



PENGEMBANGAN METODE DETEKSI DAN MENGHITUNG
JUMLAH POHON KELAPA SAWIT DARI SENTINEL 2
IMAGERY MENGGUNAKAN METODE *OBJECT-BASED*
IMAGE ANALYSIS (OBIA)

DISERTASI

GUNTUR EKA SAPUTRA

99219009

PROGRAM DOKTOR TEKNOLOGI INFORMASI

UNIVERSITAS GUNADARMA

2023



**Pengembangan Metode Deteksi Dan Menghitung Jumlah Pohon
Kelapa Sawit Dari Sentinel 2 Imagery Menggunakan Metode
*OBJECT-BASED IMAGE ANALYSIS (OBIA)***

DISERTASI

Untuk Memenuhi Salah Satu Syarat Meraih Gelar Doktor Teknologi
Informasi di bawah Pimpinan Rektor Universitas Gunadarma
Profesor Doktor E.S. Margianti, SE, MM

Laporan Rapat Komisi Pembimbing Dipertahankan dalam Sidang Terbuka
Senat Universitas Gunadarma
Pada Hari Rabu, 10 Mei 2023

GUNTUR EKA SAPUTRA

99219009

PROGRAM DOKTOR TEKNOLOGI INFORMASI

UNIVERSITAS GUNADARMA

2023

PENGEMBANGAN METODE DETEKSI DAN MENGHITUNG
JUMLAH POHON KELAPA SAWIT DARI SENTINEL 2
IMAGERY MENGGUNAKAN METODE *OBJECT-BASED*
IMAGE ANALYSIS (OBIA)

DISERTASI

GUNTUR EKA SAPUTRA

Telah disetujui oleh:

Profesor Doktor Insinyur Kudang Boro Seminar, M.Sc.
Promotor

Profesor Doktor Sarifuddin Madenda, S.Si., DEA.
Ko-Promotor

Jakarta, 10 Mei 2023

Judul Disertasi : **PENGEMBANGAN METODE DETEKSI DAN MENGHITUNG JUMLAH POHON KELAPA SAWIT DARI SENTINEL 2 IMAGERY MENGGUNAKAN METODE *OBJECT-BASED IMAGE ANALYSIS* (OBIA)**

Nama Mahasiswa : Guntur Eka Saputra
NIM : 99219009

Komite Pembimbing

Promotor : Profesor Doktor Insinyur Kudang Boro Seminar, M.Sc.
Ko-Promotor : Profesor Doktor Sarifuddin Madenda, S.Si., DEA.

Komite Penguji

Ketua : Profesor Doktor Insinyur Kudang Boro Seminar, M.Sc.

Anggota : Profesor Doktor Insinyur Sudrajat, M.Sc.
Profesor Doktor E. S. Margianti, S.E., M.M.
Profesor Suryadi Harmanto, S.Si., M.M.S.I.
Profesor Doktor Insinyur Bambang Suryawan, M.T.
Profesor Insinyur Busono Soerowirdjo, M.Sc., Ph.D.
Profesor Doktor Eri Prasetyo Wibowo, S.Si., M.M.S.I.
Doktor rer. nat. I Made Wiryantha, S.Si., S.Kom.,
M.App.Sc.
Doktor Detty Purnamasari, S.Kom., M.M.S.I.,
M.I.Kom.
Profesor Doktor Sarifuddin Madenda, S.Si., D.E.A.

PERNYATAAN ORIGINALITAS DAN PUBLIKASI

Saya yang bertanda tangan di bawah ini:

Nama : Guntur Eka Saputra

NIM : 99219009

Judul Disertasi : PENGEMBANGAN METODE DETEKSI DAN
MENGHITUNG JUMLAH POHON KELAPA SAWIT
DARI SENTINEL 2 IMAGERY MENGGUNAKAN
METODE *OBJECT-BASED IMAGE ANALYSIS*
(OBIA)

Tanggal Sidang : 10 Mei 2023

Tanggal Lulus :

Menyatakan bahwa tulisan ini adalah merupakan hasil karya saya sendiri dan dapat dipublikasikan sepenuhnya oleh Universitas Gunadarma. Segala kutipan dalam bentuk apapun telah mengikuti kaidah dan etika yang berlaku. Mengenai sisi dan tulisan adalah merupakan tanggung jawab Penulis, bukan Universitas Gunadarma.

Demikian pernyataan ini dibuat dengan sebenarnya dan dengan penuh kesadaran.

Jakarta, 10 Mei 2023

(Guntur Eka Saputra)

ABSTRAK

Penerapan teknologi informasi dan komunikasi sangat dibutuhkan, khususnya di bidang pertanian. Kelapa sawit merupakan produk pertanian yang terbesar di Indonesia dan produksi kelapa sawit sangat penting bagi perekonomian. Dalam menghasilkan produksi kelapa sawit yang baik dibutuhkan pengelolaan perkebunan kelapa sawit yang baik. Permasalahan utama dalam pengelolaan ini, yaitu luas area dalam skala besar (perusahaan perkebunan kelapa sawit memiliki luas minimum sebesar 6.000 hektar yang harus dikelola), wilayah perkebunan berada di *remote area*, akses infrastruktur yang terbatas, dan pemupukan presisi mengalami kesulitan untuk mendapatkan data secara akurat berdasarkan jumlah tegakkan pohon kelapa sawit pada suatu area lahan. Hal ini menyebabkan deteksi dan menghitung pohon kelapa sawit sangat dibutuhkan. Selama ini, penghitungan tradisional didasarkan pada catatan awal penanaman pohon kelapa sawit atau penghitungan teoritis berdasarkan jarak tanam antara pohon kelapa sawit dalam satu hektar atau blok. Metode tradisional ini lambat dan tidak akurat, serta tidak diketahui status pohon kelapa sawit yang rusak atau mati. Penggunaan teknologi dibutuhkan untuk dapat secara otomatis dan *real-time* dalam memonitoring data pohon kelapa sawit dan memperkirakan produktivitasnya. Penggunaan ini dibutuhkan dalam deteksi objek berupa citra.

