar atau *True Positive* (TP) sebanyak 17 objek, jumlah ini selisih jauh dengan jumlah *False Positive* (FP) sebanyak 67 objek. Pada waktu yang dibutuhkan untuk mendeteksi seluruh gambar, YOLOv3-*tiny* hanya membutuhkan waktu 1 detik untuk mendeteksi 107 gambar. Perolehan ini lebih cepat dibandingkan menggunakan model YOLOv3 yang dapat mendeteksi seluruh gambar dalam waktu 3 detik.

4.3 Pengujian Deteksi berdasarkan Objek Pelanggar

Dalam pengujian deteksi berdasarkan objek pelanggar bertujuan untuk mengetahui keakuratan model dalam mendeteksi kelas

kelas tanpa helm dalam berbagai macam kondisi antara lain:

1. Pelanggar Kondisi Normal
2. Pelanggar Menggunakan Masker
3. Pelanggar Menggunakan *Hoodie*
4. Pelanggar Menggunakan Jilbab
5. Pelanggar Menggunakan Topi
6. Pelanggar Menggunakan Jas Hujan
7. Pelanggar pada Penumpang
8. Pelanggar pada Anak-Anak

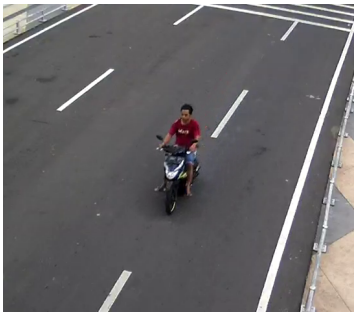
Keluaran hasil deteksi berupa *bounding box* pada objek yang dikenali yaitu Pengendara dengan keterangan Warna Ungu, Tanpa Helm dengan keterangan Warna Jingga dan kelas tanpa helm dengan keterangan Warna Hijau dan nilai *confidence*.

4.3.1 Pelanggar Kondisi Normal

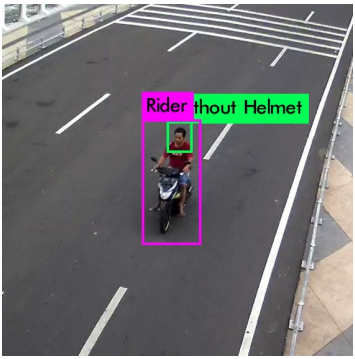
Pelanggar dengan kondisi normal merupakan pengendara yang tidak menggunakan helm tanpa memakai aksesoris apapun pada kepala. Sampel gambar yang diuji diambil dari lokasi Jembatan Surabaya. Hasil deteksi dapat dilihat pada gambar 4.5. Dari hasil deteksi pada gambar 4.5, masing-masing model dapat mendeteksi pengendara yang tidak menggunakan helm pada kondisi normal. Pada model YOLOv3 didapatkan nilai *confidence* pada kelas kelas tanpa helm sebesar 99%, sedangkan pada model YOLOv3-*tiny* didapatkan nilai *confidence* sebesar 85%.

4.3.2 Pelanggar Menggunakan Masker

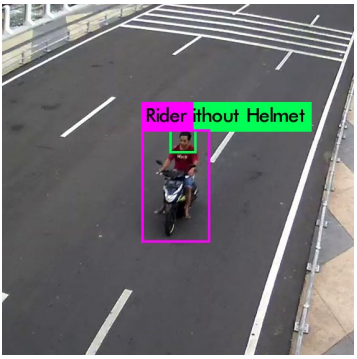
Pelanggar menggunakan masker merupakan kondisi pelanggar yang tidak menggunakan helm yang menggunakan aksesoris penutup wajah atau masker. Sampel gambar diambil dari lokasi Jembatan Surabaya pada kondisi cerah. Hasil deteksi dapat dilihat pada gambar 4.6. Dari hasil deteksi pada gambar 4.6, masing-masing model dapat mendeteksi pelanggar yang menggunakan masker. Pada model YOLOv3 didapatkan nilai *confidence* pada kelas kelas tanpa helm sebesar 92%, sedangkan pada model YOLOv3-*tiny* didapatkan nilai *confidence* sebesar 91%.



(a) Input Gambar.



(b) Deteksi menggunakan YOLOv3.



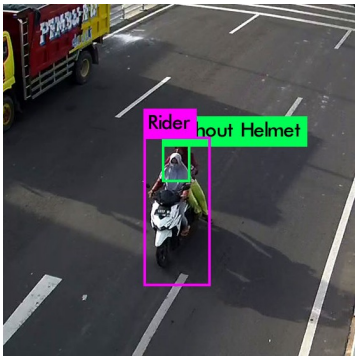
(c) Deteksi menggunakan YOLOv3-*tiny*.

Gambar 4.5: Deteksi pada Pelanggar Kondisi Normal.

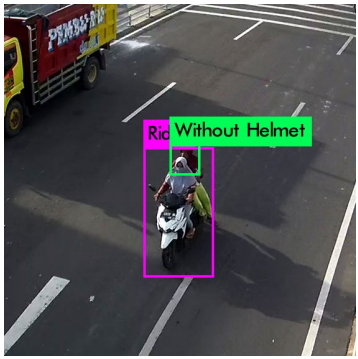
4.3.3 Pelanggar Menggunakan *Hoodie*

Pelanggar menggunakan *hoodie* merupakan kondisi pelanggar yang menggunakan *hoodie* atau menutup kepala. Sampel gambar dengan kondisi tersebut diambil pada lokasi Jembatan Surabaya. Hasil deteksi dapat dilihat pada gambar 4.7. Dari hasil deteksi pada gambar 4.7, masing-masing model dapat mendeteksi pelanggar yang menggunakan *hoodie* pada kondisi normal. Pada model

(a) Input Gambar.



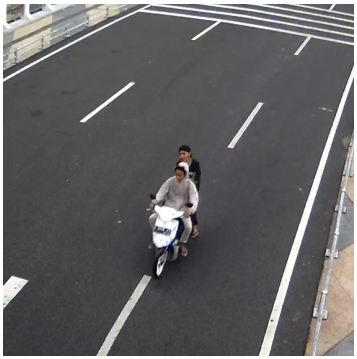
(b) Deteksi menggunakan YOLOv3.



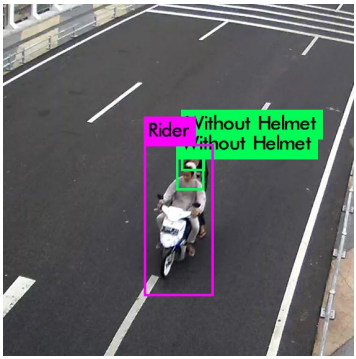
(c) Deteksi menggunakan YOLOv3-*tiny*.

Gambar 4.6: Deteksi pada Pelanggar Menggunakan Masker.

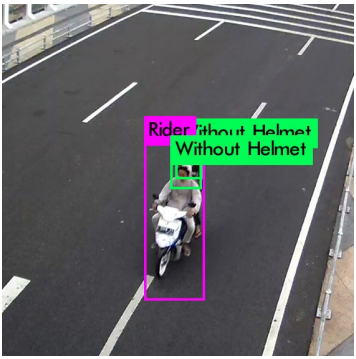
YOLOv3 didapatkan nilai *confidence* pada kelas kelas tanpa helm sebesar 98%, sedangkan pada model YOLOv3-*tiny* didapatkan nilai *confidence* sebesar 97%. Pada sampel gambar ini juga, model dapat mendeteksi pelanggaran tidak menggunakan helm pada penumpang.



(a) Input Gambar.



(b) Deteksi menggunakan YOLOv3.



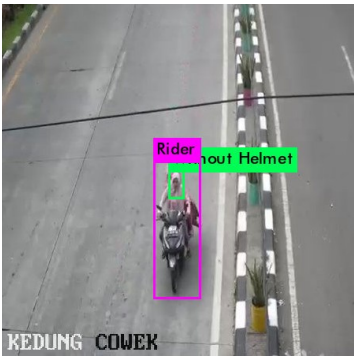
(c) Deteksi menggunakan YOLOv3-*tiny*.

Gambar 4.7: Deteksi pada Pelanggar Menggunakan *Hoodie*.

4.3.4 Pelanggar Menggunakan Jilbab

Pelanggar menggunakan jilbab merupakan kondisi pelanggar yang merupakan seorang perempuan yang menggunakan jilbab pada kepala. Sampel gambar pada kondisi ini diambil pada lokasi Kedung Cowek. Hasil deteksi dapat dilihat pada gambar 4.8 Dari hasil deteksi pada gambar 4.8, masing-masing model dapat mendeteksi pelanggar yang menggunakan jilbab. Pada model YOLOv3 dida-

(a) Input Gambar.



(b) Deteksi menggunakan YOLOv3.

(c) Deteksi menggunakan YOLOv3-*tiny*.

Gambar 4.8: Deteksi pada Pelanggar Menggunakan Jilbab.

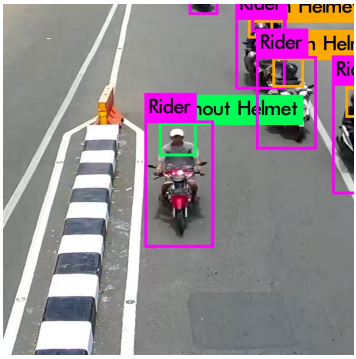
patkan nilai *confidence* pada kelas kelas tanpa helm sebesar 95%, sedangkan pada model YOLOv3-*tiny* didapatkan nilai *confidence* sebesar 55%. Secara bersamaan, pada model YOLOv3 tidak dapat mendeteksi pelanggaran pada penumpang. Sedangkan pada model YOLOv3-*tiny* dapat mendeteksi pelanggaran pada penumpang, akan tetapi terdapat hasil deteksi yang salah pada kelas kelas tanpa helm. Sehingga pada gambar secara aktual seharusnya terdeteksi 2 objek kelas kelas tanpa helm tetapi pada hasil deteksi terdapat 3

kelas kelas tanpa helm yang terdeteksi.

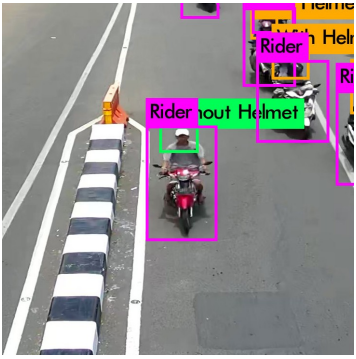
4.3.5 Pelanggar Menggunakan Topi

Pelanggar menggunakan topi merupakan kondisi pelanggar yang yang menggunakan topi atau penutup bagian atas kepala. Sampel gambar tersebut diambil pada lokasi Jalan Pandegiling. Hasil deteksi dapat dilihat pada gambar 4.9 Dari hasil deteksi pada gambar

(a) Input Gambar.



(b) Deteksi menggunakan YOLOv3.



(c) Deteksi menggunakan YOLOv3-*tiny*.

Gambar 4.9: Deteksi pada Pelanggar Menggunakan Topi.

4.9, masing-masing model dapat mendeteksi pelanggar yang meng-

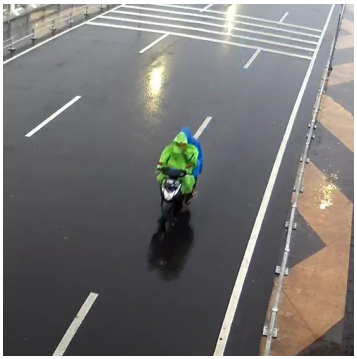
gunakan topi. Pada model YOLOv3 didapatkan nilai *confidence* pada kelas kelas tanpa helm sebesar 100%, sedangkan pada model YOLOv3-*tiny* didapatkan nilai *confidence* sebesar 99%.

4.3.6 Pelanggar Menggunakan Jas Hujan

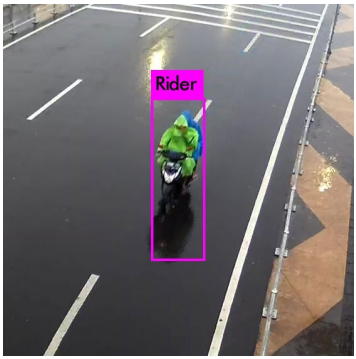
Pelanggar menggunakan topi merupakan kondisi pelanggar yang menggunakan jas hujan pada kondisi hujan sehingga jas hujan menutupi sebagian kepala. Sampel gambar tersebut diambil pada lokasi Jembatan Surabaya dalam keadaan hujan. Hasil deteksi dapat dilihat pada gambar 4.10 Seperti pada gambar 4.10, pada model YOLOv3 tidak dapat mendeteksi adanya objek kelas kelas tanpa helm pada pelanggar yang menggunakan jas hujan. Sebaliknya pada model YOLOv3-*tiny* dapat mendeteksi adanya pelanggaran pada penggunaan jas hujan dengan nilai *confidence* sebesar 86%. Hal ini disebabkan minimnya himpunan data yang didapatkan pada pengendara menggunakan jas hujan, Sementara pada YOLOv3-*tiny* mampu mendeteksi objek pengendara menggunakan jas hujan. Hal ini dikarenakan sistem objek deteksi yang menggunakan YOLOv3-*tiny* mendeteksi hampir secara keseluruhan objek, baik pelanggar maupun bukan pelanggar sebagai kelas kelas tanpa helm, hal ini terlihat ketika hasil deteksi pada masing-masing model dijalankan secara bersamaan dan dibandingkan satu dengan yang lain, sehingga jika dilihat pada pengujian lokasi maupun kondisi, pada deteksi menggunakan YOLOv3-*tiny* mendapatkan nilai *mean Average Precision* (mAP) yang kecil meski dapat mendeteksi segala objek penlanggar.

4.3.7 Pelanggaran pada Penumpang

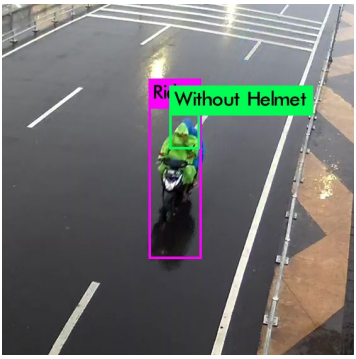
Pelanggaran pada Penumpang merupakan kondisi pengendara yang sudah menggunakan helm, tetapi pelanggaran tidak menggunakan helm terjadi pada penumpang. Sampel gambar diambil pada lokasi Jalan Pandegiling. Hasil deteksi dapat dilihat pada gambar 4.11. Seperti pada gambar 4.11, masing-masing model masing dapat mendeteksi penggunaan helm pada pengendara. Untuk pelanggaran pada penumpang, model YOLOv3 didapatkan nilai *confidence* pada kelas kelas tanpa helm sebesar 98%, sedangkan pada model YOLOv3-*tiny* didapatkan nilai *confidence* sebesar 62%.



(a) Input Gambar.



(b) Deteksi menggunakan YOLOv3.

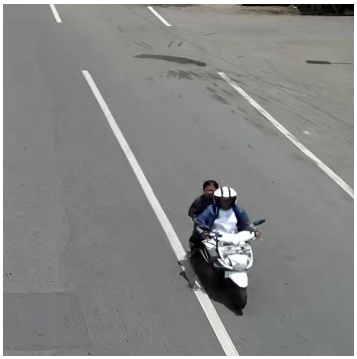


(c) Deteksi menggunakan YOLOv3-*tiny*.

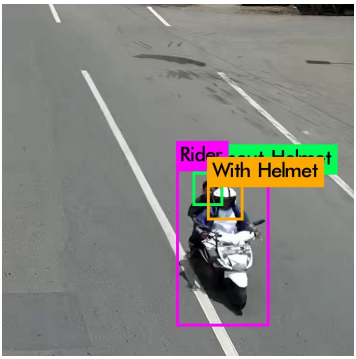
Gambar 4.10: Deteksi pada Pelanggar Menggunakan Jas Hujan.

4.3.8 Pelanggaran pada Anak-Anak

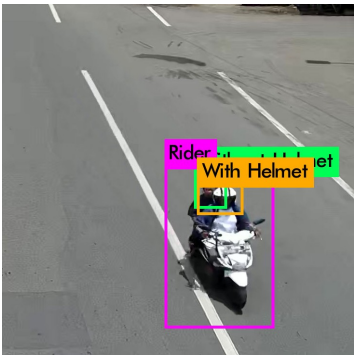
Pelanggaran pada Anak-Anak merupakan kondisi pengendara yang membawa penumpang anak yang menempati tempat duduk didepan pengendara. Sampel gambar diambil dari lokasi Jembatan Surabaya. Hasil deteksi dapat dilihat pada gambar 4.12. Dari hasil deteksi pada gambar 4.12, pada model YOLOv3-*tiny* tidak dapat mendeteksi pelanggaran pada anak-anak sehingga hanya pada mo-



(a) Input Gambar.



(b) Deteksi menggunakan YOLOv3.



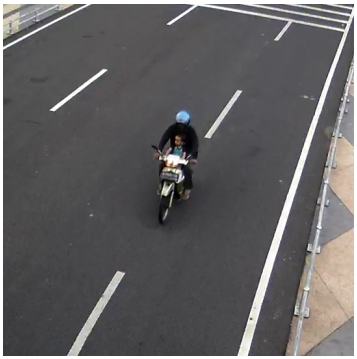
(c) Deteksi menggunakan YOLOv3-*tiny*.

Gambar 4.11: Deteksi Pelanggaran pada Penumpang.

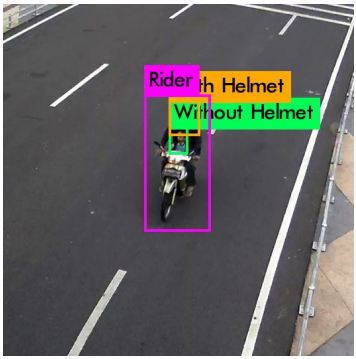
del YOLOv3 yang dapat mendeteksi pelanggaran pada anak-anak. Hasil deteksi kelas kelas tanpa helm pada model YOLOv3 didapatkan nilai *confidence* sebesar 39%.

4.4 Pengujian Sistem

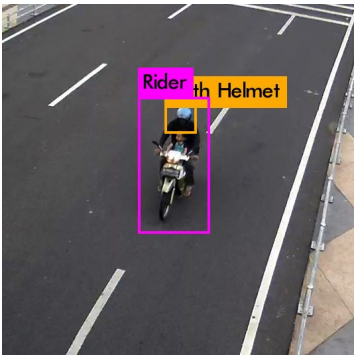
Sistem yang telah dibuat merupakan pengembangan dari *object detection* yang bertujuan untuk mencapai tujuan dan hasil yang diharapkan salah satunya dapat mengambil ROI pengendara yang



(a) Input Gambar.



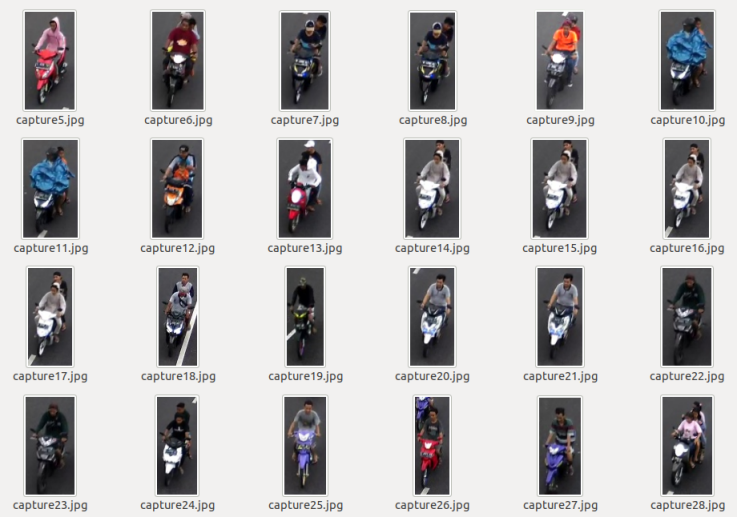
(b) Deteksi menggunakan YOLOv3.



(c) Deteksi menggunakan YOLOv3-*tiny*.

Gambar 4.12: Deteksi Pelanggaran pada Anak-Anak.

melanggar dan disimpan. Hasil potongan gambar pelanggaran pada pengendara seperti pada gambar 4.13. Pada hasil deteksi seperti pada gambar 4.13, terdapat gambar yang sama, Hal tersebut dikarenakan line detector mendeteksi lebih dari satu titik tengah kelas Pengendara pada pengendara yang sama.



Gambar 4.13: Hasil Deteksi Pelanggaran yang Disimpan.

BAB 5
PENUTUP

5.1 Kesimpulan

Berdasarkan hasil pengujian yang telah dilakukan, dapat ditarik beberapa kesimpulan sebagai berikut:

1. *Average Precision* (AP) yang didapatkan kelas tanpa helm pada semua iterasi mendapatkan presentasi kecil. Hal ini disebabkan kurangnya anotasi pada himpunan data percobaan yang berisi kelas tanpa helm yang tidak sebanding jumlahnya dengan jumlah anotasi pada kelas pengendara maupun kelas dengan helm.
2. YOLOv3 mempunyai tingkat akurasi yang lebih tinggi daripada YOLOv3-*tiny*, hal ini ditunjukkan dengan nilai *mean Average Precision* (mAP) pada pengujian masing-masing kondisi dan lokasi.
3. Pada kelas pengendara dan dengan helm tidak perlu penambahan anotasi untuk proses pelatihan selanjutnya, dikarenakan nilai Average Precision (AP) pada masing-masing model mendapatkan hasil yang baik.
4. YOLOv3-*tiny* mempunyai waktu pemrosesan yang lebih cepat daripada YOLOv3, hal ini ditunjukkan dengan waktu yang diperoleh dalam mendeteksi himpunan data yang benar dalam satuan sekon.
5. YOLOv3 dan YOLOv3-*tiny* mengalami penurunan akurasi secara drastis pada kondisi hujan dan kecerahan tinggi jika dibandingkan dengan tingkat akurasi pada tiga lokasi berbeda.
6. Masing-masing model dapat mendeteksi berbagai variasi pelanggar seperti penggunaan aksesoris pada kepala.
7. Masing-masing model mampu mendeteksi pelanggaran pada penumpang yang dibonceng.
8. Pada deteksi menggunakan YOLOv3-*tiny*, dapat mendeteksi hampir secara keseluruhan objek, baik pelanggar maupun bukan pelanggar sebagai kelas tanpa helm, hal ini terlihat ketika hasil deteksi pada masing-masing model dijalankan secara bersamaan dan dibandingkan satu dengan yang lain, sehing-

ga jika dilihat pada pengujian lokasi maupun kondisi, pada deteksi menggunakan YOLOv3-*tiny* mendapatkan nilai *mean Average Precision* (mAP) yang kecil meski dapat mendeteksi segala objek penlanggar.

9. Penggunaan garis deteksi sebagai penghitungan pelanggaran belum maksimal. Hal ini ditunjukkan terdapat lebih dari satu gambar pada pelanggar yang sama.

5.2 Saran

Untuk pengembangan lebih lanjut pada penelitian tugas akhir ini, terdapat beberapa beberapa saran yang dapat dilakukan, antara lain:

1. Memperbanyak data pelatihan pada kelas tanpa helm guna memaksimalkan hasil deteksi terutama untuk memaksimalkan tingkat akurasi YOLOv3-*tiny*.
2. Membuat tampilan antarmuka untuk mempermudah dalam pengoperasian.
3. Pemanfaatan data rekaman kamera IP pada kondisi yang melibatkan pedestrian dengan jumlah banyak untuk mendapatkan himpunan data pelatihan untuk kelas tanpa helm
4. Melanjutkan penelitian dengan pengembangan *objek tracking* sebagai pengganti sistem deteksi yang sebelumnya menggunakan garis deteksi.
5. Memperbanyak variasi karakteristik penggunaan aksesoris pada kepala pengendara yang tidak menggunakan helm.

DAFTAR PUSTAKA

[1] Badan Pusat Statistik, "Statistik transportasi darat 2017," 2018. Diakses: 2019-11-20. (Dikutip pada halaman i, iii, 1).

[2] Korlantas RI, "5 pelanggaran dengan jumlah tertinggi." http://www.korlantas-irsms.info/graph/violationTypeData, 2019. Diakses: 2019-11-20. (Dikutip pada halaman i, iii, 1).

[3] Dewan Perwakilan Rakyat (DPR), "Undang-undang republik indonesia nomor 22 tahun 2009 tentang lalu lintas dan angkutan jalan." http://www.dpr.go.id/dokjdih/document/uu/UU_2009_22.pdf, 2009. (Dikutip pada halaman 5).

[4] E. Alpaydin, Introduction to Machine Learning. The MIT Press, 2004. (Dikutip pada halaman 8).

[5] Ian Goodfellow, Yoshua Bengio, Aaron Courville, vol. 1. MIT Press, 2016. http://www.deeplearningbook.org. (Dikutip pada halaman 8, 13).

[6] J. W. G. Putra, Pengenalan Konsep Pembelajaran Mesin dan Deep Learning. 2019. (Dikutip pada halaman 9, 10, 19, 20).

[7] R. S. I Wayan Suartika E. P, Arya Yudhi Wijaya, "Klasifikasi citra menggunakan convolutional neural network (cnn) pada caltech 101," JURNAL TEKNIK ITS, vol. 5, no. 1, 2016. (Dikutip pada halaman 9, 10, 11, 12).

[8] V. Tyagi, Understanding Digital Image Processingg. CRC Press, 2018. (Dikutip pada halaman 13, 14, 15).

[9] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 779–788, 2016. (Dikutip pada halaman 14, 16, 18).

[10] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," arXiv preprint arXiv:1804.02767, 2018. (Dikutip pada halaman 17, 18, 19, 35, 38).

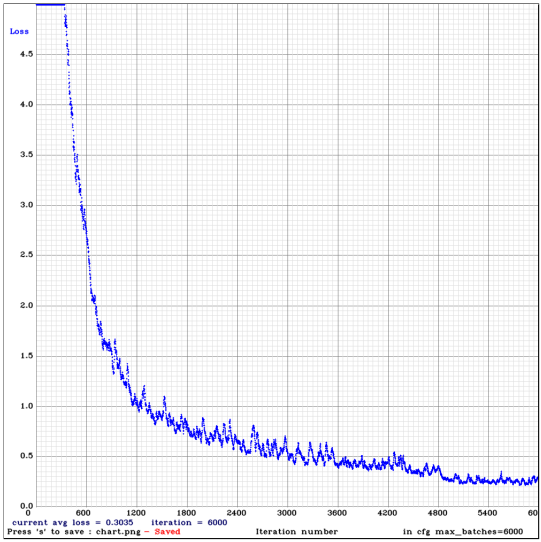
[11] Z. W. C. L. B. G. Wangpeng He, Zhe Huang, "Tf-yolo: An improved incremental network for real-time object detection," 2019. (Dikutip pada halaman 20, 36).

[12] Geeks for Geeks, "Confusion matrix in machine learning." https://www.geeksforgeeks.org/confusion-matrix-machine-learning/. Diakses: 2020-04-04. (Dikutip pada halaman 21, 24).

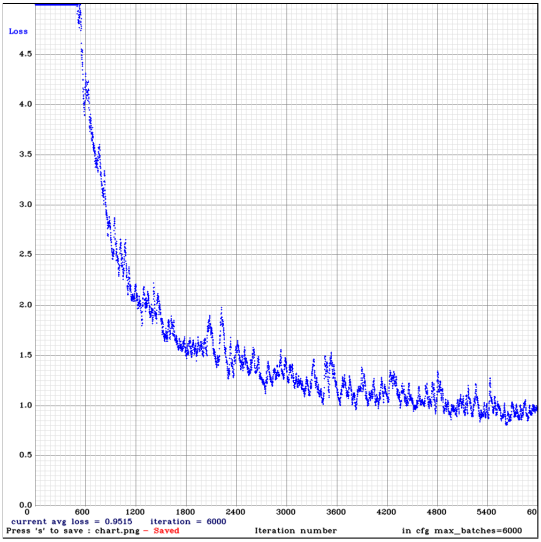
[13] A. Rosebrock, "Intersection over union (iou) for object detection." https://www.pyimagesearch.com/2016/11/07/intersection-over-union-iou-for-object/-detection/, 2016. Diakses: 2020-04-04. (Dikutip pada halaman 21, 22, 23).

[14] J. Hui, "map (mean average precision) for object detection." https://medium.com/@jonathan_hui/map-mean-average-precision-for-object-detection-45c121a31173, 2018. Diakses: 2020-04-04. (Dikutip pada halaman 22).

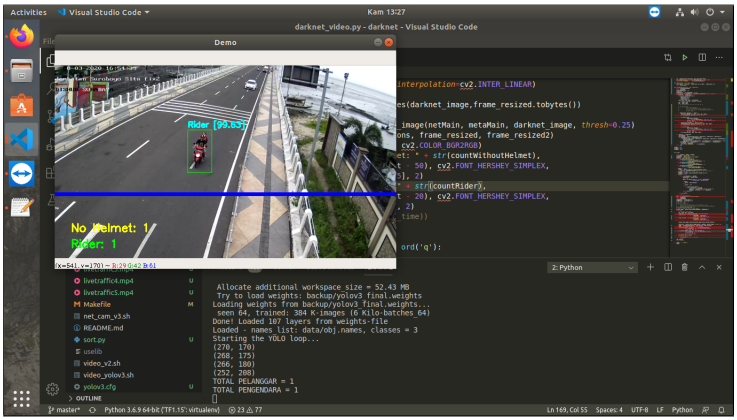
LAMPIRAN



Gambar 1: Grafik *Average Loss* terhadap Iterasi pada Proses Training YOLOv3.



Gambar 2: Grafik *Average Loss* terhadap Iterasi pada Proses Training YOLOv3-*tiny*.



Gambar 3: Proses *Running Program* Deteksi Penggunaan Helm.

BIOGRAFI PENULIS

Yusuf Umar Hanafi yang disapa Yusuf, lahir di Lamongan Jawa Timur pada tanggal 18 Juni 1998. Merupakan anak ketiga dari tiga bersaudara. Penulis lulus dari SMP Negeri 2 Sugio dan melanjutkan ke SMA Negeri 1 Babat. Penulis melanjutkan ke jenjang strata satu di Departemen Teknik Komputer Institut Teknologi Sepuluh Nopember (ITS). Dalam masa kuliah, penulis tertarik dengan pengembangan *Mobile Programming* dan *Image Proccessing*. Selain itu, penulis juga aktif dalam kegiatan organisasi seperti Badan Eksekutif Mahasiswa ITS (BEM ITS) selama 3 periode dan juga aktif dalam kegiatan komunitas sosial seperti Peduli Sekitar dan ITS Mengajar. Penulis juga aktif ikut serta dalam kompetisi pengembangan aplikasi dan menjuarai di beberapa kompetisi seperti MAGE ITS 2018, GEMASTIK XII 2019 dan Startup Weekend Bandung 2018. Bagi pembaca yang memiliki kritik, saran, atau pertanyaan mengenai tugas akhir ini dapat menghubungi penulis melalui surel flyme.yusuf@gmail.com.

Halaman ini sengaja dikosongkan