



**TUGAS AKHIR - EC 184801**

**DETEKSI DAN IDENTIFIKASI NOMOR REGISTRASI PLAT  
KENDARAAN INDONESIA BERBASIS CONVOLUTIONAL  
NEURAL NETWORK (CNN)**

Muh. Shahid Setiawan  
NRP 0721154000044

Dosen Pembimbing  
Dr. Eko Mulyanto Yuniarno, ST., MT.  
Reza Fuad Rachmadi, ST., MT., Ph.D.

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





**TUGAS AKHIR - EC 184801**

**DETEKSI DAN IDENTIFIKASI NOMOR REGISTRASI PLAT  
KENDARAAN INDONESIA BERBASIS CONVOLUTIONAL  
NEURAL NETWORK (CNN)**

Muh. Shahid Setiawan  
NRP 0721154000044

Dosen Pembimbing  
Dr. Eko Mulyanto Yuniarno, ST., MT.  
Reza Fuad Rachmadi, ST., MT., Ph.D.

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





**FINAL PROJECT - EC 184801**

**DETECTION AND IDENTIFICATION INDONESIAN  
VEHICLE LICENSE PLATE NUMBER BASED ON  
CONVOLUTIONAL NEURAL NETWORK (CNN)**

Muh. Shahid Setiawan  
NRP 0721154000044

Advisor  
Dr. Eko Mulyanto Yuniarno, ST., MT.  
Reza Fuad Rachmadi, ST., MT., Ph.D.

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



## PERNYATAAN KEASLIAN TUGAS AKHIR

Dengan ini saya menyatakan bahwa isi sebagian maupun keseluruhan Tugas Akhir saya dengan judul “**Deteksi dan Identifikasi Nomor Registrasi Plat Kendaraan Indonesia 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, Januari 2020



Muh. Shahid Setiawan  
NRP. 0721154000044

*Halaman ini sengaja dikosongkan*



LEMBAR PENGESAHAN

DETEKSI DAN IDENTIFIKASI NOMOR REGISTRASI PLAT  
KENDARAAN INDONESIA 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: Muh. Shahid Setiawan (NRP: 07211540000044)

Tanggal Ujian : 6 Januari 2020

Periode Wisuda : Maret 2020

Disetujui oleh:

Dr. Eko Mulyanto Yuniarno, ST., MT.  
NIP: 196806011995121009

(Pembimbing I)

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

(Pembimbing II)

Dr. Supeno Mardji Susiki Nugroho, ST., MT.  
NIP: 197003131995121001

(Penguji I)

Prof. Dr. Ir. Yoyon Kusnendar Suprpto M.Sc.  
NIP. 195409251978031001

(Penguji II)

Dr. Diah Puspito Wulandari, S.T., M.Sc.  
NIP. 196506181999032001

(Penguji III)



Mengetahui  
Kepala Departemen Teknik Komputer

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

*Halaman ini sengaja dikosongkan*

# ABSTRAK

Nama Mahasiswa : Muh. Shahid Setiawan  
Judul Tugas Akhir : Deteksi dan Identifikasi Nomor Registrasi  
Plat Kendaraan Indonesia Berbasis *Convolutional Neural Network* (CNN)  
Pembimbing : 1. Dr. Eko Mulyanto Yuniarno, ST., MT.  
2. Reza Fuad Rachmadi, S.T., M.T.,  
Ph.D. M.T.

Salah satu penerapan pengolahan citra ialah pendeteksian dan identifikasi nomor registrasi plat kendaraan atau biasa dikenal License Plate Recognition (LPR). Diluar negeri, LPR telah diterapkan ke berbagai sistem. Di Indonesia LPR sudah dikembangkan, namun pengembangannya tidak dilakukan secara komperhensif sehingga belum diterapkan ke berbagai sistem. Oleh karena itu dengan memanfaatkan *Convolutional Neural Network*, dibuatlah sebuah sistem yang dapat mendeteksi dan mengidentifikasi nomor registrasi plat kendaraan Indonesia yang akan diterapkan di pintu gerbang kompleks perumahan. Proses training dilakukan menggunakan YOLOv3 dengan menggunakan 1200 dataset. Data tersebut dibagi menjadi 888 *image train* dan 312 *image test*. Hasil training tertinggi yang diperoleh sebesar 55.2% mAP. Kesalahan dalam deteksi dan identifikasi nomor registrasi plat kendaraan Indonesia yaitu paling rendah hanya 1,43%. Proses komputasi dalam deteksi dan identifikasi untuk memproses 3668 *frame* rata-rata membutuhkan 0,585145763 tiap *frame* serta membutuhkan waktu 464,177 detik.

Kata Kunci: Deteksi, Identifikasi, *License Plate Recognition* (LPR), *You Only Look Once* (YOLO), *Convolutional Neural Network* (CNN)

*Halaman ini sengaja dikosongkan*

# ABSTRACT

*Name* : Muh. Shahid Setiawan  
*Title* : *Detection and Identification Indonesian Vehicle License Plate Number based on Convolutional Neural Network (CNN)*  
*Advisors* : 1. Dr. Eko Mulyanto Yuniarno, ST., MT.  
2. Reza Fuad Rachmadi, S.T., M.T., Ph.D.  
M.T.

*One application of image processing is the detection and identification of vehicle license plate numbers or commonly known as License Plate Recognition (LPR). Overseas, LPR has been applied to various systems. In Indonesia LPR has been developed, but its development is not carried out in a comprehensive manner so it has not been applied to various systems. Therefore by utilizing Convolutional Neural Network , a system can be made to detect and identify Indonesian vehicle license plates number that will be implemented at the gate of the housing complex. The training process is done using YOLOv3 with 1200 datasets. The data is divided into 888 image train and 312 image test. The highest training results obtained were 55.2% mAP. The lowest Error in detection and identification of Indonesian vehicle plates number is only 1.43 %. The computation process in detection and identification to process 3668 frames requires an average of 0.585145763 per frame and takes 464,177 seconds.*

*Keywords: Detection, Identification, License Plate Recognition (LPR), You Only Look Once (YOLO), Convolutional Neural Network (CNN)*

*Halaman ini sengaja dikosongkan*

# KATA PENGANTAR

Puji dan syukur kehadiran Allah Swt. atas segala limpahan berkah, rahmat, serta hidayah-Nya, penulis dapat menyelesaikan penelitian ini dengan judul **Deteksi dan Identifikasi Nomor Registrasi Plat Kendaraan Indoneisa 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. Supeno Mardi Susiki Nugroho, ST., MT.
3. Bapak Dr. Eko Mulyanto Yuniarno, ST., MT. dan Reza Fuad Rachmadi, S.T., M.T., Ph.D., yang senantiasa memberikan motivasi, bimbingan, dan arahan selama mengerjakan penelitian.
4. Bapak dan Ibu dosen pengajar dan staf Departemen Teknik Komputer ITS, atas pengajaran, bimbingan, serta perhatian yang diberikan kepada penulis selama 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, Januari 2020

Muh. Shahid Setiawan

*Halaman ini sengaja dikosongkan*



# DAFTAR ISI

Abstrak	i
Abstract	iii
KATA PENGANTAR	v
DAFTAR ISI	vii
DAFTAR GAMBAR	xi
DAFTAR TABEL	xiii
NUMENKLATUR	xv
<b>1 PENDAHULUAN</b>	<b>1</b>
1.1 Latar belakang . . . . .	1
1.2 Permasalahan . . . . .	2
1.3 Tujuan . . . . .	2
1.4 Batasan masalah . . . . .	2
1.5 Sistematika Penulisan . . . . .	3
<b>2 TINJAUAN PUSTAKA</b>	<b>5</b>
2.1 <i>Related Work</i> . . . . .	5
2.1.1 Pengenalan Plat Nomor Kendaraan Indonesia menggunakan <i>Principal Component Analysis</i> dan Metode <i>K-Nearest Neighbor</i> . . . . .	5
2.1.2 Deteksi Kendaraan Secara Real-Time Menggunakan Metode YOLO Berbasis Android . . . . .	6
2.1.3 Aplikasi Tilang Dengan Pengenalan Plat Nomor Kendaraan Dan Pelaku Pada Platform Mobile . . . . .	6
2.1.4 Implementasi Metode Convolutional Neural Network Untuk Klasifikasi Tanaman Pada Citra Resolusi Tinggi . . . . .	6
2.1.5 Deteksi Pengendara Sepeda Motor Menggunakan Deep Convolutional Neural Networks . . . . .	7

2.1.6	Pengembangan Deteksi Citra Mobil Untuk Mengetahui Jumlah Tempat Parkir Menggunakan Cuda Dan Modified Yolo . . . . .	7
2.1.7	Deteksi Jenis Mobil Menggunakan Metode YOLO Dan Faster R-CNN . . . . .	8
2.1.8	Pengembangan Aplikasi Android Untuk Pengenalan Citra Nomor Sertifikat Halal Mui Dengan Library Tesseract Optical Character Recognition . . . . .	8
2.1.9	Optical Character Recognition Untuk Deteksi Pelat Mobil Dan Motor Kendaraan Pada Kampus Teknik Gowa . . . . .	8
2.1.10	<i>Citizen Id Card Detection using Image Processing and Optical Character Recognition . . . . .</i>	9
2.2	<i>License Plate Recognition . . . . .</i>	9
2.3	Pengolahan Citra Digital . . . . .	9
2.3.1	Ruang Warna . . . . .	10
2.3.2	<i>Thresholding . . . . .</i>	11
2.3.3	Kontur . . . . .	12
2.4	<i>Deep Learning . . . . .</i>	12
2.5	<i>Convolutional Neural Network . . . . .</i>	13
2.5.1	<i>Convolutional Layer . . . . .</i>	13
2.5.2	<i>Pooling Layer . . . . .</i>	15
2.5.3	<i>Activation Function . . . . .</i>	16
2.6	<i>K-Nearest Neighbor . . . . .</i>	16
2.7	<i>You Only Look Once . . . . .</i>	17
2.8	<i>Optical Character Recognition . . . . .</i>	20

**3 DESAIN DAN IMPLEMENTASI SISTEM 23**

3.1	Desain Sistem . . . . .	23
3.2	Alur Kerja . . . . .	24
3.3	Pembuatan Dataset . . . . .	24
3.3.1	Pelabelan Objek . . . . .	24
3.3.2	Augmentasi Dataset . . . . .	25
3.4	<i>Training dan Deteksi Menggunakan YOLO . . . . .</i>	26
3.4.1	Pengolahan Input . . . . .	27
3.4.2	<i>Feature Extraction dan Object Classification . . . . .</i>	28
3.4.3	Pengolahan <i>Output</i> . . . . .	28

3.5	Identifikasi Karakter . . . . .	32
3.5.1	<i>Perspective Transform</i> . . . . .	34
3.5.2	<i>Invers Warna</i> . . . . .	34
3.5.3	<i>Gaussian Adaptive Threshold</i> . . . . .	35
3.5.4	<i>Resize</i> . . . . .	35
3.6	Rekognisi Karakter . . . . .	35
<b>4</b>	<b>PENGUJIAN DAN ANALISIS</b>	<b>37</b>
4.1	Pembuatan Sistem Deteksi dan Identifikasi Nomor Registrasi Plat Kendaraan Indonesia . . . . .	37
4.1.1	Pembuatan Dataset . . . . .	37
4.1.2	<i>Training Data</i> . . . . .	38
4.1.3	Hasil Deteksi . . . . .	38
4.1.4	Hasil <i>Perspective Transform</i> . . . . .	38
4.1.5	<i>Invers Warna</i> . . . . .	38
4.1.6	<i>Gaussian Adaptive Threshold</i> . . . . .	40
4.2	Pengujian Deteksi dan Identifikasi Nomor Registrasi Plat Kendaraan Indonesia Menggunakan Input Foto dan Video . . . . .	40
4.2.1	Pengujian Deteksi dan Identifikasi Nomor Registrasi Plat Kendaraan Indonesia Bagian Depan Menggunakan <i>Input Foto</i> . . . . .	42
4.2.2	Pengujian Deteksi dan Identifikasi Nomor Registrasi Plat Kendaraan Indonesia Bagian Belakang Menggunakan <i>Input Foto</i> . . . . .	44
4.2.3	Pengujian Deteksi dan Identifikasi Nomor Registrasi Plat Kendaraan Indonesia Bagian Depan Menggunakan <i>Input Video</i> . . . . .	47
4.2.4	Pengujian Deteksi dan Identifikasi Nomor Registrasi Plat Kendaraan Indonesia Bagian Belakang Menggunakan <i>Input Video</i> . . . . .	49
4.2.5	Pengujian Deteksi dan Identifikasi Nomor Registrasi Plat Kendaraan Indonesia dengan Perbedaan Pemilihan Ketinggian dan Sudut Kamera . . . . .	52
4.3	Pengujian Deteksi dan Identifikasi Nomor Registrasi Plat Kendaraan Indonesia Pada Malam Hari . . . . .	56
4.4	Pengujian Kecepatan Komputasi Sistem . . . . .	56

4.5 Pengujian Menggunakan Perbedaan Variasi Dataset	58
<b>5 PENUTUP</b>	<b>61</b>
5.1 Kesimpulan . . . . .	61
5.2 Saran . . . . .	62
<b>DAFTAR PUSTAKA</b>	<b>63</b>
<b>Biografi Penulis</b>	<b>67</b>

# DAFTAR GAMBAR

2.1	<i>Convolutional Layer</i> [20]	14
2.2	<i>Pooling Layer Layer</i> [20]	16
2.3	Sistem Pendeteksian YOLO	18
2.4	Proses Pendeteksian Objek YOLO	20
2.5	Arsitektur YOLO	21
3.1	Blok Diagram Sistem Deteksi dan Identifikasi Nomor Registrasi Plat Kendaraan Indonesia	23
3.2	Diagram Alir Pembuatan Dataset	25
3.3	Deteksi Plat Nomor Kendaraan Indonesia menggunakan YOLO	26
3.4	Susunan <i>Layer</i> Darknet53	29
3.5	Prediksi Lokasi <i>Bounding Box</i>	30
3.6	<i>Diagram Alir Identifikasi Karakter</i>	33
3.7	Arsitektur <i>Tesseract OCR</i> [26]	36
4.1	Pengambilan Data Secara Langsung Dilokasi	37
4.2	Pengambilan Data dari Internet	38
4.3	Grafik Hasil <i>Training</i>	39
4.4	Hasil Deteksi	39
4.5	Hasil <i>Perspective Transform</i>	39
4.6	<i>InversWarna</i>	40
4.7	<i>Gaussian Adaptive Threshold</i>	40
4.8	Kondisi Karakter Bagian Tengah Pada Nomor Registrasi Plat Tidak Rata	43
4.9	Deteksi dan Identifikasi Nomor Registrasi Plat Bagian Depan Menggunakan Input Foto Tanpa Kesalahan	43
4.10	Kesalahan Identifikasi Karakter	45
4.11	Kondisi Nomor Registrasi Plat Yang Kurang Bagus	46
4.12	Deteksi dan Identifikasi Nomor Registrasi Plat Bagian Belakang Menggunakan Input Foto Tanpa Kesalahan	46
4.13	Hasil Deteksi Yang Tidak Dapat Membedakan Karakter Awal dan Akhir	48