Penggunaan citra untuk deteksi objek dibutuhkan dalam persiapan data, seperti menganalisa, memberikan anotasi atau kelas dari objek tersebut. Metode dalam melakukan anotasi dataset selama ini dilakukan secara manual, satu per satu dengan memberikan kotak batas. Persiapan data ini menghabiskan lebih dari 70% waktu dalam siklus hidup *deep learning* untuk menjadi dataset yang dapat digunakan sebagai data pelatihan, validasi, dan pengujian. Hal inilah yang menjadi tantangan bagi *stakeholders*.

Penelitian ini bertujuan untuk menghasilkan pengembangan metode *object-based image analysis* (OBIA) untuk membuat dataset secara otomatis dengan memberikan label suatu kelas pada data citra. Metode yang dikembangkan menggunakan algoritma klasifikasi *template matching*. Algoritma ini sebagai template awal citra pohon kelapa sawit yang memiliki kunci nilai ambang batas dalam menentukan kelas dari objek di dalam citra. Algoritma BIRCH digunakan untuk mengurangi objek yang bukan terdeteksi ke dalam kelas pohon kelapa sawit. Hasil evaluasi performance pelatihan menunjukkan bahwa model dengan algoritma YOLOv7 lebih baik dengan akurasi best MAP sebesar 0,993 dan pada pengujian sebesar 0,997. Berdasarkan waktu pemrosesan DGX-A-100 Universitas Gunadarma lebih baik, pada pelatihan sebesar 2948 detik dibandingkan dengan Google Colab Pro sebesar 4847 detik.

Penelitian ini dihasilkan purwarupa sistem yang menggunakan model algoritma dari YOLOv7 untuk dapat mendeteksi dan menghitung pohon kelapa sawit pada area tertentu dari citra satelit yang terintegrasi dengan Google Maps API. Berdasarkan hasil pengujian 4 blok pada Kebun Pendidikan dan Pendidikan Kelapa Sawit IPB-Cargil bahwa hasil presentasi berhasil dideteksi

sebesar 97,67%, dan diketahui setiap pohon kelapa sawit yang terdeteksi diketahui letak titik koordinat untuk dapat dilakukan pemantauan, pengelolaan, dan estimasi produktivitas pohon kelapa sawit.

Kata kunci : OBIA, *Deep Learning*, YOLOv7, Kelapa Sawit.

ABSTRACT

The application of information and communication technology is needed, especially in agriculture. Palm oil is the largest agricultural product in Indonesia and palm oil production is very important for the economy. In producing good oil palm production, good oil palm plantation management is needed. The main problems in this management, namely large-scale areas (oil palm plantation companies have a minimum area of 6,000 hectares that must be managed), plantation areas in remote areas, limited infrastructure access, and precision fertilization have difficulty obtaining accurate data based on the number of standing oil palm trees in a land area. This makes detecting and counting oil palm trees necessary. So far, traditional counting has been based on early records of oil palm tree planting or theoretical counting based on the spacing between oil palm trees within a hectare or block. These traditional methods are slow and inaccurate, and the status of damaged or dead oil palm trees is unknown. The use of technology is needed to be able to automatically and real-time monitor palm oil tree data and estimate its productivity. This use is needed in object detection in the form of images.

The use of images for object detection is needed in data preparation, such as analyzing, annotating or classifying the object. The method of annotating datasets has been done manually, one by one by providing boundary boxes. This data preparation consumes more than 70% of the time in the deep learning lifecycle to become a dataset that can be used as training, validation, and testing data. This is the challenge for stakeholders.

This research aims to produce the development of object-based image analysis (OBIA) method to create datasets automatically by labeling a class on image data. The developed method uses a template matching classification algorithm. This algorithm as an initial template of palm tree image that has a key threshold value in determining the class of objects in the image. The BIRCH algorithm is used to reduce objects that are not detected into the palm tree class. The results of the training performance evaluation show that the model with the YOLOv7 algorithm is better with a best MAP accuracy of 0.993 and on testing of 0.997. Based on processing time, Gunadarma University's DGX-A-100 is better, at 2948 seconds of training compared to Google Colab Pro at 4847 seconds.

This research produced a prototype system that uses the algorithm model from YOLOv7 to be able to detect and count oil palm trees in a certain area from satellite images integrated with the Google Maps API. Based on the results of testing 4 blocks in the IPB-Cargil Oil Palm Education and Education Plantation that the presentation results were successfully detected by 97.67%, and it is known that each detected oil palm tree is known to the location of the coordinate point to be able to monitor, manage, and estimate the productivity of oil palm trees.

Key words: OBIA, Deep Learning, YOLOv7, Oil Palm.

KATA PENGANTAR

Bismillahhirrohmaanirrohim

Assalamu'alaikum Warahmatullahi Wabarakatuh

Alhamdulillah robil'aalamin

Segala Puji, Kebesaran, Kemulian, dan apa yang dilangit dan di bumi milik Allah SWT. Puji Syukur saya panjatkan kehadirat Allah SWT atas segala rahmat serta nikmat-Nya yang telah memberikan kemudahan serta kelancaran kepada Saya dalam penyelesaian Disertasi yang berjudul "Pengembangan Metode Deteksi Dan Menghitung Jumlah Pohon Kelapa Sawit Dari Sentinel 2 Imagery Menggunakan Metode *OBJECT-BASED IMAGE ANALYSIS (OBIA)*". Disertasi ini merupakan syarat untuk memperoleh gelar Doktor dalam bidang Teknologi Informasi pada Program Doktor Teknologi Informasi, Program Pascasarjana, Universitas Gunadarma, dimana penulis telah menyelesaikan seluruh rangkaian proses studi Program Doktor sejak tahun 2019. Sepanjang proses penyusunan Disertasi ini, banyak pihak yang turut membantu baik secara moril maupun materil kepada saya. Untuk itu dengan segala kerendahan dan ketulusan hati, perkenankan saya mengucapkan terima kasih kepada:

1. Ibu Profesor Doktor E. S. Margianti, S.E., MM., selaku Rektor Universitas Gunadarma.
2. Bapak Profesor Doktor Insinyur Bambang Suryawan, MT., selaku Koordinator Program Pascasarjana Universitas Gunadarma.
3. Bapak Profesor Insinyur Busono Soerowirdjo, MSc., Ph.D., selaku Direktur Program Doktor Universitas Gunadarma.
4. Profesor Doktor Sarifuddin Madenda, selaku Ketua Program Doktor Teknologi Informasi Universitas Gunadarma sekaligus Ko-Promotor dis-

elasela kesibukannya dengan sabar membimbing, mengarahkan, memberi masukan dan memotivasi dalam menyelesaikan disertasi

5. Bapak Profesor Doktor Eri Prasetyo Wibowo, selaku Sekretaris Program Doktor Teknologi Informasi Universitas Gunadarma.
6. Bapak Profesor Doktor Insinyur Kudang Boro Seminar, M.Sc. selaku Promotor yang dengan sabar memberikan membimbing, memotivasi, melakukan koreksi, memberi masukkan dan saran dalam menyelesaikan disertasi ini.
7. Bapak Profesor Doktor Insinyur Sudrajat, M.Sc., selaku Penguji Luar terima kasih atas waktu, kesediaan, dan masukkan, serta saran disertasinya bagi saya.
8. Bapak Profesor Suryadi Harmanto, SSi., MMSI, Bapak Doktor rer. nat. I Made Wiriana dan Ibu Doktor Detty Purnamasari selaku penguji dalam yang telah memberi banyak masukan dan saran perbaikan, sehingga disertasi ini semakin berkualitas.
9. Bapak Doktor Irwan Bastian yang telah memberikan dukungan kepada saya dapat melaksanakan kuliah Program Doktor Teknologi Informasi di Universitas Gunadarma.
10. Ayahanda tercinta Baikusnendro dan Ibunda tercinta Mennik Trihastuti, yang selalu memberikan doa yang terbaik dan motivasi, serta kedua adik saya Finsa Dwi Hestu Fikriansya, dan Hilmi Hestu Saputra yang selalu mendukung secara moril.
11. Rekan-rekan angkatan 25 Program Doktor Teknologi Informasi Universitas Gunadarma yang selalu memberikan semangat, dan Mas Bonang Waspadadi Ligar, serta Sumaiyah Fitriandini yang berjuang bersama, serta diskusi.

12. Ibu-ibu di Sekretariat Program Doktor Teknologi Informasi Universitas Gunadarma Ibu Doktor Diny Wahyuni, Ibu Doktor Reni Diah Kusumawati, Ibu Doktor Aini Suri Talita, dan Ibu Doktor Dety Purnamasari yang sangat membantu dalam administrasi penyelesaian disertasi ini.
13. Teman-teman Bidang Kemahasiswaan Universitas Gunadarma, komunitas GDSC UG, Gunadarma IO, UGTV, Tim Teknis UG dan sahabat saya Muhammad Rifqi Al Furqon, Manfred Michael, dan Evan Sakti Endi yang telah membantu dalam penyelesaian disertasi ini.

Semoga Allah SWT memberikan limpahan kebaikan dan pahala atas semua perhatian dan dukungan yang Bapak, Ibu, Saudara sekalian berikan kepada saya. Harapan saya agar Disertasi ini memberikan manfaat nyata bagi semua pihak yang berkepentingan. Saya mengharapkan kritik dan saran untuk perbaikan pada masa yang akan datang.

Wassalamu'alaikum Warahmatullahi Wabarakatuh

Jakarta, 10 Mei 2023

(Guntur Eka Saputra)

DAFTAR ISTILAH

Akronim

ASIC	Application Specific Integrated Circuit
CAD	Computer Aided Design
EDA	Electronic Design Automation

Singkatan

AMS	Austria Micro System
CMOS	Complementary Metal Oxide Silicon
FPGA	Field Programmable Gate Array
GDSII	Graphics Data Station Information Interchang
HDL	Hardware Description Language
IP	Intellectual Property
IC	Integrated Circuit
μm	Mikrometer, 1×10^{-6}
RTL	Register Transfer Level
SOC	System On CHIP
VHDL	VHSIC Hardware Description Language
VHSIC	Very High Speed Interated Circuit
VLSI	Very Large Scale Integration

Simbol

AMS	Austria Micro System
CMOS	Complementary Metal Oxide Silicon
FPGA	Field Programmable Gate Array
GDSII	Graphics Data Station Information Interchang

HDL	Hardware Description Language
IP	Intellectual Property
IC	Integrated Circuit
μm	Mikrometer, 1×10^{-6}
RTL	Register Transfer Level
SOC	System On CHIP
VHDL	VHSIC Hardware Description Language
VHSIC	Very High Speed Interated Circuit
VLSI	Very Large Scale Integration

DAFTAR ISI

COVER	i
HALAMAN JUDUL	ii
LEMBAR PERSETUJUAN	iii
LEMBAR PENGUJI	iv
PERNYATAAN ORIGINALITAS DAN PUBLIKASI	v
ABSTRAK	vi
ABSTRACT	viii
KATA PENGANTAR	x
DAFTAR ISTILAH	xiii
DAFTAR ISI	xv
DAFTAR TABEL	xxii
DAFTAR GAMBAR	xxix
DAFTAR ALGORITMA	xxix
I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Batasan Penelitian	13

1.3	Rumusan Masalah	14
1.4	Tujuan Penelitian	15
1.5	Kontribusi Hasil Penelitian	16
II	TELAAH PUSTAKA	17
2.1	<i>Computer Vision</i>	17
2.2	Pengertian Citra	18
2.3	Pengolahan Citra	19
2.4	<i>Artificial Intelligence</i> (AI)	20
2.5	<i>Object Detection</i>	22
2.6	Machine Learning	26
2.6.1	Supervised Learning	27
2.6.2	Unsupervised Learning	29
2.6.3	Semi-supervised learning	31
2.6.4	Reinforcement Learning	32
2.6.5	Deep Learning	34
2.6.5.1	Recurrent Neural Networks (RNN)	38
2.6.5.2	Long Short Term Memory (LSTM)	39
2.6.5.3	Convolutional Neural Network (CNN)	40
2.6.5.4	Deep Believe Networks (DBN)	47
2.6.5.5	Deep Stacking Networks (DSN)	48
2.7	You Only Look Once (YOLO)	49
2.7.1	Arsitektur YOLO	51
2.7.2	Cara kerja Deteksi Objek YOLO	52
2.7.3	Perkembangan YOLO	56
2.8	Object Based Image Analysis (OBIA)	66
2.9	Sentinel-2	68
2.10	Pertanian Presisi	69
2.11	Kelapa Sawit	71

2.12 <i>State of The Art</i> (Penelitian Terdahulu)	77
III METODE PENELITIAN	82
3.1 Tahapan Penelitian	82
3.2 Area Studi Penelitian Kelapa Sawit	83
3.3 Persiapan Data	85
3.3.1 Deskripsi Dataset dan Pemilihan Data	85
3.3.2 Membersihkan Data	91
3.3.3 Membangun Data	92
3.3.3.1 Membuat Templates	94
3.3.3.2 Menyesuaikan Template	95
3.3.3.3 Evaluasi Korelasi diantara Template	96
3.3.3.4 Penambahan Template	99
3.3.3.5 Menjalankan Template Matching	99
3.3.3.6 Analisis Kluster dengan BIRCH	99
3.3.3.7 Menjalankan Auto-Annotate Datasets	101
3.3.3.8 Integrasi Data	101
3.3.3.9 Format Data	101
3.4 Citra untuk Komputasi	102
3.5 Pelatihan Model CNN	102
3.6 Optimisasi Parameter	103
3.7 Menjalankan Model CNN	104
3.8 Hasil Deteksi dan Penghitungan Akhir	107
IV HASIL DAN PEMBAHASAN	115
4.1 Area Studi	115
4.2 Persiapan Data	118
4.2.1 Deskripsi Dataset dan Pemilihan	118
4.2.1.1 Dataset Primer	118

4.2.1.2	Dataset Sekunder	121
4.2.2	Membersihkan Data	126
4.2.3	Membangun Data	127
4.2.4	Membuat Template	128
4.2.4.1	Menyesuaikan Template	130
4.2.4.2	Evaluasi Korelasi diantara Template	131
4.2.4.3	Penambahan Template	132
4.2.4.4	Menjalankan Template Matching	133
4.2.4.5	Analisis Cluster dengan BIRCH	134
4.2.4.6	Menjalankan Auto-Annotate Datasets	135
4.2.4.7	Integrasi Data	138
4.2.4.8	Format Data	140
4.3	Citra untuk Komputasi	141
4.4	Pelatihan Model CNN	144
4.4.1	Pelatihan Model CNN YOLOv5	145
4.4.2	Pelatihan Model CNN YOLOv6	148
4.4.3	Pelatihan Model CNN YOLOv7	150
4.5	Pengujian Model CNN	153
4.5.1	Pengujian Model CNN YOLOv5	153
4.5.2	Pengujian Model CNN YOLOv6	159
4.5.3	Pengujian Model CNN YOLOv7	162
4.6	Hasil Model YOLOv7 Pelatihan dengan Inferensi	166
4.7	Hasil Deteksi dan Penghitungan Akhir	167
4.7.1	Mengubah Model CNN ke Format ONNX	168
4.7.2	Mendapatkan titik tengah koordinat (latitude - longitude)	168
4.7.3	Mendeteksi dan Menghitung berdasarkan Existing Model	171
4.7.4	Mendapatkan objek yang terdeteksi dengan format YOLO	171
4.7.5	Mengubah objek ke format COCO	174

4.7.6	Mengubah Titik Piksel ke Piksel Koordinat	174
4.7.7	Mengubah Piksel Koordinat ke Latitude-Longitude	176
V	PENUTUP	181
5.1	Kesimpulan	181
5.2	Saran	182
DAFTAR PUSTAKA		184
DAFTAR RIWAYAT HIDUP		199

DAFTAR TABEL

1.1	Statistik Areal dan Produksi Kelapa Sawit Menurut Provinsi dan Status Pengusahaan Tahun 2021 (Sumber: Ditjenbun, 2023)	4
2.1	Penerapan Arsitektur Deep Learning	38
2.2	Pengelompokan Berdasarkan Masa Berbuah	72
2.3	Produktivitas Tanaman Kelapa Sawit Menurut Umur Tanaman dalam Kondisi Kebun Percobaan Balit Marihat	73
2.4	Hubungan Jarak Tanam dan Populasi per Hektar Kelapa Sawit	76
2.5	<i>State of the Art</i> (Penelitian Terdahulu)h	77
3.1	Luas Area Dari Gambar Citra yang Ditangkap oleh drone di Kebun Pendidikan dan Penelitian Kelapa Sawit IPB-Cargil . . .	87
3.2	Augmentasi Dataset Sekunder untuk Menambahkan Data . . .	90
3.3	Sarana Penelitian Pengembangan Sistem untuk Membangun Dataset dengan Anotasi atau Label Otomatis	93
3.4	Optimisasi Parameter	103
3.5	Komponen Implementasi Prototype Perangkat Lunak	112
3.6	Komponen Implementasi Prototype Perangkat Keras	114
4.1	Data Lokasi dan Luasan Area Studi	115
4.2	Area Studi yang Berhasil Ditangkap menjadi Citra Area Pohon Kelapa Sawit dengan Drone	117
4.3	Jumlah Citra dari Dua Area Studi	119
4.4	Kegunaan Layanan Augmentasi Dataset Sekunder	123
4.5	Hasil Proses Augmentasi Citra Dataset Sekunder	124

4.6	Hasil Rerata Korelasi dan Waktu yang Dibutuhkan	131
4.7	Waktu yang Dibutuhkan untuk Rotasi	133
4.8	Hasil Waktu yang Dibutuhkan untuk Deteksi Objek sebagai Dataset	135
4.9	Hasil Waktu Yang Dibutuhkan untuk Anotasi Otomatis	136
4.10	Pengujian dengan 10 Citra Dataset Primer	137
4.11	Dataset untuk Komputasi	142
4.12	Optimisasi Parameter	144
4.13	Sarana Penelitian Mesin NVIDIA (Super Komputer)	145
4.14	Evaluasi Pelatihan Model dengan YOLOv5 pada Google Colab Pro dan DGX-A-100	146
4.15	Evaluasi Waktu Proses Pelatihan Model CNN YOLOv5	147
4.16	Evaluasi Pelatihan Model dengan YOLOv6 pada Google Colab Pro dan DGX-A-100	148
4.17	Evaluasi Waktu Proses Pelatihan Model CNN YOLOv6	150
4.18	Evaluasi Pelatihan Model dengan YOLOv7 pada Google Colab Pro dan DGX-A-100	151
4.19	Evaluasi Waktu Proses Pelatihan Model CNN YOLOv7	152
4.20	Evaluasi Pengujian Model dengan YOLOv5 pada Google Colab Pro dan DGX-A-100	154
4.21	Evaluasi Waktu Proses Pengujian Model CNN YOLOv5	156
4.22	Evaluasi Pengujian Model dengan YOLOv6 pada Google Colab Pro dan DGX-A-100	159
4.23	Evaluasi Waktu Proses Pengujian Model CNN YOLOv6	160
4.24	Evaluasi Pengujian Model dengan YOLOv7 pada Google Colab Pro dan DGX-A-100	162
4.25	Evaluasi Waktu Proses Pengujian Model CNN YOLOv7	163

4.26 Hasil Kecepatan Waktu Proses Dataset dengan Model YOLOv7	
Pelatihan dengan Inferensi	167
4.27 Nilai Titik Awal dari Titik-Titik pada Objek Pohon Kelapa Sawit	173
4.28 Nilai Komponen Format COCO	174
4.29 Hasil Presentase Prediksi Pohon Kelapa Sawit pada 4 Blok . . .	179
5.1 A sample long table.	193

DAFTAR GAMBAR

1.1	Area Lahan Kelapa Sawit IPB Cargil Jonggol Blok 3 Jarak Tanam 9x9x9	5
1.2	Pola Tanam dan Jarak Tanam Kelapa Sawit yang Tepat (Sumber: GD Morganic, 2018)	6
1.3	Peta 10 Besar Provinsi Sentra Kelapa Sawit Tahun 2021 (Sumber: Ditjenbun, 2023)	7
1.4	Estimasi Luas Area dan Produksi Minyak Sawit Indonesia 2023 Sumber: (Ditjenbun, 2023)	7
2.1	(a) Hubungan Artificial Intelligence, Machine Learning, dan Deep Learning. (b) Jenis Machine Learning (Sumber: C. Thongprayoon et al, 2020)	21
2.2	Classification, Object Detection dan Segmentation Representation (Sumber: A. Patel, 2020)	22
2.3	Segmentation, Classification+Localization, Object Detection (Sumber: A. Patel, 2020)	23
2.4	Image Classification (Sumber: A. Patel, 2020)	23
2.5	Object Detection dengan Bounding Box (Sumber: A. Patel, 2020)	24
2.6	Intersection Over Union (IoU) (Sumber: A. Patel, 2020)	25
2.7	Persamaan IoU (Sumber: A. Patel, 2020)	25
2.8	Skema Supervised Learning (Sumber: M. Kozan, 2021)	27
2.9	Skema Unsupervised Learning (Sumber: M. Kozan, 2021)	29
2.10	Skema Semi-Supervised Learning (Sumber: (?))	32

2.11 Skema Reinforcement Learning (Sumber: R. Sutton, 1998)	33
2.12 Skema Umum <i>Deep Learning</i> (Sumber: A. Feizollah et al, 2022)	36
2.13 Skema <i>Deep Learning</i> dengan Penambahan beberapa <i>hidden layer</i> (Sumber: H. Kaur et al, 2021)	37
2.14 Diagram <i>Recurrent Neural Network</i> (RNN) (Sumber: K. Dass, 2020)	39
2.15 Diagram <i>Long Short Term Memory</i> (LSTM) (Sumber: S. Yan, 2016)	40
2.16 Array dari Matriks RGB (Sumber: R. Prabhu, 2018)	41
2.17 Neural Network dengan banyak Convoluti Layer (Sumber: R. Prabhu, 2018)	41
2.18 Gambar Matriks Multiplies Kernel atau Filter Matriks (Sumber: R. Prabhu, 2018)	42
2.19 Gambar matriks 5×5 dikalikan dengan Filter matiks 3×3	42
2.20 Output Matriks 3×3	43
2.21 Beberapa Filter Umum (Sumber: R. Prabhu, 2018)	44
2.22 Stride 2 Piksel (Sumber: R. Prabhu, 2018)	45
2.23 Operasi ReLU (Sumber: P. Ratan, 2021)	46
2.24 Max Pooling	46
2.25 Setelah Pooling Layer Diratakan sebagai FC Layer	47
2.26 Arsitektur CNN Lengkap	47
2.27 Skema Diagram DBN (Sumber: H. Liu & B. Lang, 2019)	48
2.28 Skema Diagram DSN (Sumber: L. Deng et al, 2012)	49
2.29 Kecepatan YOLO dibandingkan dengan detector Objek Lainnya (Sumber: S. A. S. Hernandez et al, 2020)	50
2.30 Arsitektur YOLO dari <i>Original Paper</i> (J. Redmon et al., 2015)	51
2.31 (A) Input Image dan (B) Hasil Algoritma YOLO (Sumber: Z. Kelta, 2022)	52

2.32	Residual Blocks (Sumber: Z. Kelta, 2022)	53
2.33	Grid dengan Probabilitas (Sumber: Z. Kelta, 2022)	54
2.34	Cara <i>Bounding Box</i> (Sumber: Z. Kelta, 2022)	54
2.35	IOU (Sumber: Z. Kelta, 2022)	55
2.36	Perkembangan YOLO (Sumber: Z. Kelta, 2022)	56
2.37	Komparasi YOLOv4 dengan YOLOv3 dan <i>state-of-the-art</i> Deteksi Objek Lain (Sumber: Z. Kelta, 2022)	59
2.38	Unified Network Architecture (Sumber: C. Y. Wang et al, 2021)	60
2.39	Performance YOLOR vs YOLOv4 dan Model Lainnya (Sumber: C. Y. Wang et al, 2021)	61
2.40	Perbandingan Waktu Pelatihan antara YOLOv4 dan YOLOv5 (Sumber: J. Nelson, 2020)	63
2.41	Comparison of state-of-the-art efficient object detectors. All models are tested with TensorRT 7 except that the quantized model is with TensorRT 8 (Sumber: C. Li et al, 2022)	64
2.42	Perbandingan YOLOv7 inference time dengan real-time object detector lainnya (Sumber: C. Y. Wang et al, 2022)	65
2.43	<i>Compound scaling up depth and width for concatenation-based model</i> (Sumber: C. Y. Wang et al, 2022)	66
2.44	OBIA Segmentasi Proses Pengelompokkan Pixel yang hampir sama ke dalam Objek (Sumber: GISGeography, 2020)	67
2.45	OBIA Klasifikasi menggunakan shape, size, dan spectral properties objek untuk klasifikasi setiap objek	67
2.46	Siklus Pertanian Presisi (Sumber: A. Comparetti, 2011)	70
2.47	Buah Kelapa Sawit(Inti dan Daging Sawit)	72
2.48	Pola Jarak Tanam Bujur Sangkar atau Segiempat	74
2.49	Pola Jarak Tanam Segitiga (Sumber: GD Morganic, 2018)	75
3.1	Tahapan Penelitian yang Dilakukan	82

3.2	Lokasi Kebun Pendidikan dan Penelitian Kelapa Sawit IPB-Cargil	84
3.3	Lokasi Universitas Gunadarma PPU Campus yang ditanami Pohon Kelapa Sawit	84
3.4	Diagram Proses Persiapan Data	85
3.5	Hasil Citra Area Pohon Kelapa Sawit PPU untuk Dataset Primer	86
3.6	Hasil Citra Blok 3 untuk Dataset Primer (luas area dalam ha) .	87
3.7	Hasil Citra Blok 4 untuk Dataset Primer (luas area dalam ha) .	88
3.8	Data sekunder yang sudah Diberikan Label atau Anotasi Oil Palm (Sumber: JS, 2022)	89
3.9	(a) Gambar Dataset Sekunder; (b) Gambar Dataset Primer di Kalimantan; (c) Gambar Dataset Primer di Jonggol;.	91
3.10	Citra yang tidak digunakan sebagai dataset (a) tidak ada pohon citra kelapa sawit; (b) tangkapan citra dari drone tidak dari sisi atas yang tegak	92
3.11	Diagram Alir Pembangunan Dataset Berdasarkan Pendekatan OBIA	94
3.12	Template and Referensi Matriks	96
3.13	Cara Kerja Pendekatan Algoritma YOLOv7	104
3.14	Skema Diagram Algoritma YOLO	106
3.15	Diagram Alur Hasil Deteksi dan Penghitungan Akhir	108
3.16	<i>Bounding Box</i> pada COCO dan YOLO format	109
3.17	Diagram Arsitektur Sistem	110
4.1	Peta Sebaran Area Inisialisasi Lahan Kebun Pendidikan dan Penelitian Kelapa Sawit milik Institut Pertanian Bogor-Cargil .	116
4.2	Luas Area Blok 1 Kebun Pendidikan dan Penelitian Kelapa Sawit Institut Pertanian Bogor-Cargil	117
4.3	Luas Area Blok 2 Kebun Pendidikan dan Penelitian Kelapa Sawit Institut Pertanian Bogor-Cargil	117

4.4	Hasil Citra Sampel Universitas Gunadarma di Penajam Paser Utara, Kalimantan Timur	120
4.5	Hasil Citra Sampel Universitas Gunadarma di Penajam Paser Utara, Kalimantan Timur	121
4.6	Proses Augmentasi Data Sekunder	122
4.7	Data Citra Asli Data Sekunder (kiri: original; kanan: tampak dengan <i>bounding box</i>)	124
4.8	Data Citra Primer yang tidak Digunakan (kiri: tidak terdapat pohon kelapa sawit; kanan: tampak area yang tidak tangkapan dari atas)	127
4.9	<i>Home Page</i> Jupyter Notebook	128
4.10	Data citra untuk membuat template (5472 x 3078)	128
4.11	Membuat Template yang Ditandai dengan Titik Objek Pohon Kelapa Sawit	129
4.12	Citra Template dari Objek Pohon Kelapa Sawit	130
4.13	Menyesuaikan Template pada ID 2	131
4.14	Hasil Rotasi 30° pada Citra template	132
4.15	Objek yang terdeteksi untuk Dataset dengan <i>Template Matching</i>	133
4.16	Hasil objek yang terdeteksi untuk Dataset dengan algoritma BIRCH	134
4.17	Hasil Anotasi Otomatis dalam format file *.txt	136
4.18	Gambar dan File *.txt	138
4.19	Dataset Data Primer Drone di Roboflow	139
4.20	Banyaknya Objek yang Terdeteksi sebagai Pohon Kelapa Sawit .	140
4.21	Format Data	141
4.22	Dataset pada layanan Google Drive	141
4.23	Contoh Citra yang digunakan untuk komputasi (a) Citra dataset pelatihan; (b) Citra dataset validasi; (c) Citra dataset pengujian;	143

4.24 Hasil Evaluasi Pelatihan Model YOLOv5 antara Google Colab Pro dengan DGX-A-100	147
4.25 Hasil Evaluasi Pelatihan Model YOLOv6 antara Google Colab Pro dengan DGX-A-100	149
4.26 Hasil Evaluasi Pelatihan Model YOLOv7 antara Google Colab Pro dengan DGX-A-100	152
4.27 Hasil Evaluasi Pengujian Model YOLOv5 antara Google Colab Pro dengan DGX-A-100	155
4.28 (a) Confusion Matrix dan (b) Citra Sampel Prediksi Pohon Kelapa Sawit dari Citra Dataset Pengujian	158
4.29 Hasil Evaluasi Pengujian Model YOLOv6 antara Google Colab Pro dengan DGX-A-100	160
4.30 (a) Confusion Matrix Pengujian YOLOv6 (b) Citra Sampel Pengujian	161
4.31 Hasil Evaluasi Pengujian Model YOLOv7 antara Google Colab Pro dengan DGX-A-100	163
4.32 (a) Confusion Matrix Pengujian YOLOv7 (b) Citra Sampel Pengujian;	164
4.33 Halaman Register dan Login	169
4.34 Tampilan Citra Maps Satelit	169
4.35 Polygon untuk Mendeteksi dan Menghitung Pohon Kelapa Sawit	170
4.36 Hasil Mendeteksi dan Menghitung Pohon Kelapa Sawit	171
4.37 Hasil deteksi dan penghitungan, serta menampilkan titik-titik koordinat latitude-longitude lokasi setiap objek pohon kelapa sawit	172
4.38 Nilai Awal dari Titik-Titik pada Objek Pohon Kelapa Sawit	173
4.39 Hasil Uji Blok 1	177
4.40 Hasil Uji Blok 2	178

4.41 Hasil Uji Blok 3	178
4.42 Hasil Uji Blok 4	179

DAFTAR ALGORITMA

BAB I

PENDAHULUAN

1.1 Latar Belakang

Perkembangan teknologi yang pesat telah membantu dalam menjalani kehidupan yang lebih sederhana dan praktis. Permintaan untuk pengembangan teknologi terus meningkat dan telah menjadi salah satu bidang pekerjaan dan studi yang paling popular(Abid Haleem 2022). Perkembangan teknologi, khususnya investasi teknologi informasi dikaitkan dengan pertumbuhan produktivitas yang signifikan di negara maju dan berkembang (Jason Dedrick 2014). Di negara-negara berkembang, teknologi informasi dan komunikasi sangat membutuhkan penerapan khususnya di bidang pertanian. Pertanian memainkan peran sentral dalam pembangunan di banyak negara. Akses pasar, pembiayaan, dan pengetahuan adalah landasan pertumbuhan pertanian. Teknologi Informasi dan Komunikasi (TIK) mendukung petani dengan harga pasar secara real-time, prakiraan cuaca, hama, varietas benih dan teknik penanaman, identifikasi dan penghitungan tanaman. Pertanian juga merupakan sumber pendapatan utama bagi penduduk pedesaan di sebagian besar negara. Sektor ini menghadapi banyak tantangan untuk meningkatkan produksi. TIK memiliki potensi untuk memenuhi tantangan yang dihadapi oleh petani dan atau pemangku kepentingan yang dapat meningkatkan taraf hidup masyarakat pedesaan(Havinal 2020).

Dalam kasus ini, Indonesia adalah salah satu produsen dan eksportir pertanian terbesar di dunia, yang memasok bahan mentah penting seperti

karet alam, kopi, kakao, beras, dan minyak kelapa sawit ke seluruh dunia. Dalam beberapa dekade terakhir, industri pertanian juga telah menjadi sektor yang paling banyak menyerap tenaga kerja di Indonesia (Statista Research, 2023). Indonesia juga sebagai produsen perkebunan terbesar di dunia, seperti karet alam dan kelapa sawit, produksi tanaman Indonesia sangat penting bagi perekonomian nasional. Dari 15 produk pertanian utama Indonesia, kelapa sawit merupakan yang terbesar. Produksi kelapa sawit sangat penting bagi perekonomian, karena Indonesia produsen dan konsumen terbesar di dunia dan menyumbang sekitar setengah dari pasokan dunia. Sebagian minyak kelapa sawit berasal dari perkebunan yang dikelola oleh petani kecil dengan luas 6,02 juta hektar pada tahun 2021. Petani kecil menghasilkan sekitar 34,36% produksi minyak kelapa sawit, menjadikan petani kecil sebagai kontributor penting dalam mempromosikan industri kelapa sawit yang berkelanjutan di Indonesia (Ditjenbun, 2023).

Dalam menghasilkan produksi kelapa sawit yang baik, maka dibutuhkan penerapan teknologi yang mendukung untuk dapat memonitoring pohon kelapa sawit tersebut yang dapat menghasilkan, karena luas lahan yang begitu besar tidak memungkinkan dapat dikerjakan secara konvensional.

Pertanian presisi membutuhkan informasi yang dapat diandalkan tentang situasi terkini pada waktu yang tepat. Manajer kelapa sawit biasanya mengukur kepadatan atau jumlah pohon kelapa sawit secara manual. Tanaman kelapa sawit juga memiliki lahan yang luas, maka diperlukan pemantauan lahan secara berkala untuk mengontrol produktivitas kelapa sawit dan juga sebagai data inventaris, oleh karena itu, identifikasi otomatis dan lokasi kelapa sawit merupakan cara alternatif bagi petani untuk mengelola sumber daya mereka dengan menggunakan teknologi, bukan dengan pendekatan manual.

Kelapa sawit yang memiliki nama ilmiah (*Elaeis guineensis Jacq.*) merupakan tanaman yang berasal dari daerah Benua Afrika dan negara di Amerika

Selatan. Pada awalnya tanaman ini tumbuh liar dan setengah liar di daerah tepi sungai. Di Indonesia, tanaman ini pertama kali diperkenalkan oleh pemerintah colonial Belanda pada tahun 1848 di Kebun Raya Bogor (I. Pahan, 2018). Perkebunan kelapa sawit di Indonesia berkembang pesat, pada tahun 1939, Indonesia menjadi negara produsen dan eksportir utama kelapa sawit dunia dengan volume mencapai 244 ribu ton atau sebesar 48% total ekspor minyak kelapa sawit dunia (S. Prayitno et al, 2008).

Besar volume dari kelapa sawit yang diproduksi oleh Indonesia menjadikan sector kelapa sawit membantu ekonomi Indonesia. Dalam perekonomian makro ekonomi Indonesia, industri atau sector kelapa sawit telah memiliki peran strategis, diantaranya penghasil devisa terbesar, lokomotif perekonomian nasional, kedaulatan energy, pendukung sector ekonomi kerakyatan, dan penyerapan tenaga kerja. Hal ini dilihat dari besarnya perkebunan sawit di Indonesia. Perkebunan kelapa sawit Indonesia berkembang di 22 provinsi dari 33 provinsi. Dua pulau utama sentra perkebunan kelapa sawit yaitu Sumatera dan Kalimantan. Sekitar 90% perkebunan kelapa sawit di Indonesia berada pada dua pulau tersebut, dan kedua pulau tersebut telah menghasilkan 95% produksi minyak sawit mentah (*crude palm oil/CPO*) Indonesia. Dalam perjalannya perkebunan kelapa sawit mengalami revolusi dan berkembang dengan cepat. Dalam kurun 1990-2015, tumbuh dan berkembangnya perkebunan rakyat dengan cepat, yakni, 24% per tahun selama periode 1990-2015. Pada tahun 2015, luas perkebunan sawit di Indonesia adalah 11,3 juta hektar (KPRI, 2015). Menurut Ketua Asosiasi Industri Minyak Makan Indonesia (AIMMI) Adi Wisoko Kasman menjelaskan bahwa peningkakan konsumsi minyak kelapa sawit dalam negeri terus mengalami peningkatan (L. Yuniartha, 2019). Pada tahun 2019, konsumsi minyak sawit tumbuh hingga 23,57% atau meningkat dari 13,49 juta ton di 2018 menjadi 16,67 juta ton di tahun 2019, kemudian konsumsi minyak sawit untuk kategori makanan (food) dalam negeri mencapai

9,86 juta ton atau naik hingga 49% tahun per tahun (L. Yuniartha, 2020). Pada tahun 2021, hasil reevaluasi luas areal dari direktorat jenderal perkebunan, diketahui bahwa perkebunan besar swasta sebesar 60,64%, diikuti perkebunan rakyat 34,36%, dan perkebunan negara 5% dari 14.621.690 ha. Pada tahun 2023, CPO Indonesia diprediksi mencapai 48,2 juta ton (Ditjenbun, 2023). Luas areal dan produksi kelapa sawit menurut provinsi dan status pengusahaan tahun 2021, seperti Tabel 1.1 berikut ini:

Tabel 1.1: Statistik Areal dan Produksi Kelapa Sawit Menurut Provinsi dan Status Pengusahaan Tahun 2021
(Sumber: Ditjenbun, 2023)

Perkebunan Rakyat Smallholders		Perkebunan Negara Government Estate		Perkebunan Swasta Private Estate		Jumlah (Total)	
Luas/ Areal (Ha)	Produksi/ Production (Ton)	Luas/ Areal (Ha)	Produksi/ Production (Ton)	Luas/ Areal (Ha)	Produksi/ Production (Ton)	Luas/ Areal (Ha)	Produksi/ Production (Ton)
6.029.749	15.503.840	550.333	2.256.134	8.041.068	27.361.506	14.621.690	45.121.480

Berdasarkan Tabel 1.1 bahwa saat ini areal dan produksi kelapa sawit masih dikuasai dan dikelola oleh Perkebunan Swasta yang tersebar dari 34 provinsi di Indonesia, yaitu luas areal sebesar 8.041.068 (Ha) dan dapat memproduksi 27.361.506 (Ton) dari 45.121.480 (Ton) (Ditjenbun, 2023).

Dari data yang diperoleh menurut Ditjenbun tahun 2023 bahwa dibutuhkan kelapa sawit dalam jumlah yang besar dan produksi minyak sawit yang dihasilkan dari kelapa sawit tersebut harus baik. Pohon kelapa sawit membutuhkan waktu sekitar 4 (empat) tahun untuk menghasilkan buah yang sesuai untuk panen. Setiap pohon kelapa sawit kemudian akan terus menghasilkan buah hingga usia 25 tahun. