



TUGAS AKHIR - EC 184801

**ANALISIS PERFORMA SISTEM Pendetksi WAJAH
BERBASIS *CONVOLUTIONAL NEURAL NETWORK*
(CNN)**

Rizky Najwa
NRP 07211540000039

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

DEPARTEMEN TEKNIK KOMPUTER
Fakultas Teknologi Elektro
Institut Teknologi Sepuluh Nopember
Surabaya 2019



TUGAS AKHIR - EC 184801

**ANALISIS PERFORMA SISTEM Pendetksi WAJAH
BERBASIS CONVOLUTIONAL NEURAL NETWORK
(CNN)**

Rizky Najwa
NRP 07211540000039

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

DEPARTEMEN TEKNIK KOMPUTER
Fakultas Teknologi Elektro
Institut Teknologi Sepuluh Nopember
Surabaya 2019



FINAL PROJECT - EC 184801

**PERFORMANCE ANALYSIS OF FACE DETECTION
SYSTEM BASED ON CONVOLUTIONAL NEURAL
NETWORK (CNN)**

Rizky Najwa
NRP 07211540000039

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

Departement of Computer Engineering
Faculty of Electrical Technology
Sepuluh Nopember Institute of Technology
Surabaya 2019

PERNYATAAN KEASLIAN TUGAS AKHIR

Dengan ini saya menyatakan bahwa isi sebagian maupun keseluruhan Tugas Akhir saya dengan judul “**Analisis Performa Sistem Pendekripsi Wajah Berbasis Convolutional Neural Network (CNN)**” 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, Juni 2019



Rizky Najwa

NRP. 07211540000039

LEMBAR PENGESAHAN

ANALISIS PERFORMA SISTEM PENDETEKSI WAJAH BERBASIS CONVOLUTIONAL NEURAL NETWORK (CNN)

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

Oleh: Rizky Najwa (NRP: 07211540000039)

Tanggal Ujian : 20 Juni 2019

Periode Wisuda : September 2019

Disetujui oleh:

Reza Fuad Rachmadi, ST., MT., Ph.D.
NIP: 198504032012121001

(Pembimbing I)

(Pembimbing II)

(Pengaji I)

(Pengaji II)

(Pengaji III)

Eko Pramunanto, ST., MT.
NIP. 196612031994121001

Arief Kurniawan, ST., MT.
NIP. 197409072002121001



Mengetahui

Kepala Departemen Teknik Komputer

D.I. Herut Eddy Purwama, ST., MT.
NIP. 196907301995121001

ABSTRAK

Nama Mahasiswa	:	Rizky Najwa
Judul Tugas Akhir	:	Analisis Performa Sistem Pendekripsi Wajah Berbasis <i>Convolutional Neural Network (CNN)</i>
Pembimbing	:	<ol style="list-style-type: none">1. Reza Fuad Rachmadi, S.T., M.T., Ph.D.2. Dr. Supeno Mardi Susiki Nugroho, S.T., M.T.

Pengenalan wajah merupakan sebuah proses pengambilan citra atau data dari wajah seseorang dan kemudian menghubungkan data tersebut dengan seorang individu yang spesifik. Karena wajah merupakan salah satu karakteristik fisik manusia sulit untuk dipindah tangankan, maka sistem pengenalan wajah sering diaplikasikan ke dalam sistem autentifikasi identitas dan sistem keamanan yang berbasis biometrik. Namun, masih terdapat beberapa tantangan dalam pengenalan wajah, yang menyebabkan pengenalan wajah sulit dilakukan. Sistem pengenalan wajah terdiri dari tiga proses utama yaitu deteksi wajah, ekstraksi fitur, dan pengenalan wajah. Pada tugas akhir ini akan dianalisis performa pada tahap deteksi wajah yang dibuat dengan menggunakan sistem berbasis *Convolutional Neural Network (CNN)*, yaitu *Single Shot Multibox Detector (SSD)* dan *You Only Look Once (YOLO)*. Pengujian dilakukan dengan melakukan perekaman skenario kondisi wajah yang kemudian dilewatkan ke sistem berbasis CNN dan dianalisis dengan menggunakan nilai *mean Average Precision (mAP)*. Berdasarkan hasil pengujian, sistem berbasis SSD mampu melakukan pendekripsi pada skenario pengujian di siang hari dengan rata-rata waktu pemrosesan selama 8.15 detik, rata-rata FPS sebesar 25.82 fps, dan rata-rata nilai mAP sebesar 48.11%. Pada malam hari sistem berbasis SSD membutuhkan rata-rata waktu pemrosesan selama 7.92 detik, rata-rata FPS sebesar 26.4 fps, dan rata-rata nilai mAP sebesar 34.69%. Sedangkan sistem berbasis YOLO mampu melakukan pendekripsi pada skenario pengujian di siang hari dengan rata-rata waktu pemrosesan selama 30.49 detik, rata-rata FPS sebesar 6.89 fps, dan rata-rata nilai mAP sebesar 60.65%. Pada malam hari sistem berbasis YOLO membu-

tuhkan rata-rata waktu pemrosesan selama 30.48 detik, rata-rata FPS sebesar 6.82 fps, dan rata-rata nilai mAP sebesar 71.65%.

Kata Kunci: Analisis Performa Sistem, Sistem Pendekripsi Wajah, *Single Shot Multibox Detector* (SSD), *You Only Look Once* (YOLO), *Convolutional Neural Network* (CNN)

ABSTRACT

*Name : Rizky Najwa
Title : Performance Analysis of Face Detection System based on Convolutional Neural Network (CNN)
Advisors : 1. Reza Fuad Rachmadi, S.T., M.T., Ph.D.
2. Dr. Supeno Mardi Susiki Nugroho, S.T., M.T.*

Face recognition is a process of taking an image or visual data from someone's face and then connecting that data with a specific individual. Because faces are one of the physical characteristics of humans that are difficult to transfer, the face recognition system is often applied to identity authentication systems and biometric-based security systems. However, there are still some challenges in face recognition, which makes face recognition difficult. The face recognition system consists of three main parts, namely face detection, feature extraction, and facial recognition. In this final project, the performance of the face detection stage will be analyzed using Convolutional Neural Network (CNN) based systems, namely Single Shot Multibox Detector (SSD) and You Only Look Once (YOLO). Testing is done by recording face condition scenarios which are then passed to CNN-based systems and analyzed using mean Average Precision (mAP) value. Based on the test results, the SSD-based system is able to detect the test scenario during the day with an average processing time of 8.15 seconds, the average FPS is 25.82 fps, and the average mAP value is 48.11%. At night the SSD-based system requires an average processing time of 7.92 seconds, the average FPS is 26.4 fps, and the average mAP value is 34.69%. While the YOLO-based system is able to detect the test scenario during the day with an average processing time of 30.49 seconds, the average FPS is 6.89 fps, and the average mAP value is 60.65%. At night the YOLO-based system requires an average processing time of 30.48 seconds, the average FPS is 6.82 fps, and the average mAP value is 71.65%.

Keywords: System Performance Analysis, Face Detection System, Single Shot Multibox Detector (SSD), You Only Look Once

(YOLO), Convolutional Neural Network (CNN)

KATA PENGANTAR

Puji dan syukur kehadirat Allah Swt. atas segala limpahan berkah, rahmat, serta hidayah-Nya, penulis dapat menyelesaikan penelitian ini dengan judul **Analisis Performa Sistem Pendekripsi Wajah Berbasis *Convolutional Neural Network (CNN)***.

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. Semua keluarga dan kedua orang tua tercinta dan adik-adik saya yang telah memberikan motivasi, dorongan spiritual, dan material dalam penyelesaian buku penelitian ini.
2. Bapak Kepala Departemen Teknik Komputer ITS Dr. I Ketut Eddy Purnama, S.T., M.T.
3. Bapak Reza Fuad Rachmadi, S.T., M.T., Ph.D., dan Bapak Dr. Supeno Mardi Susiki Nugroho, S.T., M.T. yang senantiasa memberikan motivasi, bimbingan, dan arahan selama mengerjakan penelitian.
4. Ibu Dr. Diah Puspito Wulandari, S.T., M.Sc., selaku dosen wali yang telah memberikan motivasi dan dukungan dalam pelaksanaan penelitian ini.
5. Bapak dan Ibu dosen pengajar Departemen Teknik Komputer ITS, atas pengajaran, bimbingan, serta perhatian yang diberikan kepada penulis selama ini.
6. Seluruh staf Departemen Teknik Komputer ITS yang telah membantu penulis dalam hal administrasi dan perlengkapan penelitian.
7. Seluruh mahasiswa Teknik Komputer angkatan 2013, 2014, 2015, 2016, dan seterusnya yang telah menjadi saudara terbaik selama masa studi saya di Surabaya.
8. Seluruh teman-teman Lab. B507 yang menjadikan Surabaya sebuah rumah yang indah dan menyenangkan bagi saya (Arta, Habib, Rangga, dan Shahid).
9. Teman seperjuangan dalam mengerjakan penelitian berbasis *IP Camera* dan *CNN* (Andy, Luthfi, dan Randy).

10. Seluruh asisten Lab. Komputasi Multimedia B401 yang selalu membantu dan menjadi penyemangat dalam penelitian ini.

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, Juni 2019

Rizky Najwa

DAFTAR ISI

Abstrak	i
Abstract	iii
KATA PENGANTAR	v
DAFTAR ISI	vii
DAFTAR GAMBAR	xii
DAFTAR TABEL	xv
1 PENDAHULUAN	1
1.1 Latar belakang	1
1.2 Permasalahan	2
1.3 Tujuan	2
1.4 Batasan masalah	3
1.5 Sistematika Penulisan	3
2 TINJAUAN PUSTAKA	5
2.1 <i>Machine Learning</i>	5
2.2 <i>Artificial Neural Network</i>	6
2.3 <i>Deep Learning</i>	8
2.4 <i>Convolutional Neural Network</i>	8
2.4.1 <i>Convolutional Layer</i>	9
2.4.2 <i>Pooling Layer</i>	10
2.4.3 <i>Activation Function</i>	11
2.4.4 Arsitektur CNN	11
2.5 <i>Computer Vision</i>	13
2.6 <i>Image Processing</i>	14
2.7 <i>Object Detection</i>	15
2.7.1 <i>You Only Look Once</i>	16
2.7.2 <i>Single Shot Multibox Detector</i>	20
2.8 <i>Internet Protocol Camera</i>	22

3 DESAIN DAN IMPLEMENTASI SISTEM	23
3.1 Desain Sistem	23
3.2 Alur Kerja	25
3.3 Pembuatan Skenario Pengujian	25
3.3.1 Skenario Pengujian Berdasarkan Lokasi Peletakan Kamera	25
3.3.2 Skenario Pengujian Waktu Pengambilan Data	26
3.3.3 Skenario Pengujian Berdasarkan Jumlah Objek	26
3.3.4 Skenario Pengujian Berdasarkan Kecepatan Objek Bergerak	28
3.3.5 Skenario Pengujian Berdasarkan Perangkat Keras yang Digunakan	28
3.3.6 Skenario Pengujian dengan Tambahan Pencahayaan Buatan	28
3.3.7 Skenario Pengujian dengan Ketinggian Kamera Sejajar dengan Wajah	30
3.3.8 Skenario Pengujian Berdasarkan Sudut Kamera	30
3.4 Akuisisi Data Video	31
3.4.1 Posisi Peletakan Kamera	31
3.5 Lokasi dan Kondisi Akuisisi Data	31
3.5.1 <i>IP camera</i> yang Digunakan	32
3.5.2 Pengambilan Data Video	32
3.6 Pelabelan Objek	37
3.7 Pendekripsi Wajah	37
3.7.1 Proses Pendekripsi Wajah pada Sistem berbasis YOLO	38
3.7.2 Proses Pendekripsi Wajah pada Sistem berbasis SSD	41
3.8 Analisis Hasil Pendekripsi Wajah	45
3.8.1 <i>Confusion Matrix</i>	47
3.8.2 <i>Intersection over Union</i>	48
3.8.3 <i>mean Average Precision</i> (mAP)	50
4 PENGUJIAN DAN ANALISIS	51
4.1 Pengujian Berdasarkan Lokasi Peletakan Kamera	51
4.2 Pengujian Berdasarkan Waktu Pengambilan Data	58
4.3 Pengujian Berdasarkan Jumlah Objek	64
4.4 Pengujian Berdasarkan Kecepatan Objek Bergerak	69

4.5 Pengujian Berdasarkan Perangkat Keras yang Digunakan	71
4.6 Pengujian dengan Tambahan Pencahayaan Buatan	75
4.7 Pengujian dengan Ketinggian Kamera Sejajar dengan Wajah	80
4.8 Pengujian Berdasarkan Sudut Kamera	83
4.9 Rekomendasi Penerapan Sistem	86
5 PENUTUP	91
5.1 Kesimpulan	91
5.2 Saran	92
DAFTAR PUSTAKA	95
LAMPIRAN	111
Biografi Penulis	111

Halaman ini sengaja dikosongkan

DAFTAR GAMBAR

2.1	Operasi <i>Perceptron</i> pada <i>Neural Network</i>	7
2.2	Proses <i>Max Pooling</i>	11
2.3	Proses <i>Feature Extraction</i>	13
2.4	Sistem Pendekripsi YOLO	17
2.5	Proses Pendekripsi Objek YOLO	18
2.6	Arsitektur YOLO	19
2.7	Perbandingan Arsitektur SSD dengan Arsitektur YOLO	21
3.1	Diagram Alir Analisis Performa Sistem Pendekripsi Wajah	24
3.2	Skenario Pengujian Berdasarkan Jumlah Objek	29
3.3	Ilustrasi Peletakan Kamera Pertama	31
3.4	Ilustrasi Peletakan Kamera Kedua	32
3.5	Lokasi Peletakan Kamera Pertama	33
3.6	Lokasi Peletakan Kamera Kedua	33
3.7	Jenis Kamera yang Digunakan	35
3.8	Pengambilan Video pada Waktu Siang Hari	36
3.9	Pengambilan Video pada Waktu Malam Hari	36
3.10	Diagram Alir Pendekripsi Objek	38
3.11	Susunan <i>Layer</i> Darknet53	39
3.12	Prediksi Lokasi <i>Bounding Box</i>	40
3.13	Diagram Proses <i>Depthwise Separable Convolutions</i>	43
3.14	Arsitektur <i>MobileNet</i>	44
3.15	Perbandingan antara Konvolusi Standar dan <i>Depthwise Separable Convolutions</i>	44
3.16	Diagram Alir Proses Analisis	46
3.17	Contoh <i>Bounding Boxes</i> pada Objek yang Didekripsi	49
3.18	Variasi Hasil Skor IoU	49
3.19	<i>Precision-Recall Curve</i>	50
4.1	Potongan Gambar Wajah dari Data Video Kamera yang Membelakangi Sinar Matahari	56
4.2	Potongan Gambar Wajah dari Data Video Kamera yang Menghadap Sinar Matahari	57

4.3	Dokumentasi Pengujian Pada Kamera yang Membelakangi Sinar Matahari	57
4.4	Dokumentasi Pengujian Pada Kamera yang Menghadap Sinar Matahari	58
4.5	Potongan Gambar Wajah dari Data Video Kamera pada Waktu Malam Hari	63
4.6	Dokumentasi Pengujian pada Waktu Siang Hari	64
4.7	Dokumentasi Pengujian pada Waktu Malam Hari	64
4.8	Dokumentasi Pengujian Berdasarkan Jumlah Objek	67
4.9	Dokumentasi Pengujian Berdasarkan Jumlah Objek	67
4.10	Pendeteksian Wajah di Pasar Jumat ITS dengan Sistem Berbasis SSD	68
4.11	Pendeteksian Wajah di Pasar Jumat ITS dengan Sistem Berbasis YOLO	68
4.12	Dokumentasi Pengujian Berdasarkan Kecepatan Objek Bergerak	70
4.13	Ilustrasi Peletakan Kamera dan Lampu LED	77
4.14	Konfigurasi Lampu LED dan Kamera dengan Ketinggian Sejajar dengan Wajah	77
4.15	Potongan Gambar Wajah dari Data Video Kamera dengan Tambahan Pencahayaan Buatan	79
4.16	Ilustrasi Peletakan Kamera dan Lampu LED	82
4.17	Jarak Terjauh atau Minimum Pendeksi Wajah dengan Sistem Berbasis SSD	88
4.18	Jarak Terdekat atau Maksimum Pendeksi Wajah dengan Sistem Berbasis SSD	88
4.19	Jarak Terjauh atau Minimum Pendeksi Wajah dengan Sistem Berbasis YOLO	89
4.20	Jarak Terdekat atau Maksimum Pendeksi Wajah dengan Sistem Berbasis YOLO	89
1	Grafik perbandingan nilai mAP saat kamera menghadap ke arah timur	99
2	Grafik perbandingan nilai mAP saat kamera menghadap ke arah timur	100
3	Grafik perbandingan nilai mAP pada saat siang hari	101
4	Grafik perbandingan nilai mAP pada saat malam hari	102

5	Grafik perbandingan nilai mAP berdasarkan jumlah objek	103
6	Grafik perbandingan rata-rata nilai mAP berdasarkan kecepatan objek	104
7	Grafik perbandingan rata-rata waktu pemrosesan berdasarkan perangkat keras yang digunakan	105
8	Grafik perbandingan rata-rata nilai FPS berdasarkan perangkat keras yang digunakan	106
9	Grafik perbandingan rata-rata nilai mAP berdasarkan perangkat keras yang digunakan	107
10	Grafik perbandingan rata-rata nilai mAP pada pengujian dengan tambahan pencahayaan buatan . . .	108
11	Grafik perbandingan rata-rata nilai mAP pada ketinggian kamera yang berbeda	109
12	Grafik perbandingan nilai mAP Berdasarkan Sudut Kamera	110

Halaman ini sengaja dikosongkan

DAFTAR TABEL

3.1	Skenario Pengujian Berdasarkan Lokasi Peletakan Kamera.	26
3.2	Skenario Pengujian Berdasarkan Waktu Pengambilan Data.	27
3.3	Spesifikasi <i>IP Camera</i>	34
3.4	<i>Confusion Matrix</i> untuk Dua Kelas	47
4.1	Perbandingan <i>Brightness</i> dan <i>Contrast</i> pada Lokasi Peletakan Kamera yang Berbeda	53
4.2	Hasil Pengujian Berdasarkan Lokasi Peletakan Kamera Membelakangi Sinar Matahari	54
4.3	Hasil Pengujian Berdasarkan Lokasi Peletakan Kamera Menghadap Sinar Matahari	55
4.4	Perbandingan <i>Brightness</i> dan <i>Contrast</i> pada Waktu yang Berbeda	60
4.5	Hasil Pengujian pada Waktu Siang Hari	61
4.6	Hasil Pengujian pada Waktu Malam Hari	62
4.7	Hasil Pengujian Berdasarkan Jumlah Objek	66
4.8	Hasil Pengujian dengan Kecepatan Objek Bergerak Normal	72
4.9	Hasil Pengujian dengan Kecepatan Objek Bergerak Agak Cepat	72
4.10	Hasil Pengujian dengan Kecepatan Objek Bergerak Cepat	73
4.11	Perbandingan Hasil Pendekripsi Sistem Berbasis SSD berdasarkan Perangkat Keras yang Digunakan	76
4.12	Perbandingan Hasil Pendekripsi Sistem Berbasis YO-LO berdasarkan Perangkat Keras yang Digunakan	76
4.13	Perbandingan <i>Brightness</i> dan <i>Contrast</i> dengan Tambahan Pencahayaan Buatan	78
4.14	Perbandingan Hasil Pendekripsi Sistem Berbasis SSD dengan Tambahan Pencahayaan Buatan	81
4.15	Perbandingan Hasil Pendekripsi Sistem Berbasis YO-LO dengan Tambahan Pencahayaan Buatan	81

4.16 Perbandingan Hasil Pendekripsi Sistem Berbasis SSD dengan Ketinggian Kamera Sejajar dengan Wajah pada Waktu Siang Hari	84
4.17 Perbandingan Hasil Pendekripsi Sistem Berbasis SSD dengan Ketinggian Kamera Sejajar dengan Wajah pada Waktu Malam Hari	84
4.18 Perbandingan Hasil Pendekripsi Sistem Berbasis YO-LO dengan Ketinggian Kamera Sejajar dengan Wajah pada Waktu Siang Hari	85
4.19 Perbandingan Hasil Pendekripsi Sistem Berbasis YO-LO dengan Ketinggian Kamera Sejajar dengan Wajah pada Waktu Malam Hari	85
4.20 Hasil Pengujian Berdasarkan Sudut Kamera	87

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

Sebuah sistem identifikasi identitas manusia yang memiliki akurasi tinggi sangat diperlukan saat ini seiring dengan meningkatnya kebutuhan akan sistem keamanan identitas yang lebih baik. Sistem keamanan lawas berbasis token (*token-based system*) dan berbasis pengetahuan (*knowledge-based system*) memiliki risiko tinggi dalam kasus pencurian atau lupa sandi. Oleh karena itu, kebanyakan sistem keamanan saat ini sudah beralih menggunakan sistem biometrik seperti pengenalan suara, pemindaian sidik jari, dan pemindaian wajah untuk melakukan verifikasi identitas. Salah satu alasan dibalik penggunaan sistem biometrik adalah karena keunikan fitur biometrik dan karakteristik yang tidak dapat dipindah tangankan (*non-transferable characteristic*). Biometrik dibagi menjadi dua kategori yaitu berdasarkan perilaku seseorang seperti ritme mengeletik, gaya berjalan, atau suara, dan berdasarkan fisiologis seseorang seperti sidik jari, wajah, iris, atau tanda tangan. Di antara tipe biometrik fisiologis yang disebutkan sebelumnya, sistem pengenalan wajah hingga saat ini masih menjadi daerah penelitian sejak tahun 1960 dengan cakupan ruang yang luas untuk terus dilakukan penyempurnaan[1].

Dalam perkembangannya, sistem pengenalan wajah semakin canggih dengan banyaknya metode yang dapat digunakan untuk melakukan deteksi dan pengenalan objek secara *real-time*, mulai dari yang menggunakan pendekatan *Machine Learning* sampai yang menggunakan pendekatan *Deep Learning*. Salah satu metode untuk pengenalan wajah terbaru yang saat ini sedang banyak dikembangkan adalah metode *Convolutional Neural Network* (CNN). CNN merupakan Jaringan Saraf Tiruan (JST) yang mampu beradaptasi serta mengubah strukturnya untuk memecahkan masalah berdasarkan

informasi yang diberikan. Sistem pengenalan wajah terdiri dari tiga proses utama: pendekripsi wajah, klasifikasi wajah, dan pengenalan wajah. Khusus untuk proses pendekripsi wajah sendiri, telah terdapat beberapa pengembangan dari CNN yang memungkinkan proses pendekripsi wajah dilakukan secara cepat dan akurat seperti *You Only Look Once* (YOLO)[2]-[3] dan *Single Shot Multi-Box Detector* (SSD)[4]. Berbeda dengan sistem deteksi kebanyakan, YOLO dan SSD menghilangkan metode *Region Proposal* untuk mendapatkan waktu pelatihan dan pemrosesan yang lebih singkat karena masukan citra akan langsung diproses oleh sebuah CNN yang besar[5].

Meskipun sistem yang ada sudah mampu melakukan pendekripsi dengan cepat dan akurat masih terdapat beberapa tantangan dalam proses pendekripsi wajah. Tantangan dalam deteksi wajah berasal dari dua aspek: 1) variasi visual dari wajah manusia dalam latar belakang yang berantakan; 2) ruang pencarian yang besar dari kemungkinan posisi dan ukuran wajah[6]. Oleh karena itu, pada tugas akhir ini, akan dilakukan analisis performa dari sistem pendekripsi wajah yang dikembangkan dengan menggunakan metode CNN. Sistem kemudian akan diuji dengan skenario variasi visual wajah dan lingkungan yang berbeda, sehingga didapatkan kondisi optimal untuk melakukan untuk pendekripsi wajah seseorang dengan menggunakan sistem tersebut.

1.2 Permasalahan

Diperlukan analisis terhadap sistem pendekripsi wajah berbasis *Convolutional Neural Network* (CNN) pada variasi visual wajah yang berbeda untuk menguji performa sistem dalam melakukan pendekripsi wajah.

1.3 Tujuan

Tujuan dari tugas akhir ini adalah sebagai berikut:

1. Membuat sebuah sistem pendekripsi wajah berbasis *Convolutional Neural Network* (CNN).
2. Melakukan analisis performa terhadap sistem pendekripsi wajah berbasis *Convolutional Neural Network* (CNN) sehingga didapatkan kondisi optimal di mana sistem tersebut dapat

mendeteksi wajah seseorang di kamera dengan akurat.

1.4 Batasan masalah

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

1. Sistem jaringan saraf tiruan yang digunakan untuk proses pendekripsi wajah adalah *Convolutional Neural Network* (CNN).
2. Data masukan yang digunakan adalah data video rekaman atau video *real-time* dari kamera IP.
3. Proses yang dianalisis dalam penelitian ini adalah proses pendekripsi dari sistem pengenalan wajah.
4. Analisis performa dilakukan dengan cara mengukur nilai *mean Average Precision* (mAP) dari pendekripsi wajah dalam kondisi cahaya, posisi kamera, halangan wajah, jumlah objek, serta kecepatan objek yang berbeda.

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), dan teori-teori penunjang lainnya.

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 pendekripsi wajah, 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 dengan kondisi cahaya yang berbeda, posisi kamera yang berbeda, jumlah dan kecepatan objek yang berbeda, 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.

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 *Machine Learning*

Machine Learning (ML) merupakan sebuah cabang dari *Artificial Intelligence* (kecerdasan buatan) yang memungkinkan sistem komputer untuk belajar langsung dari contoh, data, dan pengalaman yang didapatkannya. Dengan membuat komputer mampu menjalankan tugas yang spesifik secara cerdas, ML dapat melakukan algoritma yang kompleks dengan cara mempelajari data yang diberikan tanpa perlu mengikuti program yang dibuat atau diinstruksikan secara manual oleh pengguna[7].

Ciri khas dari ML adalah adanya proses pembelajaran atau *training*. Oleh karena itu, ML membutuhkan data untuk dipelajari yang disebut sebagai data *training*. Setelah berhasil melakukan *training*, maka ML dapat melakukan proses klasifikasi dan prediksi terhadap data baru yang diberikan sesuai dengan hasil *training* yang telah dilakukan. Klasifikasi adalah metode dalam ML yang digunakan oleh mesin untuk memilah atau mengklasifikasikan objek berdasarkan ciri tertentu sebagaimana manusia mencoba membedakan benda satu dengan yang lain. Sedangkan prediksi atau regresi digunakan oleh mesin untuk menerka keluaran dari suatu data masukan[8]. Beberapa contoh implementasi dari ML antara lain adalah *image processing*, *search recommendation*, *text and speech recognition*, dan *health diagnosis*.

Di dalam ML, terdapat tiga kategori pembelajaran[9], yaitu:

1. *Supervised Learning*

Supervised learning merupakan metode pembelajaran dimana data yang akan dipelajari sudah diberikan label yang menunjukkan klasifikasi atau kelas data tersebut berada. Model yang dihasilkan adalah model prediksi dari data yang telah diberi label.

2. *Unsupervised Learning*

Unsupervised learning merupakan metode pembelajaran dimana data yang akan dipelajari tidak memiliki label, sehingga komputer perlu melakukan analisis dan klasifikasi berdasarkan data yang ada.

3. *Reinforcement Learning*

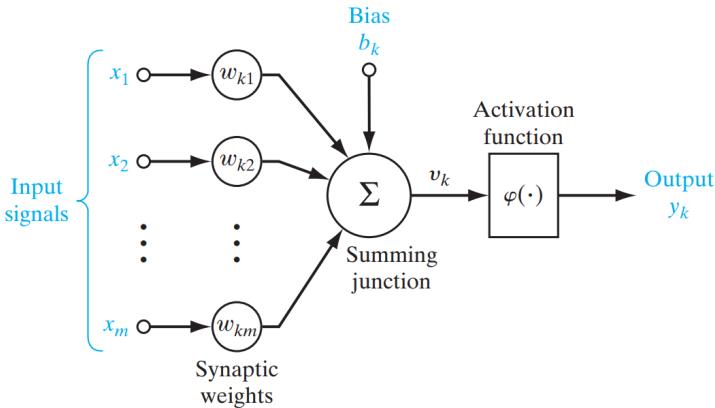
Reinforcement learning merupakan metode pembelajaran dimana diberlakukan *reward* pada hasil pembelajaran yang positif dan *punishment* pada hasil pembelajaran yang negatif. Komputer tidak diberitahu apa yang harus dilakukan, namun memiliki tujuan untuk mendapatkan *reward* yang maksimal sehingga memaksa komputer untuk belajar secara terus-menerus.

2.2 *Artificial Neural Network*

Artificial Neural Network (NN) adalah teknik dalam ML yang bertujuan untuk menirukan saraf manusia yang merupakan bagian fundamental dari otak. NN terdiri atas lapisan masukan (*input layer*) dan lapisan keluaran (*output layer*). Setiap *layer* terdiri atas satu atau beberapa unit neuron yang mempunyai sebuah fungsi aktivasi yang menentukan *output* dari unit tersebut. Kita bisa menambahkan lapisan tersembunyi (*hidden layer*) untuk menambah kemampuan dari NN tersebut. Neuron adalah unit pengolahan informasi dasar bagi operasi NN[8]. Pada tahun 1958, Frank Rosenblatt mengenalkan jaringan saraf tiruan ini sebagai *perceptron*. Gambar 2.1 memperlihatkan detail dari operasi *perceptron*. Pada Gambar 2.1, terdapat tiga elemen dasar dari neuron atau *perceptron*[10] yaitu:

1. *Synapses* atau penghubung yang dikategorikan berdasarkan bobot atau kekuatannya.
2. *Adder* yang berfungsi untuk menjumlahkan sinyal *input* yang telah diberikan bobot oleh *synapses*.
3. *Activation function* yang berfungsi untuk membatasi amplitudo *output* dari sebuah neuron. Biasanya, batas amplitudo normal dari *output* ditulis dalam bentuk interval satuan tertutup $[0, 1]$ atau $[-1, 1]$.

Secara matematis, neuron k yang terdapat di Gambar 2.1 da-



Gambar 2.1: Operasi *Perceptron* pada *Neural Network*[10].

pat dijelaskan dengan persamaan 2.1 dan persamaan 2.2.

$$u_k = \sum_{j=1}^m w_{kj} \cdot x_j \quad (2.1)$$

dan

$$y_k = \varphi(u_k + b_k) \quad (2.2)$$

dimana:

x_1, x_2, \dots, x_m = sinyal *input*

$w_{k1}, w_{k2}, \dots, w_{km}$ = bobot *synapses* dari neuron k

u_k = kombinasi linier *output* yang disebabkan oleh sinyal *input*

y_k = sinyal *output* dari neuron

b_k = bias

$\varphi(\cdot)$ = fungsi aktivasi

Penggunaan bias b_k bertujuan untuk menambahkan sebuah input *dummy* yang bernilai satu yang mana perhitungannya direpresentasikan dengan persamaan 2.3.

$$v_k = u_k + b_k \quad (2.3)$$

Bias b_k memberikan efek *affine transformation* kepada *output* u_k sehingga dihasilkan *activation potential* v_k . Dari kombinasi persamaan 2.1 dan 2.2 maka didapatkan persamaan 2.4 dan persamaan 2.5.

$$v_k = \sum_{j=0}^m w_{kj} \cdot x_j \quad (2.4)$$

$$y_k = \varphi(v_k) \quad (2.5)$$

NN bisa dilatih dengan menggunakan data *training*. Semakin banyak data *training* maka akan semakin bagus performa dari NN tersebut. Namun, kemampuan NN juga terbatas pada jumlah *layer*, semakin banyak jumlah *layer* semakin tinggi kapasitas NN tersebut. Semakin banyak *layer* juga membawa kekurangan yaitu semakin banyaknya jumlah iterasi atau *training* yang dibutuhkan. Untuk mengatasi hal ini, dikembangkanlah teknik *deep learning*[8].

2.3 Deep Learning

Deep Learning (DL) adalah salah satu bidang yang muncul dari penelitian ML. DL memungkinkan model komputasi yang tersusun dari beberapa lapisan pemrosesan untuk mempelajari representasi dari data dengan berbagai level abstraksi. DL menemukan struktur sulit yang terdapat di dalam kumpulan data yang besar dengan menggunakan algoritma *backpropagation*. Struktur yang didapatkan menunjukkan parameter internal apa yang harus diubah oleh mesin agar dapat menghitung representasi di setiap *layer* berdasarkan representasi dari *layer* sebelumnya[11].

DL melakukan pendekatan dalam penyelesaian masalah dengan menggunakan konsep hierarki. Dengan konsep tersebut, komputer mampu mempelajari sebuah konsep yang kompleks dengan menggabungkan konsep-konsep yang lebih sederhana[12].

2.4 Convolutional Neural Network

Convolutional Neural Network (CNN) adalah variasi dari *Multilayer Perceptron* yang terinspirasi dari jaringan saraf manusia. Pemberian nama *convolutional neural network* mengindikasikan bah-

wa jaringan tersebut menggunakan operasi matematika yang disebut konvolusi. Konvolusi sendiri adalah sebuah operasi linier. CNN merupakan pengembangan dari *artificial neural network* yang saat ini diklaim sebagai model terbaik untuk memecahkan masalah seputar *object recognition* dan *detection*.

Secara teknis, CNN adalah arsitektur yang bisa di *training* dan terdiri dari beberapa tahap. *Input* dan *output* dari masing-masing tahap berupa *array* yang disebut *feature map* atau peta fitur. *Output* dari masing-masing tahap adalah *feature map* hasil pengolahan dari semua lokasi pada *input*. Struktur CNN dibangun dari tiga jenis *layer* utama yaitu *convolution layer*, *pooling layer*, dan *activation function*[13].

2.4.1 Convolutional Layer

Convolutional Layer merupakan *layer* pertama yang menerima *input* citra langsung pada arsitektur. Operasi pada *layer* ini adalah operasi konvolusi yang melakukan operasi kombinasi linier filter terhadap daerah lokal. Filter merupakan representasi bidang reseptif dari neuron yang terhubung ke dalam daerah lokal pada *input* citra. *Convolutional layer* melakukan operasi konvolusi pada *output* dari *layer* sebelumnya. *Convolutional layer* adalah proses utama yang mendasari sebuah CNN.

Tujuan dilakukannya konvolusi pada data citra adalah untuk mengekstraksi fitur dari citra *input*. Secara umum operasi konvolusi dapat dituliskan dengan persamaan 2.6.

$$s(t) = (x \times w)(t) \quad (2.6)$$

Pada fungsi $s(t)$ menghasilkan *output* tunggal yaitu *feature map*. Argumen pertama berupa *input* yang merupakan x dan argumen kedua yang merupakan w sebagai *kernel* atau filter. Jika melihat *input* sebagai citra dua dimensi, maka (t) bisa diasumsikan sebagai sebuah piksel dan menggantinya dengan i dan j . Oleh karena itu, untuk operasi konvolusi dengan lebih dari satu dimensi dapat digunakan persamaan 2.7.

$$S_{(i,j)} = (K \times I)_{(i,j)} = \Sigma \Sigma I_{(i-m,j-n)} K_{(m,n)} \quad (2.7)$$

Persamaan 2.7 merupakan perhitungan dalam operasi konvolusi dengan i dan j sebagai piksel dari sebuah citra. Perhitungannya bersifat komutatif dan muncul ketika $K_{(m,n)}$ sebagai *kernel*-nya serta I sebagai *input* dan *kernel* yang dapat dibalik relatif terhadap *input*. Operasi konvolusi dapat dilihat sebagai perkalian matriks antara citra *input* dan filter dimana *output*-nya dapat dihitung dengan *dot product*.

2.4.1.1 *Stride*

Stride adalah parameter yang menentukan berapa jumlah pgeseran filter. Jika nilai *stride* adalah satu, maka *feature map* akan bergeser sebanyak 1 piksel secara horizontal lalu vertikal. Semakin kecil *stride* yang digunakan, maka semakin detail informasi yang didapatkan dari sebuah *input*, namun membutuhkan komputasi lebih jika dibandingkan dengan *stride* yang besar.

2.4.1.2 *Padding*

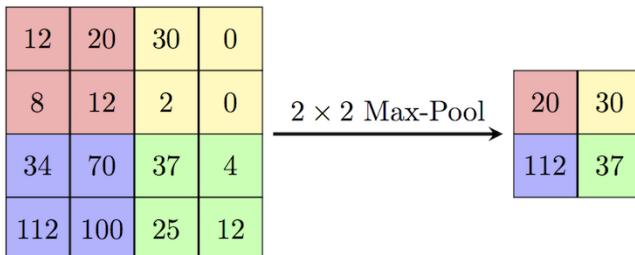
Padding atau *zero padding* adalah parameter yang menentukan jumlah piksel (berisi nilai nol) yang akan ditambahkan di setiap sisi dari *input*. Hal ini digunakan dengan tujuan untuk memanipulasi dimensi *output* dari *feature map*. Penggunaan *padding* dapat digunakan untuk mengatur dimensi *output* agar tetap sama seperti dimensi *input* atau setidaknya tidak kurang drastis sehingga dapat dilakukan ekstraksi fitur yang lebih mendalam.

2.4.1.3 *Loss Function*

Loss function merupakan fungsi yang menggambarkan kerugian yang terkait dengan semua kemungkinan yang dihasilkan oleh sebuah model. *Loss function* ini bekerja ketika model pembelajaran memberikan kesalahan yang harus diperhatikan. *Loss function* yang baik memberikan nilai *error* yang rendah.

2.4.2 *Pooling Layer*

Pooling atau *subsampling* adalah pengurangan ukuran matriks dengan menggunakan operasi *pooling*. *Pooling layer* biasanya dilakukan setelah *convolutional layer*. Terdapat dua macam *pooling* yang sering dipakai yaitu *average pooling* dan *max pooling*. Dalam *average pooling*, nilai yang diambil adalah nilai rata-rata, sementara



Gambar 2.2: Proses *Max Pooling*[14].

pada *max pooling*, nilai yang diambil adalah nilai maksimal.

Gambar 2.2 menunjukkan adanya operasi *max pooling* untuk citra berukuran 4×4 dengan *pooling mask* yang berukuran 2×2 . *Output* dari proses *pooling* adalah matriks dengan dimensi yang lebih kecil dibanding dengan matriks awal. Proses konvolusi dan *pooling* dilakukan beberapa kali sehingga didapatkan *feature map* dengan ukuran yang dikehendaki. *Feature map* tersebut akan menjadi *input* bagi *fully connected neural network*[14].

2.4.3 Activation Function

Activation function atau fungsi aktivasi merupakan sebuah *node* yang ditambahkan di akhir *output* dari setiap *neural network*. Fungsi aktivasi juga dikenal sebagai *transfer function* yang digunakan untuk menentukan *output* dari *neural network*. Fungsi aktivasi dibagi menjadi dua tipe yaitu linier dan *non-linier*. Pada arsitektur CNN, fungsi aktivasi terletak pada perhitungan akhir *output feature map*, sesudah proses perhitungan konvolusi atau *pooling*, untuk menghasilkan suatu pola fitur. Beberapa macam fungsi aktivasi yang sering digunakan dalam penelitian antara lain fungsi sigmoid, tanh, dan *Rectified Linear Unit* (ReLU).

2.4.4 Arsitektur CNN

Arsitektur dari CNN terbagi menjadi dua bagian utama, yaitu *feature extraction layer* yang mengubah citra menjadi *feature map* dan *fully connected layer* yang melakukan transformasi pada dimensi data *feature map* agar data dapat diklasifikasikan secara linear.

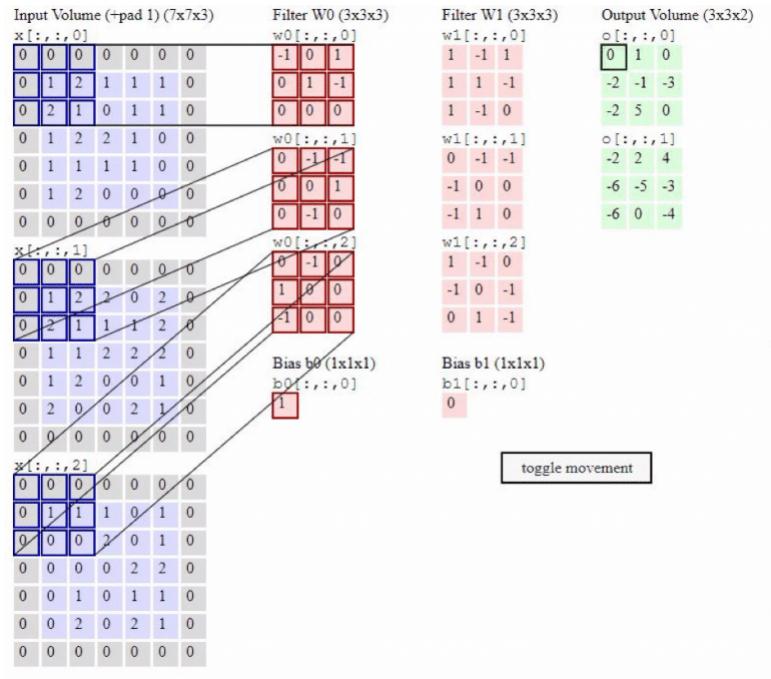
2.4.4.1 *Feature Extraction Layer*

Proses yang terjadi pada *feature extraction layer* adalah melakukan *encoding* dari sebuah citra menjadi *feature map* yang berupa angka-angka yang mempresentasikan citra tersebut dengan menggunakan *convolutional layer*. *Convolutional layer* terdiri dari neuron yang tersusun sehingga membentuk sebuah filter dengan panjang dan tinggi (piksel). Gambar 2.3 menunjukkan visualisasi dari proses *feature extraction* yang terjadi pada *convolutional layer*. Sebagai contoh, pada *layer* pertama pada tahap *feature extraction layer* biasanya terdapat *convolutional layer* dengan ukuran $5 \times 5 \times 3$. Panjang 5 piksel, tinggi 5 piksel, dan jumlahnya 3 buah sesuai dengan *channel* dari citra tersebut, yaitu *Red*, *Green*, *Blue* (RGB). Ketiga filter ini akan digeser ke seluruh bagian dari citra. Setiap pergeseran akan dilakukan operasi *dot matrix* antara *input* dan nilai dari filter tersebut, sebagaimana yang terdapat dalam persamaan 2.7, sehingga menghasilkan sebuah *output* atau biasa disebut dengan *activation map* atau *feature map*[15].

2.4.4.2 *Fully Connected Layer*

Feature map yang dihasilkan dari *feature extraction layer* masih berbentuk *multidimensional array*, sehingga harus dilakukan *flatten* atau *reshape* *feature map* menjadi sebuah *vector* agar bisa digunakan sebagai *input* dari *fully connected layer*. *Layer* ini memiliki *hidden layer*, *activation function*, *output layer*, dan *loss function*. *Layer* ini adalah *layer* yang biasanya digunakan dalam penerapan *multi layer perceptron* dan bertujuan untuk melakukan transformasi pada dimensi data agar data dapat diklasifikasikan secara linear[15].

Fully connected layer adalah *layer* dimana semua neuron aktifasi dari *layer* sebelumnya terhubung satu sama lain dengan neuron di *layer* selanjutnya. Perbedaan antara *fully connected layer* dan *convolutional layer* biasa adalah neuron di *convolutional layer* terhubung hanya ke daerah tertentu pada *input*, sementara *fully connected layer* memiliki neuron yang secara keseluruhan terhubung. Namun, kedua *layer* tersebut masih mengoperasikan produk *dot*, sehingga fungsinya tidak begitu berbeda.



Gambar 2.3: Proses Feature Extraction[15].

2.5 Computer Vision

Vision secara bahasa dapat diartikan sebagai penglihatan. *Vision* juga dapat diartikan sebagai suatu proses pengamatan apa yang ada pada dunia nyata melalui panca indra penglihatan manusia. Adapun *computer vision* adalah suatu pembelajaran menganalisis citra dan video oleh komputer untuk memperoleh hasil sebagaimana yang bisa dilakukan manusia. Pada hakikatnya, *computer vision* mencoba meniru cara kerja sistem visual manusia (*Human Vision*). Manusia melihat objek dengan indra penglihatan (mata), lalu citra objek diteruskan ke otak untuk diinterpretasi sehingga manusia mengerti objek apa yang tampak dalam pandangan matanya. Hasil interpretasi ini kemudian digunakan untuk pengambilan keputusan seperti menghindari pejalan kaki ketika sedang mengen-

darai sebuah mobil atau memanggil nama seseorang ketika mengenali wajahnya. Pada komputer, hal ini dilakukan dengan melakukan penangkapan citra atau video melalui kamera, lalu dilakukan proses analisis terhadap gambar tersebut. Hasil analisis digunakan untuk melakukan keputusan-keputusan yang dibuat berdasarkan kondisi citra atau video yang ditangkap oleh kamera. *Computer vision* dibuat agar dapat membantu manusia melakukan proses pengamatan dan pengambilan keputusan yang sulit jika dilakukan dalam kondisi yang spesifik[16].

2.6 *Image Processing*

Di dalam bidang komputer terdapat tiga bidang studi yang berkaitan dengan data citra, namun dengan tujuan yang berbeda[16], yaitu:

1. Grafika Komputer

Grafika Komputer bertujuan menghasilkan citra dengan primitif-primitif geometri seperti garis, lingkaran dan sebagainya. Primitif-primitif geometri tersebut memerlukan data deskriptif untuk melukis elemen-elemen gambar. Contoh data deskriptif adalah koordinat titik, panjang garis, jari-jari lingkaran, tebal garis, warna, dan sebagainya. Grafika komputer memainkan peranan penting dalam visualisasi dan *virtual reality*.

2. Pengolahan Citra

Pengolahan Citra bertujuan memperbaiki kualitas citra agar mudah diinterpretasi oleh manusia atau mesin (komputer). Teknik-teknik pengolahan citra mentransformasikan citra menjadi citra lain. Jadi, masukannya adalah citra dan keluarannya juga citra, namun dengan kualitas yang lebih baik daripada citra masukan.

3. Pengenalan Pola

Pengenalan pola mengelompokkan data numerik dan simbolik (termasuk citra) secara otomatis oleh mesin (dalam hal ini komputer). Tujuan pengelompokan adalah untuk mengenali suatu objek di dalam citra. Manusia bisa mengenali objek yang dilihatnya karena otak manusia telah belajar mengklasifikasi objek-objek di alam sehingga mampu membedakan suatu objek dengan objek lainnya. Kemampuan sistem visual

manusia inilah yang dicoba ditiru oleh mesin. Komputer menerima masukan berupa citra objek yang akan diidentifikasi, memproses citra tersebut, dan memberikan keluaran berupa deskripsi objek di dalam citra.

2.7 Object Detection

Perkembangan object detection berjalan beriringan dengan berkembangnya *convolutional networks* yang dimulai pada tahun 1998 oleh LeCun et al. Mereka mengembangkan *LeNet-5*, pioner dari jaringan konvolusional 7 level, yang dapat melakukan klasifikasi digit dan diterapkan oleh beberapa bank untuk mengenali nomor tulisan tangan pada cek yang didigitalkan dalam bentuk *input* citra *grayscale* berukuran 32×32 piksel. Namun pada saat itu kemampuan memproses gambar dengan resolusi yang lebih tinggi masih belum dapat dilakukan karena keterbatasan sumber daya komputasi.

ImageNet adalah database visual besar yang dirancang untuk digunakan dalam penelitian perangkat lunak pada bidang pengenalan objek visual. *ImageNet* menjalankan kontes perangkat lunak tahunan, *ImageNet Large Scale Visual Recognition Challenge* (IL-SVRC), dimana berbagai metode bersaing untuk mengklasifikasikan dan mendeteksi objek dan adegan dengan benar. Pada tahun 2012, AlexNet memenangkan ILSVRC 2012, dimana mereka menggunakan arsitektur yang mirip dengan LeNet namun dengan lapisan *layer* yang lebih dalam, filter yang lebih banyak pada tiap *layer*-nya, dan tumpukan *convolutional layers*. AlexNet berhasil memenangkan IL-SVRC dengan *top-5 error* sebesar 15,3%. Sejak keberhasilan AlexNet, ILSVRC selalu dimenangkan oleh metode berbasis CNN, mulai dari ZFNet (2013), GoogleNet/Inception dan VGGNet sebagai runner-up (2014), dan ResNet (2015) yang top-5 error-nya berhasil mengalahkan level kemampuan manusia dengan nilai 3,57% [17].

Hasil-hasil tersebut membuat metode klasifikasi lama berbasis *Histogram of Oriented Gradient* (HOG) mulai digantikan oleh program klasifikasi berbasis CNN yang lebih akurat. Namun, masih terdapat sebuah masalah, yaitu CNN masih terlalu lambat dan biaya komputasinya sangat mahal. R-CNN memecahkan masalah ini dengan menggunakan algoritma *object proposal* yang disebut dengan *selective search* yang bertujuan untuk mengurangi jumlah *bounding box* yang diberikan kepada *classifier* sampai sekitar 2000 *region pro-*

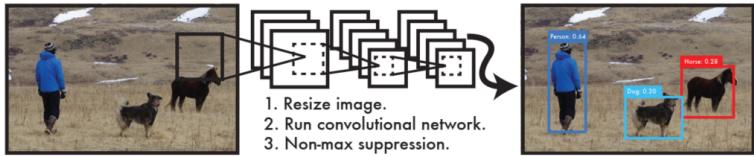
posal. R-CNN kemudian dikembangkan lagi menjadi Fast R-CNN dan Faster R-CNN yang memberikan peningkatan kecepatan hingga 250 kali lipat dari R-CNN biasa.

Beberapa metode di atas mengangani deteksi sebagai masalah klasifikasi dengan cara membangun *pipeline*, dimana dilakukan proposal terlebih dahulu terhadap objek yang ingin dideteksi lalu kemudian proposal tersebut dimasukkan ke dalam sistem klasifikasi. Namun, seiring perkembangan sistem pengenalan objek visual, terdapat beberapa metode yang menjadikan proses pendekripsi sebagai masalah regresi, dua metode yang paling populer adalah *You Only Look Once* (YOLO) dan *Single Shot Multibox Detector* (SSD)[18].

2.7.1 You Only Look Once

You Only Look Once (YOLO) merupakan sistem pendekripsi objek berbasis *convolutional neural network*. Berbeda dari sistem pendekripsi objek pada umumnya yang memiliki proses yang kompleks untuk melakukan pendekripsi, YOLO menggantikan keseluruhan proses tersebut dengan melakukan deteksi dengan menggunakan *single neural network* sehingga dapat mendekripsi objek dengan sangat cepat dan akurat. Secara umum, proses pendekripsi yang dilakukan oleh YOLO terbagi ke dalam tiga langkah utama sebagaimana yang ditunjukkan oleh Gambar 2.4, yaitu mengubah ukuran citra menjadi 448×448 piksel, lalu menjalankan *single convolutional network* pada citra, dan terakhir melakukan *threshold* pada hasil deteksi berdasarkan pada *confidence score* yang didapatkan oleh model.

YOLO mendekripsi objek dengan menggunakan *unified model* dimana sebuah *single convolutional network* memprediksi beberapa *bounding boxes* (kotak pembatas) serta probabilitas kelas di dalam kotak-kotak tersebut secara bersamaan. Pertama-tama, sistem YOLO membagi citra *input* ke dalam *grid* $S \times S$. Jika pusat dari sebuah objek jatuh di dalam salah satu sel *grid*, maka sel *grid* itu bertanggung jawab untuk mendekripsi objek tersebut. Setiap sel *grid* memprediksi *bounding boxes* B dan *confidence score* dari tiap *bounding box* tersebut. *Confidence score* merefleksikan seberapa yakin dan akurat model bahwa terdapat sebuah objek di dalam kotak tersebut. Setiap *bounding box* terdiri dari 5 prediksi: x , y , w , h ,



Gambar 2.4: Sistem Pendektsian YOLO[2].

dan *confidence*. Koordinat (x, y) mewakili pusat dari kotak relatif ke batas sel *grid*. (w, h) atau lebar dan tinggi mewakili pusat dari kotak relatif ke gambar. Dan terakhir adalah *confidence* yang mewakili *Intersection over Union* (IoU) antara kotak prediksi dan kotak *ground-truth*. Setiap sel *grid* juga memprediksi probabilitas kelas C . Probabilitas dikondisikan pada sel *grid* yang memuat objek dan hanya satu kelas probabilitas yang dideteksi per sel *grid* tanpa memperhitungkan jumlah *bounding boxes* B . Ketika melakukan pendektsian, probabilitas kelas dikalikan dengan *confidence* pada kotak prediksi sesuai dengan persamaan 2.8.

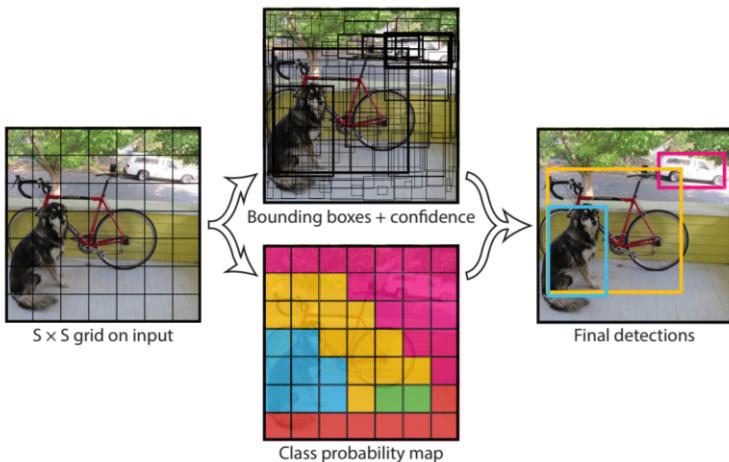
$$Pr(Class_i|Object) \times Pr(Object) \times_{truth} IOU_{pred} \quad (2.8)$$

Atau yang dapat disederhanakan menjadi persamaan 2.9.

$$Pr(Class_i) \times_{truth} IOU_{pred} \quad (2.9)$$

Dari persamaan 2.9 didapatkan *confidence score* dari kelas yang spesifik. Skor tersebut menyimpan probabilitas dari kelas yang muncul di dalam kotak dan seberapa baik kotak prediksi tersebut menyesuaikan objek. Keseluruhan proses pendektsian objek ini divisualisasikan oleh Gambar 2.5

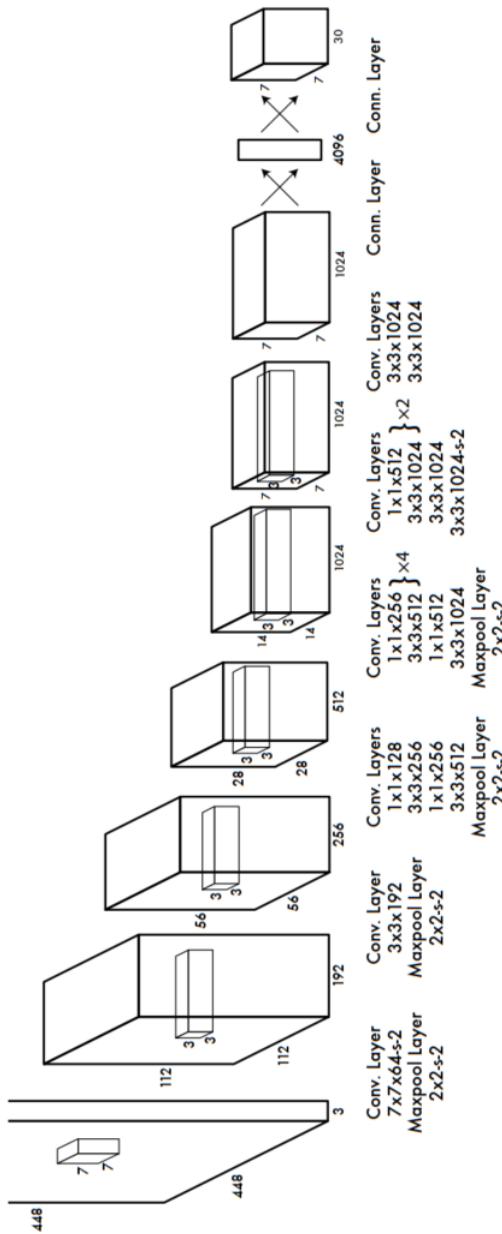
Terdapat beberapa keuntungan dari penggunaan *unified model* pada YOLO. Pertama, pendektsian dapat dilakukan dengan sangat cepat karena tidak menggunakan proses yang kompleks. Kedua, YOLO melihat citra secara global sehingga dapat mengurangi *background error* yang terjadi ketika pendektsian dilakukan dengan menggunakan *sliding windows* dan *region proposal* seperti yang di-



Gambar 2.5: Proses Pendektsian Objek YOLO[2].

gunakan pada pendektsi objek lain.

Arsitektur YOLO terdiri dari dua bagian, 24 *convolution layer* pada awal jaringan yang berfungsi untuk melakukan ekstraksi fitur dari citra dan dua 2 *fully connected layer* yang berfungsi untuk melakukan prediksi probabilitas dan koordinat. Jaringan penuh YOLO ditunjukkan oleh Gambar 2.6. Meskipun sudah mampu melakukan pendektsian dengan cepat dibandingkan dengan sistem yang lain, YOLO masih memiliki beberapa kekurangan. YOLO memberikan batasan spasial yang kuat pada prediksi *bounding box* karena setiap sel *grid* hanya dapat memprediksi dua *bounding boxes* dan satu kelas saja. Batasan ini membatasi jumlah objek yang dapat diprediksi oleh YOLO, khususnya benda-benda kecil yang muncul secara berkelompok seperti kawanan burung. Selain itu YOLO juga masih mengalami kesulitan dalam menyesuaikan ukuran *bounding box* dengan objek yang diprediksi karena arsitektur YOLO memiliki beberapa *downsampling* layer yang menurunkan kualitas dari citra *input*[2]. Namun, YOLO terus mengalami perkembangan menjadi YOLOv2 (YOLO9000) dan yang terbaru adalah YOLOv3, dimana performa yang dihasilkan semakin meningkat seiring dengan berkembangnya metode ini.



Gambar 2.6: Arsitektur YOLO[2].

2.7.2 Single Shot Multibox Detector

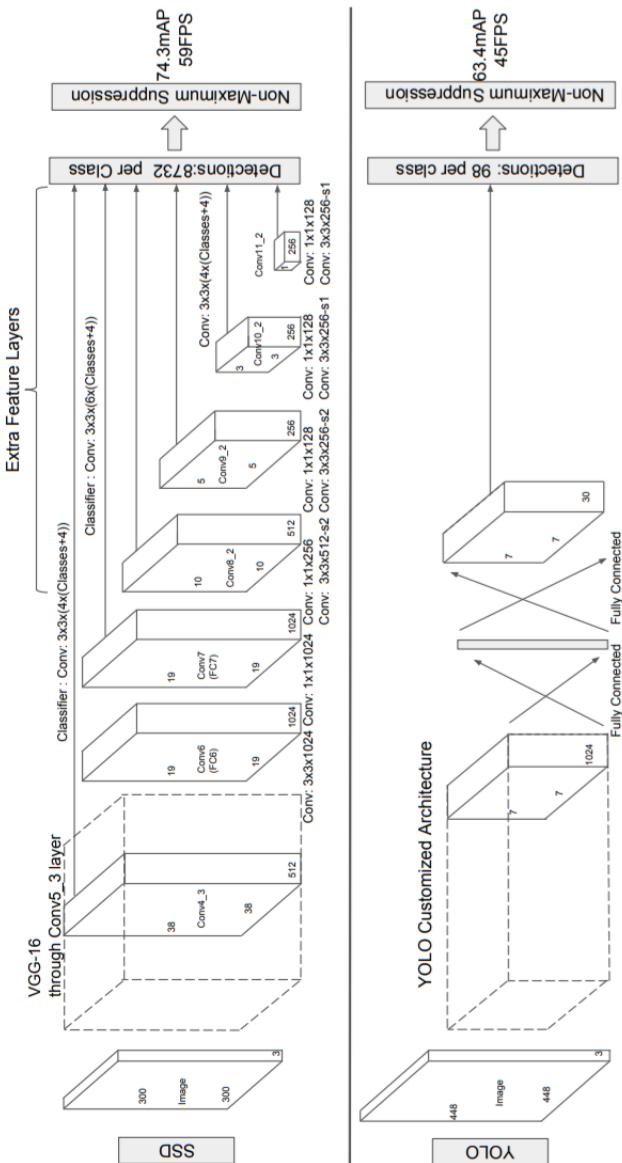
Single Shot Multibox Detector (SSD) merupakan sistem pendeteksi objek berbasis *convolutional neural network*. SSD menjadikan ruang *output* dari *bounding box* menjadi satu set *default box* pada berbagai rasio dan skala aspek pada tiap lokasi *feature map*. Saat melakukan prediksi, jaringan dari SSD menghasilkan skor untuk keberadaan setiap kategori objek di setiap *default box* dan melakukan penyesuaian agar bentuk kotak lebih sesuai dengan bentuk objek. Selain itu, jaringan dari SSD menggabungkan prediksi dari beberapa *feature map* dengan resolusi yang berbeda sehingga dapat mendeteksi objek dalam berbagai ukuran dengan natural. SSD menghilangkan menghilangkan metode *object proposals* dan *feature resampling stages* yang biasanya terdapat pada pendekripsi objek generasi lama. Hal ini membuat SSD mudah dilatih dan untuk diintegrasikan ke dalam sistem yang membutuhkan sebuah komponen deteksi.

SSD bekerja dengan cara menerima *input* citra dan *ground-truth boxes* untuk setiap objek yang akan di-*training*. Dengan menggunakan proses konvolusi, SSD melakukan evaluasi pada satu set *default boxes* dengan rasio aspek yang berbeda di setiap lokasi pada beberapa *feature map* dengan skala yang berbeda (misalnya 8×8 dan 4×4). Untuk setiap *default box*, SSD memprediksi bentuk dan *confidence* dari setiap kategori objek (c_1, c_1, \dots, c_p).

Pendekatan SSD dibuat berdasarkan pada *feed-forward convolutional neural network* yang menghasilkan beberapa *bounding box* dan skor kehadiran objek yang terdapat pada *bounding box* tersebut, dilanjutkan dengan proses *non-maximum suppression* untuk menghasilkan deteksi akhir. Lapisan awal jaringan SSD menggunakan arsitektur standar yang digunakan untuk melakukan klasifikasi pada gambar (dalam hal ini SSD menggunakan VGG16) yang disebut dengan jaringan dasar (*base network*)[4]. Dari Gambar 2.7 dapat disimpulkan bahwa arsitektur SSD berbeda dengan YOLO, khususnya pada bagian *object detection layer*. SSD meletakkan struktur tambahan ke dalam jaringan sehingga menghasilkan deteksi dengan beberapa fitur yaitu:

1. *Multi-scale feature map for detection*

SSD menambahkan lapisan fitur konvolisional pada akhir ja-



Gambar 2.7: Perbandingan Arsitektur SSD dengan Arsitektur YOLO[4].

ringan dasar. Tiap layer mengalami penurunan ukuran dan memungkinkan prediksi deteksi dilakukan dalam beberapa skala. Model konvolusi untuk melakukan prediksi deteksi berbeda pada tiap lapisan fiturnya.

2. *Convolutional predictors for detection*

Setiap lapisan fitur yang ditambahkan dapat menghasilkan satu set tetap dari prediksi deteksi menggunakan satu set filter konvolusional seperti yang terdapat pada Gambar 2.7. Tiap lapisan fitur dengan ukuran $m \times n \times p$ dihasilkan skor untuk sebuah kategori objek atau sebuah bentuk yang dengan koordinat relatif terhadap *default box*. Tahapan ini tentu berbeda dengan YOLO yang langsung meletakkan *fully-connected layer* daripada filter konvolusi.

3. *Default boxes and aspect ratio*

Untuk setiap kotak k yang terdapat pada lokasi tertentu pada *feature map*, dilakukan komputasi skor kelas c dan 4 *offset* relatif terhadap bentuk asli *default box*. Hal ini menghasilkan total $(c + 4)k$ filter yang diaplikasikan pada tiap lokasi di *feature map*, yang kemudian menghasilkan $(c + 4)kmn$ output untuk sebuah $m \times n$ *feature map*.

2.8 *Internet Protocol Camera*

Internet Protocol Camera atau *IP Camera* merupakan pengembangan dari kamera CCTV analog. *IP camera* mampu mengirim data dan melakukan konversi dari video yang ditangkap menjadi sebuah file digital yang berisi rekaman video dari *IP camera* tersebut secara otomatis. Hasil rekaman tersebut dapat dilihat secara *online* melalui internet dengan mengakses IP *address* yang telah ditentukan melalui *web browser* atau aplikasi yang mampu melakukan pemutaran video dari *network stream*. Perbedaan yang paling mendasar antara *IP camera* dan CCTV analog adalah pada kabel. Jika CCTV analog menggunakan jenis kabel *coaxial*, maka *IP camera* menggunakan kabel jenis UTP. Rekaman yang dihasilkan oleh *IP camera* juga lebih baik dan jelas karena menggunakan resolusi gambar yang lebih tinggi dibandingkan dengan CCTV analog[19].

BAB 3

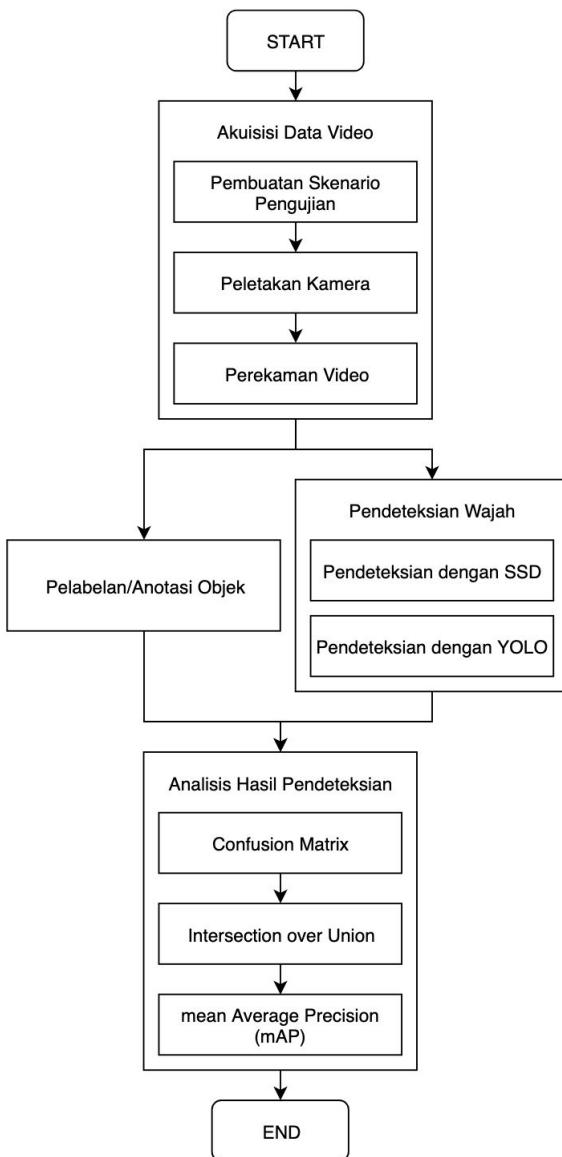
DESAIN DAN IMPLEMENTASI SISTEM

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.

3.1 Desain Sistem

Tugas akhir ini merupakan penelitian dalam bidang visi komputer yang bertujuan untuk menganalisis performa dari sistem pendekripsi wajah berbasis *Convolutional Neural Network* (CNN). Analisis ini dilakukan dengan memanfaatkan sistem pendekripsi wajah yang dibangun pada sebuah komputer dan *input* video yang ditangkap menggunakan sebuah *IP camera*. Analisis dilakukan dengan membandingkan nilai *mean Average Precision* (mAP) pada hasil deteksi oleh sistem yang digunakan. Proses kerja dari analisis sistem ditunjukkan pada Gambar 3.1.

Berdasarkan diagram alir pada Gambar 3.1, proses pertama yang dilakukan adalah akuisisi data video. Tujuan dari akuisisi data ini adalah untuk mengambil data video yang akan dianalisis. Data video yang telah diambil kemudian diberikan label atau anotasi berupa *ground-truth bounding box* yang menunjukkan dimana posisi objek serta kelas wajah berada di dalam video yang telah diambil. Proses akuisisi data dilakukan secara paralel dengan proses pembuatan sistem pendekripsi wajah. Sistem pendekripsi wajah yang digunakan pada penelitian ini merupakan sistem yang menggunakan metode pendekripsi objek berbasis CNN, yaitu *You Only Look Once* (YOLO) dan *Single Shot Multibox Detector* (SSD). Sistem pendekripsi wajah akan memberikan hasil pendekripsi berupa *confidence* dan *predicted bounding box*. Hasil dari pendekripsi wajah tersebut kemudian dianalisis dengan menggunakan *confusion matrix* dan *Intersection over Union* (IoU). Terakhir, kedua nilai tersebut diguna-



Gambar 3.1: Diagram Alir Analisis Performa Sistem Pendekripsi Wajah.

kan untuk membentuk sebuah kurva bernama *precision-recall curve*. Nilai *Average Precision* (AP) didapatkan dari area di bawah kurva tersebut. Kemudian, nilai AP dari seluruh kelas dirata-ratakan sehingga didapatkan nilai *mean Average Precision* (mAP).

3.2 Alur Kerja

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

1. Pembuatan Skenario Pengujian.
2. Akuisisi Data Video.
3. Pelabelan Objek.
4. Pendekripsi Wajah.
5. Analisis Hasil Pendekripsi Wajah.

3.3 Pembuatan Skenario Pengujian

Penelitian ini bertujuan untuk melakukan analisis performa terhadap sistem pendekripsi wajah berbasis CNN. Oleh karena itu, diperlukan pembuatan skenario pengambilan data pengujian yang akan dianalisis sehingga didapatkan hasil dari pengujian tersebut. Dari hasil pengujian, akan diketahui seberapa baik performa sistem dalam mendekripsi wajah di skenario-skenario yang telah ditentukan.

3.3.1 Skenario Pengujian Berdasarkan Lokasi Peletakan Kamera

Pengujian berdasarkan pada lokasi peletakan kamera bertujuan untuk mengetahui pengaruh lokasi peletakan kamera terhadap hasil pendekripsi wajah. Berdasarkan desain sistem, terdapat dua kamera yang terletak di dua lokasi yang berbeda. Kamera pertama terletak di atas pintu Laboratorium Telematika B201 dan membelakangi sinar matahari, sedangkan kamera kedua terletak di palang besi yang ditempelkan pada tiang di depan area Laboratorium Telematika B201 dan menghadap ke arah sinar matahari.

Selain posisi kamera, pengujian juga dilakukan dengan memberikan objek penghalang yang berbeda pada wajah dengan total 16 kombinasi objek penghalang. Waktu pengambilan data dilakukan pada waktu yang sama yaitu pada waktu siang hari. Detail skenario pengambilan data untuk pengujian berdasarkan lokasi peletakan kamera terdapat pada Tabel 3.1.

Tabel 3.1: Skenario Pengujian Berdasarkan Lokasi Peletakan Kamera.

Letak Kamera	Halangan 1	Halangan 2	Halangan 3	Halangan 4
Kamera Reolink (menghadap sinar matahari) dan Kamera Hikvision (membelaangi sinar matahari)	-	-	-	-
	Kacamata	-	-	-
	Masker	-	-	-
	Topi	-	-	-
	<i>Hoodie</i>	-	-	-
	Kacamata	Masker	-	-
		Topi	-	-
		<i>Hoodie</i>	-	-
	Masker	Topi	-	-
		<i>Hoodie</i>	-	-
	Topi	<i>Hoodie</i>	-	-
	Kacamata	Masker	Topi	-
		<i>Hoodie</i>	-	-
		Topi	<i>Hoodie</i>	-
	Masker	Topi	<i>Hoodie</i>	-
	Kacamata	Masker	Topi	<i>Hoodie</i>

3.3.2 Skenario Pengujian Waktu Pengambilan Data

Pengujian berdasarkan waktu pengambilan data bertujuan untuk mengetahui pengaruh waktu pengambilan data terhadap performa komputasi dan hasil pendekripsi wajah. Pada pengujian ini hanya digunakan satu kamera yang mampu memberikan hasil lebih baik pada pengujian berdasarkan lokasi peletakan kamera sebelumnya. Pengujian ini dilakukan dengan mengambil data sesuai dengan skenario pada Tabel 3.2. Waktu pengambilan data dilakukan pada waktu siang dan malam hari sehingga didapatkan data dengan kondisi cahaya yang sangat jauh berbeda.

3.3.3 Skenario Pengujian Berdasarkan Jumlah Objek

Pengujian berdasarkan pada jumlah objek bertujuan untuk mengetahui pengaruh jumlah objek terhadap kemampuan sistem

Tabel 3.2: Skenario Pengujian Berdasarkan Waktu Pengambilan Data.

Waktu	Halangan 1	Halangan 2	Halangan 3	Halangan 4
	-	-	-	-
	Kacamata	-	-	-
	Masker	-	-	-
	Topi	-	-	-
	<i>Hoodie</i>	-	-	-
Siang dan Malam		Masker	-	-
	Kacamata	Topi	-	-
		<i>Hoodie</i>	-	-
		Masker	-	-
		Topi	-	-
		<i>Hoodie</i>	-	-
	Topi	<i>Hoodie</i>	-	-
		Masker	Topi	-
			<i>Hoodie</i>	-
	Masker	Topi	<i>Hoodie</i>	-
Kacamata Hitam	Kacamata	Masker	Topi	<i>Hoodie</i>
		-	-	-
		Masker	-	-
		Topi	-	-
		<i>Hoodie</i>	-	-
		Masker	Topi	-
			<i>Hoodie</i>	-
		Topi	<i>Hoodie</i>	-

dalam mendeteksi wajah. Pada pengujian ini hanya digunakan satu kamera yang mampu memberikan hasil lebih baik pada pengujian berdasarkan lokasi peletakan kamera. Waktu pengambilan data dilakukan pada waktu siang hari. Terdapat sembilan kondisi jumlah objek yang digunakan, antara lain seperti yang terdapat pada Gambar 3.2.

3.3.4 Skenario Pengujian Berdasarkan Kecepatan Objek Bergerak

Pengujian berdasarkan kecepatan objek bergerak bertujuan untuk mengetahui pengaruh kecepatan objek terhadap kemampuan sistem dalam mendeteksi wajah. Pada pengujian ini hanya digunakan satu kamera yang mampu memberikan hasil lebih baik pada pengujian berdasarkan lokasi peletakan kamera. Waktu pengambilan data dilakukan pada waktu siang hari. Terdapat tiga skenario kondisi kecepatan objek yang digunakan, antara lain:

1. Objek yang bergerak dengan kecepatan normal.
2. Objek yang bergerak dengan kecepatan agak cepat.
3. Objek yang bergerak dengan kecepatan cepat.

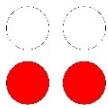
3.3.5 Skenario Pengujian Berdasarkan Perangkat Keras yang Digunakan

Pengujian berdasarkan perangkat keras yang digunakan bertujuan untuk mengetahui pengaruh perangkat keras yang digunakan untuk menjalankan sistem terhadap kemampuan sistem dalam mendeteksi wajah. Pada pengujian ini, data yang digunakan adalah data yang didapatkan dari pengujian kedua, yaitu data dari pengujian berdasarkan waktu pengambilan data. Terdapat tiga skenario kondisi perangkat keras pada pengujian ini, antara lain:

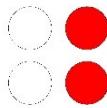
1. Menjalankan sistem dengan menggunakan CPU laptop.
2. Menjalankan sistem dengan menggunakan GPU laptop.
3. Menjalankan sistem dengan menggunakan GPU desktop.

3.3.6 Skenario Pengujian dengan Tambahan Pencahayaan Buatan

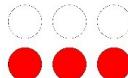
Pengujian dengan tambahan pencahayaan buatan dilakukan dengan melakukan pengambilan data yang di lingkungan yang di-



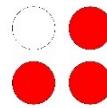
(a) Dua orang berjalan sejajar.



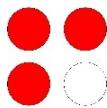
(b) Dua orang berjalan beriringan.



(c) Tiga orang berjalan sejajar.



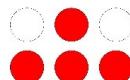
(d) Dua orang berjalan sejajar diikuti oleh satu orang berjalan sendiri di belakangnya.



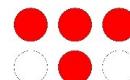
(e) Satu orang berjalan sendiri diikuti oleh dua orang berjalan sejajar di belakangnya.



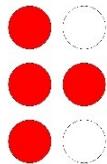
(f) Empat orang berjalan sejajar.



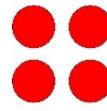
(g) Tiga orang berjalan sejajar diikuti oleh satu orang berjalan sendiri di belakangnya.



(h) Satu orang berjalan sendiri diikuti oleh tiga orang berjalan sejajar di belakangnya.



(i) Satu orang berjalan sendiri diikuti oleh dua orang yang berjalan sejajar dan diikuti lagi oleh satu orang berjalan sendiri di belakangnya.



(j) Dua orang berjalan sejajar diikuti oleh dua orang yang juga berjalan sejajar di belakangnya.

Gambar 3.2: Skenario Pengujian Berdasarkan Jumlah Objek.

kondisikan pencahayaannya. Cahaya buatan yang digunakan dalam skenario ini didapatkan dari lampu LED untuk fotografi dengan ukuran lux sebesar 860 lumens dan daya sebesar 9 watt. Pengujian ini dilakukan dengan mengambil data sesuai dengan skenario pada Tabel 3.2 dan waktu pengambilan data dilakukan pada waktu siang dan malam hari. Data yang didapatkan kemudian dibandingkan dengan pengujian tanpa menggunakan pencahayaan buatan, sehingga dapat terlihat perbedaan antara hasil pendekripsi dengan dan tanpa bantuan pencahayaan buatan.

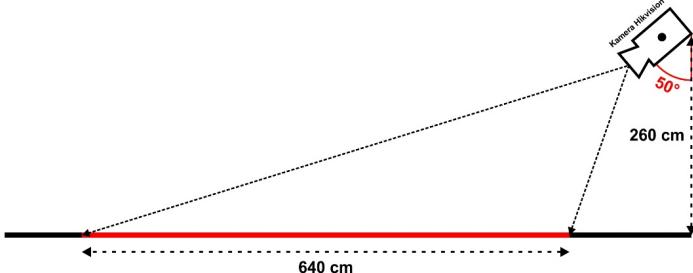
3.3.7 Skenario Pengujian dengan Ketinggian Kamera Sejajar dengan Wajah

Pengujian ini dilakukan dengan melakukan pengambilan data menggunakan kamera yang diletakkan pada ketinggian yang sejajar dengan tinggi dari wajah objek yang akan diambil datanya. Pengujian ini dilakukan dengan mengambil data sesuai dengan skenario pada Tabel 3.2 dan waktu pengambilan data dilakukan pada waktu siang dan malam hari. Pengujian ini juga dilakukan dengan menggunakan pencahayaan buatan. Data yang didapatkan kemudian dibandingkan dengan pengujian pada posisi kamera yang berada di atas objek, sehingga dapat terlihat perbedaan antara hasil pendekripsi kamera yang terletak di atas dan kamera yang terletak sejajar dengan wajah objek.

3.3.8 Skenario Pengujian Berdasarkan Sudut Kamera

Pengujian berdasarkan sudut kamera yang berbeda bertujuan untuk mengetahui pengaruh perubahan sudut kamera terhadap kemampuan sistem dalam mendekripsi wajah. Pada pengujian ini hanya digunakan satu kamera yang membelakangi sinar matahari. Waktu pengambilan data dilakukan pada waktu siang hari. Terdapat lima skenario kondisi sudut kamera yang digunakan, antara lain:

1. Sudut kamera 30° .
2. Sudut kamera 40° .
3. Sudut kamera 50° .
4. Sudut kamera 60° .



Gambar 3.3: Ilustrasi Peletakan Kamera Pertama.

5. Sudut kamera 70° .

3.4 Akuisisi Data Video

Pada tahap akuisisi data video dilakukan proses pengambilan data berupa video dari beberapa skenario yang direkam dengan menggunakan *IP camera* yang diletakkan di arah depan atas objek pada lokasi yang telah ditentukan.

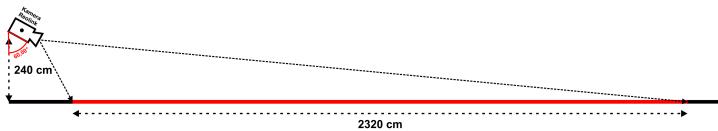
3.4.1 Posisi Peletakan Kamera

Akuisisi data video pada penelitian ini menggunakan dua buah *IP camera* yang diletakkan pada dua tempat yang berbeda. Kamera yang pertama diletakkan pada sebuah kayu yang terdapat di atas pintu dan kamera yang kedua diletakkan pada palang besi yang ditempelkan pada tiang.

Berdasarkan Gambar 3.3, *IP camera* yang pertama diletakkan pada ketinggian 240 cm dari lantai, dengan kemiringan sebesar 60° , membelakangi sinar matahari, dan dengan cakupan *frame* yang dapat ditangkap sepanjang 640 cm. Sedangkan *IP camera* yang kedua diletakkan pada ketinggian 260 cm dari lantai, dengan kemiringan sebesar 50° , menghadap ke arah sinar matahari, dan dengan cakupan *frame* yang dapat ditangkap sepanjang 2.320 cm seperti yang diilustrasikan pada Gambar 3.4.

3.5 Lokasi dan Kondisi Akuisisi Data

Pengambilan data video ini menggunakan dua buah *IP camera* yang telah terpasang di area lingkungan Kampus ITS. Kamera



Gambar 3.4: Ilustrasi Peletakan Kamera Kedua.

pertama terletak di atas pintu Laboratorium Telematika B201 dan membelakangi sinar matahari, sedangkan kamera kedua terletak di palang besi yang ditempelkan pada tiang di depan area Laboratorium Telematika B201 dan menghadap ke arah sinar matahari. Area di depan laboratorium tersebut merupakan salah satu area yang sering dilewati oleh mahasiswa dan dosen.

Sesuai dengan topik penelitian yang terkait dengan pengenalan wajah, maka area di depan Laboratorium Telematika B201 merupakan tempat yang sangat cocok untuk lokasi penelitian ini. Gambar 3.5 dan 3.6 menunjukkan lokasi kamera yang terdapat pada area depan Laboratorium Telematika B201 dimana proses akuisisi data video pada penelitian ini dilakukan.

Pengambilan data video dilakukan pada waktu siang dan malam hari agar dapat membandingkan kemampuan pendekripsi wajah di waktu yang berbeda, dimana kondisi cahaya sangat memengaruhi proses pendekripsi wajah oleh komputer.

3.5.1 *IP camera* yang Digunakan

Dalam penelitian ini terdapat dua buah *IP camera* yang digunakan yaitu Hikvision DS-2CD2055FWD-I dan Reolink RLC-411WS dengan spesifikasi dan bentuk seperti yang terdapat pada Tabel 3.3 dan Gambar 3.7.

3.5.2 Pengambilan Data Video

Pengambilan data video dilakukan sesuai dengan skenario yang telah dibuat sebelumnya. Pada tahap awal, dilakukan pengambilan 32 data video berdasarkan skenario untuk menentukan letak kamera yang tepat, yaitu 16 data video dari kamera Hikvision dan 16 data video dari kamera Reolink, yang mana kedua jenis data tersebut



Gambar 3.5: Lokasi Peletakan Kamera Pertama.



Gambar 3.6: Lokasi Peletakan Kamera Kedua.

Tabel 3.3: Spesifikasi IP Camera

Merk Kamera	Hikvision DS-2CD2055FWD-I	Reolink RLC-411WS
Image Sensor	1/2.9" Progressive Scan CMOS	1/3" 4MP CMOS
Min. Illumination	Color: 0.01 lux @ (F1.2, AGC ON), 0.028 lux @ (F2.0, AGC ON), 0 lux with IR on)	Color: 0.05LUX/F1.4, B/W: 0Lux (IR on)
Shutter Time	1/3 s ~1/100,000 s	Auto /Manual 1/50 ~ 1/1000 s
Lens	2.8 mm 97°, 4 mm 78°, 6 mm 60°, 8 mm 39.5°, 12 mm 19°	f = 2.8 ~ 12mm motorized, F = 1.2
Focus	Fixed	Automatic
Angle of View	19° ~ 97°	36° ~ 100°
Max. Resolution	2944 × 1656	2560 × 1440
Video Compression	H.265+/H.265/H.264+/H.264/MJPEG	H.264 High Profile/JPEG Snapshot
Video Bit Rate	32Kbps ~ 16Mbps	1024Kbps ~ 8192Kbps
Video Frame Rate	20fps at 5MP	30fps at 4MP
Network Protocols	TCP/IP, UDP, ICMP, HTTP, HTTPS, FTP, DHCP, DNS, DDNS, RTP, RTSP, RTCP, PPPoE, NTP, UPnP, SMTP, SNMP, IGMP, 802.1X, QoS, IPv6	TCP/IP, HTTP, HTTPS, uPNP, RTSP, SMTP, NTP, DHCP, DNS, FTP, ONVIF, URL, P2P
Comm. Interface	1 RJ45 10M/100M ethernet interface	RJ-45, Wi-Fi IEEE802.11 a/b/g/n 2T2R
Power Supply	12 VDC ± 25%, PoE (802.3af Class3)	DC12V
Power Consumption	12 VDC ± 25%, 6W, PoE(802.3af, class 3), 7W	12 VDC ± 25%, 6W, PoE(802.3af, class 3), 7W
Dimension	Φ70 × 155.03 mm	Φ75 × 250 mm
Weight	410g	800g



(a) Kamera Hikvision.

(b) Kamera Reolink.

Gambar 3.7: Jenis Kamera yang Digunakan.

diambil pada waktu siang hari. Dari analisis kedua jenis data tersebut didapatkan lokasi mana yang menghasilkan hasil prediksi lebih baik, sehingga dapat digunakan sebagai lokasi utama pengambilan data.

Selanjutnya diambil 48 data video dalam dua waktu yang berbeda yaitu pada 24 data video pada waktu siang hari dan 24 video pada waktu malam hari dengan menggunakan kamera yang terletak di lokasi yang menghasilkan pendekripsi lebih baik, yaitu kamera Hikvision yang terletak di atas pintu Laboratorium Telematika B201. Setelah itu, pengambilan data dilanjutkan dengan skenario berdasarkan jumlah, kecepatan objek, dan posisi kamera. Perekaman video dilakukan dengan menggunakan program FFMPEG yang merupakan *framework* yang memungkinkan pengguna untuk merekam, mengonversi, dan memutar video dalam berbagai format di platform Linux.

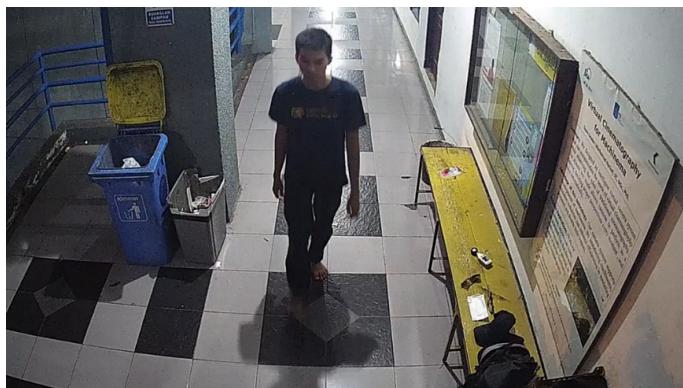
Gambar 3.8 merupakan hasil pengambilan video pada waktu siang hari pukul 12.57 WIB. Terlihat dari data video tersebut bahwa kondisi cahaya yang terdapat di area depan Laboratorium Telematika B201 membuat data video yang diambil tampak jelas dan bersih dari *noise*. Hal ini merupakan kondisi yang cukup ideal bagi sistem untuk dapat mengenali wajah dengan baik pada tahap selanjutnya.

Sedangkan Gambar 3.9 merupakan hasil pengambilan video pada waktu malam hari pukul 20.31 WIB. Terlihat dari data vi-

deo tersebut bahwa kondisi cahaya yang terdapat di area depan Laboratorium Telematika B201 hanya dibantu oleh cahaya buatan yaitu lampu, hal tersebut berdampak pada hasil rekaman yang kurang bagus karena banyak terdapat *noise* pada data video. Kondisi ini merupakan kondisi yang kurang ideal bagi sistem untuk dapat mengenali wajah dengan baik.



Gambar 3.8: Pengambilan Video pada Waktu Siang Hari.



Gambar 3.9: Pengambilan Video pada Waktu Malam Hari.

3.6 Pelabelan Objek

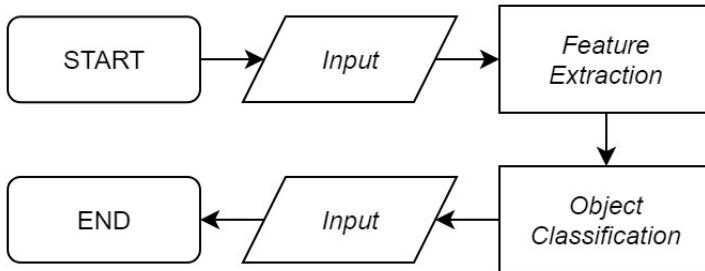
Tahap pelabelan atau anotasi objek merupakan proses pembeiran informasi berupa kelas dan posisi dari objek yang akan dideteksi. Dari data hasil rekaman yang didapatkan, dilakukan pelabelan satu per satu terhadap objek wajah sehingga didapatkan koordinat *ground-truth bounding box* yang akan dibandingkan dengan *predicted bounding box*. Dari kedua jenis *bounding box* yang ada, akan didapatkan nilai *Intersection over Union* (IoU) berdasarkan dari perbandingan kedua *bounding box* tersebut.

Proses pelabelan *bounding box* dilakukan satu per satu secara manual pada tiap *frame* yang terdapat objek wajah di dalamnya. Informasi mengenai label yang telah dibuat kemudian disimpan dalam bentuk file TXT atau XML.

3.7 Pendekstrian Wajah

Proses pendekstrian wajah dilakukan dengan menggunakan program yang telah dibuat sebelumnya dan kemudian dilakukan pengubahan parameter serta penambahan kode agar sesuai dengan kebutuhan analisis. Dalam penelitian ini, digunakan dua jenis metode pendekstrian objek berbasis CNN, yaitu *You Only Look Once* (YOLO) dan *Single Shot Multibox Detector* (SSD). Kedua metode ini merupakan metode yang menjadikan proses pendekstrian sebagai masalah regresi, sehingga hanya perlu dilakukan sekali penangkapan terhadap *input* untuk dapat melakukan pendekstrian objek. Secara umum, proses pendekstrian wajah oleh metode pendekstrian objek berbasis CNN ditunjukkan oleh Gambar 3.10.

Pada tahap pendekstrian wajah, data video dari hasil rekaman sebelumnya digunakan sebagai *input* yang kemudian dilewatkan pada sistem pendekstrian wajah berbasis CNN, yaitu YOLO dan SSD. *Output* dari proses pendekstrian wajah berupa *confidence*, nama kelas, serta posisi *bounding box* dari wajah yang terdeteksi oleh sistem. Ketiga nilai yang diperoleh tersebut adalah nilai yang dibutuhkan pada proses analisis performa dari metode yang digunakan.



Gambar 3.10: Diagram Alir Pendektsian Objek.

3.7.1 Proses Pendektsian Wajah pada Sistem berbasis YOLO

3.7.1.1 Pengolahan *Input*

YOLO membutuhkan *input* gambar dengan ukuran 416×416 piksel, yang masih dapat diubah selama masih ada di dalam kelipatan 32. Semakin besar ukuran gambar, maka semakin akurat pula hasil pendektsian yang didapatkan. Namun, semakin besar ukuran *input* gambar akan membuat proses komputasi menjadi lebih lambat. Hal ini juga berlaku sebaliknya, apabila *input* gambar yang diberikan semakin kecil maka hasil pendektsian semakin berkurang akurasinya, namun proses komputasi akan berjalan dengan lebih cepat.

3.7.1.2 *Feature Extraction* dan *Object Classification*

YOLOv3 menggunakan *feature extractor* yang bernama Darknet53, yang memiliki 53 *layer* yang telah dilatih di ImageNet. Untuk tugas pendektsian, ditambahkan lagi 53 *layer* ke dalam jaringan, sehingga terdapat 106 *layer* yang mendasari arsitektur *convolutional* dari YOLOv3. Hal ini menyebabkan proses pendektsian lebih lambat, namun memberikan hasil yang lebih akurat. Gambar 3.11 menunjukkan susunan *layer* dari Darknet53.

Sedangkan untuk prediksi kelasnya, YOLO generasi awal menggunakan fungsi *softmax* yang menghitung probabilitas setiap kelas target dari semua kemungkinan kelas target. Namun pada YOLOv3

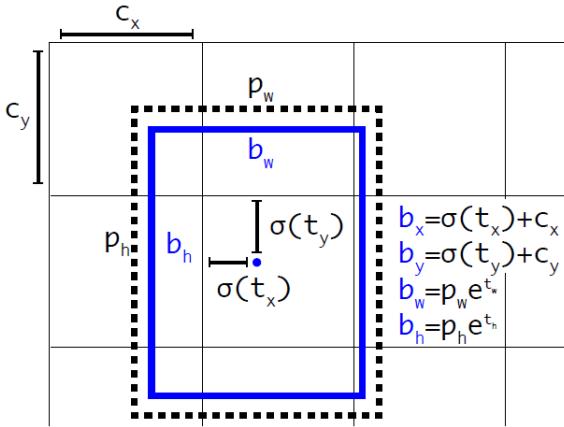
	Type	Filters	Size	Output
1x	Convolutional	32	3×3	256×256
	Convolutional	64	$3 \times 3 / 2$	128×128
Convolutional	32	1×1		
	64	3×3		
2x	Residual			128×128
	Convolutional	128	$3 \times 3 / 2$	64×64
Convolutional	64	1×1		
	128	3×3		
8x	Residual			64×64
	Convolutional	256	$3 \times 3 / 2$	32×32
Convolutional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

Gambar 3.11: Susunan Layer Darknet53[3].

digunakan *multilabel classification* pada setiap objek yang terdeteksi pada gambar. Hal ini dilakukan karena fungsi *softmax* hanya memberikan satu kelas saja pada objek yang dideteksi. Sedangkan pada *multilabel classification*, setiap skor kelas diprediksi dengan menggunakan *logistic regression* dan *threshold* untuk memprediksi beberapa label dari suatu objek. Kelas yang memiliki skor lebih tinggi dari *threshold* akan menjadi kelas yang diberikan kepada objek yang dideteksi.

3.7.1.3 Pengolahan Output

YOLO menghasilkan prediksi berupa koordinat *bounding box* (t_x, t_y, t_w, t_h), *confidence*, dan *class probability*. *Confidence* merupakan tingkat keyakinan ada atau tidaknya objek pada gambar, sedangkan *class probability* merupakan jenis kelas dari objek yang



Gambar 3.12: Prediksi Lokasi *Bounding Box*[3].

terdeteksi. YOLO membuat *bounding box* pada lokasi *feature map* dengan menggunakan *logistic activation sigmoid*. Pengolahan *bounding box* dari hasil deteksi diilustrasikan pada Gambar 3.12. Dimana (c_x, c_y) merupakan lokasi dari *grid*. (b_x, b_y) merupakan lokasi *bounding box* yang didapatkan dari lokasi (c_x, c_y) dengan hasil fungsi sigmoid dari t_x dan t_y , (p_w, p_h) merupakan *anchor box* yang didapatkan dari proses *clustering*, sedangkan (b_w, b_h) merupakan dimensi akhir dari *bounding box* yang didapatkan dari perkalian *anchor box* (p_w, p_h) dengan (t_w, t_h) .

Agar tidak terdapat *bounding box* yang berulang pada objek yang sama dan untuk menghindari terjadinya kesalahan deteksi, maka digunakan *logistic regression*, yang bernilai 1 jika terdapat *predicted bounding box* yang tumpang tindih dengan *ground-truth object* lebih baik daripada *predicted bounding box* yang lain. Jika terdapat *predicted bounding box* yang tidak memiliki nilai *confidence* yang baik namun tumpang tindih dengan *ground-truth object* dan memenuhi syarat *threshold*, yaitu 0.5, maka prediksi tersebut akan diabaikan. Hanya terdapat satu *predicted bounding box* yang digunakan untuk setiap *ground-truth object* yang ada.

Pada pengembangan YOLO terbaru, yaitu YOLOv3, jika *input* gambar yang diberikan sebesar 416×416 , maka dihasilkan *output*

dalam tiga skala dalam ukuran dimensi 13×13 , 26×26 , dan 52×52 piksel. Dari perbedaan skala tersebut, YOLOv3 dapat menghasilkan *predicted box* sebanyak 10,647 kotak. Hal ini dapat dilakukan karena YOLOv3 melakukan deteksi dengan cara mengaplikasikan *kernel* 1×1 pada *feature map* pada tiga ukuran yang berbeda di tiga tempat yang berbeda pada jaringan. Bentuk dari *kernel* deteksi tersebut adalah $1 \times 1 \times (B \times (5 + C))$, dimana B adalah jumlah *bounding box* yang dapat diprediksi oleh *feature map*, angka 5 mewakili 4 *bounding box* dan 1 *object confidence*, dan C adalah jumlah kelas.

3.7.2 Proses Pendekripsi Wajah pada Sistem berbasis SSD

3.7.2.1 Pengolahan *Input*

SSD membutuhkan *input* gambar dengan ukuran 300×300 piksel untuk SSD300 dan 512×512 untuk SSD512. Semakin besar *input* yang diberikan maka semakin baik kemampuan SSD, khususnya untuk mendekripsi objek-objek yang kecil. Namun, semakin tinggi ukuran *input* yang diberikan maka semakin berat juga proses komputasi yang perlu dilakukan.

3.7.2.2 Feature Extraction

Standar arsitektur *feature extractor* yang digunakan oleh SSD adalah VGG16. Namun, pada penelitian ini *feature extractor* yang digunakan adalah *MobileNet*. *MobileNet* adalah sebuah arsitektur yang dioptimalkan untuk perangkat dengan kemampuan komputasi rendah. Arsitektur *MobileNet* menggunakan *depthwise separable convolutions* yang dapat mengurangi jumlah parameter secara signifikan jika dibandingkan dengan arsitektur konvolusi standar yang memiliki kedalaman jaringan yang sama. Hal ini menghasilkan sebuah sistem *deep neural network* yang ringan untuk dijalankan [20].

Konvolusi standar biasa yang terdapat pada jaringan digantikan dengan *depthwise convolution* dan diikuti oleh *pointwise convolution*. Gabungan dari kedua konvolusi ini disebut dengan *depthwise separable convolutions*. *Depthwise separable convolutions* merupakan bentuk konvolusi yang memfaktorisasi konvolusi standar menjadi sebuah *depthwise convolution* dan sebuah konvolusi 1×1 yang disebut *pointwise convolution*. Pada *MobileNet*, *depthwise convolution*

mengaplikasikan sebuah filter tunggal pada tiap saluran *input*. *Pointwise convolution* kemudian mengaplikasikan konvolusi 1×1 untuk menyatukan *output* dari *depthwise separable convolutions*. Proses tersebut diilustrasikan pada Gambar 3.13. Apabila konvolusi standar memiliki biaya komputasi sebesar 3.1.

$$D_K \times D_K \times M \times N \times D_F \times D_F \quad (3.1)$$

Maka biaya komputasi dari *depthwise separable convolutions* adalah sebesar 3.2.

$$D_K \times D_K \times M \times D_F \times D_F + M \times N \times D_F \times D_F \quad (3.2)$$

Dimana M adalah jumlah saluran *input*, N adalah saluran *output*, D_K adalah ukuran *kernel*, dan D_F adalah ukuran *feature map*. Sehingga berdasarkan kedua persamaan di atas, didapatkan total pengurangan biaya komputasi sebesar

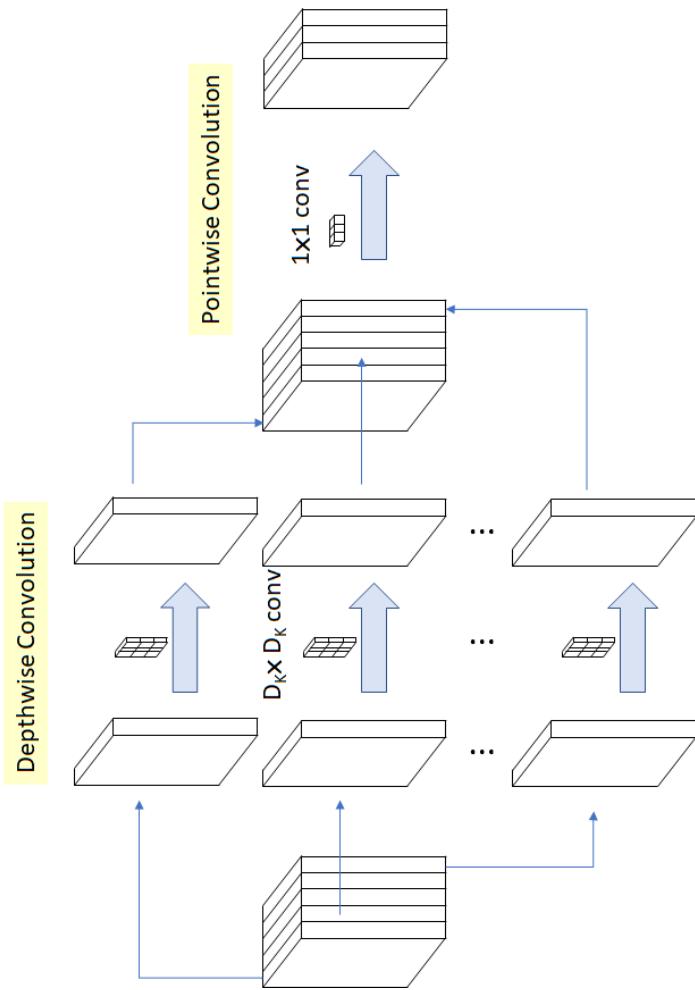
$$\frac{D_K \times D_K \times M \times D_F \times D_F + M \times N \times D_F \times D_F}{D_K \times D_K \times M \times N \times D_F \times D_F} = \frac{1}{N} = \frac{1}{D_K^2} \quad (3.3)$$

Gambar 3.14 menunjukkan arsitektur lengkap dari *MobileNet*. Sedangkan Gambar 3.15 menunjukkan perbandingan antara arsitektur konvolusi standar *depthwise separable convolutions*.

3.7.2.3 Object Classification

SSD menambahkan lapisan fitur konvolisional pada akhir jaringan dasar. Tiap layer mengalami penurunan ukuran dan memungkinkan prediksi deteksi dilakukan dalam beberapa skala. Model konvolusi untuk melakukan prediksi deteksi berbeda pada tiap lapisan fiturnya. Setiap lapisan fitur yang ditambahkan dapat menghasilkan satu set tetap dari prediksi deteksi menggunakan satu set filter konvolisional. Tiap lapisan fitur dengan ukuran $m \times n \times p$ dihasilkan skor untuk sebuah kategori objek atau sebuah bentuk yang dengan koordinat relatif terhadap *default box*. Tahapan ini berbe-

¹<https://towardsdatascience.com/review-mobilenetv1-depthwise-separable-convolution-light-weight-model-a382df364b69>



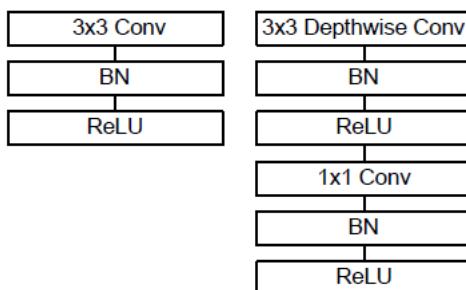
Gambar 3.13: Diagram Proses Depthwise Separable Convolution^a

^a<https://towardsdatascience.com/review-mobilenetv1-depthwise-separable-convolution-light-weight-model-a382df364b69>

Table 1. MobileNet Body Architecture

Type / Stride	Filter Shape	Input Size
Conv / s2	$3 \times 3 \times 3 \times 32$	$224 \times 224 \times 3$
Conv dw / s1	$3 \times 3 \times 32$ dw	$112 \times 112 \times 32$
Conv / s1	$1 \times 1 \times 32 \times 64$	$112 \times 112 \times 32$
Conv dw / s2	$3 \times 3 \times 64$ dw	$112 \times 112 \times 64$
Conv / s1	$1 \times 1 \times 64 \times 128$	$56 \times 56 \times 64$
Conv dw / s1	$3 \times 3 \times 128$ dw	$56 \times 56 \times 128$
Conv / s1	$1 \times 1 \times 128 \times 128$	$56 \times 56 \times 128$
Conv dw / s2	$3 \times 3 \times 128$ dw	$56 \times 56 \times 128$
Conv / s1	$1 \times 1 \times 128 \times 256$	$28 \times 28 \times 128$
Conv dw / s1	$3 \times 3 \times 256$ dw	$28 \times 28 \times 256$
Conv / s1	$1 \times 1 \times 256 \times 256$	$28 \times 28 \times 256$
Conv dw / s2	$3 \times 3 \times 256$ dw	$28 \times 28 \times 256$
Conv / s1	$1 \times 1 \times 256 \times 512$	$14 \times 14 \times 256$
5× Conv dw / s1	$3 \times 3 \times 512$ dw	$14 \times 14 \times 512$
	$1 \times 1 \times 512 \times 512$	$14 \times 14 \times 512$
Conv dw / s2	$3 \times 3 \times 512$ dw	$14 \times 14 \times 512$
Conv / s1	$1 \times 1 \times 512 \times 1024$	$7 \times 7 \times 512$
Conv dw / s2	$3 \times 3 \times 1024$ dw	$7 \times 7 \times 1024$
Conv / s1	$1 \times 1 \times 1024 \times 1024$	$7 \times 7 \times 1024$
Avg Pool / s1	Pool 7×7	$7 \times 7 \times 1024$
FC / s1	1024 × 1000	$1 \times 1 \times 1024$
Softmax / s1	Classifier	$1 \times 1 \times 1000$

Gambar 3.14: Arsitektur MobileNet[20].



Gambar 3.15: Perbandingan antara Konvolusi Standar dan *Depthwise Separable Convolutions*[20].

da dengan YOLO yang langsung meletakkan *fully-connected layer* daripada filter konvolusi.

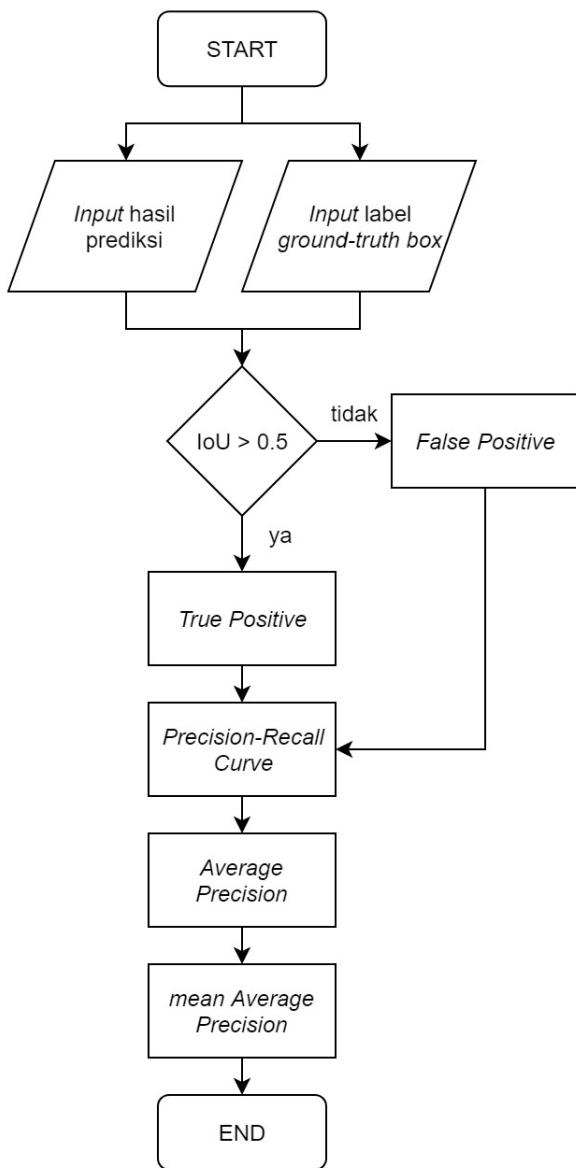
3.7.2.4 Pengolahan *Output*

Pendekatan SSD dibuat berdasarkan pada *feed-forward convolutional neural network* yang menghasilkan beberapa *bounding box* dan skor kehadiran objek yang terdapat pada *bounding box* tersebut, dilanjutkan dengan proses *non-maximum suppression* untuk menghasilkan deteksi akhir. Untuk setiap kotak k yang terdapat pada lokasi tertentu pada *feature map*, dilakukan komputasi skor kelas c dan 4 *offset* relatif terhadap bentuk asli *default box*. Hal ini menghasilkan total $(c + 4) \cdot k$ filter yang diaplikasikan pada tiap lokasi di *feature map*, yang kemudian menghasilkan $(c + 4) \cdot k \cdot m \cdot n$ output untuk sebuah $m \times n$ *feature map*.

3.8 Analisis Hasil Pendekripsi Wajah

Pada tahap ini dilakukan analisis dari hasil pendekripsi wajah oleh sistem. Proses analisis ini diilustrasikan pada Gambar 3.16. Sistem pendekripsi wajah akan memberikan hasil pendekripsi berupa *confidence* dan *predicted bounding box*. Hasil dari pendekripsi wajah tersebut kemudian dibandingkan dengan *ground-truth bounding box* yang telah diberikan sebelumnya pada proses pelabelan. Perbandingan ini dilakukan dengan cara menghitung nilai *Intersection over Union* (IoU) atau nilai *overlap* antara *ground-truth bounding box* dan *predicted bounding box*. Apabila nilai IoU dari kedua *bounding box* tersebut bernilai lebih dari *threshold* yang ditentukan dalam penelitian ini, yaitu 0.5, maka hasil pendekripsi tersebut dianggap benar. Sebaliknya, apabila ternyata nilai IoU yang didapatkan kurang dari 0.5, maka hasil pendekripsi tersebut dianggap salah.

Dari hasil penghitungan IoU didapatkan nilai *True Positive*, *False Positive*, dan *False Negative* yang digunakan untuk mencari nilai *precision* dan *recall* dari hasil pendekripsi wajah. Nilai *precision* dan *recall* kemudian digambarkan ke dalam sebuah kurva yang dinamakan *precision-recall curve*. Nilai *Average Precision* (AP) sendiri didapatkan dari hasil penghitungan area di bawah kurva dari tiap kelas yang terdeteksi oleh sistem. Sedangkan untuk nilai *mean Average Precision* (mAP) didapatkan dengan cara merata-ratakan



Gambar 3.16: Diagram Alir Proses Analisis.

nilai AP dari seluruh kelas yang terdeteksi. Namun, karena pada penelitian ini hanya terdapat satu kelas yang akan dideteksi, yaitu wajah, maka dapat ditarik kesimpulan bahwa nilai mAP yang dihasilkan sama dengan nilai dari AP.

3.8.1 *Confusion Matrix*

Confusion Matrix merupakan salah satu metode yang dapat digunakan untuk mengukur kinerja suatu sistem klasifikasi. *Confusion matrix* berisi informasi detail mengenai hasil klasifikasi (prediksi) oleh sistem terhadap data testing yang telah diketahui kelasnya (aktual) dan biasanya disusun dalam bentuk matriks. Pada Tabel 3.4, elemen pada diagonal utama (kiri atas ke kanan bawah) menunjukkan jumlah data testing yang dikenali dengan benar atau sesuai kelasnya, sedangkan yang di luar diagonal utama (kanan atas ke kiri bawah) adalah data yang salah dikenali oleh sistem[21].

Tabel 3.4: *Confusion Matrix* untuk Dua Kelas.

		Aktual	
		Positif	Negatif
Prediksi	Positif	TP	FP
	Negatif	FN	TN

Tabel 3.4 merupakan *confusion matrix* untuk dua kelas. Berikut adalah penjelasan dari kondisi-kondisi di atas:

1. *True Positive* (TP) merupakan jumlah data positif yang terklasifikasi positif oleh sistem.
2. *False Positive* (FP) merupakan yaitu jumlah data negatif yang terklasifikasi positif oleh sistem (*type I error*).
3. *False Negative* (FN) merupakan yaitu jumlah data positif yang terklasifikasi negatif oleh sistem (*type 2 error*).
4. *True Negative* (TN) merupakan jumlah data negatif yang terklasifikasi negatif oleh sistem.

Beberapa instrumen pengukuran untuk mengevaluasi kinerja sistem yang tercantum pada *confusion matrix* adalah:

1. *Precision*: Persentase prediksi positif yang benar dari seluruh hasil prediksi.

2. *Recall*: Persentase data testing (aktual) positif yang diprediksi benar (sebagai positif).
3. *Specificity*: Persentase data testing (aktual) negatif yang diprediksi benar (sebagai negatif).
4. *Accuracy*: Persentase jumlah total prediksi benar.

Pada penelitian ini instrumen pengukuran yang digunakan untuk mengukur kinerja sistem pendekripsi wajah adalah *precision* dan *recall*. Nilai *precision* dan *recall* dari sistem dapat diketahui dengan menggunakan persamaan 3.4 dan persamaan 3.5.

$$Precision = \frac{TP}{TP + FP} \quad (3.4)$$

$$Recall = \frac{TP}{TP + FN} \quad (3.5)$$

3.8.2 *Intersection over Union*

Intersection over Union (IoU) adalah metrik evaluasi yang digunakan untuk mengukur keakuratan pendekripsi objek pada dataset tertentu. IoU dapat digunakan untuk menilai sistem pendekripsi objek dengan ketentuan sebagai berikut:

1. Modelnya menghasilkan prediksi koordinat (x, y) , misalnya *bounding box* bagi objek di dalam gambar.
2. Memiliki *ground-truth bounding box* di dalam *dataset* objek.
3. Memiliki *predicted bounding box* dari model.

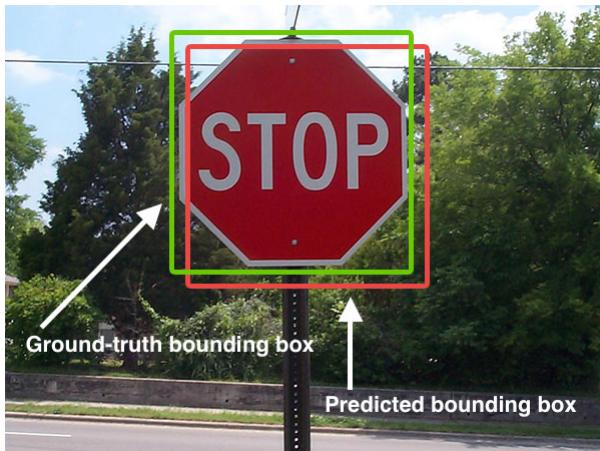
Biasanya IoU digunakan untuk mengevaluasi HOG+*Linear SVM* dan pendekripsi objek berbasis CNN. Tetapi selama model tersebut memiliki dua set *bounding box* sesuai dengan ketentuan poin 2 dan 3 di atas, maka IoU dapat diterapkan di dalam model tersebut[22]. Pada Gambar 3.17, *predicted bounding box* digambarkan dengan kotak berwarna merah sementara *ground-truth bounding box* digambarkan dengan kotak berwarna hijau.

IoU merupakan perbandingan antara area potongan dengan area gabungan dari *predicted bounding box* dan *ground-truth boun-*

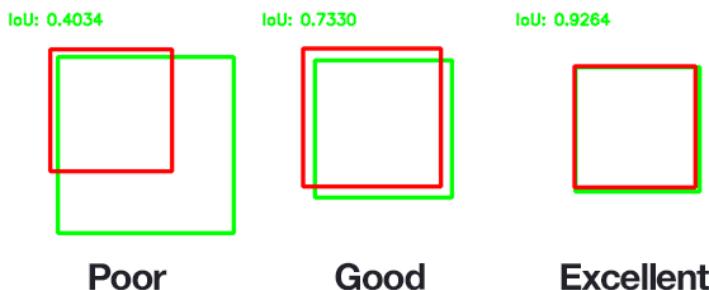
ding box yang dapat dituliskan dengan persamaan 3.6.

$$IoU = \frac{\text{Area of Overlap}}{\text{Area of Union}} \quad (3.6)$$

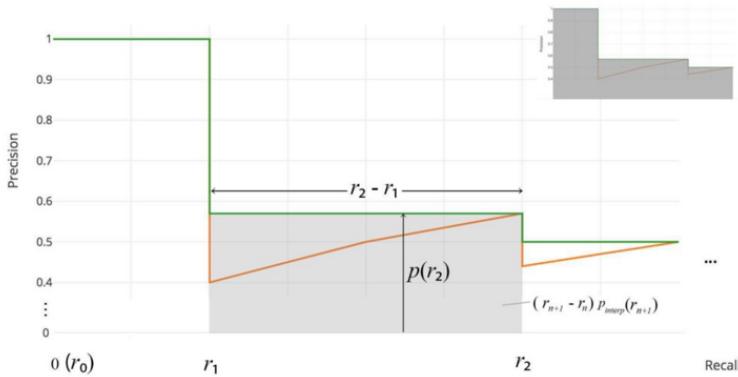
Normalnya, sebuah model dapat dikatakan bagus apabila skor IoU yang didapatkan dari prediksinya lebih dari 0,5. Gambar 3.18 menggambarkan contoh perbandingan antara skor IoU yang buruk, bagus, dan sangat bagus.



Gambar 3.17: Contoh *Bounding Boxes* pada Objek yang Dideteksi².



Gambar 3.18: Variasi Hasil Skor IoU².



Gambar 3.19: *Precision-Recall Curve*³.

3.8.3 *mean Average Precision* (mAP)

mean Average Precision (mAP) adalah metrik evaluasi yang mengukur keakuratan dari sebuah pendekripsi objek dalam melaksanakan proses pendekripsi. Nilai mAP merupakan rata-rata dari nilai *Average Precision* (AP) dari tiap kelas objek. AP didapatkan dengan cara menghitung area di bawah *precision-recall curve* yang dibuat berdasarkan nilai *precision* dan *recall* seperti yang terdapat pada persamaan 3.4 dan 3.5. Setelah kedua nilai tersebut didapatkan, maka AP dapat dihitung dengan persamaan 3.7 dan 3.8. Kurva *precision-recall* ditunjukkan oleh gambar 3.19.

$$AP = \sum (recall_{n+1} - recall_n) \times precision_{interp} \times (recall_{n+1}) \quad (3.7)$$

$$precision_{interp}(r_{n+1}) = \max_{\tilde{r} \geq r_{n+1}} p(\tilde{r}) \quad (3.8)$$

²<https://www.pyimagesearch.com/2016/11/07/intersection-over-union-iou-for-object-detection/>

³<https://medium.com/jonathanhui/map-mean-average-precision-for-object-detection-45c121a31173>

BAB 4

PENGUJIAN DAN ANALISIS

Pada penelitian ini, dipaparkan hasil pengujian serta analisis dari desain sistem dan implementasi. Data yang digunakan dalam pengujian diperoleh dengan merekam wajah dari orang yang lewat pada area di depan Laboratorium Telematika B201 dengan menggunakan *IP camera* yang terletak di atas pintu Laboratorium. Pengujian dilakukan dalam beberapa bagian sebagai berikut:

1. Pengujian Berdasarkan Lokasi Peletakan Kamera.
2. Pengujian Berdasarkan Waktu Pengambilan Data.
3. Pengujian Berdasarkan Jumlah Objek.
4. Pengujian Berdasarkan Kecepatan Objek Bergerak.
5. Pengujian Berdasarkan Perangkat Keras yang Digunakan.
6. Pengujian dengan Tambahan Pencahayaan Buatan.
7. Pengujian dengan Ketinggian Kamera Sejajar dengan Wajah.
8. Pengujian Berdasarkan Sudut Kamera.

4.1 Pengujian Berdasarkan Lokasi Peletakan Kamera

Pengujian berdasarkan pada lokasi peletakan kamera bertujuan untuk mengetahui pengaruh lokasi peletakan kamera terhadap hasil pendekripsi wajah. Berdasarkan desain sistem, terdapat dua kamera yang terletak di dua lokasi yang berbeda. Kamera pertama terletak di atas pintu Laboratorium Telematika B201 dan membelakangi sinar matahari, sedangkan kamera kedua terletak di palang besi yang ditempelkan pada tiang di depan area Laboratorium Telematika B201 dan menghadap ke arah sinar matahari. Waktu pengambilan data dilakukan pada waktu yang sama yaitu pada waktu siang hari. Selain posisi kamera, pengujian juga dilakukan dengan memberikan objek penghalang yang berbeda pada wajah dengan total 16 kombinasi objek penghalang, yaitu:

1. Normal tanpa objek penghalang.
2. Kacamata.
3. Masker.
4. Topi.

5. *Hoodie*.
6. Kacamata, masker.
7. Kacamata, topi.
8. Kacamata, *hoodie*.
9. Masker, topi.
10. Masker, *hoodie*.
11. Topi, *hoodie*.
12. Kacamata, masker, topi.
13. Kacamata, masker, *hoodie*.
14. Kacamata, topi, *hoodie*.
15. Masker, topi, *hoodie*.
16. Kacamata, masker, topi, *hoodie*.

Dari data video yang telah diambil, didapatkan rata-rata nilai *brightness* dan intensitas *contrast* dari wajah yang akan mendekripsi sebagaimana yang terdapat pada Gambar 4.1 dan 4.2 serta perbandingan keduanya pada Tabel 4.1. Setelah dilewatkan pada sistem mendekripsi wajah berbasis SSD dan YOLO, serta dilakukan analisis menggunakan mAP, didapatkan hasil seperti yang tertera pada Tabel 4.2 dan 4.3. Grafik dari kedua tabel dapat dilihat pada Gambar 1 dan 2 yang terdapat pada bagian lampiran. Jika dilihat berdasarkan hasil pengujian dari skenario yang diberikan, tidak ada metode yang benar-benar konsisten memberikan nilai mAP yang stabil.

Pada posisi kamera yang menghadap ke arah sinar matahari, seperti pada Gambar 4.3, rata-rata *brightness* dari wajah bernilai 59.681 dan intensitas *contrast*-nya bernilai 0.257. Sistem berbasis SSD cukup kesulitan untuk mendekripsi wajah dengan penghalang masker yang dikombinasikan dengan objek penghalang lain, dimana sistem berbasis SSD bahkan tidak mampu melakukan mendekripsi sama sekali dengan nilai mAP 0%. Sedangkan sistem berbasis YOLO mengalami kesulitan untuk mendekripsi wajah dengan penghalang topi, dimana sistem berbasis YOLO mendapatkan nilai 12.5%. Berkebalikan dari sistem berbasis SSD, sistem berbasis YOLO mampu melakukan mendekripsi dengan baik terhadap wajah yang dihalangi masker dengan nilai mAP sebesar 77.91%, demikian juga ketika masker dikombinasikan dengan penghalang lain, sistem berbasis YOLO tetap mampu memberikan performa yang lebih baik jika dibandingkan dengan sistem berbasis SSD.

Tabel 4.1: Perbandingan *Brightness* dan *Contrast* pada Lokasi Peletakan Kamera yang Berbeda

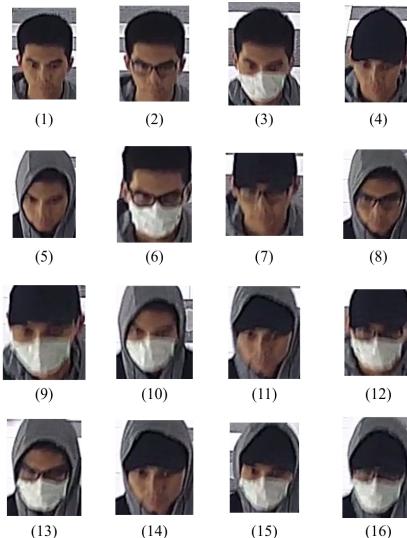
No.	Skenario	Membelakangi Sinar Matahari		Menghadap Sinar Matahari	
		<i>Brightness</i>	<i>Contrast</i>	<i>Brightness</i>	<i>Contrast</i>
1	Normal	80.889	0.131	62.452	0.002
2	Kacamata	79.961	0.120	60.496	0.375
3	Masker	96.174	0.009	64.631	0.649
4	Topi	76.554	0.513	65.255	0.506
5	Hoodie	87.305	0.639	51.577	0.596
6	Kacamata, Masker	87.546	0.649	64.869	0.091
7	Kacamata, Topi	68.910	0.055	67.467	0.031
8	Kacamata, Hoodie	87.697	0.643	53.268	0.369
9	Masker, Topi	85.117	0.641	79.567	0.471
10	Masker, Hoodie	101.270	0.591	59.105	0.117
11	Topi, Hoodie	75.661	0.694	55.485	0.146
12	Kacamata, Masker, Topi	79.979	0.686	74.581	0.126
13	Kacamata, Masker, Hoodie	102.594	0.182	50.373	0.188
14	Kacamata, Topi, Hoodie	79.190	0.667	41.233	0.016
15	Masker, Topi, Hoodie	87.200	0.232	48.403	0.273
16	Kacamata, Masker, Topi, Hoodie	95.066	0.511	56.138	0.160
Rata-rata		85.695	0.435	59.681	0.257

Tabel 4.2: Hasil Pengujian Berdasarkan Lokasi Peletakan Kamera Membelakangi Sinar Matahari

No	SSD			YOLO		
	Waktu Pemrosesan (s)	FPS	mAP	Waktu Pemrosesan (s)	FPS	mAP
1	8.368	22.788	57.34%	33.682	6.264	84.75%
2	133.893	28.048	74.03%	34.199	6.287	78.46%
3	10.299	21.846	64.35%	35.456	6.346	80.00%
4	8.352	26.819	48.35%	34.489	6.495	12.28%
5	8.702	26.432	75.39%	35.947	6.398	70.49%
6	7.719	25.262	41.30%	30.185	6.46	93.22%
7	7.504	25.321	49.55%	29.536	6.433	18.18%
8	7.672	26.719	70.35%	31.04	6.604	79.37%
9	7.55	27.153	34.02%	32.062	6.394	40.35%
10	8.084	27.216	40.00%	33.621	6.544	63.94%
11	9.196	25.012	54.69%	35.052	6.562	56.25%
12	8.494	26.489	14.84%	34.147	6.589	11.48%
13	7.14	27.311	31.25%	30.137	6.471	60.94%
14	9.765	23.553	54.69%	36.435	6.313	56.25%
15	7.763	25.764	30.07%	31.016	6.448	47.69%
16	8.729	26.349	29.82%	36.274	6.341	30.08%
Rata-rata	8.368	25.755	48.13%	33.330	6.434	55.23%
Total	133.893	-	-	533.278	-	-

Tabel 4.3: Hasil Pengujian Berdasarkan Lokasi Peletakan Kamera Menghadap Sinar Matahari

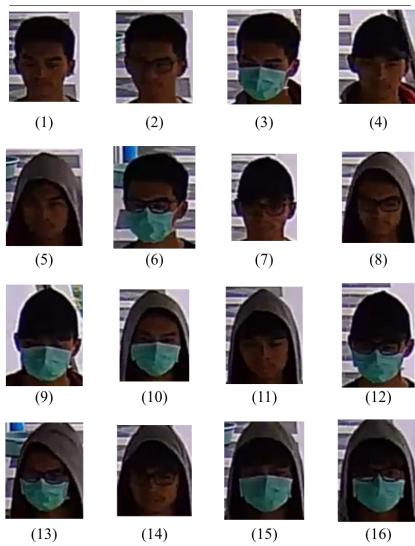
No	SSD			YOLO		
	Waktu Pemrosesan (s)	FPS	mAP	Waktu Pemrosesan (s)	FPS	mAP
1	11.552	22.247	72.01%	42.913	5.989	47.22%
2	11.055	24.243	53.27%	44.054	6.083	35.11%
3	13.789	25.237	38.76%	56.499	6.159	77.91%
4	12.06	25.042	43.29%	48.677	6.204	12.50%
5	12.555	22.302	37.91%	45.229	6.191	22.22%
6	10.798	21.949	41.30%	38.261	6.194	70.65%
7	11.696	23.426	45.40%	44.258	6.191	22.77%
8	14.009	23.842	60.98%	53.867	6.2	43.53%
9	12.875	23.767	0.84%	49.445	6.189	44.85%
10	13.296	23.842	5.49%	51.867	6.112	67.84%
11	14.885	23.446	49.25%	56.497	6.177	35.79%
12	11.522	24.214	1.23%	44.882	6.216	26.25%
13	10.873	23.911	0.00%	41.923	6.202	54.06%
14	13.059	24.121	53.45%	51.555	6.11	37.93%
15	13.379	23.918	0.00%	51.523	6.211	61.16%
16	13.408	23.866	0.00%	51.559	6.206	25.23%
Rata-rata	12.551	23.711	31.45%	48.313	6.165	43%
Total	200.811	379.373	-	773.009	98.634	-



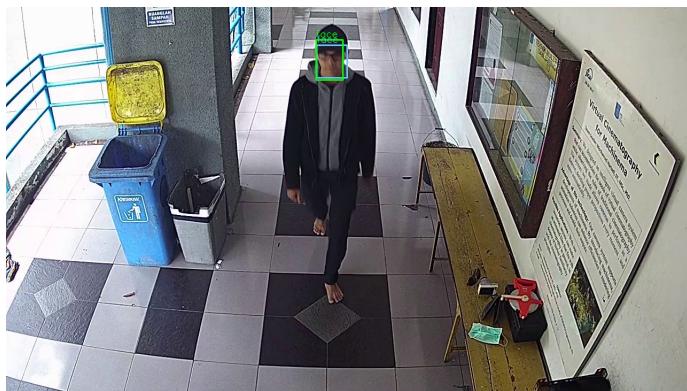
Gambar 4.1: Potongan Gambar Wajah dari Data Video Kamera yang Membelakangi Sinar Matahari.

Hal ini cukup berbeda pada posisi kamera yang membelakangi sinar matahari, seperti pada Gambar 4.4, dimana rata-rata *brightness* dari wajah bernalilai 85.695 dan intensitas *contrast*-nya bernalilai 0.435. Performa sistem berbasis SSD mengalami peningkatan nilai mAP sebesar $\pm 17\%$ dan sistem berbasis YOLO mengalami peningkatan nilai mAP sebesar $\pm 12\%$. Hal ini membuktikan bahwa lokasi peletakan kamera sangat memengaruhi performa pendekripsi sistem. Salah satu hal yang menyebabkan terjadinya perbedaan performa ini adalah paparan sinar matahari yang berbeda. Kamera pertama membelakangi sinar matahari sehingga objek terlihat lebih terang pada kamera, sedangkan kamera kedua menghadap ke arah sinar matahari yang menyebabkan terjadinya *highlight* sehingga membuat video yang dihasilkan menjadi terlihat lebih gelap.

Secara keseluruhan, pada posisi kamera menghadap yang menghadap sinar matahari, sistem berbasis YOLO berhasil mengungguli rata-rata nilai mAP sistem berbasis SSD dengan selisih sebesar



Gambar 4.2: Potongan Gambar Wajah dari Data Video Kamera yang Menghadap Sinar Matahari.



Gambar 4.3: Dokumentasi Pengujian Pada Kamera yang Membelaangi Sinar Matahari.



Gambar 4.4: Dokumentasi Pengujian Pada Kamera yang Menghadap Sinar Matahari.

9.55% dan pada posisi kamera membelakangi matahari, sistem berbasis YOLO tetap berhasil mengungguli rata-rata nilai mAP sistem berbasis SSD dengan selisih sebesar 7.1%. Namun, performa sistem berbasis YOLO yang baik tidak diimbangi dengan waktu pemrosesan yang cepat, dimana sistem berbasis YOLO membutuhkan waktu proses hampir empat kali lebih lambat jika dibandingkan dengan sistem berbasis SSD. Begitu pula dengan fps yang didapatkan oleh sistem berbasis YOLO hanya mencapai 6 fps, yang berbeda jauh jika dibandingkan dengan sistem berbasis SSD yang mencapai 25 fps.

4.2 Pengujian Berdasarkan Waktu Pengambilan Data

Pengujian berdasarkan waktu pengambilan data bertujuan untuk mengetahui pengaruh waktu pengambilan data terhadap performa komputasi dan hasil pendekripsi wajah. Pada pengujian ini hanya digunakan satu kamera yang mampu memberikan hasil lebih baik pada pengujian berdasarkan lokasi peletakan kamera sebelumnya, yaitu kamera yang membelakangi sinar matahari. Waktu pengambilan data dilakukan pada waktu siang dan malam hari sehingga didapatkan data dengan kondisi cahaya yang sangat jauh berbeda. Pengujian ini dilakukan dengan mengambil data sesuai

dengan skenario yang dibuat sebelumnya, yaitu:

1. Normal.
2. Kacamata.
3. Masker.
4. Topi.
5. Hoodie.
6. Kacamata, masker.
7. Kacamata, topi.
8. Kacamata, *hoodie*.
9. Masker, topi.
10. Masker, *hoodie*.
11. Topi, *hoodie*.
12. Kacamata, masker, topi.
13. Kacamata, masker, *hoodie*.
14. Kacamata, topi, *hoodie*.
15. Masker, topi, *hoodie*.
16. Kacamata, masker, topi, *hoodie*.
17. Kacamata hitam.
18. Kacamata Hitam, masker.
19. Kacamata Hitam, topi.
20. Kacamata Hitam, *hoodie*.
21. Kacamata Hitam, masker, topi.
22. Kacamata Hitam, masker, *hoodie*.
23. Kacamata Hitam, topi, *hoodie*.
24. Kacamata Hitam, masker, topi, *hoodie*.

Dari data video yang telah diambil, didapatkan rata-rata nilai *brightness* dan intensitas *contrast* dari wajah yang akan dideteksi sebagaimana yang terdapat pada Gambar 4.1 dan 4.5 serta perbandingan keduanya pada Tabel 4.4. Seluruh skenario di atas diwakilkan oleh nomor yang sama pada Tabel 4.5 dan 4.6. Grafik dari kedua tabel dapat dilihat pada Gambar 3 dan 4 yang terdapat pada bagian lampiran.

Berdasarkan hasil pengujian, terlihat ketika pengujian dilakukan pada waktu siang hari, dimana rata-rata *brightness* dari wajah bernilai 85.695 dan rata-rata intensitas *contrast*-nya bernilai 0.435, rata-rata nilai mAP yang dihasilkan oleh sistem berbasis SSD dan sistem berbasis YOLO terpaut sekitar 7.63%. Tapi masih dengan

Tabel 4.4: Perbandingan *Brightness* dan *Contrast* pada Waktu yang Berbeda

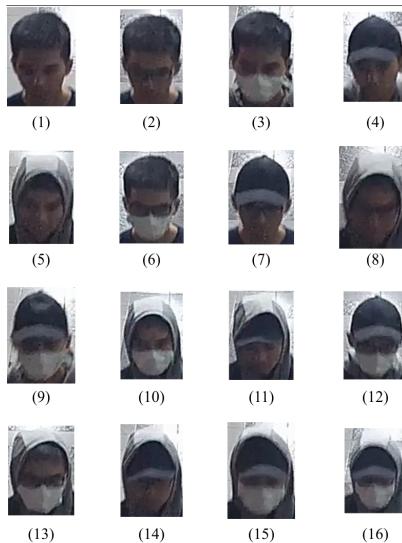
No.	Skenario	Siang Hari		Malam Hari	
		<i>Brightness</i>	<i>Contrast</i>	<i>Brightness</i>	<i>Contrast</i>
1	Normal	80.889	0.131	94.547	0.351
2	Kacamata	79.961	0.120	87.970	0.405
3	Masker	96.174	0.009	92.757	0.330
4	Topi	76.554	0.513	97.307	0.612
5	Hoodie	87.305	0.639	96.795	0.265
6	Kacamata, Masker	87.546	0.649	96.403	0.180
7	Kacamata, Topi	68.910	0.055	97.530	0.331
8	Kacamata, Hoodie	87.697	0.643	96.740	0.281
9	Masker, Topi	85.117	0.641	105.109	0.291
10	Masker, Hoodie	101.270	0.591	116.840	0.244
11	Topi, Hoodie	75.661	0.694	103.954	0.241
12	Kacamata, Masker, Topi	79.979	0.686	104.482	0.318
13	Kacamata, Masker, Hoodie	102.594	0.182	102.167	0.182
14	Kacamata, Topi, Hoodie	79.190	0.667	94.708	0.311
15	Masker, Topi, Hoodie	87.200	0.232	100.006	0.314
16	Kacamata, Masker, Topi, Hoodie	95.066	0.511	111.182	0.276
Rata-rata		85.695	0.435	99.906	0.308

Tabel 4.5: Hasil Pengujian pada Waktu Siang Hari

No	SSD			YOLO		
	Waktu Pemrosesan (s)	FPS	mAP	Waktu Pemrosesan (s)	FPS	mAP
1	9.259	22.788	57.34%	33.682	6.264	84.75%
2	7.665	28.048	74.03%	34.199	6.287	78.46%
3	10.299	21.846	64.35%	35.456	6.346	80.00%
4	8.352	26.819	48.35%	34.489	6.495	12.28%
5	8.702	26.432	75.39%	35.947	6.398	70.49%
6	7.719	25.262	41.30%	30.185	6.46	93.22%
7	7.504	25.321	49.55%	29.536	6.433	18.18%
8	7.672	26.719	70.35%	31.04	6.604	79.37%
9	7.55	27.153	34.02%	32.062	6.394	40.35%
10	8.084	27.216	40.00%	33.621	6.544	63.94%
11	9.196	25.012	54.69%	35.052	6.562	56.25%
12	8.494	26.489	14.84%	34.147	6.589	11.48%
13	7.14	27.311	31.25%	30.137	6.471	60.94%
14	9.765	23.553	54.69%	36.435	6.313	56.25%
15	7.763	25.764	30.07%	31.016	6.448	47.69%
16	8.729	26.349	29.82%	36.274	6.341	30.08%
17	8.128	27.069	60.74%	33.833	6.503	80.22%
18	8.227	23.704	29.03%	30.737	6.344	80.65%
19	8.28	25.361	40.86%	32.907	6.382	31.75%
20	9.004	24.435	47.36%	34.205	6.432	30.60%
21	8.874	25.355	23.82%	35.045	6.42	24.24%
22	9.394	21.823	28.44%	31.768	6.453	64.29%
23	8.222	27.972	37.44%	35.169	6.54	33.33%
24	11.636	26.298	15.87%	46.742	6.547	7.94%
Rata-rata	8.569	25.587	43.90%	33.904	6.440	51.53%
Total	205.658	-	-	813.684	-	-

Tabel 4.6: Hasil Pengujian pada Waktu Malam Hari

No	SSD			YOLO		
	Waktu Pemrosesan (s)	FPS	mAP	Waktu Pemrosesan (s)	FPS	mAP
1	9.259	25.156	14.93%	34.183	6.29	73.13%
2	7.665	27.32	21.32%	29.775	6.549	62.12%
3	10.299	27.438	7.58%	31.601	6.487	71.21%
4	8.352	26.419	23.33%	28.508	6.489	73.33%
5	8.702	26.82	43.79%	29.205	6.506	81.82%
6	7.719	27.988	3.03%	30.548	6.547	59.09%
7	7.504	27.161	4.72%	29.706	6.564	73.49%
8	7.672	26.744	22.89%	29.605	6.418	81.67%
9	7.55	27.452	0.00%	31.151	6.581	93.63%
10	8.084	27.357	29.58%	32.083	6.546	70.34%
11	9.196	27.657	16.82%	29.757	6.553	65.45%
12	8.494	26.132	0.00%	28.962	6.56	76.78%
13	7.14	27.83	5.97%	31.355	6.538	66.81%
14	9.765	26.707	18.18%	31.119	6.523	66.67%
15	7.763	26.005	4.00%	30.467	6.565	99.89%
16	8.729	24.824	3.27%	30.095	6.48	85.97%
17	8.128	25.486	13.43%	32.081	6.546	67.16%
18	8.227	28.118	6.06%	33.213	6.473	59.09%
19	8.28	26.355	9.99%	29.803	6.543	87.69%
20	9.004	27.27	29.81%	32.326	6.496	64.04%
21	8.874	26.508	0.00%	31.927	6.578	72.31%
22	9.394	27.461	16.67%	33.494	6.568	58.97%
23	8.222	27.069	17.65%	28.264	6.546	86.27%
24	11.636	27.567	22.81%	33.143	6.487	82.35%
Rata-rata	8.569	26.869	13.99%	30.932	6.518	74.14%
Total	205.658	-	-	742.371	-	-



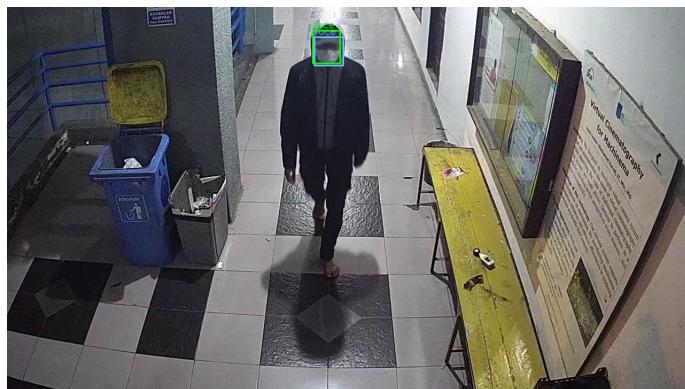
Gambar 4.5: Potongan Gambar Wajah dari Data Video Kamera pada Waktu Malam Hari.

masalah yang sama seperti pengujian sebelumnya, waktu pemrosesan sistem berbasis YOLO selama 813.684 detik masih lebih lambat empat kali lipat jika dibandingkan dengan sistem berbasis SSD yang hanya memerlukan waktu 205.658 detik untuk memproses seluruh skenario. Gambar 4.6 menunjukkan dokumentasi dari pengambilan data pada waktu siang hari.

Meskipun sistem berbasis YOLO tergolong lambat untuk melakukan pemrosesan data, namun hal tersebut justru baru sangat terasa ketika pengujian dilakukan pada waktu malam hari, dimana rata-rata *brightness* dari wajah bernilai 99.906 dan rata-rata intensitas *contrast*-nya bernilai 0.308. Rata-rata nilai mAP dari sistem berbasis YOLO sebesar 74.14% unggul jauh jika dibandingkan rata-rata nilai mAP dari sistem berbasis SSD yang hanya sebesar 13.99%, yang berarti sistem berbasis YOLO mampu memberikan performa sekitar lima kali lipat lebih baik dari sistem berbasis SSD. Gambar 4.7 menunjukkan dokumentasi dari pengambilan data pada waktu



Gambar 4.6: Dokumentasi Pengujian pada Waktu Siang Hari.



Gambar 4.7: Dokumentasi Pengujian pada Waktu Malam Hari.

malam hari.

4.3 Pengujian Berdasarkan Jumlah Objek

Pengujian berdasarkan pada jumlah objek bertujuan untuk mengetahui pengaruh jumlah objek terhadap kemampuan sistem dalam mendeteksi wajah. Pada pengujian ini hanya digunakan satu kamera yang membelakangi sinar matahari dan waktu pengambilan data dilakukan pada waktu siang hari. Terdapat sembilan kondisi

jumlah objek yang digunakan seperti yang terdapat pada Gambar 3.2, antara lain:

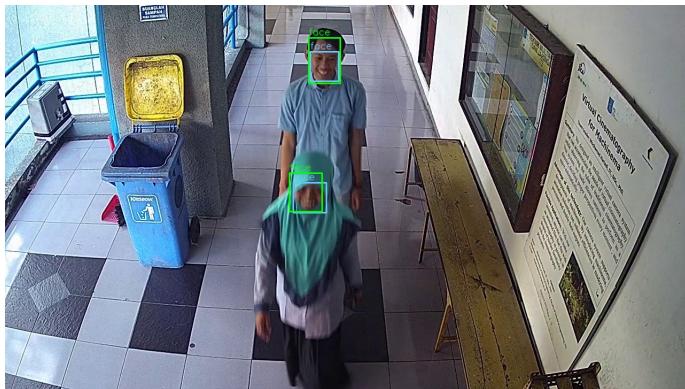
1. Dua orang berjalan sejajar.
2. Dua orang berjalan beriringan.
3. Tiga orang berjalan sejajar.
4. Dua orang berjalan sejajar diikuti oleh satu orang berjalan sendiri di belakangnya.
5. Satu orang berjalan sendiri diikuti oleh dua orang berjalan sejajar di belakangnya.
6. Empat orang berjalan sejajar.
7. Tiga orang berjalan sejajar diikuti oleh satu orang berjalan sendiri di belakangnya.
8. Satu orang berjalan sendiri diikuti oleh tiga orang berjalan sejajar di belakangnya.
9. Satu orang berjalan sendiri diikuti oleh dua orang yang berjalan sejajar dan diikuti lagi oleh satu orang berjalan sendiri di belakangnya.
10. Dua orang berjalan sejajar diikuti oleh dua orang yang juga berjalan sejajar di belakangnya.

Berdasarkan skenario di atas, didapatkan hasil pengujian pada Tabel 4.7 dan grafik pada Gambar 5 yang terdapat pada bagian lampiran. Dari hasil pengujian, semua skenario hampir berhasil mendeteksi seluruh wajah yang terdapat pada skenario dengan nilai mAP yang relatif stabil. Terdapat sebuah galat pada SSD dimana pada skenario nomor 10 terdapat tangan yang terdeteksi sebagai wajah, sehingga menyebabkan sistem mendeteksi 5 wajah pada video. Pada YOLO juga terdapat galat pada skenario nomor 5 dimana hanya 2 dari 3 wajah yang terdeteksi yang disebabkan karena YOLO gagal mendeteksi wajah salah satu objek yang terdapat pada video. Gambar 4.8 dan 4.9 merupakan dokumentasi dari pengambilan data berdasarkan jumlah objek.

Selain itu, pengujian juga dilakukan dengan cara melakukan pengambilan data langsung yang dilakukan di Pasar Jumat ITS dengan menggunakan kamera dari perangkat telepon genggam yang dikonfigurasi dengan resolusi yang sama dengan *IP camera*. Dari pengujian tersebut, didapatkan jumlah maksimal wajah yang mampu terdeteksi oleh sistem pendekripsi wajah berbasis SSD adalah

Tabel 4.7: Hasil Pengujian Berdasarkan Jumlah Objek

No	SSD				YOLO			
	Waktu Pemrosesan (s)	FPS	mAP	Wajah Terdeteksi	Waktu Pemrosesan (s)	FPS	mAP	Wajah Terdeteksi
1	10.358	25.489	57.78%	2	44.034	5.995	35.64%	2
2	9.163	29.249	55.25%	2	42.345	6.329	35.51%	2
3	8.334	28.556	49.16%	3	36.935	6.444	44.74%	3
4	8.203	28.648	51.44%	3	36.569	6.426	43.86%	3
5	8.488	28.394	53.47%	3	37.56	6.416	40.27%	2
6	8.614	29.021	47.78%	4	39.013	6.408	50.41%	4
7	8.731	27.717	50.71%	4	37.983	6.371	51.69%	4
8	9.249	29.083	50.70%	4	42.265	6.365	49.83%	4
9	9.508	29.237	47.83%	4	43.607	6.375	55.47%	4
10	10.142	28.396	52.19%	5	45.905	6.274	51.22%	4
Rata-rata	9.079	28.379	51.63%	-	40.622	6.340	45.86%	-
Total	90.79	-	-	-	406.216	-	-	-



Gambar 4.8: Dokumentasi Pengujian Berdasarkan Jumlah Objek.



Gambar 4.9: Dokumentasi Pengujian Berdasarkan Jumlah Objek.

sebanyak 30 wajah dan jumlah maksimal yang mampu terdeteksi oleh sistem pendekripsi wajah berbasis YOLO adalah sebanyak 11 wajah. Kedua hasil tersebut didapatkan dari video yang sama, dimana terdapat sekitar 50 orang lebih yang ada di dalam frame. Gambar 4.10 dan 4.11 menunjukkan dokumentasi dari pendekripsi wajah yang dilakukan di Pasar Jumat ITS.



Gambar 4.10: Pendektsian Wajah di Pasar Jumat ITS dengan Sistem Berbasis SSD.



Gambar 4.11: Pendektsian Wajah di Pasar Jumat ITS dengan Sistem Berbasis YOLO.

4.4 Pengujian Berdasarkan Kecepatan Objek Bergerak

Pengujian berdasarkan kecepatan objek bergerak bertujuan untuk mengetahui pengaruh kecepatan objek saat bergerak terhadap kemampuan sistem dalam mendeteksi wajah. Pada pengujian ini hanya digunakan satu kamera yang membelakangi sinar matahari dan waktu pengambilan data dilakukan pada waktu siang hari. Ukuran kecepatan didapatkan dengan mengukur waktu tempuh objek dari ujung atas sampai ujung bawah *frame*, waktu tempuh tersebut kemudian dikalkulasikan dengan jangkauan panjang kamera sehingga didapatkan kecepatan dari objek. Terdapat tiga kondisi kecepatan objek bergerak yang masing-masing diuji sebanyak tiga kali, antara lain:

1. Satu orang bergerak dengan kecepatan normal.
2. Satu orang bergerak dengan kecepatan agak cepat.
3. Satu orang bergerak dengan kecepatan cepat.

Dari hasil pengujian yang terdapat pada Tabel 4.8, objek bergerak dengan kecepatan rata-rata 0.68 m/s. Masing-masing sistem mampu melakukan pendekripsi terhadap objek pada kecepatan tersebut. Terdapat perbedaan antara kedua sistem, pada sistem berbasis SSD rata-rata waktu pemrosesan lebih cepat jika dibandingkan dengan sistem berbasis YOLO, perbedaan yang dihasilkan mencapai 21 detik. Rata-rata FPS yang dihasilkan oleh sistem berbasis SSD sebesar 24.24 fps lebih cepat jika dibandingkan dengan rata-rata FPS yang dihasilkan sistem berbasis YOLO yaitu sebesar 7.095 fps. Sedangkan untuk nilai mAP, sistem berbasis YOLO jauh mengungguli sistem berbasis SSD. Sistem berbasis SSD menghasilkan rata-rata nilai mAP sebesar 62.45%, sedangkan sistem berbasis YOLO menghasilkan rata-rata nilai mAP sebesar 92.31%.

Pada pengujian dengan objek yang bergerak agak cepat yang terdapat pada Tabel 4.9, objek bergerak dengan kecepatan rata-rata 1 m/s. Masing-masing sistem mampu melakukan pendekripsi terhadap objek pada kecepatan tersebut. Sistem berbasis SSD menghasilkan rata-rata waktu pemrosesan lebih cepat jika dibandingkan dengan sistem berbasis YOLO, perbedaan yang dihasilkan sebesar 14 detik. Rata-rata FPS yang dihasilkan oleh sistem ber-



Gambar 4.12: Dokumentasi Pengujian Berdasarkan Kecepatan Objek Bergerak.

basis SSD sebesar 24.512 fps lebih cepat jika dibandingkan dengan rata-rata FPS yang dihasilkan sistem berbasis YOLO yaitu sebesar 7.188 fps. Sedangkan untuk nilai mAP, sistem berbasis YOLO masih mengungguli sistem berbasis SSD. Sistem berbasis SSD menghasilkan rata-rata nilai mAP sebesar 62.56%, sedangkan sistem berbasis YOLO menghasilkan rata-rata nilai mAP sebesar 87.78%.

Pada pengujian dengan objek yang bergerak cepat yang terdapat pada Tabel 4.10, objek bergerak dengan kecepatan rata-rata 1.64 m/s. Masing-masing sistem mampu melakukan pendekripsi terhadap objek pada kecepatan tersebut. Sistem berbasis tetap SSD menghasilkan rata-rata waktu pemrosesan lebih cepat jika dibandingkan dengan sistem berbasis YOLO, perbedaan yang dihasilkan sebesar 8 detik. Rata-rata FPS yang dihasilkan oleh sistem berbasis SSD sebesar 22.699 fps juga masih lebih cepat jika dibandingkan dengan rata-rata FPS yang dihasilkan sistem berbasis YOLO yaitu sebesar 7.206 fps. Sedangkan untuk nilai mAP, sistem berbasis YOLO tetap mengungguli sistem berbasis SSD. Sistem berbasis SSD menghasilkan rata-rata nilai mAP sebesar 44.34%, sedangkan sistem berbasis YOLO menghasilkan rata-rata nilai mAP sebesar 71.73%. Gambar 4.12 menunjukkan dokumentasi pengujian objek tunggal dengan kecepatan 1.62 m/s.

Grafik perbandingan rata-rata nilai mAP dari ketiga penguji-

an berdasarkan kecepatan objek bergerak terdapat pada Gambar 6 yang terdapat pada bagian lampiran. Dari ketiga hasil pengujian tersebut, dapat disimpulkan bahwa semakin cepat kecepatan gerak dari suatu objek, maka semakin berkurang pula nilai mAP yang dihasilkan. Sedangkan nilai FPS yang dihasilkan kedua sistem cenderung stabil meskipun kecepatan dari objek berbeda-beda. Sistem berbasis SSD mampu memberikan waktu pemrosesan dan FPS yang lebih cepat jika dibandingkan dengan sistem berbasis YOLO, namun sistem berbasis YOLO mampu menghasilkan nilai mAP yang jauh lebih baik jika dibandingkan dengan sistem berbasis SSD. Hal tersebut menunjukkan bahwa sistem berbasis SSD lebih baik jika digunakan untuk melakukan pendekripsi objek yang bergerak dengan cepat, karena FPS yang dihasilkan mendekati FPS *real-time*. Sedangkan sistem berbasis YOLO kurang cocok digunakan untuk mendekripsi objek yang bergerak dengan cepat karena harus mengorbankan waktu pemrosesan dan FPS, meskipun memiliki hasil pendekripsi yang lebih akurat.

4.5 Pengujian Berdasarkan Perangkat Keras yang Digunakan

Pengujian berdasarkan perangkat keras yang digunakan bertujuan untuk mengetahui pengaruh perangkat keras yang digunakan untuk menjalankan sistem pendekripsi terhadap kemampuan sistem dalam mendekripsi wajah. Pada pengujian ini, data yang digunakan adalah data yang didapatkan dari pengujian kedua, yaitu data dari pengujian berdasarkan waktu pengambilan data. Terdapat tiga kondisi perangkat keras yang digunakan pada pengujian ini, antara lain:

1. Menjalankan sistem dengan menggunakan CPU Laptop (Intel Core i7-7700HQ).
2. Menjalankan sistem dengan menggunakan GPU Laptop (NVIDIA GTX1050 2GB).
3. Menjalankan sistem dengan menggunakan GPU Desktop (NVIDIA GTX1050Ti 4GB).

Dari hasil pengujian didapatkan perbandingan kemampuan dari masing-masing perangkat keras yang digunakan seperti yang terdapat pada Tabel 4.11 dan 4.12. Pada sistem berbasis SSD, per-

Tabel 4.8: Hasil Pengujian dengan Kecepatan Objek Bergerak Normal

Nomor Percobaan	SSD			YOLO			
	Kecepatan (m/s)	Waktu Pemrosesan (s)	FPS	mAP	Waktu Pemrosesan (s)	FPS	mAP
1	0.71	9.638	21.684	62.50%	30.404	6.874	90.62%
2	0.70	9.772	25.582	64.57%	34.668	7.211	95.89%
3	0.63	7.617	25.468	60.27%	26.948	7.199	90.41%
Rata-rata	0.68	9.009	24.24	62.45%	30.673	7.095	92.31%
Total	-	27.027	-	-	92.020	-	-

Tabel 4.9: Hasil Pengujian dengan Kecepatan Objek Bergerak Agak Cepat

Nomor Percobaan	Kecepatan (m/s)	SSD			YOLO		
		Waktu Pemrosesan (s)	FPS	mAP	Waktu Pemrosesan (s)	FPS	mAP
1	0.99	6.382	21.935	59.57%	19.328	7.243	85.11%
2	1.07	5.214	25.890	56.58%	18.708	7.216	83.67%
3	0.94	5.640	25.711	71.54%	20.409	7.105	94.55%
Rata-rata	1.00	5.745	24.512	62.56%	19.482	7.188	87.78%
Total	-	17.236	-	-	58.445	-	-

Tabel 4.10: Hasil Pengujian dengan Kecepatan Objek Bergerak Cepat

Nomor Percobaan	Kecepatan (m/s)	SSD			YOLO		
		Waktu Pemrosesan (s)	FPS	mAP	Waktu Pemrosesan (s)	FPS	mAP
1	1.62	3.686	23.059	40.62%	11.792	7.208	75.00%
2	1.60	3.712	22.900	53.12%	11.791	7.209	68.75%
3	1.71	3.388	22.138	39.29%	10.414	7.202	71.43%
Rata-rata	1.64	3.595	22.699	44.34%	11.332	7.206	71.73%
Total	-	10.786	-	-	33.997	-	-

bedaan utama dari hasil pendektsian dengan perangkat keras yang berbeda terdapat pada waktu pemrosesan dan FPS. Pada pendektsian dengan CPU, sistem berbasis SSD membutuhkan rata-rata waktu pemrosesan sebesar 26.139 detik dan menghasilkan rata-rata FPS sebesar 8.355 fps. Pada pendektsian dengan GPU laptop, sistem berbasis SSD membutuhkan rata-rata waktu pemrosesan sebesar 8.569 detik dan menghasilkan rata-rata FPS sebesar 25.587 fps. Pada pendektsian dengan GPU desktop, sistem berbasis SSD membutuhkan rata-rata waktu pemrosesan sebesar 6.541 detik dan menghasilkan rata-rata FPS sebesar 33.908 fps. Sedangkan pada nilai mAP hampir tidak terdapat perbedaan sama sekali pada setiap perangkat keras. Pada pendektsian dengan CPU dan GPU laptop nilai mAP yang dihasilkan sama persis, sedangkan pada pendektsian dengan GPU desktop rata-rata nilai mAP yang dihasilkan hanya bertambah sebesar 0.07%.

Pada sistem berbasis YOLO, perbedaan utama dari hasil pendektsian dengan perangkat keras yang berbeda terdapat pada waktu pemrosesan dan FPS. Pada pendektsian dengan CPU, sistem berbasis YOLO membutuhkan rata-rata waktu pemrosesan sebesar 80.803 detik dan menghasilkan rata-rata FPS sebesar 2.704 fps. Pada pendektsian dengan GPU laptop, sistem berbasis YOLO membutuhkan rata-rata waktu pemrosesan sebesar 33.904 detik dan menghasilkan rata-rata FPS sebesar 6.44 fps. Pada pendektsian dengan GPU desktop, sistem berbasis YOLO membutuhkan rata-rata waktu pemrosesan sebesar 24.555 detik dan menghasilkan rata-rata FPS sebesar 8.898 fps. Sedangkan pada nilai mAP hampir tidak terdapat perbedaan sama sekali pada setiap perangkat keras. Pada pendektsian dengan GPU laptop dan GPU desktop nilai mAP yang dihasilkan sama persis, sedangkan pada pendektsian dengan CPU rata-rata nilai mAP yang dihasilkan bertambah sebesar 3.23%.

Dari perbandingan antara sistem berbasis SSD dan YOLO, sistem berbasis SSD dengan GPU desktop masih belum mampu mengungguli nilai mAP dari sistem berbasis YOLO dengan CPU. Sistem berbasis YOLO dengan GPU desktop juga berhasil mengungguli sistem berbasis SSD dengan CPU dalam hal waktu pemrosesan dan FPS, dimana sistem berbasis YOLO dengan GPU desktop membutuhkan rata-rata waktu pemrosesan sebesar 24.555 detik dan menghasilkan FPS sebesar 8.898 fps, sedangkan sistem berbasis

SSD dengan CPU membutuhkan rata-rata waktu pemrosesan sebesar 26.139 detik dan menghasilkan FPS sebesar 8.355 fps. Namun ketika sistem berbasis SSD melakukan deteksi dengan menggunakan GPU laptop dan desktop, sistem berbasis YOLO tidak mampu mengejar waktu pemrosesan dan FPS, bahkan tertinggal sampai tiga kali lipat dari sistem berbasis SSD.

Dari hasil pengujian di atas, dapat ditarik kesimpulan bahwa perbedaan perangkat keras yang digunakan untuk pendekripsi tidak terlalu memengaruhi nilai mAP yang dihasilkan, namun sangat memengaruhi waktu pemrosesan dan nilai FPS. Pendekripsi dengan menggunakan GPU menghasilkan waktu pemrosesan yang tiga kali lebih cepat dibandingkan pendekripsi dengan menggunakan CPU. Begitu pula dengan FPS yang dihasilkan oleh pendekripsi dengan GPU lebih besar empat kali lipat dibandingkan pendekripsi dengan menggunakan CPU. Sehingga, semakin bagus jenis GPU dan semakin besar VRAM yang dimilikinya, maka semakin cepat pula waktu pemrosesan yang dibutuhkan dan semakin baik pula FPS yang didapatkan. Perbandingan rata-rata nilai dari seluruh hasil pengujian berdasarkan perangkat keras yang digunakan disajikan dalam bentuk grafik pada Gambar 7, 8, dan 9 yang terdapat pada bagian lampiran.

4.6 Pengujian dengan Tambahan Pencahayaan Buatan

Pengujian dengan tambahan pencahayaan buatan dilakukan dengan melakukan pengambilan data yang di lingkungan yang telah ditambahkan pencahayaan buatan. Cahaya buatan yang digunakan dalam skenario ini didapatkan dari lampu LED untuk fotografi dengan ukuran lux sebesar 860 lumens dan daya sebesar 9 watt. Pengujian ini dilakukan dengan mengambil data sesuai dengan skenario pada Tabel 3.2 dan waktu pengambilan data dilakukan pada waktu malam hari. Ilustrasi penempatan lampu LED dapat dilihat pada Gambar 4.13 dan konfigurasi lampu LED dapat dilihat pada Gambar 4.14.

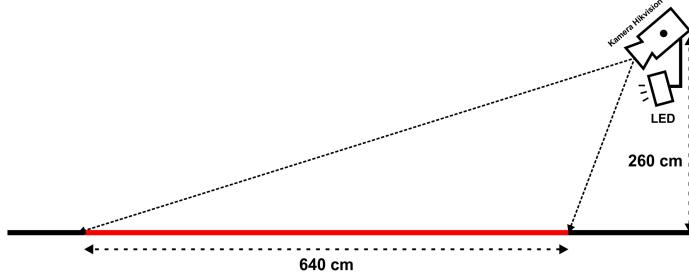
Dari data video yang telah diambil, didapatkan rata-rata nilai *brightness* dan intensitas *contrast* dari wajah yang akan dideteksi sebagaimana yang terdapat pada Gambar 4.5 dan 4.15 serta perbandingannya antara keduanya pada Tabel 4.13.

Tabel 4.11: Perbandingan Hasil Pendekripsi Sistem Berbasis SSD berdasarkan Perangkat Keras yang Digunakan

Nilai	SSD CPU			SSD GPU Laptop			SSD GPU Desktop		
	Waktu Pemrosesan (s)	FPS	mAP	Waktu Pemrosesan (s)	FPS	mAP	Waktu Pemrosesan (s)	FPS	mAP
Maksimum	36.723	8.435	75.39%	11.636	28.048	75.39%	11.741	35.822	75.39%
Minimum	22.706	8.122	14.84%	7.140	21.823	14.84%	5.545	17.971	14.84%
Rata-rata	26.139	8.355	43.90%	8.569	25.587	43.90%	6.541	33.908	43.97%
Total	627.330	-	-	205.658	-	-	156.981	-	-

Tabel 4.12: Perbandingan Hasil Pendekripsi Sistem Berbasis YOLO berdasarkan Perangkat Keras yang Digunakan

Nilai	YOLO CPU			YOLO GPU Laptop			YOLO GPU Desktop		
	Waktu Pemrosesan (s)	FPS	mAP	Waktu Pemrosesan (s)	FPS	mAP	Waktu Pemrosesan (s)	FPS	mAP
Maksimum	114.281	2.837	93.94%	46.742	6.604	93.22%	34.000	9.003	93.22%
Minimum	70.739	2.633	6.35%	29.536	6.264	7.94%	21.196	7.718	7.94%
Rata-rata	80.803	2.704	54.76%	33.904	6.44	51.53%	24.555	8.898	51.53%
Total	1939.280	-	-	813.684	-	-	589.312	-	-



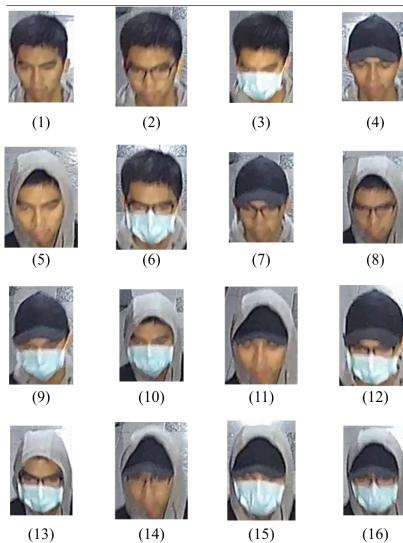
Gambar 4.13: Ilustrasi Peletakan Kamera dan Lampu LED.



Gambar 4.14: Konfigurasi Lampu LED dan Kamera dengan Ketinggian Sejajar dengan Wajah.

Tabel 4.13: Perbandingan *Brightness* dan *Contrast* dengan Tambahan Pencahaayaan Buatan

No.	Skenario	Tanpa Tambahan Pencahaayaan Buatan		Dengan Tambahan Pencahaayaan Buatan	
		<i>Brightness</i>	<i>Contrast</i>	<i>Brightness</i>	<i>Contrast</i>
1	Normal	94.547	0.351	121.582	0.187
2	Kacamata	87.970	0.405	110.846	0.066
3	Masker	92.757	0.330	125.537	0.024
4	Topi	97.307	0.612	120.752	0.193
5	Hoodie	96.795	0.265	141.831	0.046
6	Kacamata, Masker	96.403	0.180	133.682	0.162
7	Kacamata, Topi	97.530	0.331	107.421	0.045
8	Kacamata, Hoodie	96.740	0.281	131.737	0.119
9	Masker, Topi	105.109	0.291	123.490	0.075
10	Masker, Hoodie	116.840	0.244	143.085	0.170
11	Topi, Hoodie	103.954	0.241	119.928	0.188
12	Kacamata, Masker, Topi	104.482	0.318	131.193	0.126
13	Kacamata, Masker, Hoodie	102.167	0.182	155.016	0.065
14	Kacamata, Topi, Hoodie	94.708	0.311	119.956	0.074
15	Masker, Topi, Hoodie	100.006	0.314	137.314	0.178
16	Kacamata, Masker, Topi, Hoodie	111.182	0.276	135.796	0.133
Rata-rata		99.906	0.308	123.698	0.116



Gambar 4.15: Potongan Gambar Wajah dari Data Video Kamera dengan Tambahan Pencahayaan Buatan.

Dari Tabel 4.13, video yang didapatkan dari kamera dengan tambahan pencahayaan buatan memiliki rata-rata *brightness* dari wajah bernilai 128.698 dan intensitas *contrast*-nya bernilai 0.116. Pada hasil pendekripsi sistem berbasis SSD, sebagaimana yang ditunjukkan oleh Tabel 4.14, terdapat perbedaan besar antara hasil pendekripsi sistem yang diberikan tambahan lampu LED dengan sistem yang tidak diberikan tambahan lampu LED. Pada waktu pemrosesan dan FPS yang dihasilkan, nilainya rata-ratanya tidak jauh berbeda. Namun pada nilai mAP, sistem berbasis SSD dengan tambahan lampu LED mampu memberikan rata-rata nilai mAP sebesar 33.35% yang lebih besar hampir tiga kali lipat jika dibandingkan dengan rata-rata nilai mAP dari sistem berbasis SSD tanpa tambahan lampu LED yang hanya sebesar 13.99%. Sistem dengan tambahan lampu LED juga mampu melakukan deteksi pada seluruh skenario objek penghalang pada wajah dimana sistem tanpa tambahan lampu LED mengalami kegagalan pendekripsi di beberapa

skenario yang menggunakan tiga objek penghalang pada wajah. Hal ini memberikan gambaran awal bahwa tambahan pencahayaan buatan dapat memberikan perbedaan yang sangat signifikan terhadap hasil pendektsian.

Namun, pada hasil pendektsian sistem berbasis YOLO dengan tambahan lampu LED memberikan hasil yang sebaliknya. Seperti yang dapat dilihat pada Tabel 4.15, nilai mAP mengalami penurunan dari yang awalnya pada sistem tanpa tambahan lampu LED mampu memberikan nilai mAP sebesar 74.14% menjadi hanya sebesar 67.2% pada sistem dengan tambahan lampu. Salah satu hal yang dapat memengaruhi terjadinya penurunan nilai mAP ini adalah karena pengujian antara dua kondisi dilakukan pada selang waktu yang relatif lama, sehingga ada kemungkinan terjadinya perubahan, baik dari lingkungan tempat pengambilan data dilakukan, seperti terdapat tambahan objek pada *background*, atau dari objek yang diambil datanya, seperti perbedaan cara dan kecepatan berjalan, maupun cara pemasangan objek penghalang.

Dari seluruh pengujian ini dapat ditarik kesimpulan bahwa tambahan pencahayaan buatan dapat memberikan perbedaan yang sangat signifikan terhadap hasil pendektsian. Namun, terkadang pencahayaan bantuan juga belum tentu dapat meningkatkan performa dari hasil pendektsian. Terdapat faktor-faktor eksternal lain yang dapat memengaruhi performa sistem, seperti cara bergerak objek dan kondisi *background*. Perbandingan rata-rata nilai mAP dari seluruh hasil pengujian dengan tambahan pencahayaan buatan disajikan dalam bentuk grafik pada Gambar 10 yang terdapat pada bagian lampiran.

4.7 Pengujian dengan Ketinggian Kamera Sejajar dengan Wajah

Pengujian ini dilakukan dengan melakukan pengambilan data menggunakan kamera yang diletakkan pada ketinggian yang sejajar dengan tinggi dari wajah objek yang akan diambil datanya. Pengujian ini dilakukan dengan mengambil data sesuai dengan skenario pada Tabel 3.2 dan waktu pengambilan data dilakukan pada waktu siang dan malam hari. Pengujian ini juga dilakukan dengan menggunakan tambahan pencahayaan buatan. Ilustrasi penempatan kamera dapat dilihat pada Gambar 4.16 dan konfigurasi kamera dapat

Tabel 4.14: Perbandingan Hasil Pendeteksian Sistem Berbasis SSD dengan Tambahan Pencahayaan Buatan

Nilai	SSD Tanpa Lampu LED			SSD dengan Lampu LED		
	Waktu Pemrosesan (s)	FPS	mAP	Waktu Pemrosesan (s)	FPS	mAP
Maksimum	8.547	28.118	43.79%	14.013	28.132	61.44%
Minimum	6.834	24.824	0.00%	7.233	13.701	1.25%
Rata-rata	7.508	26.869	13.99%	8.260	25.922	33.35%
Total	180.202	-	-	198.234	-	-

Tabel 4.15: Perbandingan Hasil Pendeteksian Sistem Berbasis YOLO dengan Tambahan Pencahayaan Buatan

Nilai	YOLO Tanpa Lampu LED			YOLO dengan Lampu LED		
	Waktu Pemrosesan (s)	FPS	mAP	Waktu Pemrosesan (s)	FPS	mAP
Maksimum	34.183	6.581	99.89%	38.669	7.178	86.49%
Minimum	28.264	6.290	58.97%	27.641	4.965	42.11%
Rata-rata	30.932	6.518	74.14%	30.602	6.899	67.20%
Total	742.371	-	-	734.454	-	-



Gambar 4.16: Ilustrasi Peletakan Kamera dan Lampu LED.

dilihat pada Gambar 4.14.

Data hasil pengujian yang didapatkan kemudian dibandingkan dengan hasil pendektsian pada posisi kamera yang berada di atas objek dari pengujian sebelumnya, sehingga didapatkan perbandingan antara hasil pendektsian dengan kamera yang berada di atas objek dan kamera yang sejajar dengan wajah objek. Pada hasil pengujian dengan sistem berbasis SSD pada waktu siang hari, sebagaimana yang terdapat pada Tabel 4.16, dapat dilihat bahwa seluruh aspek mulai dari waktu pemrosesan, FPS, dan nilai mAP cenderung stabil dan tidak terdapat banyak perbedaan. Waktu pemrosesan dan FPS pada hasil pendektsian dengan kamera yang berada sejajar dengan wajah objek memiliki hasil yang lebih baik, namun nilai mAP yang dihasilkan lebih kecil 2% daripada hasil pendektsian dengan kamera yang berada di atas objek. Sedangkan pada hasil pengujian sistem berbasis SSD pada waktu malam hari, sebagaimana yang terdapat pada Tabel 4.17, nilai mAP dari hasil pendektsian dengan kamera yang berada sejajar dengan wajah objek mengalami peningkatan yang signifikan, dimana nilai mAP-nya unggul $\pm 23\%$ jika dibandingkan dengan hasil pendektsian dengan kamera yang berada di atas objek.

Pada hasil pengujian dengan sistem berbasis YOLO pada waktu siang hari, sebagaimana yang terdapat pada Tabel 4.18, nilai mAP dari hasil pendektsian dengan kamera yang berada sejajar dengan wajah objek menghasilkan nilai sebesar 69.11% yang jauh lebih baik jika dibandingkan dengan hasil pendektsian dengan ka-

mera yang berada di atas objek yang hanya menghasilkan nilai sebesar 56.82%. Sedangkan pada hasil pengujian dengan sistem berbasis YOLO pada waktu malam hari, sebagaimana yang terdapat pada Tabel 4.19, juga terjadi peningkatan nilai mAP sebesar 6.4% pada hasil pendekripsi yang menggunakan kamera yang berada sejajar dengan wajah objek.

Dari keempat pengujian di atas, dapat ditarik kesimpulan bahwa mengubah letak kamera menjadi sejajar dengan wajah objek dapat meningkatkan performa dari sistem, khususnya pada nilai mAP. Selain itu, dengan menggunakan kamera menjadi sejajar dengan wajah objek, sistem juga dapat melakukan pendekripsi pada skenario dengan empat objek penghalang wajah lebih baik daripada ketika menggunakan kamera yang berada di atas objek. Perbandingan rata-rata nilai mAP dari seluruh hasil pengujian dengan ketinggian kamera yang berbeda disajikan dalam bentuk grafik pada Gambar 11 yang terdapat pada bagian lampiran.

4.8 Pengujian Berdasarkan Sudut Kamera

Pengujian berdasarkan perbedaan sudut kamera bertujuan untuk mengetahui pengaruh perubahan sudut kamera terhadap kemampuan sistem dalam mendekripsi wajah. Pada pengujian ini hanya digunakan satu kamera yang membelakangi sinar matahari dan waktu pengambilan data dilakukan pada waktu siang hari. Berdasarkan Gambar 3.3, dilakukan eksperimen terhadap sudut kamera. Terdapat lima skenario kondisi sudut kamera yang masing-masing diuji sebanyak dua kali, antara lain:

1. Sudut kamera 30° .
2. Sudut kamera 40° .
3. Sudut kamera 50° .
4. Sudut kamera 60° .
5. Sudut kamera 70° .

Dari hasil yang didapatkan pada Tabel 4.20, sistem pendekripsi wajah berbasis berbasis SSD mampu memberikan hasil pendekripsi terbaik pada sudut 60° dengan nilai mAP tertinggi sebesar 74.48%. Namun, sistem berbasis SSD juga tidak mampu melakukan pendekripsi sama sekali ketika kamera berada pada sudut 30° . Sedangkan sistem pendekripsi wajah berbasis YOLO mampu mem-

Tabel 4.16: Perbandingan Hasil Pendeketeksian Sistem Berbasis SSD dengan Ketinggian Kamera Sejajar dengan Wajah pada Waktu Siang Hari

Nilai	SSD Kamera Atas			SSD Kamera Bawah		
	Waktu Pemrosesan (s)	FPS	mAP	Waktu Pemrosesan (s)	FPS	mAP
Maksimum	9.962	28.145	72.81%	9.225	28.850	74.72%
Minimum	6.751	21.079	1.30%	6.630	21.680	2.35%
Rata-rata	7.878	25.438	43.69%	7.566	26.445	41.88%
Total	189.068	-	-	181.590	-	-

Tabel 4.17: Perbandingan Hasil Pendeketeksian Sistem Berbasis SSD dengan Ketinggian Kamera Sejajar dengan Wajah pada Waktu Malam Hari

Nilai	SSD Kamera Atas			SSD Kamera Bawah		
	Waktu Pemrosesan (s)	FPS	mAP	Waktu Pemrosesan (s)	FPS	mAP
Maksimum	14.013	28.132	61.44%	10.670	28.369	92.30%
Minimum	7.233	13.701	1.25%	7.224	21.088	13.37%
Rata-rata	8.260	25.922	33.35%	7.995	26.423	56.73%
Total	198.234	-	-	191.884	-	-

Tabel 4.18: Perbandingan Hasil Pendeteksian Sistem Berbasis YOLO dengan Ketinggian Kamera Sejajar dengan Wajah pada Waktu Siang Hari

Nilai	YOLO Kamera Atas			YOLO Kamera Bawah			
	Waktu	Pemrosesan (s)	FPS	mAP	Pemrosesan (s)	FPS	mAP
Maksimum	29.139	7.274	94.87%		29.812	7.145	92.13%
Minimum	24.843	6.865	1.33%		25.191	6.711	22.32%
Rata-rata	27.656	7.208	56.82%		28.267	7.053	69.11%
Total	663.750	-	-		678.404	-	

Tabel 4.19: Perbandingan Hasil Pendeteksian Sistem Berbasis YOLO dengan Ketinggian Kamera Sejajar dengan Wajah pada Waktu Malam Hari

Nilai	YOLO Kamera Atas			YOLO Kamera Bawah			
	Waktu	Pemrosesan (s)	FPS	mAP	Waktu	Pemrosesan (s)	FPS
Maksimum	38.669	7.178	86.49%		32.782	7.142	91.75%
Minimum	27.641	4.965	42.11%		27.302	6.609	37.91%
Rata-rata	30.602	6.899	67.20%		29.913	7.034	73.60%
Total	734.454	-	-		717.904	-	

berikan hasil pendektsian terbaik pada sudut 40° dengan nilai mAP tertinggi sebesar 84.94%. Dan pada sudut 60° , sistem berbasis YOLO juga menghasilkan nilai mAP yang tidak terpaut jauh dibandingkan dengan nilai mAP pada sudut 40° , yaitu sebesar 80.82% untuk nilai mAP tertingginya.

Dari hasil pengujian tersebut, dapat disimpulkan bahwa sudut kamera terbaik adalah sudut 60° . Hal ini terjadi karena ketika kamera berada pada sudut 60° , jangkauan panjang frame yang didapatkan lebih panjang dibandingkan dengan ketika kamera berada pada sudut 30° sampai 40° . Dengan semakin panjang jangkauan *frame*, maka semakin banyak pula objek wajah yang bisa terdeteksi, sehingga dapat menghasilkan nilai mAP yang lebih baik. Pengecualian pada sudut 70° , nilai mAP yang didapatkan mengalami penurunan yaitu dengan nilai mAP tertinggi hanya sebesar 31.12% pada sistem berbasis SSD dan 26.77% pada sistem berbasis YOLO. Hal ini disebabkan karena jangkauan *frame* terlalu jauh, sehingga objek wajah hanya terlihat seperti piksel kecil pada video. Karena objek wajah terlalu kecil, maka sistem pendektsi wajah tidak dapat mendektsi objek tersebut. Perbandingan nilai mAP dari seluruh hasil pengujian berdasarkan sudut kamera disajikan dalam bentuk grafik pada Gambar 12 yang terdapat pada bagian lampiran.

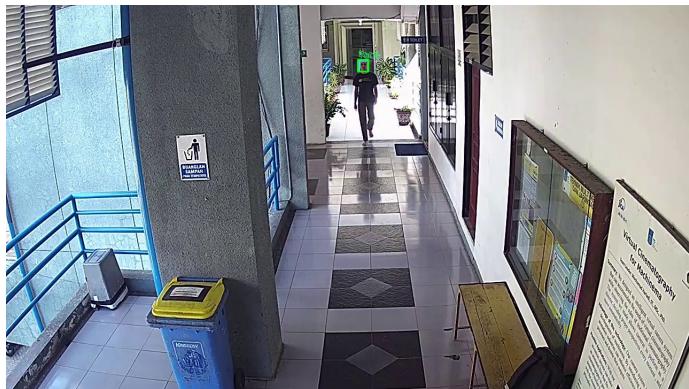
Selain itu, berdasarkan dari pengujian ini juga dapat disimpulkan jarak terjauh pendektsian objek yang dapat dilakukan. Pada sistem pendektsi wajah berbasis SSD, jarak terjauh atau jarak minimum dapat dilakukannya pendektsian adalah pada jarak 10.8 meter, seperti yang ditunjukkan oleh Gambar 4.17, dan jarak terdekat atau jarak maksimumnya adalah pada jarak 1.6 meter, seperti yang ditunjukkan oleh Gambar 4.18. Sedangkan pada sistem pendektsi wajah berbasis YOLO, jarak terjauh atau jarak minimum dapat dilakukannya pendektsian adalah pada jarak 8 meter, seperti yang ditunjukkan oleh Gambar 4.19, dan jarak terdekat atau jarak maksimumnya adalah pada jarak 1.2 meter, seperti yang ditunjukkan oleh Gambar 4.20.

4.9 Rekomendasi Penerapan Sistem

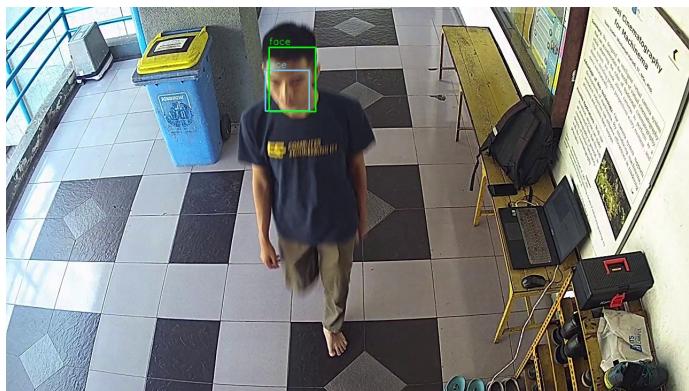
Dari seluruh hasil pengujian yang telah dilakukan, terdapat beberapa rekomendasi yang dapat diterapkan agar sistem pendektsi wajah berbasis CNN dapat memberikan performa pendektsian

Tabel 4.20: Hasil Pengujian Berdasarkan Sudut Kamera

Sudut ($^{\circ}$)	SSD				YOLO		
	Waktu Pemrosesan (s)	FPS	mAP	Pemrosesan (s)	FPS	mAP	
30	13.520	9.541	0.00%	22.867	5.641	17.17%	
	5.317	25.955	0.00%	21.080	6.546	10.04%	
40	5.414	24.936	17.63%	20.522	6.578	66.01%	
	6.045	25.642	17.95%	23.396	6.625	54.94%	
50	8.144	26.276	36.84%	32.553	6.574	84.21%	
	8.363	25.59	51.64%	32.434	6.598	84.94%	
60	17.148	26.709	74.48%	69.776	6.564	80.82%	
	15.541	27.026	68.61%	63.718	6.592	80.23%	
70	19.258	26.794	24.85%	78.139	6.604	26.77%	
	23.992	26.843	31.12%	97.335	6.616	25.94%	



Gambar 4.17: Jarak Terjauh atau Minimum Pendeksi Wajah dengan Sistem Berbasis SSD.



Gambar 4.18: Jarak Terdekat atau Maksimum Pendeksi Wajah dengan Sistem Berbasis SSD.



Gambar 4.19: Jarak Terjauh atau Minimum Pendeksi Wajah dengan Sistem Berbasis YOLO.



Gambar 4.20: Jarak Terdekat atau Maksimum Pendeksi Wajah dengan Sistem Berbasis YOLO.

yang baik, antara lain:

1. Sistem berbasis SSD lebih cocok jika digunakan untuk melakukan pendektsian dengan cepat terhadap objek yang bergerak. Sedangkan sistem berbasis YOLO lebih cocok untuk melakukan pendektsian secara akurat.
2. Lokasi peletakan kamera diletakkan pada posisi di depan atas wajah objek yang akan dideteksi dengan sudut kamera dari dinding sebesar 60° .
3. Kondisi pencahayaan yang digunakan untuk melakukan pendektsian adalah pada waktu siang hari dengan arah kamera membelakangi sinar matahari.
4. Pendektsian tetap dapat dilakukan pada waktu malam hari apabila kondisi lingkungan tempat dilakukannya pendektsian diberikan pencahayaan tambahan dengan menggunakan lampu.
5. Sistem pendektsi wajah berbasis SSD selalu kesulitan dalam mendektsi wajah dengan objek penghalang masker, topi, *hoodie*, dan kacamata hitam. Sedangkan sistem berbasis YOLO yang mampu melakukan pendektsian dengan baik pada wajah meskipun terdapat beberapa kombinasi objek penghalang. Oleh karena itu, sistem pendektsi wajah berbasis SSD lebih baik jika diterapkan di tempat yang terdapat aturan untuk melepaskan masker, topi, jaket, dan kacamata hitam.
6. Apabila sistem ingin digunakan untuk melakukan pendektsian secara cepat dan *real-time*, maka perlu digunakan GPU yang bagus dengan ukuran VRAM yang besar untuk mempercepat waktu pemrosesan dan FPS yang dihasilkan.

BAB 5

PENUTUP

5.1 Kesimpulan

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

1. Sistem pendekripsi wajah berbasis SSD mampu memberikan waktu pemrosesan yang lebih cepat dan nilai FPS yang lebih baik hingga empat kali lipat dibandingkan sistem pendekripsi wajah berbasis YOLO. Sedangkan sistem berbasis YOLO mampu memberikan nilai mAP lebih tinggi hingga dua kali lipat daripada sistem berbasis SSD.
2. Berdasarkan hasil pengujian, posisi peletakan kamera berpengaruh terhadap performa sistem pendekripsi wajah. Perbandingan nilai mAP dari pengujian menunjukkan bahwa video dari kamera yang membelakangi sinar matahari dengan rata-rata *brightness* sebesar 85.695 dan rata-rata intensitas *contrast* sebesar 0.435 menghasilkan nilai mAP rata-rata tertinggi sebesar 55.23% yang lebih baik dibandingkan video dari kamera yang menghadap ke arah sinar matahari dengan rata-rata *brightness* sebesar 59.681 dan rata-rata intensitas *contrast* sebesar 0.257 yang menghasilkan nilai mAP rata-rata tertinggi hanya sebesar 42.81%. Hal ini terjadi karena pada kamera yang menghadap sinar matahari terjadi *highlight* pada video yang menyebabkan video terlihat lebih gelap.
3. Perbedaan ketinggian peletakan kamera juga berpengaruh terhadap nilai mAP yang didapatkan, dimana ketika menggunakan kamera yang diletakkan sejajar dengan wajah objek, nilai mAP yang didapatkan lebih tinggi, dengan nilai mAP rata-rata tertinggi sebesar 73.60%, jika dibandingkan dengan nilai ketika kamera diletakkan di atas objek, dengan nilai maksimal hanya sebesar 56.82%.
4. Kedua sistem pendekripsi wajah tidak dapat melakukan pendeteksian terhadap wajah dengan kombinasi objek penghalang yang terdapat pada wajah berjumlah lebih dari tiga buah, khususnya wajah dengan objek penghalang masker dan topi.

5. Waktu pengambilan data video berpengaruh terhadap nilai mAP. Ketika *input* video yang diberikan adalah video pada waktu malam hari dengan rata-rata *brightness* sebesar 99.906 dan rata-rata intensitas *contrast* sebesar 0.308, sistem tidak dapat mendeteksi objek wajah yang terdapat pada video. Hal ini bisa diatasi dengan memberikan tambahan pencahayaan buatan pada lokasi pendekripsi yang dapat mengubah nilai rata-rata *brightness* menjadi sebesar 128.698 dan intensitas *contrast* menjadi sebesar 0.116, serta meningkatkan nilai mAP hingga $\pm 20\%$ lebih tinggi.
6. Berdasarkan hasil pengujian pada objek yang berjumlah lebih dari satu, kedua sistem pendekripsi wajah berbasis CNN mampu melakukan pendekripsi pada seluruh wajah dengan galat hanya sebesar 3,03%. Dan jumlah maksimal wajah yang dapat dideteksi oleh sistem adalah sebanyak 30 objek wajah.
7. Pada pengujian berdasarkan kecepatan objek bergerak, sistem berbasis YOLO mampu memberikan nilai mAP lebih baik dengan selisih hingga 27% dari sistem berbasis SSD. Namun sistem berbasis YOLO masih belum mampu untuk melakukan pendekripsi secara *real-time* karena memiliki FPS yang rendah, yaitu sebesar 7 fps. Hal ini terjadi karena bahasa pemrograman yang digunakan untuk membangun sistem adalah Phyton yang memuat banyak fitur di dalamnya, sehingga menyebabkan eksekusi program menjadi lebih lambat.
8. Berdasarkan hasil pengujian berdasarkan sudut kamera, kedua sistem pendekripsi wajah berbasis CNN mampu melakukan pendekripsi dengan baik pada sudut 60° dengan nilai mAP rata-rata sebesar 71.55% untuk sistem berbasis SSD dan 80.53% pada sistem berbasis YOLO.
9. Nilai *brightness* dan intensitas *contrast* yang terdapat pada hasil pengujian belum dapat mewakili nilai yang sebenarnya karena konfigurasi *auto white balance* dari *IP camera* tidak dapat dinonaktifkan.

5.2 Saran

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

1. Melakukan pengujian berdasarkan ekspresi wajah yang berbeda.
2. Melakukan pengujian pada lokasi yang telah dikondisikan dan benar-benar steril dari objek-objek lain yang mengganggu proses pendekripsi.
3. Membuat *user interface* untuk mempermudah pengoperasian sistem.
4. Melanjutkan penelitian ke tahap pengenalan wajah dari objek yang dideteksi.

Halaman ini sengaja dikosongkan

DAFTAR PUSTAKA

- [1] M. Zufar and B. Setiyono, “Convolutional neural networks untuk pengenalan wajah secara real-time,” Jurnal Sains dan Seni ITS, vol. 5, no. 2, 2016. (Dikutip pada halaman 1).
- [2] 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 2, 17, 18, 19).
- [3] J. Redmon and A. Farhadi, “Yolov3: An incremental improvement,” arXiv preprint arXiv:1804.02767, 2018. (Dikutip pada halaman 2, 39, 40).
- [4] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, “Ssd: Single shot multibox detector,” in European conference on computer vision, pp. 21–37, Springer, 2016. (Dikutip pada halaman 2, 20, 21).
- [5] A. Yicheng, J. Wu, and C. Yue, “Cnns for face detection and recognition.” <http://cs231n.stanford.edu/reports/2017/pdfs/222.pdf>. Diakses: 2018-12-16. (Dikutip pada halaman 2).
- [6] H. Li, Z. Lin, X. Shen, J. Brandt, and G. Hua, “A convolutional neural network cascade for face detection,” in Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 5325–5334, 2015. (Dikutip pada halaman 2).
- [7] D. Sharma and N. Kumar, “A review on machine learning algorithms, tasks and applications,” International Journal of Advanced Research in Computer Engineering & Technology (IJARCET), vol. 6, no. 10, 2017. (Dikutip pada halaman 5).
- [8] A. Ahmad, “Mengenal artificial intelligence, machine learning, neural network, dan deep learning,” no. October, 2017. (Dikutip pada halaman 5, 6, 8).

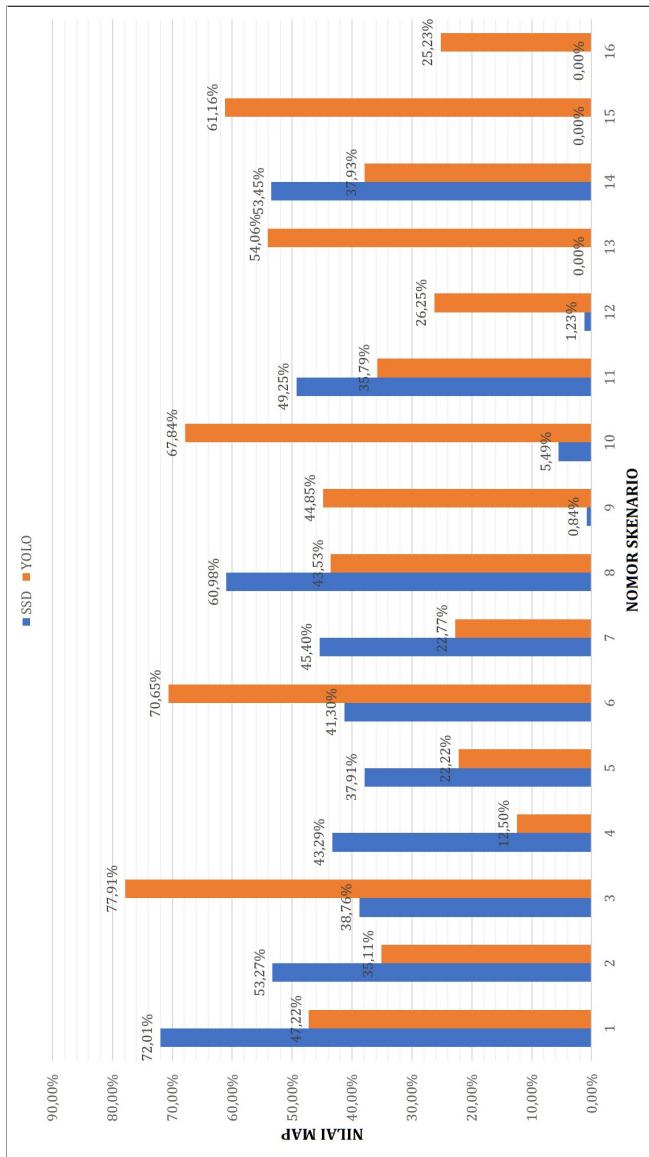
- [9] J. DeMuro, “What is a neural network?.” <https://www.techradar.com/news/what-is-a-neural-network>, 2018. Diakses: 2019-04-06. (Dikutip pada halaman 5).
- [10] S. S. Haykin, S. S. Haykin, S. S. Haykin, K. Elektroingenieur, and S. S. Haykin, Neural networks and learning machines, vol. 3. Pearson education Upper Saddle River, 2009. (Dikutip pada halaman 6, 7).
- [11] Y. Bengio et al., “Learning deep architectures for ai,” Foundations and trends® in Machine Learning, vol. 2, no. 1, pp. 1-127, 2009. (Dikutip pada halaman 8).
- [12] I. Goodfellow, Y. Bengio, and A. Courville, Deep learning. MIT press, 2016. (Dikutip pada halaman 8).
- [13] S. R. Dewi et al., “Deep learning object detection pada video menggunakan tensorflow dan convolutional neural network,” 2018. (Dikutip pada halaman 9).
- [14] MacKenty, “Max-pooling / pooling.” https://computersciencewiki.org/index.php/Max-pooling_-_Pooling, 2018. Diakses: 2019-05-01. (Dikutip pada halaman 11).
- [15] S. Sena, “Pengenalan deep learning part 7 : Convolutional neural network (cnn).” <https://medium.com/@samuelsena/pengenalan-deep-learning-part-7/-convolutional-neural-network-cnn-b003b477dc94>, 2017. Diakses: 2019-05-03. (Dikutip pada halaman 12, 13).
- [16] K. Umam and B. S. Negara, “Deteksi obyek manusia pada basis data video menggunakan metode background subtraction dan operasi morfologi,” Jurnal CoreIT: Jurnal Hasil Penelitian Ilmu Komputer dan Teknologi Informasi, vol. 2, no. 2, pp. 31-40, 2016. (Dikutip pada halaman 14).
- [17] S. Das, “Cnn architectures: Lenet, alexnet, vgg, googlenet, resnet and more” <https://medium.com/@sidereal/cnns-architectures-lenet-alexnet-vgg-googlenet/>

-resnet-and-more-666091488df5, 2017. Diakses: 2019-05-05. (Dikutip pada halaman 15).

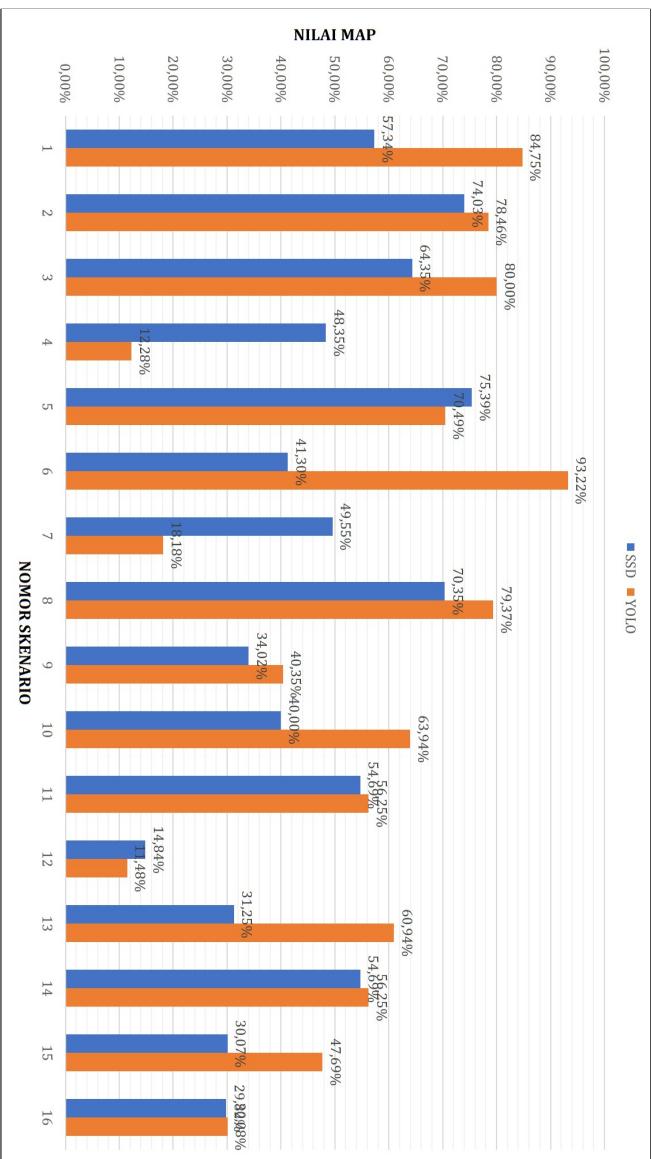
- [18] K. Sinhal and A. Sachan, “Zero to hero: Guide to object detection using deep learning: Faster r-cnn, yolo, ssd.” <https://cv-tricks.com/object-detection/faster-r-cnn-yolo-ssd/>, 2017. Diakses: 2019-05-05. (Dikutip pada halaman 16).
- [19] A. Pambudhi, “Monitoring dan analisis ip camera pada jaringan internet,” Jurnal Teknik Elektro Universitas Tanjungpura, vol. 1, no. 1. (Dikutip pada halaman 22).
- [20] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, “Mobilenets: Efficient convolutional neural networks for mobile vision applications,” arXiv preprint arXiv:1704.04861, 2017. (Dikutip pada halaman 41, 44).
- [21] A. Solichin, “Mengukur kinerja algoritma klasifikasi dengan confusion matrix.” <https://achmatim.net/2017/03/19/mengukur-kinerja-algoritma-klasifikasi-dengan-confusion-matrix/>, 2017. Diakses: 2019-04-04. (Dikutip pada halaman 47).
- [22] 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: 2019-04-17. (Dikutip pada halaman 48).

Halaman ini sengaja dikosongkan

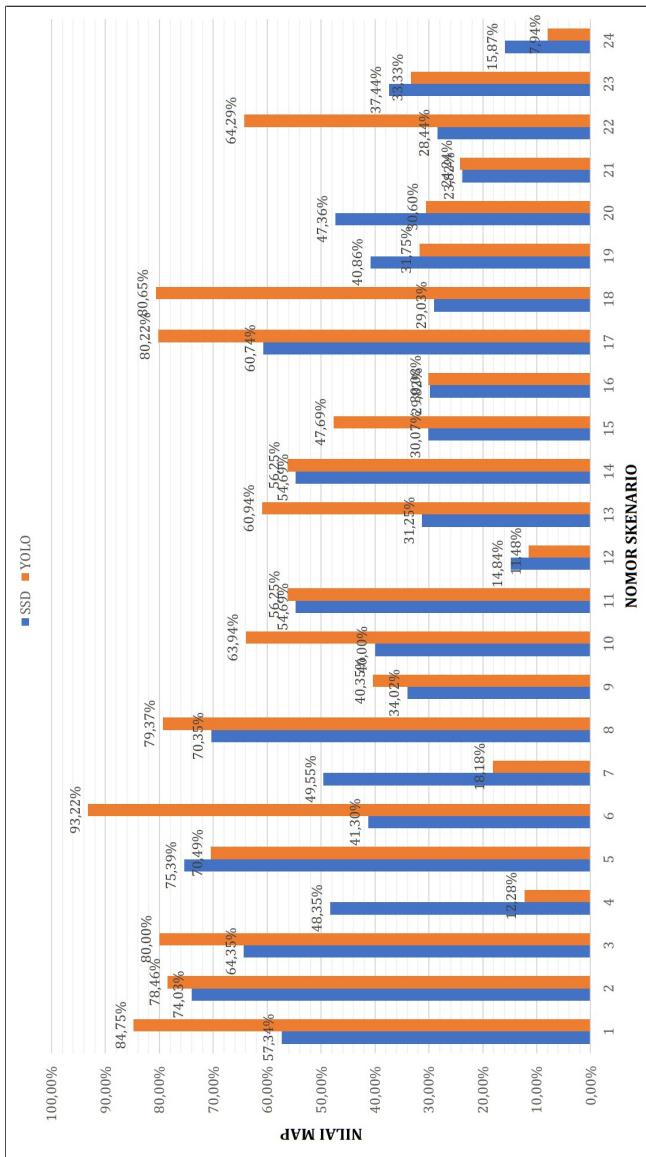
LAMPIRAN



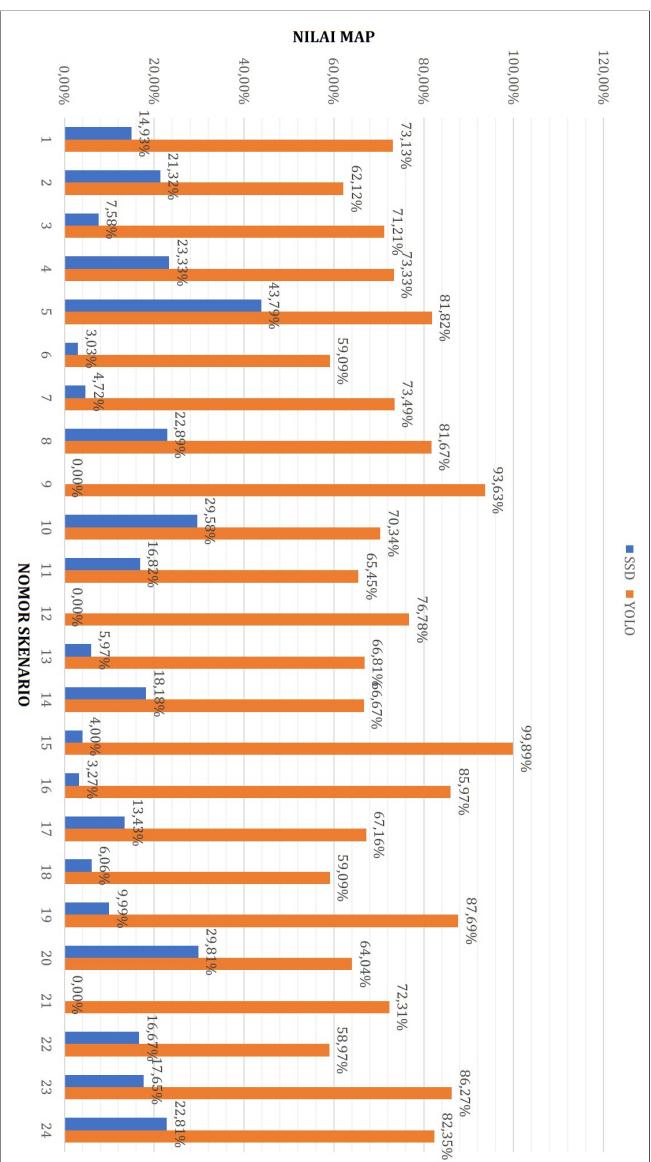
Gambar 1: Grafik perbandingan nilai mAP saat kamera menghadap ke arah timur.



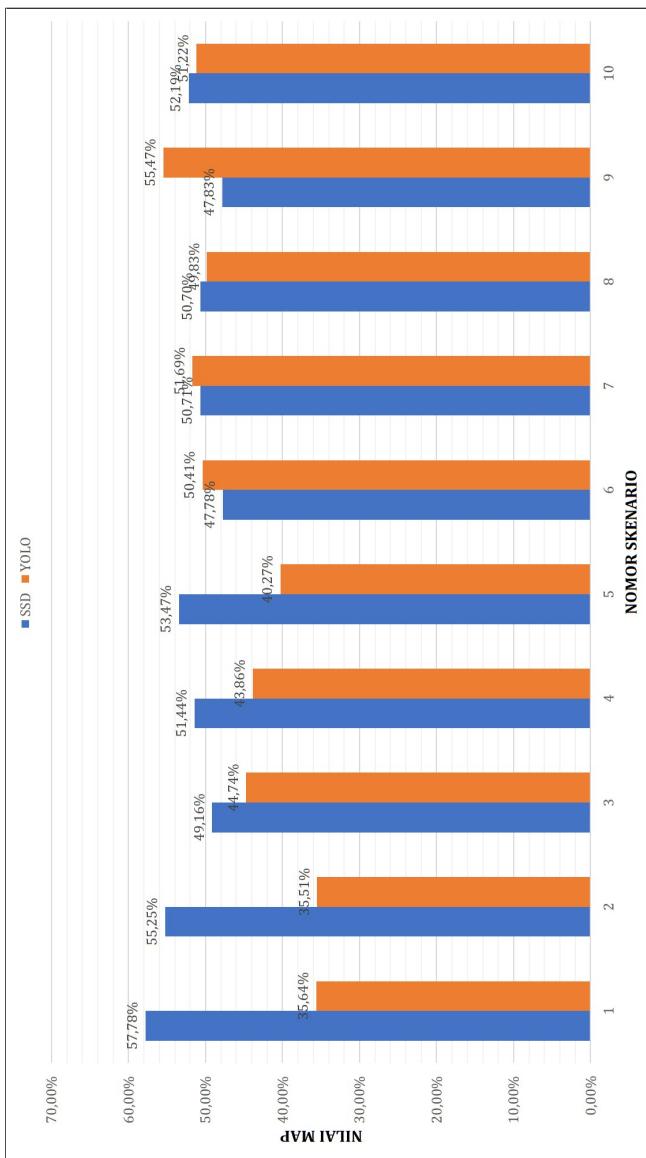
Gambar 2: Grafik perbandingan nilai mAP saat kamera menghadap ke arah barat.



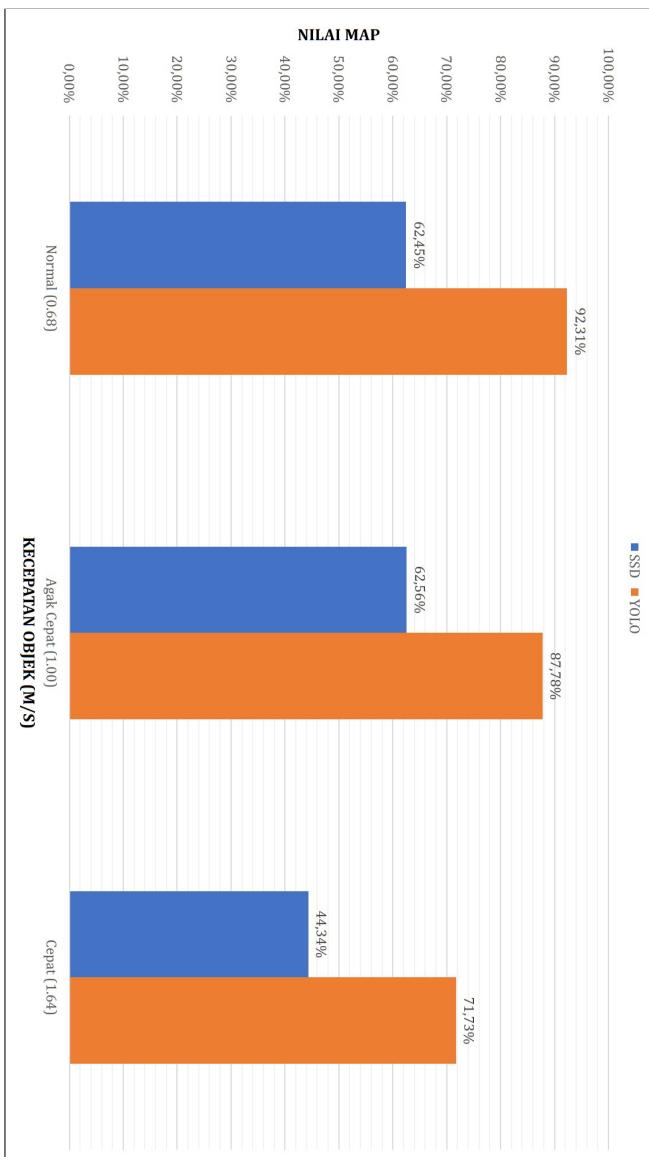
Gambar 3: Grafik perbandingan nilai mAP pada saat siang hari.



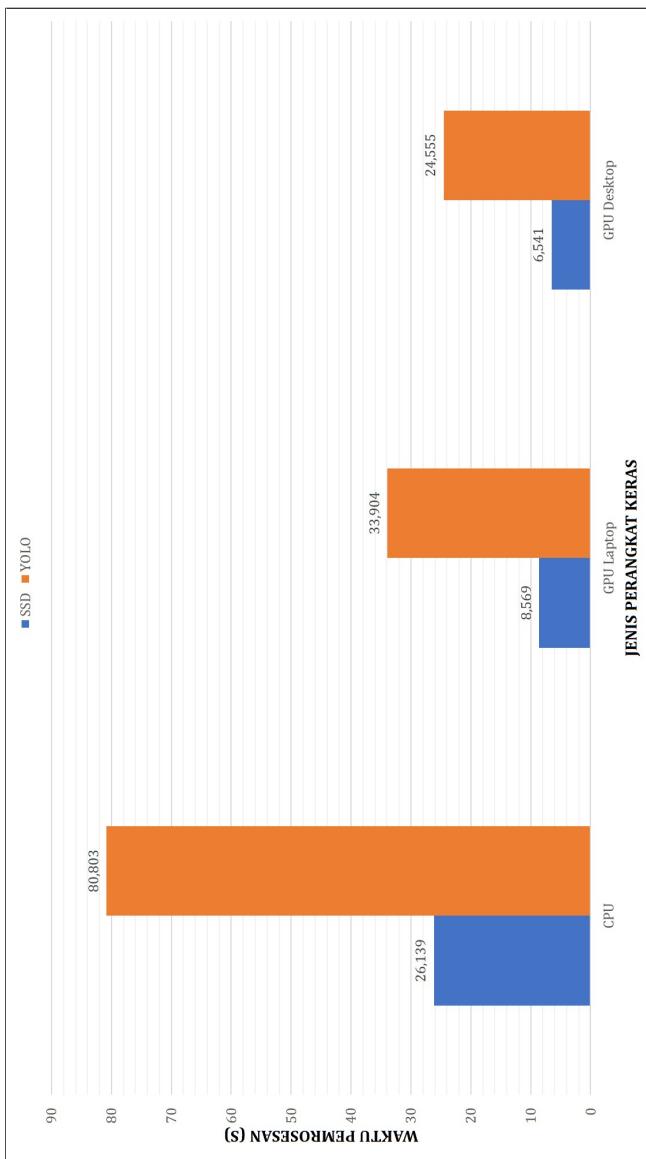
Gambar 4: Grafik perbandingan nilai mAP pada saat malam hari.



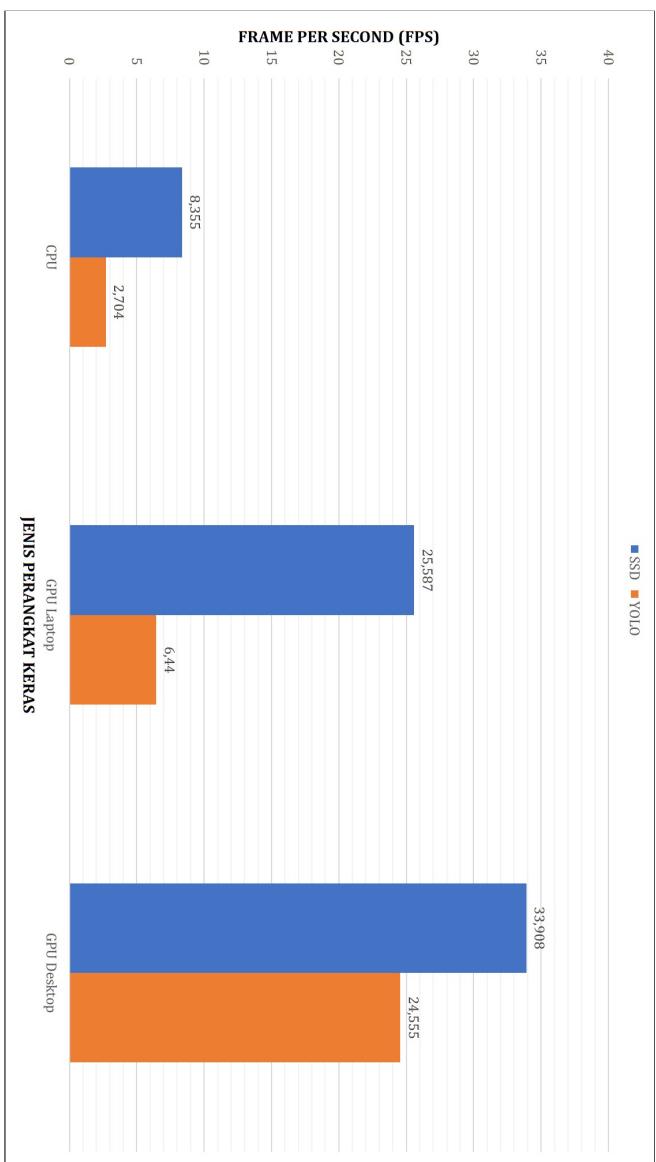
Gambar 5: Grafik perbandingan nilai mAP berdasarkan jumlah objek.



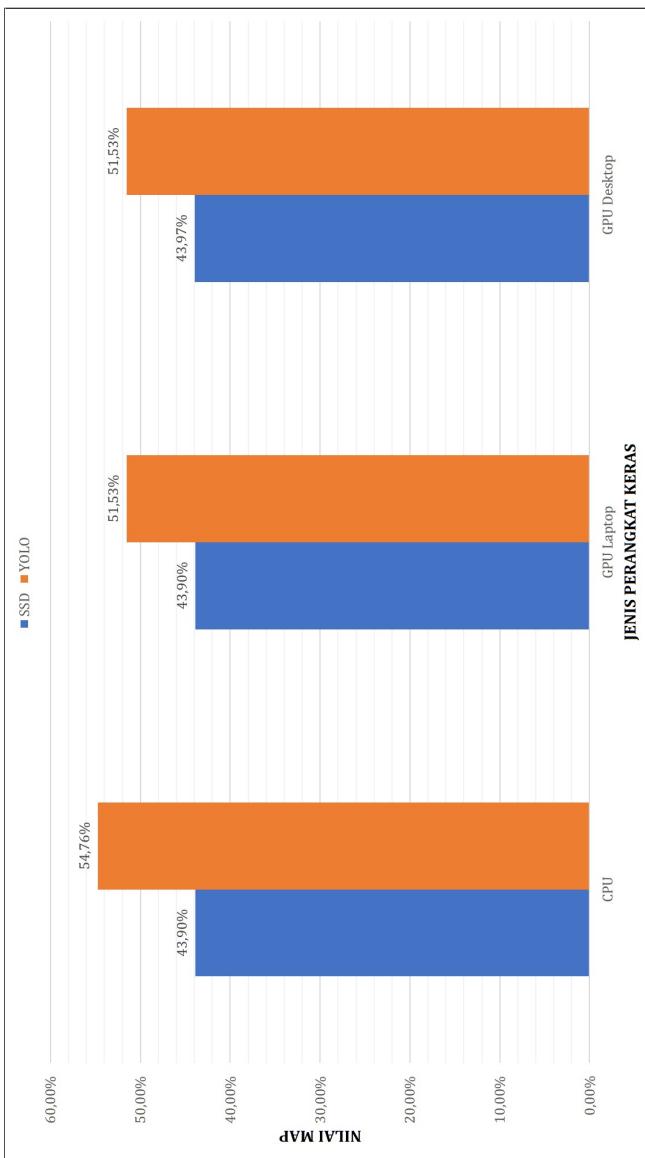
Gambar 6: Grafik perbandingan rata-rata nilai mAP berdasarkan kecepatan objek.



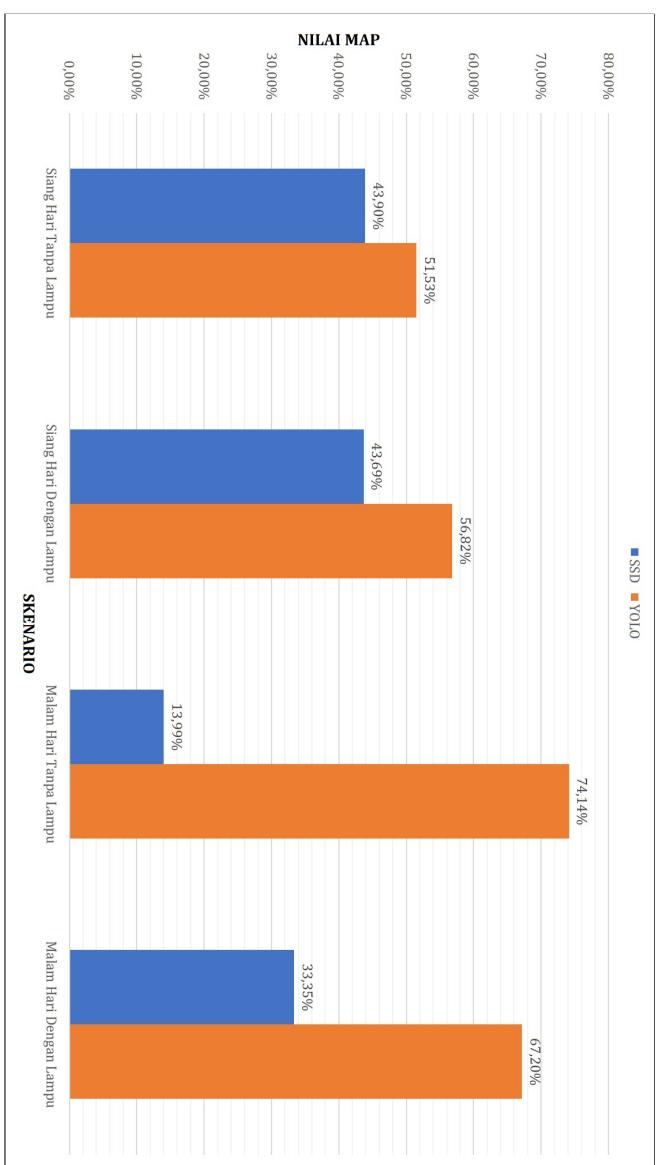
Gambar 7: Grafik perbandingan rata-rata waktu pemrosesan berdasarkan perangkat keras yang digunakan.



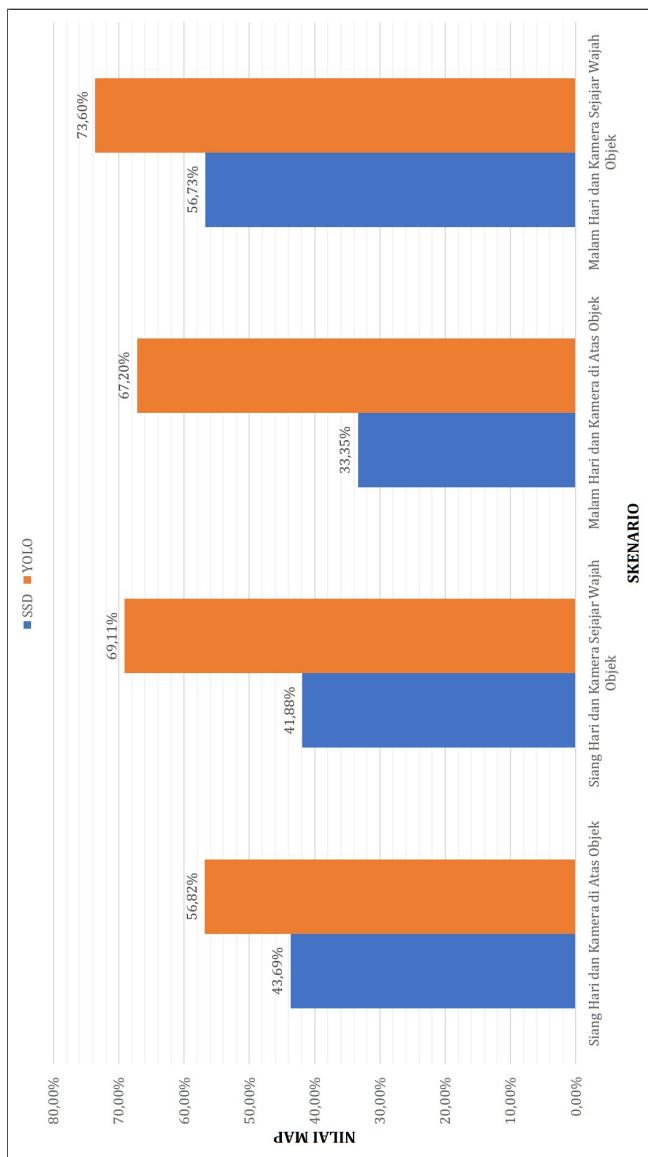
Gambar 8: Grafik perbandingan rata-rata nilai FPS berdasarkan perangkat keras yang digunakan.



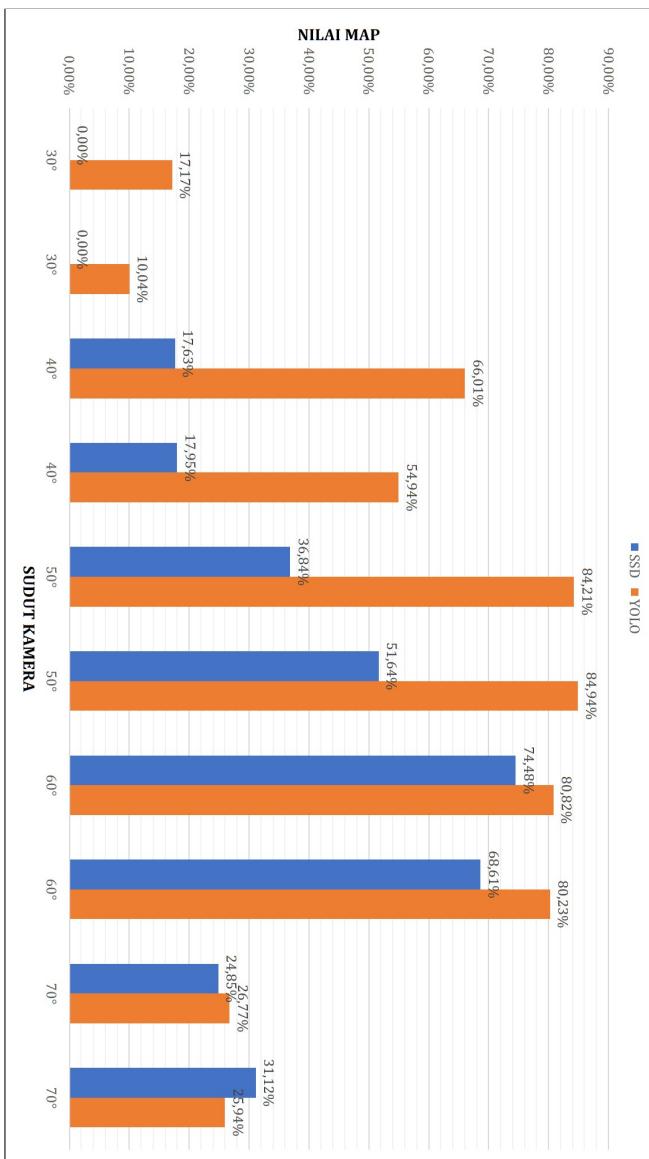
Gambar 9: Grafik perbandingan rata-rata nilai mAP berdasarkan perangkat keras yang digunakan.



Gambar 10: Grafik perbandingan rata-rata nilai mAP pada pengujian dengan tambahan pencahaayaan buatan.



Gambar 11: Grafik perbandingan rata-rata nilai mAP pada ketiggaan kamera yang berbeda.



Gambar 12: Grafik perbandingan nilai mAP Berdasarkan Sudut Kamera.

BIOGRAFI PENULIS



Rizky Najwa, lahir pada 12 Oktober 1997 di Kecamatan Kotawaringin Lama, Provinsi Kalimantan Tengah. Penulis merupakan anak pertama dari tiga bersaudara. Saat ini penulis tinggal di Kota Palangkaraya, Provinsi Kalimantan Tengah. Pada tahun 2009 menyelesaikan pendidikan di SD Negeri Mendawai 3 Sukamara. Tahun 2012 lulus dari SMP Negeri 1 Sukamara. Dan pada tahun 2015 lulus dari SMA Global Islamic Boarding School di Kalimantan Selatan. Penulis sedang menjalani studi di Departemen Teknik Komputer, Fakultas Teknologi Elektro, Institut Teknologi Sepuluh Nopember. Saat ini penulis aktif menjadi Asisten Laboratorium Komputasi Multimedia B401. Penulis memiliki hobi dalam bidang sepakbola dan *travelling*. Penulis berhasil menyelesaikan tugas akhir dengan judul "**Analisis Performa Sistem Pendekripsi Wajah Berbasis Convolutional Neural Network (CNN)**". Bagi pembaca yang memiliki kritik, saran, atau pertanyaan mengenai tugas akhir ini dapat menghubungi penulis melalui surel rizkynajwa333@gmail.com.

Halaman ini sengaja dikosongkan