4.14	Deteksi dan Identifikasi Nomor Registrasi Plat Bagian Depan Menggunakan <i>Input</i> Video Tanpa Kesalahan	48
4.15	Hasil Deteksi Yang Hanya Mendeteksi Salah Satu Dari Karakter Awal dan Akhir . . . . .	50
4.16	Nomor Registrasi Plat Tidak Terlihat Jelas Oleh Kamera . . . . .	51
4.17	Beberapa Bagian Karakter Tidak Terdeteksi . . . . .	51
4.18	Deteksi dan Identifikasi Nomor Registrasi Plat Bagian Belakang Menggunakan Input Video Tanpa Kesalahan . . . . .	52
4.19	Nomor Registrasi Plat Kurang Terlihat Jelas oleh Kamera . . . . .	53
4.20	Hasil Deteksi Tidak Sesuai . . . . .	54
4.21	Kesalahan Rekognisi Karakter Karena Cetakan Nomor Registrasi Plat Tidak Lurus . . . . .	54
4.22	Deteksi dan Identifikasi Nomor Registrasi Plat Dengan Perbedaan Kamera, Tinggi, dan Sudut Kamera Tanpa Kesalahan . . . . .	55
4.23	Nomor Registrasi Plat Kendaraan Tidak Terlihat Jelas Pada Malam Hari . . . . .	56

## DAFTAR TABEL

4.1	Spesifikasi Kamera Canon EOS M10[30] . . . . .	41
4.2	Spesifikasi Kamera Samsung A70[31] . . . . .	42
4.3	Tabel Pengujian Deteksi dan Identifikasi Nomor Registrasi Plat Kendaraan Indonesia Bagian Depan Menggunakan <i>Input</i> Foto . . . . .	42
4.4	Tabel Pengujian Deteksi dan Identifikasi Nomor Registrasi Plat Kendaraan Indonesia Bagian Belakang Menggunakan <i>Input</i> Foto . . . . .	44
4.5	Tabel Pengujian Deteksi dan Identifikasi Nomor Registrasi Plat Kendaraan Indonesia Bagian Depan Menggunakan <i>Input</i> Video . . . . .	47
4.6	Tabel Pengujian Deteksi dan Identifikasi Nomor Registrasi Plat Kendaraan Indonesia Bagian Belakang Menggunakan <i>Input</i> Video . . . . .	49
4.7	Tabel Pengujian dengan Perbedaan Pemilihan Ketinggian dan Sudut Camera. . . . .	53
4.8	Spesifikasi ASUS ROG GL503GE[32] . . . . .	57
4.9	Kecepatan Komputasi Sistem Terhadap Foto . . . . .	58
4.10	Kecepatan Komputasi Sistem Terhadap Video . . . . .	58
4.11	Tabel Pengujian Perbedaan Variasi Dataset-2 di Parkiran . . . . .	59

*Halaman ini sengaja dikosongkan*



# NUMENKLATUR

$Pr$	: <i>Predection</i>
$IOU$	: <i>Intersection Over Union</i>
$b_x$	: <i>Lokasi Boundix Box x</i>
$b_y$	: <i>Lokasi Boundix Box y</i>
$b_w$	: <i>Dimensi Akhir Boundix Box Weight</i>
$b_h$	: <i>Dimensi Akhir Boundix Box Height</i>
$t_x$	: <i>Fungsi Signoid x</i>
$t_y$	: <i>Fungsi Signoid y</i>
$c_x$	: <i>Lokasi Grid x</i>
$c_y$	: <i>Lokasi Grid y</i>
$p_w$	: <i>Anchor Box Weight</i>
$p_h$	: <i>Anchor Box Height</i>
$n$	: <i>Jumlah kelas</i>
$B$	: <i>Jumlah Bounding Box</i>
$C$	: <i>Jumlah Kelas</i>
$AO$	: <i>Area of Overlap</i>
$AU$	: <i>Area of Union</i>
$P$	: <i>Precision</i>
$R$	: <i>Recall</i>
$TP$	: <i>True Positive</i>
$FP$	: <i>False Positive</i>
$FN$	: <i>True Negative</i>
$src(f(x,y))$	: <i>Source Pixel</i>
$dst(x,y)$	: <i>Destination Pixel</i>

*Halaman ini sengaja dikosongkan*

# BAB 1

## PENDAHULUAN

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

### 1.1 Latar belakang

Ilmu dan teknologi pengolahan citra merupakan suatu ilmu yang bermanfaat dalam kehidupan. Pengolahan citra adalah sebuah disiplin ilmu yang mempelajari tentang teknik-teknik mengolah citra. Citra yang dimaksud disini adalah gambar diam berupa foto maupun gambar bergerak [1]. Beberapa manfaat pengolahan citra ialah untuk mendapatkan informasi berupa objek apa saja yang terdapat dalam sebuah gambar atau pendeteksian objek yang apabila dengan mata sulit dilihat dengan jelas. Salah satu penerapan pengolahan citra ialah pendeteksian dan identifikasi nomor registrasi plat kendaraan atau biasa dikenal *License Plate Recognition* (LPR). Di luar negeri, LPR telah diterapkan ke berbagai sistem[2]-[3]. Di Indonesia LPR sudah dikemabgkan, namun pengembangannya tidak dilakukan secara komperhensif. Salah satu permasalahan untuk LPR kendaraan di Indonesia yaitu untuk tahap pendeteksian. Meskipun sudah banyak pengembangan yang dilakukan di luar negeri kita tidak dapat mengandalkannya karena struktur plat nomor setiap negara berbeda-beda. Di Indonesia, LPR juga telah dikembangkan[4]-[5]. Salah satu pengembangan LPR kendaraan Indonesia yaitu dengan menggunakan metode *K-Nearest Neighbor*. Input yang digunakan pada penelitian tersebut tidak menggunakan foto atau video kendaraan secara utuh untuk mendeteksi nomor registrasi plat, tetapi hanya foto secara manual nomor registrasi plat kendaraannya. Hasil tersebut juga tidak diuji coba dilapangan sehingga tidak dapat diketahui apakah sistem tersebut bisa diimplementasikan atau tidak.

LPR dapat diterapkan ke berbagai sistem misalnya sebagai access control kompleks perumahan yang memungkinkan hanya kendaraan penghuni yang dapat masuk kompleks. Access control pada

kompleks perumahan di Indonesia masih dilakukan secara konvensional. Satpam kompleks perumahan harus terus berada pada lokasi gerbang kompleks untuk membuka atau menutup serta mengawasi kendaraan yang masuk melewati gerbang secara manual.

Berdasarkan uraian diatas, dibuatlah sebuah sistem secara komperhensif yang dapat mendeteksi dan mengidentifikasi nomor registrasi plat kendaraan Indonesia berbasis *Convolutional Neural Network* (CNN) yang akan diterapkan di gerbang kompleks perumahan. CNN adalah pengembangan dari *Multilayer Perceptron* (MLP) yang didesain untuk mengolah data dua dimensi. CNN termasuk dalam jenis *Deep Neural Network* karena kedalaman jaringan yang tinggi dan banyak diaplikasikan pada data citra[6].

## 1.2 Permasalahan

Permasalahan dalam tugas akhir ini adalah kurangnya pengembangan LPR di Indonesia secara komperhensif untuk mendeteksi dan mengidentifikasi nomor registrasi plat kendaraan Indonesia serta belum dapat diterapkan ke berbagai sistem.

## 1.3 Tujuan

Tujuan dari tugas akhir ini adalah untuk mengembangkan sebuah sistem secara komperhensif untuk mendeteksi dan mengidentifikasi nomor registrasi plat kendaraan Indonesia berbasis *Convolutional Neural Network* yang dapat diterapkan di pintu gerbang komplek perumahan agar nantinya sistem ini dapat dikembangkan dan digunakan kedalam sistem lainnya.

## 1.4 Batasan masalah

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

1. Kondisi nomor registrasi plat kendaraan Indonesia yang akan dideteksi adalah menghadap ke arah depan kamera.
2. Metode yang digunakan untuk proses deteksi dan identifikasi adalah *Convolutional Neural Network* (CNN).
3. Nomor registrasi plat kendaraan yang akan dideteksi dan diidentifikasi adalah plat kendaraan roda 4 pribadi atau dinas dengan ukuran yaitu 395x135 mm.

4. Proses deteksi dan identifikasi nomor plat kendaraan Indonesia pada intensitas cahaya yang cukup.
5. Hanya dapat mengidentifikasi 1 nomor registrasi plat kendaraan Indonesia dalam sebuah citra.

## 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 *License Plate Recognition*, *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 dan proses pendeteksian dan identifikasi nomor registrasi plat kendaraan Indonesia. 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.
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 juga dituliskan pada bab ini.

*Halaman ini sengaja dikosongkan*

# BAB 2

## TINJAUAN PUSTAKA

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

### **2.1 *Related Work***

Berikut akan dijelaskan mengenai beberapa penelitian yang berelasi dengan tugas akhir ini.

#### **2.1.1 Pengenalan Plat Nomor Kendaraan Indonesia menggunakan *Principal Component Analysis* dan Metode *K-Nearest Neighbor***

Tujuan dari penelitian yang dilakukan Rohimah[5] adalah untuk pengenalan plat nomor kendaraan menggunakan Metode K-Nearest Neighbor agar plat kendaraan yang melintas tersimpan di-database. Penelitian ini terdiri dari beberapa tahapan yaitu memasukkan *input* berupa *image* plat nomor kendaraan Indonesia yang diambil dari webcam laptop, kemudian *image* tersebut di *pre-processing* dengan melakukan *resizing* dan *grayscale*. Setelah itu *image* tersebut akan di segmentasi dan di ekstraksi fitur sehingga dapat diklasifikasi menggunakan *K-Nearest Neighbor* agar plat nomor kendaraan tersebut dapat dikenali. Hasil pengujian telah dilakukan pada aplikasi Deteksi Plat Kendaraan pada *image* menggunakan algoritma K-Nearest Neighbor, dapat diperoleh nilai akurasi dengan rata-rata 92,86% dari 25 plat nomor kendaraan yang dikenali. Akurasi yang diperoleh cukup bagus tetapi pengambilan *input* dengan kondisi ideal yaitu menggunakan latar belakang putih disekitar plat nomor, tidak diimplementasikan dilokasi. Karena apabila dilakukan diimplementasikan dilokasi, akurasi yang diperoleh akan berkurang karena salah satunya disebabkan oleh warna dari disekitar plat nomor sangat bervariasi sehingga sangat sulit atau tidak bisa untuk mendeteksi plat nomor kendaraan tersebut.

### **2.1.2 Deteksi Kendaraan Secara Real-Time Menggunakan Metode YOLO Berbasis Android**

Tujuan dari penelitian yang dilakukan Sri[7] adalah merancang sebuah aplikasi deteksi kendaraan secara *real-time* berbasis android menggunakan metode YOLO (*You Only Look Once*). Analisa yang dilakukan menggunakan jumlah dataset sebanyak 200, 4 kelas, 10 batch, dan 200 epoch. Proses pelatihan dilakukan hingga 4.000 step, dan penyimpanan checkpoint ke bentuk protobuf file dilakukan pada step 800, 1.000, 1.200, 1.400, 1.600, 1.800, 2.000, 3.000, dan 4.000. Hasil yang diperoleh yaitu kendaraan dapat dideteksi dengan nilai akurasi 83.3% dari 5/6 objek yang dideteksi.

### **2.1.3 Aplikasi Tilang Dengan Pengenalan Plat Nomor Kendaraan Dan Pelaku Pada Platform Mobile**

Tujuan dari penelitian yang dilakukan Ariasa[8], dkk adalah membuat aplikasi berbasis *android* dengan menggunakan *library Leptonica* yang berfungsi untuk memproses gambar hasil dari *capture* kamera dan akan di-ekstrak menjadi teks menggunakan *Library Tesseract*, kemudian hasil dari ekstraksi tersebut akan dicocokkan dengan data yang telah ada pada server. Jika info STNK tercatat pada database maka polisi lalu lintas bisa melakukan penilangan ditempat. Hasil yang diperoleh yaitu dari 168 karakter, yang berhasil dikenali berjumlah 137 karakter dan yang tidak dikenal berjumlah 31 karakter atau 81,54% dikenali dan 18,45% tidak dikenali. Hal itu disebabkan karena adanya kemiripan karakter satu sama lain, seperti B dan 8, 2 dan Z, 1 dan I, D dan 0, A dan 4. Selain itu terdapat segmentasi yang tidak sempurna karena posisi plat nomor kendaraan miring.

### **2.1.4 Implementasi Metode Convolutional Neural Network Untuk Klasifikasi Tanaman Pada Citra Resolusi Tinggi**

Tujuan dari penelitian yang dilakukan Arrofiqoh[9], dkk adalah untuk menerangkan algoritma CNN dalam membedakan jenis tanaman dengan memberikan label semantik dari objek jenis tanaman. Penelitian menggunakan 5 kelas jenis tanaman, yaitu kelas tanaman padi, bawang merah, kelapa, pisang, dan cabai. Proses le-



arning jaringan menghasilkan akurasi 100% terhadap data training . Pengujian terhadap data validasi menghasilkan akurasi 93% dan akurasi terhadap data tes 82%. Hasil penelitian tersebut menunjukkan bahwa penggunaan metode CNN berpotensi untuk pendekatan pengenalan objek secara otomatis dalam membedakan jenis tanaman sebagai bahan pertimbangan interpreter dalam menentukan objek pada citra.

### **2.1.5 Deteksi Pengendara Sepeda Motor Menggunakan Deep Convolutional Neural Networks**

Tujuan dari penelitian yang dilakukan Limantoro[10], dkk adalah memahami arsitektur CNN dengan hasil yang terbaik. Pada penelitian tersebut, *dash cam* digunakan untuk merekam video di jalan raya Surabaya pada siang hari atau cuaca cerah dengan resolusi 1080 HD. Video dengan warna RGB diekstrak menjadi beberapa citra dan diberi label. Lalu, data citra dilatih dengan convolutional neural networks menghasilkan bobot. Eksperimen yang dilakukan adalah membagi arsitektur CNN menjadi tiga, yaitu kecil, sedang, dan besar. Lalu, arsitektur CNN tersebut saling dibandingkan. Pada hasil eksperimen, arsitektur CNN besar memiliki akurasi yang tertinggi sebesar 93% dan *f1-score* sebesar 94,9%. Arsitektur CNN yang kompleks dan besar cenderung memiliki performa yang lebih baik.

### **2.1.6 Pengembangan Deteksi Citra Mobil Untuk Mengetahui Jumlah Tempat Parkir Menggunakan Cuda Dan Modified Yolo**

Tujuan dari penelitian yang dilakukan Jupriadi[11], dkk adalah untuk mengetahui jumlah slot pada lahan parkir dengan akurat sehingga memudahkan pengelola parkir. Selain itu sistem tersebut juga dikembangkan agar waktu pencarian lahan parkir oleh pengguna parkir bisa sangat cepat. Sistem tersebut menggunakan penerapan pemrograman GPU yang dikombinasi dengan *Modified Yolo* (M-Yolo). GPU pada MYolo dibutuhkan untuk mengolah citra sekaligus mengolah data untuk mendeteksi citra mobil dan jumlah mobil secara paralel. Hasil uji coba menunjukkan bahwa dengan menggunakan GPU dibandingkan dengan CPU dapat mempercepat waktu komputasi rata-rata sebesar 0,179 detik dengan rata-rata akurasi sebesar 100

### **2.1.7 Deteksi Jenis Mobil Menggunakan Metode YOLO Dan Faster R-CNN**

Tujuan dari penelitian yang dilakukan Shianto[12], dkk adalah untuk melakukan identifikasi secara otomatis. Untuk identifikasi gambar, salah satu metode yang terkenal adalah *Faster R-CNN* yang cukup cepat dan tepat untuk melakukan identifikasi gambar. Namun untuk ketepatannya masih belum maksimal. Metode lain yang tersedia adalah YOLO dimana metode ini akan lebih cepat dalam melakukan identifikasi. Penggunaan kedua metode dalam arsitektur dibangun bertujuan untuk meningkatkan akurasi kebenaran identifikasi jenis mobil. Hasil uji coba menunjukkan bahwa YOLO memiliki akurasi prediksi yang lebih rendah dari *Faster R-CNN*, namun dapat memprediksi lebih akurat dari *Faster RCNN*. Penambahan metode YOLO dapat menambah akurasi deteksi *Faster R-CNN*.

### **2.1.8 Pengembangan Aplikasi Android Untuk Pengenalan Citra Nomor Sertifikat Halal Mui Dengan Library Tesseract Optical Character Recognition**

Tujuan dari penelitian yang dilakukan Muhtadii[13] adalah membuat aplikasi yang dapat mengidentifikasi nomor sertifikat halal secara manual dan otomatis menggunakan citra. Untuk dapat mengenali karakter pada sebuah citra akan digunakan sebuah sistem *Optical Character Recognition (OCR)*. OCR merupakan sistem komputer yang dapat membaca karakter pada sebuah citra. Dari sekian banyak sistem OCR yang ada, pada penelitian tersebut menggunakan tesseract OCR karena tesseract memiliki tingkat akurasi yang cukup besar yaitu diatas 94%. Hasil penelitian tersebut adalah sebuah aplikasi android yang dapat mengidentifikasi nomor sertifikat halal secara manual dan otomatis menggunakan citra.

### **2.1.9 Optical Character Recognition Untuk Deteksi Pelat Mobil Dan Motor Kendaraan Pada Kampus Teknik Gowa**

Tujuan dari penelitian yang dilakukan Mansyur[14] adalah mengembangkan sistem yang mampu mengidentifikasi pelat nomor kendaraan dalam area kampus Teknik Unhas yang berada di Kabupaten

Gowa melalui data video. Metode yang digunakan adalah *Local Binary Pattern Cascade Classifier* untuk mendeteksi pelat dan *Optical Character Recognition* (OCR) untuk rekognisi teks pada pelat kendaraan dengan memanfaatkan aplikasi Matlab 2016a. Data video yang diolah beresolusi 1280x720 pixel yang telah direkam menggunakan static camera. Hasil penelitian tersebut menunjukkan bahwa tingkat akurasi pendeteksian dan rekognisi pelat kendaraan area kampus Teknik Unhas mampu mencapai akurasi sebesar 80.9%. Dari *profiler* matlab, untuk video riil berdurasi 10 detik, dibutuhkan waktu proses selama 150 detik.

### **2.1.10 Citizen Id Card Detection using Image Processing and Optical Character Recognition**

Tujuan penelitian yang dilakukan Satyawan[15], dkk adalah untuk mendeteksi kartu identitas elektronik menggunakan kombinasi dari *Image Processing* dan *Optical Character Recognition*. Hasil penelitian tersebut memperoleh akurasi 98% dari deteksi kartu identitas menggunakan metode kombinasi tersebut. Penelitian tersebut diterapkan dalam situs *web* yang digunakan oleh perusahaan otomotif.

## **2.2 License Plate Recognition**

*License Plate Recognition* (LPR) adalah modul perangkat keras dan perangkat lunak yang terintegrasi yang mengidentifikasi isi pelat yaitu nomor identifikasi kendaraan dan menghasilkan urutan karakter ASCII yang tepat. Pengenalan plat nomor kendaraan bermanfaat untuk mengenali identitas dari kendaraan. Pengenalan plat nomor pada kendaraan tersebut dilakukan secara otomatis. Dari citra nomor plat yang tertangkap sistem akan mengenali karakter apa saja yang terdapat pada citra[16].

## **2.3 Pengolahan Citra Digital**

Pengolahan Citra Digital atau (*Digital Image Processing*) merupakan sebuah disiplin ilmu yang mempelajari tentang teknik-teknik dalam pengolahan citra. Arti citra sendiri adalah penelitian pada suatu gambar statis yang berasal dari sensor vision bisa berupa kamera ataupun webcam. Tetapi secara matematis, citra adalah fungsi kontinu dengan intensitas cahaya pada bidang 2 dimensi. Kemudian pengolahan citra digital agar bisa dikalkulasikan pada komputer maka akan bersifat numerik dan diskrit. Sebuah citra digital

dapat diwakili oleh sebuah matriks dua dimensi  $f(x,y)$  yang terdiri dari  $M$  kolom dan  $N$  baris[1]. Citra juga tersusun dari sekumpulan piksel yang memiliki koordinat  $f(x,y)$  dan amplitudo  $f(x,y)$ . Fungsi koordinat sendiri sebagai posisi piksel pada suatu citra, sedangkan amplitudo mengacu pada intensitas warna pada citra. Dan besarnya nilai tingkat kecerahan tersebut juga dinamakan grayscale atau tingkat keabuan suatu citra. Dan terdapat elemen pada pengolahan citra digital salah satunya ruang warna, *thresholding* dan *contour*.

Dalam ilmu komputer, pemrosesan gambar digital adalah penggunaan algoritma komputer untuk melakukan pemrosesan gambar pada gambar digital. Sebagai subkategori atau bidang pemrosesan sinyal digital, pemrosesan gambar digital memiliki banyak keunggulan dibandingkan pemrosesan gambar analog. Ini memungkinkan jangkauan algoritma yang jauh lebih luas untuk diterapkan pada data *input* dan dapat menghindari masalah seperti penumpukan kebisingan dan distorsi sinyal selama pemrosesan. Karena gambar didefinisikan lebih dari dua dimensi (mungkin lebih) pemrosesan gambar digital dapat dimodelkan dalam bentuk sistem multidimensi.

### 2.3.1 Ruang Warna

Pada pengolahan warna gambar atau ruang warna ada bermacam-macam model salah satunya adalah model rgb atau normalisasi RGB. Pada color *image* ini masing-masing piksel memiliki warna tertentu, warna tersebut adalah merah (*Red*), hijau (*Green*) dan biru (*Blue*). Model pengolahan ini merupakan pengolahan warna dengan menghitung persentase warna RGB dalam sebuah citra. Dengan menggunakan model ini, sebuah objek dengan warna tertentu dapat dideteksi dan terbebas dari pengaruh perubahan intensitas cahaya dari luar. Kelebihan dari RGB sendiri adalah paling mudah dipahami dikarenakan warna-warna tersebut penyusun sinar tampak. Kelemahan dari pengolahan warna model ini adalah tidak dapat membedakan warna hitam dan putih[1]. Ruang warna yang kedua yaitu ruang warna HSV (*Hue, Saturation, Value*). HSV adalah sebuah ruang warna yang seringkali lebih natural terhadap persepsi manusia terhadap warna. HSV merupakan hasil transformasi dari ruang RGB sehingga komponen-komponennya merupakan turunan dari ruang RGB dan sangat berkaitan.

*Hue* (H) dari sebuah warna merujuk kepada kemurnian warna tersebut. Setiap variasi warna merah memiliki hue yang sama. Dideskripsikan sebagai sebuah angka yang menentukan kemurnian warna pada lingkaran warna. Lingkaran warna dibagi menjadi 6 bagian oleh warna primer beserta komplemennya. *Saturation* (S) dari sebuah warna menunjukkan seberapa “putih” warna tersebut. Semakin “putih” saturasinya, maka warna tersebut semakin tidak pekat. Nilai saturasi ‘0’ berarti warna tersebut “putih”. Nilai ‘1’ berarti warna tersebut *fully saturated*. *Value* (V) dari sebuah warna menunjukkan seberapa gelap atau terang warna tersebut. Nilai *value* ‘0’ berarti warna tersebut sangat gelap (hitam), sebaliknya, nilai *value* ‘1’ maka warna tersebut sangat cerah.

### **2.3.2 Thresholding**

*Thresholding* merupakan suatu teknik atau elemen pada pengolahan citra yang digunakan untuk mengubah citra *grayscale* atau keabuan menjadi citra hitam putih. Kemudian proses *thresholding* sendiri disebut dengan proses binerisasi. Karena dalam melakukan *thresholding* hasil outputnya berupa citra biner. *Thresholding* memisahkan antara objek dengan background dalam suatu citra berdasarkan pada perbedaan tingkat kecerahannya atau gelap terangnya. terdapat 2 jenis *threshold* yaitu *global thresholding* dan *adaptive thresholding*.

#### **2.3.2.1 Global Thresholding**

*Global thresholding* merupakan proses *thresholding* dengan menentukan *threshold* derajat keabuan terlebih dahulu. Setelah itu, *threshold* akan diaplikasikan ke semua piksel pada suatu citra. Artinya, dalam satu citra, proses *thresholding* dilakukan secara menyeluruh pada setiap piksel dengan *threshold* yang sama.

#### **2.3.2.2 Adaptive Thresholding**

Pada gambar dengan pencahayaan yang tidak merata, hasil *thresholding* dengan metode global *thresholding* akan terlihat buruk. Misal, dilakukan *thresholding* pada sebuah dokumen dengan pencahayaan yang tidak merata (sisi kanan lebih gelap). *Global thresholding* bisa menyebabkan hasil *thresholding* pada sisi yang lebih gelap tersebut menjadi hitam, padahal pada sisi tersebut bisa saja terdapat karakter tulisan. Karakter tersebut menjadi tidak terbaca karena menyatu dengan warna backgroundnya. Dengan

*adaptive thresholding*, hal ini bisa diatasi. *Adaptive threshold* secara statistik mengecek nilai intensitas piksel-piksel yang berdekatan (local neighborhood) dengan suatu piksel acuan. Perhitungan ini bisa dilakukan dengan menghitung nilai rata-rata, median, ataupun rata-rata dari nilai maksimum dan minimum dari piksel-piksel yang berdekatan dan mengesetnya sebagai nilai *threshold*.

### 2.3.3 Kontur

Kontur adalah suatu proses Rangkaian pixel-pixel tepi yang membentuk batas daerah (*region boundary*). Kontur dapat berupa senarai tepi (*edge list*) atau berupa kurva. Senarai tepi merupakan himpunan terurut piksel-piksel tepi. Representasi kontur ke dalam kurva merupakan representasi yang kompak dan mangkus untuk analisis citra. Misalnya, rangkaian piksel tepi yang membentuk garis dapat direpresentasikan hanya dengan sebuah persamaan garis lurus. Representasi semacam ini menyederhanakan perhitungan selanjutnya seperti arah dan panjang garis. Akan tetapi untuk dapat mencari kontur diharuskan untuk melakukan *thresholding* terlebih dahulu supaya hasil gambar yang akan diproses adalah biner. Deteksi kontur ini sangat berguna dalam melokalisasi nomor registrasi plat kendaraan pada proses pengolahan citra dalam sistem ini dan menentukan *region of interestnya* sebab itu akan mempermudah dalam pencarian letak dari nomor registrasi plat kendaraan tersebut.

## 2.4 Deep Learning

Deep Learning (DL) adalah salah satu bidang yang muncul dari penelitian Machine Learning (ML). Deep Learning merupakan sebuah metode implementasi machine learning yang bertujuan meniru cara kerja otak manusia menggunakan artificial neural network atau jaringan saraf buatan. Deep Learning memungkinkan model komputasi yang tersusun dari beberapa lapisan pemrosesan untuk mempelajari representasi dari data dengan berbagai level abstraksi. Deep Learning menemukan struktur sulit yang terdapat dalam kumpulan data yang besar dengan menggunakan algoritma backpropagation. Struktur yang didapatkan menunjukkan parameter internal yang harus diubah oleh mesin agar dapat menghitung representasi di setiap layer berdasarkan representasi dari layer sebelumnya[17].

Deep Learning melakukan pendekatan dalam penyelesaian masalah dengan menggunakan konsep hierarki. Dengan konsep terse-

but, komputer mampu mempelajari sebuah konsep yang kompleks dengan menggabungkan konsep-konsep yang lebih sederhana[18].

## 2.5 *Convolutional Neural Network*

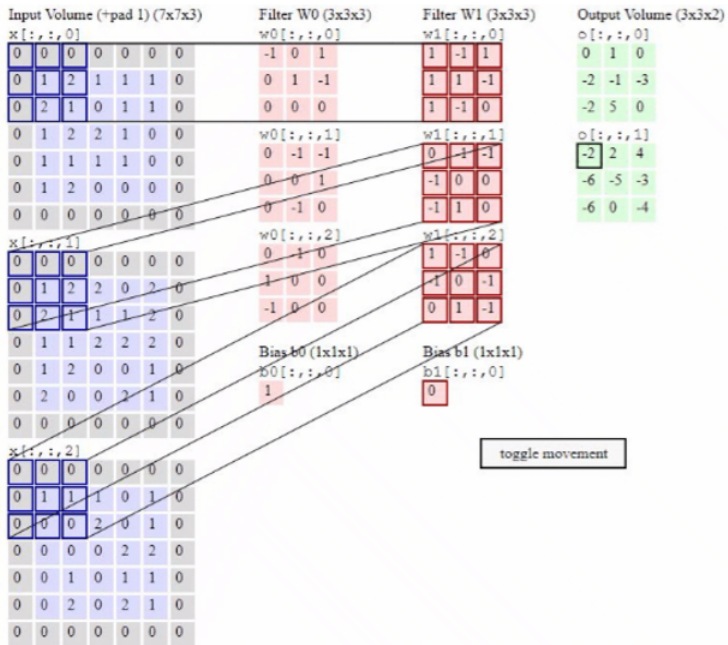
*Convolutional Neural Network* (CNN) adalah variasi dari Multilayer Perceptron yang terinspirasi dari jaringan saraf manusia. Pemberian nama convolutional neural network mengindikasikan bahwa jaringan tersebut menggunakan operasi matematika yang disebut konvolusi. Konvolusi sendiri adalah sebuah operasi linier. CNN merupakan pengembangan dari artificial neural network yang saat ini diklaim sebagai model terbaik untuk memecahkan masalah seperti object recognition dan detection. CNN termasuk dalam kategori Supervised Learning dimana dilakukan proses pembelajaran (learning) terhadap komputer menggunakan masukan data latih yang telah diberi label.

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[19].

### 2.5.1 *Convolutional Layer*

*Convolutional Layer* adalah bagian dari tahap pada arsitektur CNN. Tahap ini melakukan operasi konvolusi pada output dari layer sebelumnya. Layer tersebut adalah proses utama yang mendasari jaringan arsitektur CNN. Konvolusi adalah istilah matematis dimana pengaplikasian sebuah fungsi pada *output* fungsi lain secara berulang. Operasi konvolusi merupakan operasi pada dua fungsi argumen bernilai nyata. Operasi ini menerapkan fungsi *output* sebagai *Feature Map* dari input citra[21]. *Convolutional layer* biasanya dijadikan layer pertama pada *feature learning* arsitektur CNN. Pada 2.1 ukuran *convolutional layer* adalah 7x7x3. Panjang 7 pixels, tinggi 7 pixels dan tebal 3 channel.

Ketiga filter ini akan digeser keseluruhan bagian dari citra. Setiap pergeseran akan dilakukan operasi *dot product* antara *input* dan nilai dari filter tersebut sehingga menghasilkan sebuah *output* atau



Gambar 2.1: Convolutional Layer[20].

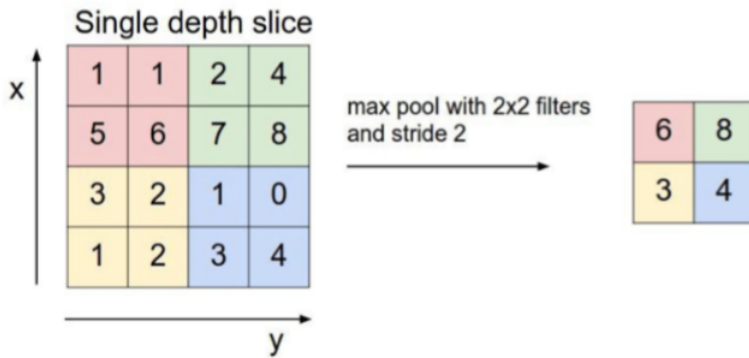


biasa disebut sebagai *activation map* atau *feature map*. Parameter yang menentukan berapa jumlah pergeseran filter disebut "*Stride*". Jika nilai *stride* adalah 1, maka convolutional filter akan bergeser sebanyak 1 *pixels* secara horizontal lalu vertical. Semakin kecil *stride* maka akan semakin detail informasi yang akan didapatkan dari sebuah input, namun membutuhkan proses komputasi yang lebih jika dibandingkan dengan *stride* yang besar. Perlu diperhatikan bahwa dengan menggunakan *stride* bernilai kecil tidak selalu akan mendapatkan performa yang bagus

*Padding* atau *Zero Padding* adalah parameter yang menentukan jumlah *pixels* (berisi nilai 0) yang akan ditambahkan di setiap sisi dari input. Dimensi *output* dari *convolutional layer* selalu lebih kecil dari *inputnya* (kecuali penggunaan 1x1 filter jumlah dengan *stride* 1). *Output* ini akan digunakan kembali sebagai *input* dari *convolutional layer* selanjutnya, sehingga informasi akan banyak yang terbuang. Dengan menggunakan *padding* dapat mengatur dimensi *output* agar tetap sama seperti dimensi *input* atau setidaknya tidak berkurang secara drastis, sehingga penggunaan *convolutional layer* yang lebih dalam dapat menghasilkan lebih banyak *features* yang di-*extract*[20].

### 2.5.2 *Pooling Layer*

*Pooling layer* terdapat setelah *convolutional layer*. Terdiri dari sebuah filter dengan ukuran dan *stride* tertentu yang akan bergeser pada seluruh *activation map*. Fungsi dari *pooling* ini adalah untuk mereduksi *input* secara spasial (mengurangi jumlah parameter) dengan operasi *down-sampling*. Dalam *pooling layer* terdapat dua macam *pooling* yang biasa digunakan yaitu *average pooling* dan *maxpooling*. Nilai yang diambil pada *average pooling* adalah nilai rata-rata, sedangkan pada *max-pooling* adalah nilai maksimal. Lapisan *pooling* yang dimasukkan diantara lapisan konvolusi secara berturut-turut dalam arsitektur model CNN dapat secara progresif mengurangi ukuran volume output pada *feature map*, sehingga mengurangi jumlah parameter dan perhitungan di jaringan, untuk mengendalikan *overfitting*[21]. Berdasarkan Gambar 2.2 menunjukkan proses dari *max-pooling*. *Output* dari proses *pooling* adalah sebuah matriks dengan dimensi yang lebih kecil dibandingkan dengan dimensi citra awal. Lapisan *pooling* tersebut akan beroperasi pa-



Gambar 2.2: *Pooling Layer*[20].

da setiap irisan kedalaman volume *input* secara bergantian. Jika dilihat dari Gambar 2.2 operasi *max-pooling* menggunakan ukuran filter 2x2. Masukan pada proses tersebut berukuran 4x4, dari masing-masing 4 angka pada *input* operasi tersebut diambil nilai maksimalnya kemudian dilanjutkan membuat ukuran *output* baru menjadi ukuran 2x2[21].

### 2.5.3 *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).

### 2.6 *K-Nearest Neighbor*

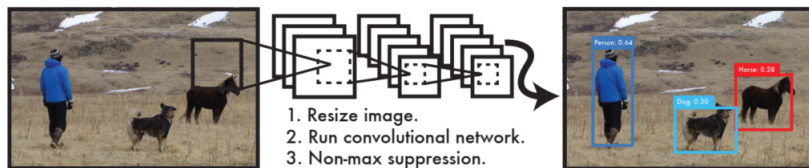
*K-Nearest Neighbor* (KNN) adalah metode yang digunakan untuk melakukan klasifikasi terhadap suatu objek berdasarkan k buah

data yang dilatih dengan jarak paling dekat dengan objek tersebut. Syarat nilai  $k$  adalah tidak boleh lebih besar dari jumlah data latih, dan nilai  $k$  harus ganjil dan lebih dari satu[22]. Klasifikasi dilakukan tanpa menggunakan model tetapi hanya berdasarkan memori. Algoritma *K-Nearest Neighbour* menggunakan klasifikasi ketetanggaan sebagai prediksi terhadap data baru. Pada fase *training*, algoritma ini hanya melakukan penyimpanan vektor-vektor fitur dan klasifikasi data training sampel. Pada fase klasifikasi, fitur-fitur yang sama dihitung untuk *testing* data (klasifikasinya belum diketahui). Jarak dari vektor yang baru terhadap seluruh vektor *training* sampel dihitung, dan sejumlah  $k$  buah yang paling dekat diambil. Titik yang baru klasifikasinya diprediksikan termasuk pada klasifikasi terbanyak dari titik-titik tersebut[23].

Nilai  $K$  yang terbaik untuk algoritma ini tergantung pada data, secara umum nilai  $K$  yang tinggi akan mengurangi efek pada klasifikasi. Namun membuat batasan antara setiap klasifikasi menjadi lebih kabur sehingga nilai  $K$  yang bagus dapat dipilih dengan optimasi parameter. Contohnya dengan menggunakan *cross-validation*, dimana klasifikasi diprediksikan berdasarkan data pembelajaran yang paling dekat. Cara kerja berdasarkan jarak minimum dari data baru terhadap  $K$  tetangga terdekat yang telah ditetapkan. Setelah diperoleh  $K$  tetangga terdekat, prediksi kelas dari data baru akan ditentukan berdasarkan mayoritas  $K$  tetangga terdekat. Data untuk *K-Nearest Neighbour* terdiri dari beberapa atribut  $X_i$  yang akan digunakan untuk mengklasifikasikan  $Y$ . Data dapat berupa data ordinal, nominal sampai dengan skala kuantitatif, namun dalam penelitian ini data yang digunakan adalah biner (nominal)  $Y$ .

## 2.7 *You Only Look Once*

*You Only Look Once* (YOLO) merupakan sistem pendeteksi objek berbasis *convolutional neural network*. Berbeda dari sistem pendeteksi objek pada umumnya yang memiliki proses yang kompleks untuk melakukan pendeteksian, YOLO menggantikan keseluruhan proses tersebut dengan melakukan deteksi dengan menggunakan *single neural network* sehingga dapat mendeteksi objek dengan sangat cepat dan akurat. Secara umum, proses pendeteksian yang dilakukan oleh YOLO terbagi ke dalam tiga langkah utama



**Gambar 2.3:** Sistem Pendeteksian YOLO[24].

sebagaimana yang ditunjukkan oleh Gambar 2.3, yaitu mengubah ukuran citra menjadi  $448 \times 448$  piksel, lalu menjalankan *single convolutional network* pada citra, dan terakhir melakukan *threshold* pada hasil deteksi berdasarkan pada *confidence score* yang didapatkan oleh model.

YOLO mendeteksi 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 mendeteksi 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:  $x$ ,  $y$ ,  $w$ ,  $h$ , 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 pendeteksian, probabilitas kelas dikalikan dengan *confidence* pada kotak prediksi sesuai dengan persamaan 2.1.

$$Pr(Class_i|Object) \times Pr(Object) \times_{truth} IOU_{pred} \quad (2.1)$$

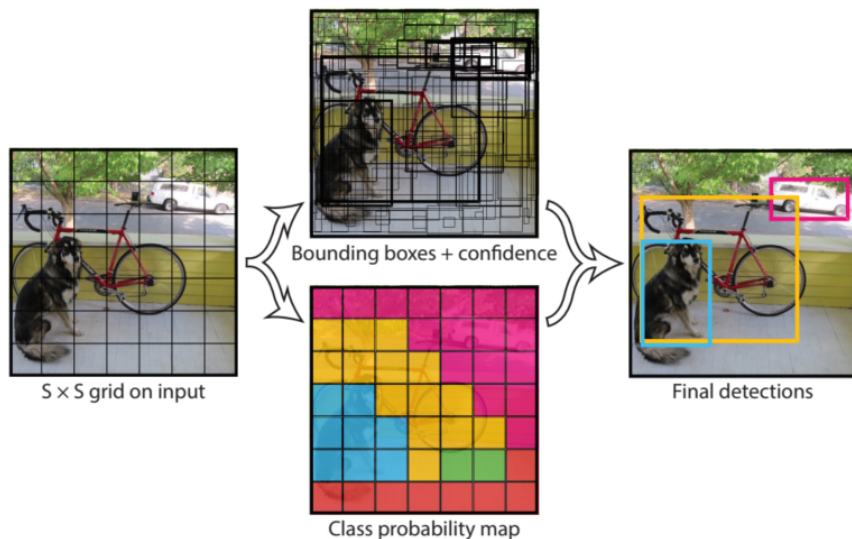
Atau yang dapat disederhanakan menjadi persamaan 2.2.

$$Pr(Class_i) \times_{truth} IOU_{pred} \quad (2.2)$$

Dari persamaan 2.2 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 pendeteksian objek ini divisualisasikan oleh Gambar 2.4

Terdapat beberapa keuntungan dari penggunaan *unified model* pada YOLO. Pertama, pendeteksian 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 pendeteksian dilakukan dengan menggunakan *sliding windows* dan *region proposal* seperti yang digunakan pada pendeteksi 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 2.5. Meskipun sudah mampu melakukan pendeteksian 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*[24]. Namun, YOLO terus mengalami perkembangan menjadi YOLOv2 (YOLO9000) dan yang terbaru adalah YOLOv3, dimana performa yang dihasilkan semakin meningkat seiring dengan ber-



**Gambar 2.4:** Proses Pendeteksian Objek YOLO[24].

kembangnya metode ini.

## 2.8 *Optical Character Recognition*

*Optical Character Recognition* (OCR) adalah proses konversi gambar huruf menjadi karakter ASCII yang dikenali oleh komputer. Gambar huruf yang dimaksud dapat berupa hasil scan dokumen, hasil print-screen halaman web, hasil foto, dan lain-lain[25]. Dalam pemrosesan OCR, gambar atau *bitmap* yang dipindai dianalisis untuk area terang dan gelap untuk mengidentifikasi setiap huruf alfabet atau digit angka. Ketika sebuah karakter dikenali, itu dikonversi menjadi kode ASCII. Papan sirkuit khusus dan cip komputer yang dirancang khusus untuk OCR digunakan untuk mempercepat proses pengenalan.

Teknik ini seperti kombinasi dari mata dan otak manusia. Sebuah mata dapat melihat sebuah teks dari gambar tetapi sebenarnya otak yang memproses dan mengolah teks tersebut sehingga dapat dibaca oleh mata. Dalam pengembangan OCR terdapat beberapa



masalah, diantaranya yang pertama adalah terdapat sedikit perbedaan antara beberapa huruf dengan angka yang sulit dibedakan. Contohnya antara angka 0 dengan huruf o. Faktor kedua yaitu faktor cahaya dalam citra yang mempersulit sistem dalam mengenali huruf[26].



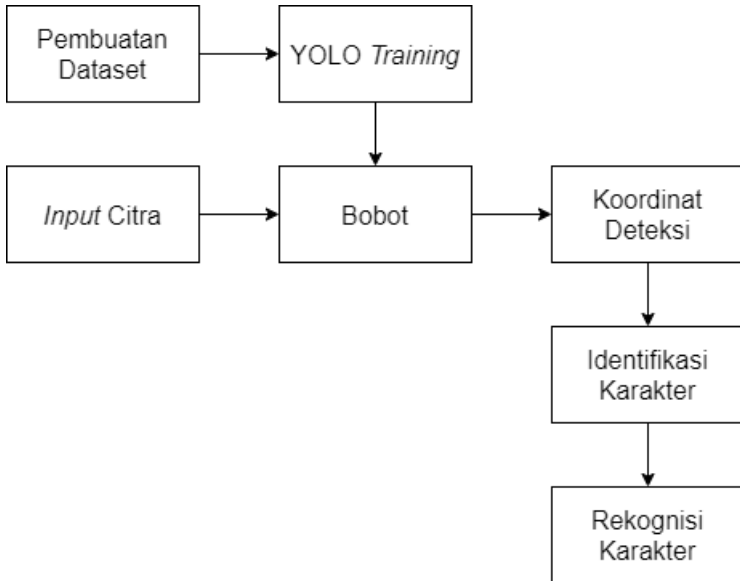
# BAB 3

## DESAIN DAN IMPLEMENTASI SISTEM

### 3.1 Desain Sistem

Tugas akhir ini bertujuan untuk mendeteksi dan mengidentifikasi nomor registrasi plat kendaraan Indonesia berbasis *Convolutional Neural Network* (CNN). Proses kerja dari sistem ini ditunjukkan pada Gambar 3.1.

Berdasarkan Gambar 3.1, proses pertama yang dilakukan adalah deteksi nomor registrasi plat kendaraan Indonesia. Tujuan dari deteksi tersebut adalah untuk mengetahui posisi atau letak nomor registrasi plat kendaraan Indonesia dalam sebuah citra. Untuk mencapai hal tersebut, diperlukan pembuatan dataset. Tujuan dari pembuatan dataset adalah untuk mempersiapkan data berupa plat



**Gambar 3.1:** Blok Diagram Sistem Deteksi dan Identifikasi Nomor Registrasi Plat Kendaraan Indonesia.

nomor yang akan dilatih. Data yang telah diambil kemudian diberikan label atau anotasi berupa *ground-truth bounding box* yang menunjukkan dimana posisi nomor registrasi plat nomor serta kelasnya. Karena data yang diambil masih cukup kurang, maka dilakukan augmentasi data untuk meningkatkan jumlah dataset. *Training* dan pendeteksi nomor registrasi plat kendaraan Indonesia yang digunakan pada penelitian ini berbasis CNN, yaitu *You Only Look Once* (YOLO). Hasil *training* yang diperoleh berupa *weights* dan hasil tersebut digunakan untuk mendeteksi nomor registrasi plat kendaraan Indonesia. Proses kedua yaitu Identifikasi. Untuk mengidentifikasi hasil dari deteksi tersebut, perlu dilakukan Pre-Processing terlebih dahulu. Pre-Processing yang digunakan yaitu *Perspective Transform*, invers warna, *Gaussian Adaptive Threshold*, *resize*, dan *border*. Hasil dari pre-processing tersebut akan disegmentasi pada masing-masing karakter pada nomor registrasi plat kendaraan dan karakternya akan direkognisi.

## 3.2 Alur Kerja

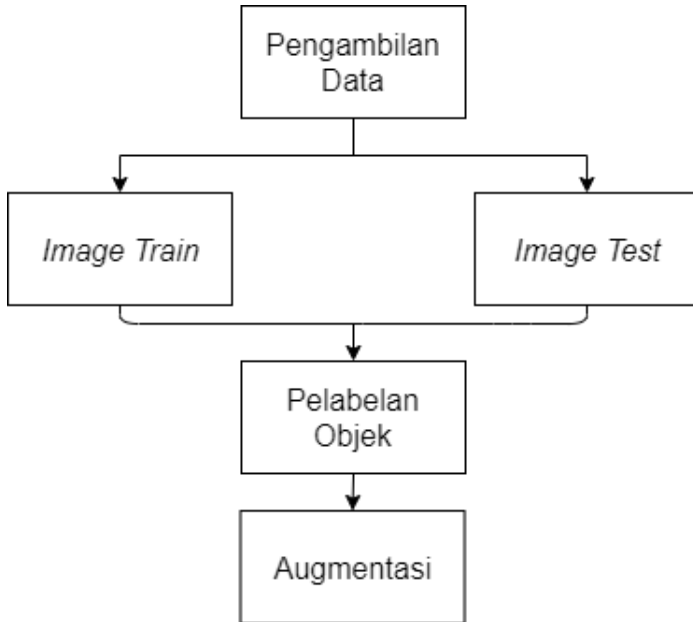
1. Pembuatan Dataset.
2. *Training*.
3. Deteksi.
4. Identifikasi.

## 3.3 Pembuatan Dataset

Pada pembuatan dataset, dilakukan pengambilan data dengan cara foto secara langsung, pengambilan gambar dari internet baik berupa gambar maupun video. Agar dataset tersebut dapat dilatih menggunakan YOLO, data tersebut perlu diberi label dan juga perlu dilakukan augmentasi untuk meningkatkan jumlah dataset. Gambar 3.2 adalah proses pembuatan dataset.

### 3.3.1 Pelabelan Objek

Tahap pelabelan atau anotasi objek merupakan proses pemberian informasi berupa kelas dan posisi dari nomor registrasi plat kendaraan yang akan dideteksi. Dari dataset yang didapatkan, dilakukan pelabelan satu per satu sehingga didapatkan koordinat *ground-truth bounding box* yang akan dibandingkan dengan *predicted bounding box*. Dari kedua jenis *bounding box* yang ada, akan

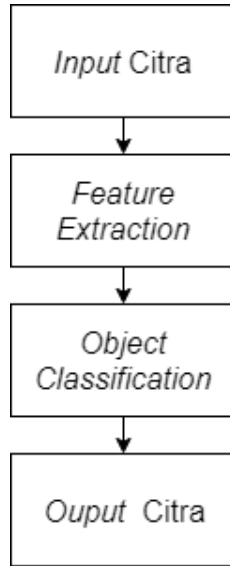


**Gambar 3.2:** Diagram Alir Pembuatan Dataset.

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 gambar yang terdapat nomor registrasi plat kendaraan Indonesia di dalamnya. Informasi mengenai label yang telah dibuat kemudian disimpan dalam bentuk file TXT.

### 3.3.2 Augmentasi Dataset

Tahap augmentasi dataset merupakan proses penambahan dataset agar dataset yang diperoleh bisa menjadi lebih banyak tanpa perlu mengambil data secara langsung. Dari dataset yang telah didapatkan, data tersebut akan disalin dan kemudian dilakukan *translation*, *rotation*, *shear*, *scale*, *reflection*, *HSV saturation*, *HSV intensity*.



**Gambar 3.3:** Deteksi Plat Nomor Kendaraan Indonesia menggunakan YOLO.

### 3.4 *Training* dan Deteksi Menggunakan YOLO

Pada tahap ini dataset yang telah dilabeli dan diaugmentasi akan dilatih sehingga membentuk sebuah pola yang hasilnya berbentuk Bobot. Bobot tersebut akan digunakan untuk mengenali objek dalam citra. *Training* yang digunakan berbasis CNN yaitu *You Only Look Once* (YOLO). Perangkat yang digunakan dalam proses *training* tersebut menggunakan *Google Colabs*. YOLO yang digunakan yaitu *Yolov3* dengan jumlah 3 kelas.

Proses deteksi dilakukan dengan menggunakan program yang telah dibuat sebelumnya dan kemudian dilakukan pengubahan parameter serta penambahan kode agar dapat digunakan ke tahap selanjutnya. Pada tahap ini, citra baik berupa gambar atau video digunakan sebagai input yang kemudian dilewatkan sistem berbasis CNN, yaitu YOLO. Output dari proses pendeteksian berupa confidence, nama kelas, serta posisi koordinat bounding box dari nomor registrasi plat kendaraan Indonesia yang terdeteksi oleh

sistem. Koordinat bounding box tersebut akan digunakan untuk mengidentifikasi citra tersebut. Gambar 4.4 adalah proses deteksi nomor registrasi plat kendaraan Indonesia menggunakan YOLO.

### 3.4.1 Pengolahan Input

YOLO membutuhkan input gambar dengan ukuran piksel dengan kelipatan 32 untuk mendeteksi objek. Semakin besar ukuran gambar, maka semakin akurat pula hasil pendeteksian yang didapatkan. Namun, semakin besar ukuran input gambar akan membuat proses komputasi menjadi lebih lambat. Hal ini juga berlaku sebaliknya, apabila *input* citra yang diberikan semakin kecil maka hasil pendeteksian semakin berkurang akurasi, namun proses komputasi akan berjalan dengan lebih cepat. Oleh sebab itu, ukuran piksel input untuk melakukan training dan deteksi ditentukan berdasarkan perangkat yang digunakan untuk melakukan komputasi.

YOLO juga membutuhkan pengaturan nilai *batch size*. *Batch size* adalah jumlah sampel data yang disebarkan ke Neural Network. Contoh: jika terdapat 100 dataset dan batch size adalah 5, maka algoritma ini akan menggunakan 5 sampel data pertama dari 100 data yang dimiliki (ke1, ke2, ke3, ke4, dan ke5) lalu disebarkan atau dilatih oleh Neural Network sampai selesai kemudian mengambil kembali 5 sampel data kedua dari 100 data (ke6, ke7, ke8, ke9, dan ke10), dan begitu seterusnya sampai 5 sampel data ke 20 ( $100/5=20$ ). Semakin tinggi nilai *batch size* yang digunakan, maka proses *training* akan berlangsung semakin cepat. Nilai tertinggi *batch size* yang digunakan berdasarkan kelipatan 2 dari nilai VRAM perangkat yang digunakan untuk melakukan *training*. Misalnya nilai VRAM perangkat yang dimiliki adalah 8gb, maka nilai *batch size* terbesar adalah 16.

Untuk dapat mengenal object yang akan dideteksi, YOLO perlu mengetahui nama dan jumlah kelas yang akan *ditraining* dan dideteksi. Setelah itu, YOLO juga membutuhkan pengaturan nilai *filters size*. *Filters size* adalah berapa banyak informasi yang dapat diperoleh ketika memproses *layer*. Nilai *filter size* didapatkan dari 4 *box coordinates*, 1 *object confidence* dan nilai *n* yaitu jumlah kelas kemudian dikalikan 3 nilai *anchors* atau dapat dilihat dalam persamaan 3.1. Nilai anchors adalah rasio nilai *bounding boxes* dengan

*height* dan *weight* dari beberapa data yang akan *ditraining*. Apabila jumlah kelas adalah 3, maka nilai *filters* adalah 24.

$$[4 + 1 + n] * 3 = 15 + 3 * n \quad (3.1)$$

### 3.4.2 Feature Extraction dan Object Classification

YOLOv3 menggunakan *feature extraction* yang bernama Darknet53, yang memiliki 53 *layer* yang telah dilatih di ImageNet. Untuk tugas pendeteksian, ditambahkan lagi 53 *layer* ke dalam jaringan, sehingga terdapat 106 *layer* yang mendasari arsitektur *convolutional* dari YOLOv3. Hal ini menyebabkan proses pendeteksian lebih lambat, namun memberikan hasil yang lebih akurat. Gambar 3.4 menunjukkan susunan *layer* dari Darknet53.

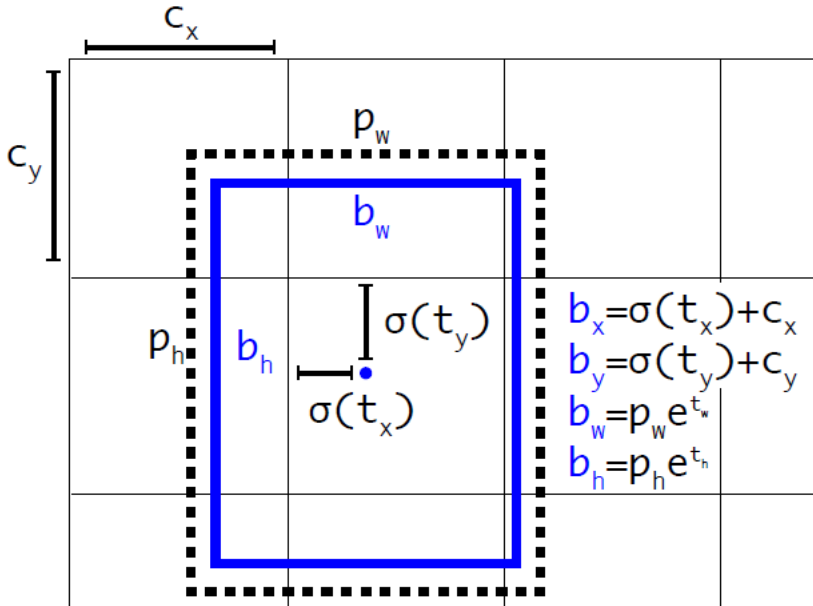
Sedangkan untuk prediksi kelasnya, YOLO generasi awal menggunakan fungsi *softmax* yang menghitung probabilitas setiap kelas target dari semua kemungkinan kelas target. Namun pada YOLOv3 digunakan *multilabel classification* pada setiap objek yang terdeteksi pada gambar. Hal ini dilakukan karena fungsi *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.

### 3.4.3 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 terdeteksi. YOLO membuat *bounding box* pada lokasi *feature map* dengan menggunakan *logistic activation sigmoid*. Pengolahan *bounding box* dari hasil deteksi diilustrasikan pada Gambar 3.5. 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 dida-

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

Gambar 3.4: Susunan *Layer* Darknet53[27].



Gambar 3.5: Prediksi Lokasi *Bounding Box*[27].

patkan 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  $(t_w, t_h)$ . Semua perhitungan tersebut dapat dilihat pada persamaan 3.2, persamaan 3.3, persamaan 3.4, dan persamaan 3.5

$$b_x = \sigma(t_x) + c_x \quad (3.2)$$

$$b_y = \sigma(t_y) + c_y \quad (3.3)$$

$$b_w = p_w e^{t_w} \quad (3.4)$$



$$b_h = p_h e^{t_h} \quad (3.5)$$

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  $416 \times 416$ , maka dihasilkan *output* dalam tiga skala dalam ukuran dimensi  $13 \times 13$ ,  $26 \times 26$ , dan  $52 \times 52$  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*  $1 \times 1$  pada *feature map* pada tiga ukuran yang berbeda di tiga tempat yang berbeda pada jaringan. Bentuk dari *kernel* deteksi tersebut dapat dilihat pada persamaan 3.6. 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.

$$1 \times 1 \times (B \times (5 + C)) \quad (3.6)$$

Hasil dari pendeteksian 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* yang dapat dituliskan pada persamaan 3.7.

$$IoU = \frac{AO}{AU} \quad (3.7)$$

Apabila nilai IoU dari kedua *bounding box* tersebut bernilai lebih dari *threshold* yang ditentukan dalam penelitian ini, yaitu 0.5, maka hasil pendeteksian tersebut dianggap benar. Sebaliknya, apabila ternyata nilai IoU yang didapatkan kurang dari 0.5, maka hasil pendeteksian 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 pendeteksian. Nilai *precision* dan *recall* yang diperoleh dengan menggunakan persamaan 3.8 dan 3.9.

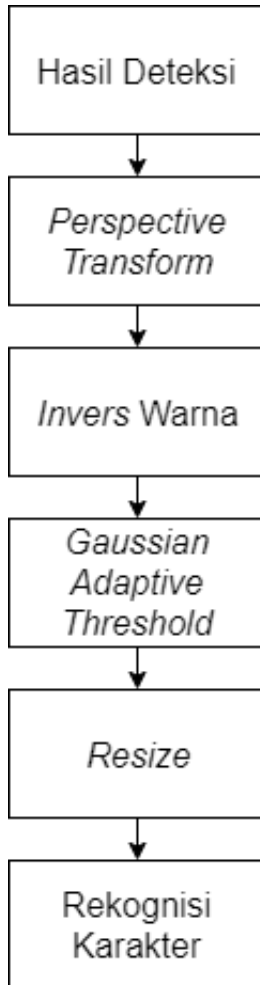
$$P = \frac{TP}{TP + FP} \quad (3.8)$$

$$R = \frac{TP}{TP + FN} \quad (3.9)$$

Nilai tersebut 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 kemudian dari hasil tersebut diperoleh nilai *mean Average Precision* (mAP). Nilai mAP adalah metrik evaluasi yang mengukur keakuratan dari sebuah pendeteksi objek dalam melakukan proses pendeteksian. Nilai mAP diperoleh dari rata-rata nilai *Average Precision* (AP) dari tiap kelas objek.

### 3.5 Identifikasi Karakter

Untuk mengidentifikasi karakter, diperlukan hasil deteksi pada karakter awal dan karakter akhir pada nomor registrasi plat kendaraan. Apabila terdapat kesalahan pada deteksi, maka nomor registrasi plat tidak bisa diidentifikasi. Hasil deteksi tersebut akan dilakukan *Pre-processing*. *Pre-processing* adalah sebuah proses memanipulasi citra untuk mendapatkan fitur penting yang dapat digunakan sebagai indikator dalam proses rekognisi pada citra yaitu citra atau gambar nomor registrasi plat kendaraan Indonesia yang telah dideteksi menggunakan YOLO. Terdapat beberapa tahap *Pre-processing* yang digunakan yaitu *Perspective Transform*,



**Gambar 3.6:** Diagram Alir Identifikasi Karakter.

invers warna, *Gaussian Adaptive Threshold*, dan *Resize*. Hasil dari *Pre-processing* akan diidentifikasi menggunakan *tesseract ocr*. Gambar 3.6 adalah proses identifikasi nomor registrasi plat.

### 3.5.1 *Perspective Transform*

Fungsi *Perspective Transform* membuat proses manipulasi gambar yang terlihat miring diubah menjadi lurus. Fungsi tersebut tidak mengubah konten gambar tetapi menghilangkan *pixel grid* yang tidak digunakan dan memetakan *grid* ke *image* yang dituju. Untuk setiap piksel  $(x, y)$  dari bagian *image* yang dituju, fungsi *perspective transform* menghitung koordinat piksel yang sesuai dengan *image* yang dituju dan menyalin nilai piksel atau dapat dilihat dari persamaan 3.10[28]:

$$\text{dst}(x, y) = \text{src}(f_x(x, y), f_y(x, y)) \quad (3.10)$$

Untuk dapat melakukan fungsi *Perspective Transform* diperlukan 4 titik koordinat yang merupakan posisi yang akan dilihat ketika gambar tersebut lurus. Nilai titik-titik koordinat tersebut diperoleh dari hasil deteksi yang telah diperoleh menggunakan YOLO dengan mengambil kelas pada karakter awal dan akhir pada nomor registrasi plat kendaraan yang diperoleh. 4 titik koordinat tersebut merupakan koodinat dari karakter awal dan karakter akhir dari nomor registrasi plat kendaraan. Apabila ada kesalahan deteksi misalnya hasil deteksi tidak dapat membedakan karakter awal dan akhir nomor registrasi plat, maka *Perspective Transform* transform tidak dapat dilakukan atau hasil dari *Perspective Transform* tidak sesuai dengan yang diinginkan.

### 3.5.2 *Invers Warna*

Digunakannya invers warna pada *image* karena pada penelitian ini hanya dapat menggunakan nomor registrasi plat dari negara-negara eropa yang menggunakan nomor registrasi plat dengan latar putih dan karakter hitam. Sedangkan pada negara Indonesia sendiri menggunakan nomor registrasi plat yang mempunyai latar hitam dan karakter putih, sehingga diperlukan invers warna pada foto pada saat proses pembacaan *image*. Warna yang dipakai untuk invers warna sendiri adalah RGB (Red, Green, Blue) dengan masing

masing mempunyai titik 255 yang akan menggunakan perhitungan invers  $f(x,y) = 255 - f(x,y)$ .

### 3.5.3 *Gaussian Adaptive Threshold*

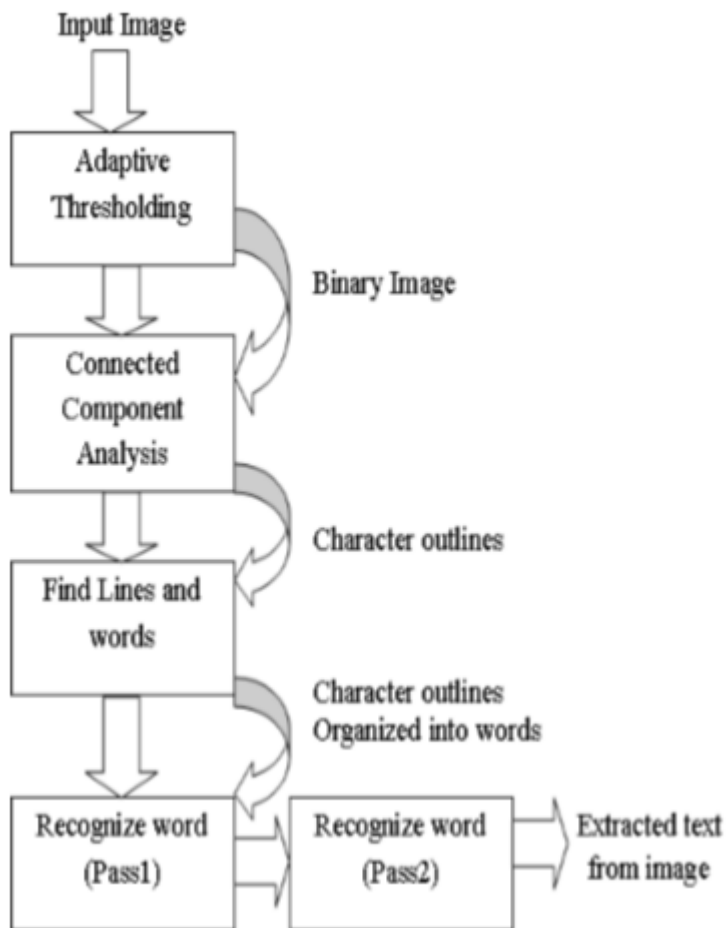
Fungsi *Gaussian Adaptive Threshold* manipulasi *image* dengan cara memberi batasan yang berbeda untuk berbagai wilayah pada gambar sehingga memberi hasil yang lebih baik dengan iluminasi yang bervariasi. Digunakannya *Gaussian Adaptive Threshold* karena input data yang diperoleh memiliki kondisi pencahayaan yang berbeda-beda.

### 3.5.4 *Resize*

*Resize* merupakan proses manipulasi *image* dengan mengubah dimensi *image* menjadi lebih kecil atau lebih besar. Pada tahap ini dilakukan mengubah dimensi *image* menjadi lebih kecil dengan menjaga nilai rasio dan tidak terlalu banyak kehilangan kualitasnya. *image* diubah menjadi 200 dpi karena *tesseract ocr* dapat berfungsi dengan baik jika *image* yang dimiliki DPI minimal 300 dpi [29].

## 3.6 *Rekognisi Karakter*

Pada sistem ini, rekognisi karakter dilakukan menggunakan *tesseract ocr*. *Tesseract ocr* merupakan library open-source. Berdasarkan gambar 3.7, langkah pertama adalah Adaptive Thresholding, yang mengubah *image* menjadi *image* biner. Langkah selanjutnya adalah *connected compenent analysis*, yang digunakan untuk mengekstrak *character outlines*. Kemudian setelah itu, *character outlines* diubah menjadi *blobs*. Blobs adalah *small region* dari sebuah *image*. Blobs disusun dalam *text lines*, dan *lines* serta *region* dianalisis untuk menetapkan beberapa area atau ukuran teks yang sesuai. Teks dibagi menjadi beberapa kata menggunakan *definite spaces* and *fuzzy spaces*. Rekognisi teks kemudian dibagi menjadi 2 tahap seperti yang ditunjukkan pada gambar 3.7. Pada tahap pertama, proses yang dilakukan adalah untuk mengenali setiap kata dari teks. Setiap kata yang sesuai akan diteruskan ke *adaptive classifier* sebagai *training data*. *Adaptive classifier* mencoba mengenali teks dengan cara yang lebih akurat. Karena *adaptive classifier* telah mendapatkan beberapa *training data*, data tersebut akan *ditraining* kemudian digunakan untuk mengekstraksi teks dari *image* dan dapat merekognisi karakter dari *image* tersebut[26].



**Gambar 3.7:** Arsitektur Tesseract OCR[26].

## BAB 4

# PENGUJIAN DAN ANALISIS

Pada penelitian ini, dipaparkan hasil pengujian serta analisis dari desain sistem dan implementasi yaitu terhadap sistem deteksi dan identifikasi nomor registrasi plat kendaraan Indonesia berbasis *Convolutional Neural Network* (CNN):

### 4.1 Pembuatan Sistem Deteksi dan Identifikasi Nomor Registrasi Plat Kendaraan Indonesia

#### 4.1.1 Pembuatan Dataset

Pembuatan dataset bertujuan untuk menyiapkan data-data yang akan digunakan untuk training. Data berupa image nomor registrasi plat kendaraan Indonesia yang diperoleh dengan cara mengambil data berupa foto atau video langsung dilokasi maupun mengambil dari internet. Apabila data berupa video, video tersebut dikonversikan menjadi beberapa image. Gambar 4.1 merupakan salah satu contoh pengambilan data langsung dilokasi dan gambar 4.2 merupakan salah satu contoh pengambilan data dari internet. Data yang diperoleh berjumlah 1200. Data tersebut dibagi menjadi 888 *image train* dan 312 *image test*.



**Gambar 4.1:** Pengambilan Data Secara Langsung Dilokasi.



**Gambar 4.2:** Pengambilan Data dari Internet.

### 4.1.2 *Training* Data

*Training* data bertujuan untuk menghasilkan sebuah bobot yang akan digunakan dalam proses deteksi nomor registrasi plat kendaraan Indonesia. Proses *training* data dilakukan menggunakan YOLOv3 dengan *pytorch*. *Training* dilakukan menggunakan *Google Collabs*. Proses training berlangsung selama 1 hari dengan *training* menggunakan 1200 epoch, dengan *img size* 416 dan *batch size* 16 dengan jumlah 3 class. Gambar 4.3 adalah grafik hasil dari *training*. Berdasarkan grafik tersebut, mAP terbaik yang diperoleh sebesar 52.3%.

### 4.1.3 Hasil Deteksi

Gambar 4.4 adalah contoh dari hasil deteksi plat, karakter awal, dan karakter akhir dari nomor registrasi plat kendaraan Indonesia.

### 4.1.4 Hasil *Perspective Transform*

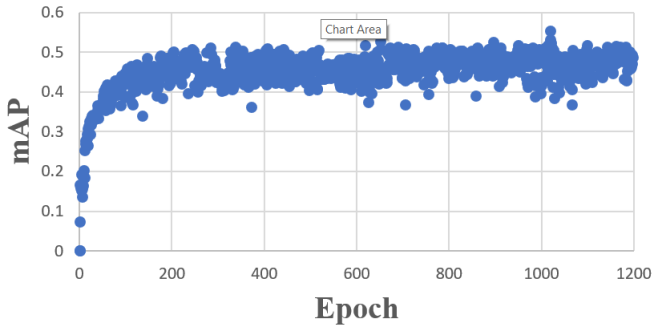
Gambar 4.5 adalah contoh dari hasil *Perspective Transform*.

### 4.1.5 *Invers* Warna

Gambar 4.6 adalah contoh dari hasil *invers* warna.



## Hasil Training



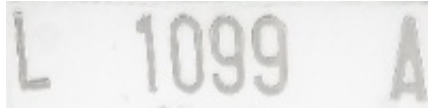
Gambar 4.3: Grafik Hasil *Training*.



Gambar 4.4: Hasil Deteksi.



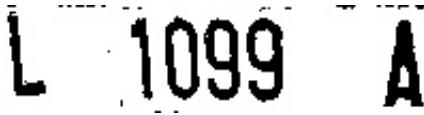
Gambar 4.5: Hasil *Perspective Transform*.



**Gambar 4.6:** *Invers Warna.*

#### 4.1.6 *Gaussian Adaptive Threshold*

Gambar 4.7 adalah contoh dari hasil *Gaussian Adaptive Threshold*.



**Gambar 4.7:** *Gaussian Adaptive Threshold.*

### 4.2 Pengujian Deteksi dan Identifikasi Nomor Registrasi Plat Kendaraan Indonesia Menggunakan Input Foto dan Video

Pengujian input foto dilakukan dengan 2 cara yaitu pengujian deteksi dan identifikasi nomor registrasi plat kendaraan Indonesia bagian depan dan belakang. Untuk menghitung akurasi deteksi dan identifikasi menggunakan persamaan 4.1 sedangkan untuk menghitung akurasi kesalahan karakter, menggunakan persamaan 4.2.

$$Akurasi \ Deteksi/Identifikasi = \frac{\sum Deteksi/Identifikasi}{\sum Kendaraan} \times 100\% \quad (4.1)$$

$$Akurasi \ Kesalahan = 100\% - \frac{\sum Karakter - \sum Kesalahan}{\sum Kendaraan} \times 100\% \quad (4.2)$$

Pengujian dilakukan menggunakan 2 jenis kamera yaitu kamera Samsung A70 dan kamera Canon EOS M10. Kamera Samsung

digunakan untuk mengambil video kendaraan pada bagian depan dan belakang. Sedangkan Kamera Canon EOS M10 digunakan untuk mengambil video kendaraan dengan menggunakan perbedaan tinggi dan sudut kamera. Pengambilan video kamera Samsung A70 dilakukan dengan menggunakan resolusi full HD 1080p 60 fps sedangkan pengambilan video kamera Canon EOS M10 menggunakan resolusi full HD 1080p 30 fps. Tabel 4.2 adalah spesifikasi dari kamera Samsung A70 dan tabel 4.1 adalah spesifikasi dari kamera Canon EOS M10.

**Tabel 4.1:** Spesifikasi Kamera Canon EOS M10[30]

Type	Type	Digital single-lens non-reflex AF/AE camera
	Image sensor size	Approx. 22.3 x 14.9 mm (APS-C Size)
Image Sensor	Type	CMOS sensor
	Effective pixels	Approx. 18.0 megapixels
Recording System	Type	Design rule for Camera File system 2.0
	Max Frame Rate	60Hz: 30fps @(1920 x 1080, 1280 x 960, 1280 x 720)
	Max Resolution	1920 x 1080
Dimensions	Dimensions (W x H x D)	Approx. 108.0 x 66.6 x 35.0 mm (4.25 x 2.62 x 1.38 in.) RAW + JPEG simultaneous recording possible

**Tabel 4.2:** Spesifikasi Kamera Samsung A70[31]

Rear Camera - Resolution (Multiple)	32.0 MP + 5.0 MP + 8.0 MP
Rear Camera - F Number (Multiple)	F1.7 , F2.2 , F2.2
Main Camera - Auto Focus	Yes
Rear Camera - OIS	Yes
Main Camera - Flash	Yes
Max Resolusi Rekaman Video	UHD 4K (3840 x 2160) @30fps
Slow Motion	240fps @HD

#### 4.2.1 Pengujian Deteksi dan Identifikasi Nomor Registrasi Plat Kendaraan Indonesia Bagian Depan Menggunakan *Input* Foto

**Tabel 4.3:** Tabel Pengujian Deteksi dan Identifikasi Nomor Registrasi Plat Kendaraan Indonesia Bagian Depan Menggunakan *Input* Foto

Mobil	Deteksi	Identifikasi	Kesalahan
Ke-1	Ya	Ya	0/6 Karakter
Ke-2	Ya	Ya	0/7 Karakter
Ke-3	Ya	Ya	0/8 Karakter
Ke-4	Ya	Ya	1/6 Karakter
Ke-5	Ya	Ya	0/7 Karakter
Ke-6	Ya	Ya	0/7 Karakter
Ke-7	Ya	Ya	0/7 Karakter
Ke-8	Ya	Ya	0/7 Karakter
Ke-9	Ya	Ya	0/7 Karakter
Ke-10	Ya	Ya	0/8 Karakter
<b>Total Akurasi</b>	100%	100%	1,43%

Berdasarkan hasil tabel 4.3, hasil deteksi dan identifikasi pada mobil ke-4 memiliki 1 kesalahan identifikasi yang disebabkan oleh kondisi karakter bagian tengah pada nomor registrasi plat mobil tersebut tidak rata sehingga terjadi kesalahan identifikasi yang dapat dilihat pada gambar 4.8. Selain itu, deteksi dan identifikasi nomor registrasi plat mobil lainnya berhasil dilakukan tanpa kesalahan. Gambar 4.12 adalah salah satu contoh hasil deteksi dan identifikasi nomor registrasi plat kendaraan indonesia bagian depan menggunakan foto tanpa kesalahan.



**Gambar 4.8:** Karakter Bagian Tengah Pada Nomor Registrasi Plat Tidak Rata.



**Gambar 4.9:** Deteksi dan Identifikasi Nomor Registrasi Plat Bagian Depan Menggunakan Input Foto Tanpa Kesalahan.

## 4.2.2 Pengujian Deteksi dan Identifikasi Nomor Registrasi Plat Kendaraan Indonesia Bagian Belakang Menggunakan *Input* Foto

**Tabel 4.4:** Tabel Pengujian Deteksi dan Identifikasi Nomor Registrasi Plat Kendaraan Indonesia Bagian Belakang Menggunakan *Input* Foto

Mobil	Deteksi	Identifikasi	Kesalahan
Ke-1	Ya	Ya	1/7 Karakter
Ke-2	Ya	Ya	0/7 Karakter
Ke-3	Ya	Ya	1/8 Karakter
Ke-4	Ya	Ya	0/8 Karakter
Ke-5	Ya	Ya	1/8 Karakter
Ke-6	Ya	Ya	1/7 Karakter
Ke-7	Ya	Ya	0/8 Karakter
Ke-8	Ya	Ya	0/7 Karakter
Ke-9	Ya	Ya	1/7 Karakter
Ke-10	Ya	Ya	0/7 Karakter
<b>Total Akurasi</b>	100%	100%	5,4%

Berdasarkan hasil tabel 4.4, hasil deteksi dan identifikasi terdapat beberapa kesalahan yang disebabkan oleh sulitnya membedakan beberapa karakter yang mirip misalnya huruf "D" dan "0" yang dapat dilihat pada gambar 4.10 . Selain itu, kesalahan tersebut disebabkan oleh kondisi nomor registrasi plat mobil yang kurang bagus sehingga tidak teridentifikasi yang dapat dilihat pada gambar 4.11. Selain itu, deteksi dan identifikasi nomor registrasi plat mobil lain-

nya berhasil dilakukan tanpa kesalahan. Gambar 4.12 adalah salah satu contoh hasil deteksi dan identifikasi nomor registrasi plat kendaraan indonesia bagian belakang menggunakan input foto tanpa kesalahan.



**Gambar 4.10:** Kesalahan Identifikasi Karakter.



**Gambar 4.11:** Kondisi Nomor Registrasi Plat Yang Kurang Bagus.



**Gambar 4.12:** Deteksi dan Identifikasi Nomor Registrasi Plat Bagian Belakang Menggunakan Input Foto Tanpa Kesalahan.



### 4.2.3 Pengujian Deteksi dan Identifikasi Nomor Registrasi Plat Kendaraan Indonesia Bagian Depan Menggunakan *Input Video*

**Tabel 4.5:** Tabel Pengujian Deteksi dan Identifikasi Nomor Registrasi Plat Kendaraan Indonesia Bagian Depan Menggunakan *Input Video*

Mobil	Deteksi	Identifikasi	Kesalahan
Ke-1	Ya	Ya	0/7 Karakter
Ke-2	Ya	Ya	0/7 Karakter
Ke-3	Ya	Ya	0/7 Karakter
Ke-4	Ya	Ya	0/7 Karakter
Ke-5	Ya	Tidak	7/7 Karakter
Ke-6	Ya	Ya	0/7 Karakter
Ke-7	Ya	Ya	0/7 Karakter
Ke-8	Ya	Ya	0/7 Karakter
Ke-9	Ya	Ya	0/7 Karakter
Ke-10	Ya	Ya	0/7 Karakter
<b>Total Akurasi</b>	100%	90%	10%

Berdasarkan hasil tabel 4.5, hasil deteksi dan identifikasi terhadap mobile ke-5 pada nomor registrasi platnya dapat dideteksi tetapi tidak dapat diidentifikasi. Hal tersebut disebabkan oleh hasil deteksi yang tidak dapat membedakan karakter awal dan akhir dari nomor registrasi plat mobil tersebut yang dapat dilihat pada gambar 4.13. Selain itu, deteksi dan identifikasi nomor registrasi plat mobil lainnya berhasil dilakukan tanpa kesalahan. Gambar 4.14 adalah

salah satu contoh hasil deteksi dan identifikasi nomor registrasi plat kendaraan indonesia bagian depan menggunakan input video tanpa kesalahan.



**Gambar 4.13:** Hasil Deteksi Yang Tidak Dapat Membedakan Karakter Awal dan Akhir.



**Gambar 4.14:** Deteksi dan Identifikasi Nomor Registrasi Plat Bagian Depan Menggunakan Input Video Tanpa Kesalahan.

**Tabel 4.6:** Tabel Pengujian Deteksi dan Identifikasi Nomor Registrasi Plat Kendaraan Indonesia Bagian Belakang Menggunakan *Input* Video

Mobil	Deteksi	Identifikasi	Kesalahan
Ke-1	Ya	Ya	3/7 Karakter
Ke-2	Ya	Ya	2/7 Karakter
Ke-3	Ya	Ya	3/7 Karakter
Ke-4	Ya	Tidak	7/7 Karakter
Ke-5	Ya	Ya	0/7 Karakter
Ke-6	Ya	Tidak	7/7 Karakter
Ke-7	Ya	Tidak	7/7 Karakter
Ke-8	Ya	Ya	1/7 Karakter
Ke-9	Ya	Tidak	7/7 Karakter
Ke-10	Ya	Ya	1/7 Karakter
<b>Total Akurasi</b>	100%	60%	54.29%

#### 4.2.4 Pengujian Deteksi dan Identifikasi Nomor Registrasi Plat Kendaraan Indonesia Bagian Belakang Menggunakan *Input* Video

Berdasarkan hasil tabel 4.6, hasil deteksi dan identifikasi terdapat beberapa mobil yang tidak diidentifikasi, serta memiliki kesalahan rekognisi karakter yang banyak. Mobil yang tidak bisa diidentifikasi disebabkan oleh hasil deteksi yang hanya mendeteksi salah satu dari karakter awal dan akhir dari nomor registrasi plat mobil tersebut yang dapat dilihat pada gambar 4.15. Kesalahan rekognisi karakter yang banyak disebabkan oleh nomor registrasi plat mobil tersebut terlihat kabur atau tidak terlihat jelas oleh kamera sehing-

ga hasil identifikasinya menjadi tidak sesuai seperti pada gambar 4.16. Selain itu hal tersebut juga disebabkan oleh posisi kamera yang cukup jauh dengan nomor registrasi plat kendaraan sehingga hasil deteksi yang diperoleh tidak sesuai misalnya miring atau terdapat beberapa bagian karakter tersebut yang tidak terdeteksi yang dapat menyebabkan tidak dapat teridentifikasi karena hasil *perspective image* menjadi tidak sesuai seperti pada gambar 4.17. Namun, terdapat 1 mobil yang dapat terdeteksi dan teridentifikasi dengan baik tanpa kesalahan yaitu pada mobil ke-5 yang dapat dilihat pada gambar 4.18



**Gambar 4.15:** Hasil Deteksi Yang Hanya Mendeteksi Salah Satu Dari Karakter Awal dan Akhir.



**Gambar 4.16:** Nomor Registrasi Plat Tidak Terlihat Jelas Oleh Kamera.



**Gambar 4.17:** Beberapa Bagian Karakter Tidak Terdeteksi.



**Gambar 4.18:** Deteksi dan Identifikasi Nomor Registrasi Plat Bagian Belakang Menggunakan Input Video Tanpa Kesalahan.

#### 4.2.5 Pengujian Deteksi dan Identifikasi Nomor Registrasi Plat Kendaraan Indonesia dengan Perbedaan Pemilihan Ketinggian dan Sudut Kamera

Pengujian ini dilakukan dengan memberikan *input* video dengan mengatur posisi ketinggian yang berbeda-beda menggunakan tripod yang dapat disesuaikan ketinggiannya, yang digunakan sebagai penyangga kamera serta dikombinasikan dengan perubahan sudut untuk melakukan proses pengambilan citra video dan dijalankan dengan sistem yang telah dibuat.

Pengujian dilakukan dengan menggunakan kamera Canon EOS M10 dengan spesifikasi yang terdapat pada tabel 4.1 Proses perekaman video dilakukan menggunakan resolusi full HD 1080p 30 fps dengan menggantikan berbagai posisi sudut dan ketinggian sesuai dengan tabel 4.7 dengan mengatur tinggi tiang 1 meter dan 1,1 meter. Pengambilan video dilakukan dengan perputaran sudut kamera dari 90 derajat hingga 80 derajat.

**Tabel 4.7:** Tabel Pengujian dengan Perbedaan Pemilihan Ketinggian dan Sudut Camera.

Ketinggian	Sudut	Deteksi	Identifikasi	Kesalahan
1 Meter	90°	4/5	4/5	7/34 Karakter
	85°	5/5	5/5	0/35 Karakter
	80°	5/5	3/5	14/35 Karakter
1.1 Meter	90°	3/5	1/5	30/36 Karakter
	85°	5/5	3/5	19/36 Karakter
	80°	5/5	4/5	10/36 Karakter
<b>Total Akurasi</b>		90%	66,67%	37,74%



**Gambar 4.19:** Nomor Registrasi Plat Kurang Terlihat Jelas Oleh Kamera.



**Gambar 4.20:** Hasil Deteksi Tidak Sesuai.



**Gambar 4.21:** Kesalahan Rekognisi Karakter karena Cetakn Nomor Registrasi Plat Tidak Lurus.





**Gambar 4.22:** Deteksi dan Identifikasi Nomor Registrasi Plat Kendaraan Dengan Perbedaan Kamera, Tinggi, dan Sudut Kamera Tanpa Kesalahan.

Berdasarkan hasil tabel 4.7, terdapat beberapa mobil yang tidak dapat terdeteksi dan teridentifikasi. Hal tersebut karena nomor registrasi plat mobil tersebut tidak terlihat atau kurang terlihat jelas oleh kamera seperti pada gambar 4.19. Hal tersebut disebabkan beberapa faktor misalnya kurangnya cahaya pada nomor registrasi plat dan jarak mobil yang cukup jauh dari kamera. Selain itu, terdapat nomor registrasi plat kendaraan yang sudah terdeteksi namun tidak teridentifikasi. Hal tersebut karena hasil deteksi tidak sesuai yang hanya mendeteksi salah satu dari karakter awal atau akhir dari nomor registrasi plat mobil tersebut seperti pada gambar 4.20. Kesalahan deteksi tersebut disebabkan kurangnya variasi dataset khususnya dataset di area tersebut. Selain itu, berdasarkan hasil tabel 4.7 terdapat kesalahan 1 atau 2 rekognisi karakter. Hal itu disebabkan oleh nomor registrasi plat kendaraan Indonesia yang bervariasi misalnya cetakan nomor registrasi plat tersebut tidak lurus seperti pada gambar 4.21. Selain itu, deteksi dan identifikasi nomor registrasi plat mobil lainnya berhasil dilakukan tanpa kesalahan sehingga perbedaan kamera, tinggi maupun sudut tidak berpengaruh

selama nomor registrasi plat kendaraan terlihat jelas oleh kamera. Gambar 4.22 adalah salah satu contoh hasil deteksi dan identifikasi nomor registrasi plat kendaraan Indonesia menggunakan perbedaan kamera, tinggi dan sudut tanpa kesalahan.

### 4.3 Pengujian Deteksi dan Identifikasi Nomor Registrasi Plat Kendaraan Indonesia Pada Malam Hari

Berdasarkan hasil uji coba yang telah dilakukan di pintu masuk ITS, nomor registrasi plat mobil tidak dapat dideteksi dan diidentifikasi karena nomor registrasi plat terlihat silau, samar-samar, bahkan tidak terlihat. Hal tersebut disebabkan karena pencahayaan pada malam hari di daerah tersebut kurang bagus yang menyebabkan nomor registrasi plat tidak terlihat oleh kamera. Selain kondisi cahaya, keterbatasan perangkat kamera yang dimiliki kurang memadai karena tidak adanya fitur-fitur khusus untuk malam hari. Gambar 4.23 adalah salah satu contoh uji coba yang dilakukan dimana nomor registrasi plat mobil tidak terlihat jelas oleh kamera.



**Gambar 4.23:** Nomor Registrasi Plat Kendaraan Tidak Terlihat Jelas Pada Malam Hari.

### 4.4 Pengujian Kecepatan Komputasi Sistem

Pengujian kecepatan komputasi sistem bertujuan untuk mengetahui kecepatan komputasi sistem dalam mendeteksi dan mengidentifikasi nomor registrasi plat kendaraan Indonesia. Pengujian ini

**Tabel 4.8:** Spesifikasi ASUS ROG GL503GE[32]

Category	Esports Gaming
Operating System	Windows 10 Home
Processor	Intel Core i7-8750H
Graphics	NVIDIA GeForce GTX 1050 Ti
	4GB GDDR5 VRAM
Memory	8GB Up to DDR4 2666MHz SDRAM (Dual-channel)
	*8GB Up to 16GB
SSD	M.2 NVMe PCIe 3.0 256GB SSD
HDD	2.5" 5400rpm 1TB SSHD
Display	15.6-inch Full HD (1920x1080) TN panel, 120Hz, 3ms, 94% NTSC
VR Ready	No
Peripheral Recommended	ROG Strix Impact and ROG Gladius II gaming mice are designed for marathon gaming sessions.
Power	4 Cells 64 Whrs Battery

dilakukan dengan menggunakan Laptop ASUS ROG GL503GE dengan spesifikasi yang terdapat pada tabel 4.8.

Berdasarkan hasil tabel 4.9, terdapat hasil running time yang melebihi 1 detik yang disebabkan oleh tingkat kesulitan dalam mendeteksi dan mengidentifikasi nomor registrasi plat yang disebabkan beberapa faktor misalnya membutuhkan waktu yang lebih lama dalam melakukan proses perspective transform karena foto nomor registrasi plat kendaraan tersebut terlalu miring. Berdasarkan hasil tabel 4.10, nilai rata-rata waktu deteksi dan identifikasi tidak dipengaruhi oleh banyaknya jumlah frame. Dengan spesifikasi perangkat yang digunakan dapat mendeteksi dan mengidentifikasi dengan rata-rata kurang dari 1 detik terhadap foto maupun video sehingga sudah layak digunakan ke berbagai sistem lainnya. Hasil rata-rata kecepatan komputasi sistem terhadap video lebih cepat daripada terhadap foto karena terdapat kondisi dimana tidak adanya kenda-

**Tabel 4.9:** Kecepatan Komputasi Sistem Terhadap Foto

Mobil	Running Time (s)
Ke-1	1,256
Ke-2	0,424
Ke-3	0,745
Ke-4	0,406
Ke-5	0,728
Ke-6	0,728
Ke-7	0,732
Ke-8	0,398
Ke-9	0,741
Ke-10	0,747
Rata-rata	0,6905

**Tabel 4.10:** Kecepatan Komputasi Sistem Terhadap Video

Frame	Rata-rata Waktu (s/frame)	Running Time (s)
3668	0,047846	296,637
7612	0,064928825	763.789

raan pada frame tersebut sehingga tidak dilakukan proses deteksi dan identifikasi. Untuk meningkatkan waktu deteksi dan identifikasi menjadi lebih cepat dibutuhkan perangkat yang memiliki spesifikasi yang lebih tinggi.

## 4.5 Pengujian Menggunakan Perbedaan Variasi Dataset

Pengujian menggunakan perbedaan variasi dilakukan untuk mengetahui apakah ada pengaruh hasil deteksi terhadap penggunaan variasi dataset yang berbeda. Pengujian ini dilakukan dengan cara melakukan training dan deteksi menggunakan variasi dataset yang berbeda. Dataset-1 diperoleh dari pengambilan foto kendaraan yang ada di parkir dan pintu masuk ITS. Dataset-2 diperoleh dari pengambilan foto kendaraan yang ada di pintu masuk ITS saja. Pengujian perbedaan variasi dataset-1 di parkir dapat dilihat pada tabel 4.3 dan pengujian dataset-1 dan dataset-2 di pintu masuk

**Tabel 4.11:** Tabel Pengujian Perbedaan Variasi Dataset-2 di Parkiran

Mobil	Deteksi	Identifikasi	Kesalahan
Ke-1	Ya	Tidak	6/6 Karakter
Ke-2	Tidak	Tidak	7/7 Karakter
Ke-3	Tidak	Tidak	8/8 Karakter
Ke-4	Ya	Tidak	6/6 Karakter
Ke-5	Ya	Tidak	7/7 Karakter
Ke-6	Ya	Tidak	7/7 Karakter
Ke-7	Ya	Tidak	7/7 Karakter
Ke-8	Tidak	Tidak	7/7 Karakter
Ke-9	Tidak	Tidak	7/7 Karakter
Ke-10	Tidak	Tidak	8/8 Karakter
<b>Total Akurasi</b>	50%	0%	100%

ITS dapat dilihat pada tabel 4.5. Pengujian menggunakan dataset-2 di parkiran dapat dilihat pada tabel 4.11.

Berdasarkan hasil tabel 4.3 dan 4.5, kendaraan dapat dideteksi dan diidentifikasi dengan baik karena didalam dataset-1 terdapat data yang diambil di parkiran dan pintu masuk ITS. Sedangkan pada hasil dari tabel 4.11, kendaraan tidak bisa dideteksi sama sekali. Hal tersebut disebabkan karena pada dataset-2 tidak terdapat data yang diambil di parkiran dan variasi dataset kurang. Agar dapat mendeteksi dan mengidentifikasi menggunakan dataset-2, diperlukan penambahan dataset yang diambil di parkiran dan menambahkan variasi dataset kendaraan.

*Halaman ini sengaja dikosongkan*

# BAB 5

## PENUTUP

### 5.1 Kesimpulan

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

1. Pada penelitian ini berhasil mendeteksi dan mengidentifikasi nomor registrasi plat kendaraan Indonesia berbasis Convolutional Neural Network (CNN) yang telah diuji coba di pintu masuk ITS.
2. Berdasarkan hasil pengujian deteksi dan identifikasi bagian depan, dengan menggunakan input foto memiliki kesalahan hanya 1,43% dan dengan menggunakan input video memiliki kesalahan 10%.
3. Berdasarkan hasil pengujian deteksi dan identifikasi bagian belakang, dengan menggunakan input foto memiliki ke 5,4% dan dengan menggunakan input video memiliki kesalahan yang cukup tinggi yaitu 54,29%. Hal tersebut disebabkan karena kurangnya jumlah dan variasi dataset pada bagian belakang kendaraan serta posisi kamera atau kendaraan yang kurang bagus.
4. Berdasarkan hasil pengujian perbedaan ketinggian dan sudut kamera didapatkan kesalahan yang cukup tinggi yaitu 37,74%. Posisi kamera yang semakin tinggi menyebabkan nomor registrasi plat kendaraan tidak terlihat jelas oleh kamera.
5. Berdasarkan pengujian kecepatan komputasi sistem, kecepatan komputasi tertinggi terhadap foto mencapai 1,256 s sedangkan rata-rata kecepatan komputasi tertinggi terhadap video yaitu 0,064928825 s. Untuk meningkatkan kecepatan komputasi tersebut dibutuhkan perangkat dengan spesifikasi yang lebih tinggi.
6. Berdasarkan hasil pengujian perbedaan variasi dataset, kesalahan pada dataset-2 yang diuji coba di parkir mencapai 100% yang disebabkan oleh pada dataset-2 tidak memiliki data yang diambil di parkir.

## 5.2 Saran

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

1. Menambah dataset seluruh variasi kendaraan yang ada di Indonesia.
2. Menggunakan kamera dengan spesifikasi yang tinggi agar nomor registrasi plat kendaraan dapat terlihat lebih jelas.
3. Menggunakan spesifikasi perangkat yang tinggi untuk memproses sistem agar proses yang dilakukan dapat berjalan lebih cepat.
4. Melanjutkan penelitian menjadi sebuah sistem yang telah diaplikasikan.



## DAFTAR PUSTAKA

- [1] R. Kusumanto and A. N. Tompunu, “pengolahan citra digital untuk mendeteksi obyek menggunakan pengolahan warna model normalisasi rgb,” Semantik, vol. 1, no. 1, 2011. (Dikutip pada halaman 1, 10).
- [2] R. Laroca, E. Severo, L. A. Zanlorensi, L. S. Oliveira, G. R. Gonçalves, W. R. Schwartz, and D. Menotti, “A robust real-time automatic license plate recognition based on the yolo detector,” in 2018 International Joint Conference on Neural Networks (IJCNN), pp. 1–10, IEEE, 2018. (Dikutip pada halaman 1).
- [3] P. Shivakumara, D. Tang, M. Asadzadehkaljahi, T. Lu, U. Pal, and M. H. Anisi, “Cnn-rnn based method for license plate recognition,” CAAI Transactions on Intelligence Technology, vol. 3, no. 3, pp. 169–175, 2018. (Dikutip pada halaman 1).
- [4] Y. S. Hariyani and I. Wijayanto, “Indonesian vehicle plate recognition and identification based on digital image processing and artificial neural network,” ICSIT 2012, p. 83, 2012. (Dikutip pada halaman 1).
- [5] W. Rohimah, “Pengenalan plat nomor kendaraan indonesia menggunakan principal component analysis dan metode k-nearest neighbor,” 2017. (Dikutip pada halaman 1, 5).
- [6] I. Putra, “Klasifikasi citra menggunakan convolutional neural network (cnn) pada caltech 101,” 2016. (Dikutip pada halaman 2).
- [7] J. W. H. Sri, “Deteksi kendaraan secara real-time menggunakan metode yolo berbasis android,” DETEKSI KENDARAAN SECARA REAL-TIME MENGGUNAKAN METODE YOLO BERBASIS ANDROID. (Dikutip pada halaman 6).
- [8] K. Ariasa, I. M. D. Susila, K. Budiarta, and T. K. S. S. BALI, “Aplikasi tilang dengan pengenalan plat nomor kendaraan dan pelaku pada platform mobile,” (Dikutip pada halaman 6).

- [9] E. N. Arrofiqoh and H. Harintaka, “Implementasi metode convolutional neural network untuk klasifikasi tanaman pada citra resolusi tinggi,” GEOMATIKA, vol. 24, no. 2, pp. 61–68, 2018. (Dikutip pada halaman 6).
- [10] S. E. Limantoro, Y. Kristian, and D. D. Purwanto, “Deteksi pengendara sepeda motor menggunakan deep convolutional neural networks,” in Seminar Nasional Teknologi Informasi dan Komunikasi, pp. 79–86, 2017. (Dikutip pada halaman 7).
- [11] S. Jupiyandi, F. R. Saniputra, Y. Pratama, M. R. Dharmawan, and I. Cholissodin, “Pengembangan deteksi citra mobil untuk mengetahui jumlah tempat parkir menggunakan cuda dan modified yolo,” Jurnal Teknologi Informasi dan Ilmu Komputer, vol. 6, no. 4, pp. 413–419, 2019. (Dikutip pada halaman 7).
- [12] K. A. Shianto, K. Gunadi, and E. Setyati, “Deteksi jenis mobil menggunakan metode yolo dan faster r-cnn,” Jurnal Infra, vol. 7, no. 1, pp. 157–163, 2019. (Dikutip pada halaman 8).
- [13] M. Muhtadii and H. A. Tawakal, “Pengembangan aplikasi android untuk pengenalan citra nomor sertifikat halal mui dengan library tesseract optical character recognition (ocr),” Jurnal Informatika Terpadu (JIT), vol. 2, no. 1, 2016. (Dikutip pada halaman 8).
- [14] Y. MANSYUR, “Optical character recognition untuk deteksi pelat mobil dan motor kendaraan pada kampus teknik gowa,” (Dikutip pada halaman 8).
- [15] W. Satyawan, M. O. Pratama, R. Jannati, G. Muhammad, B. Fajar, H. Hamzah, R. Fikri, and K. Kristian, “Citizen id card detection using image processing and optical character recognition,” in Journal of Physics: Conference Series, vol. 1235, p. 012049, IOP Publishing, 2019. (Dikutip pada halaman 9).
- [16] C.-N. E. Anagnostopoulos, “License plate recognition: A brief tutorial,” IEEE Intelligent transportation systems magazine, vol. 6, no. 1, pp. 59–67, 2014. (Dikutip pada halaman 9).

- [17] 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 12).
- [18] I. Goodfellow, Y. Bengio, and A. Courville, Deep learning. MIT press, 2016. (Dikutip pada halaman 13).
- [19] S. R. Dewi et al., “Deep learning object detection pada video menggunakan tensorflow dan convolutional neural network,” 2018. (Dikutip pada halaman 13).
- [20] S. Sena, “Pengenalan deep learningpart 7: Convolutional neural network(cnn),” 2017. (Dikutip pada halaman xi, 14, 15, 16).
- [21] T. Nurhikmat et al., “Implementasi deep learning untuk image classification menggunakan algoritma convolutional neural network (cnn) pada citra wayang golek,” 2018. (Dikutip pada halaman 13, 15, 16).
- [22] M. Rivki and A. M. Bachtiar, “Implementasi algoritma k-nearest neighbor dalam pengklasifikasian follower twitter yang menggunakan bahasa indonesia,” Jurnal Sistem Informasi, vol. 13, no. 1, pp. 31–37, 2017. (Dikutip pada halaman 17).
- [23] A. Sukma, D. Ramadhan, B. P. Santoso, T. R. Sari, and N. Wiraswari, “K-nearest neighbor information retrieval (sistem temu kembali informasi),” Universitas Airlangga Fakultas Sains dan Teknologi, Surabaya, 2014. (Dikutip pada halaman 17).
- [24] 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 18, 19, 20, 21).
- [25] F. Mohammad, J. Anarase, M. Shingote, and P. Ghanwat, “Optical character recognition implementation using pattern matching,” International Journal of Computer Science and Information Technologies, vol. 5, no. 2, pp. 2088–2090, 2014. (Dikutip pada halaman 20).

- [26] C. Patel, A. Patel, and D. Patel, “Optical character recognition by open source ocr tool tesseract: A case study,” International Journal of Computer Applications, vol. 55, no. 10, pp. 50–56, 2012. (Dikutip pada halaman xi, 22, 35, 36).
- [27] J. Redmon and A. Farhadi, “Yolov3: An incremental improvement,” arXiv preprint arXiv:1804.02767, 2018. (Dikutip pada halaman 29, 30).
- [28] OpenCv, “Geometric image transformations.” [https://docs.opencv.org/2.4/modules/imgproc/doc/geometric\\_transformations.html](https://docs.opencv.org/2.4/modules/imgproc/doc/geometric_transformations.html). Diakses: 2018-11-22. (Dikutip pada halaman 34).
- [29] G. T. Data, “Building a naïve ocr system,” (Dikutip pada halaman 35).
- [30] “Spesifikasi canon eos m10.” <https://global.canon/en/c-museum/product/dslr835.html>. Diakses: 2019-11-22. (Dikutip pada halaman xiii, 41).
- [31] “Spesifikasi samsung a70.” <https://www.samsung.com/id/smartphones/galaxy-a70-a705/SM-A705FZBGXID/>. Diakses: 2019-11-22. (Dikutip pada halaman xiii, 42).
- [32] “Spesifikasi asus rog gl503ge.” <https://www.asus.com/id/ROG-Republic-Of-Gamers/ROG-Strix-GL503/Tech-Specs/>. Diakses: 2019-12-22. (Dikutip pada halaman xiii, 57).

## BIOGRAFI PENULIS



Muh. Shahid Setiawan, lahir pada 26 Juni 1996 di Kota Kendari, Provinsi Sulawesi Tenggara. Penulis merupakan anak pertama dari tiga bersaudara. Saat ini penulis tinggal di Kota Kendari, Sulawesi Tenggara. Pada tahun 2008 menyelesaikan pendidikan di SD Negeri 10 Poasia. Tahun 2011 lulus dari SMP Negeri 1 Kendari. Dan pada tahun 2014 lulus dari SMA Negeri 4 Kendari. Penulis diterima di Program Studi S-1 Departemen Teknik Komputer Fakultas Teknologi Elektro ITS pada tahun 2015. Penulis pernah aktif mejadi Bendahara KALAM Elektro ITS tahun 2017-2018 serta menjadi Ketua E-Sport Competition MAGE 2018. Penulis memiliki hobi dalam bermain *game online*. Penulis berhasil menyelesaikan tugas akhir dengan judul "**Deteksi dan Identifikasi Nomor Registrasi Plat Kendaraan Indonesia Berbasis *Convolutional Neural Network* (CNN)**". Bagi pembaca yang memiliki kritik, saran, atau pertanyaan mengenai tugas akhir ini dapat menghubungi penulis melalui surel [muhammadshahidsetiawan@gmail.com](mailto:muhammadshahidsetiawan@gmail.com).

*Halaman ini sengaja dikosongkan*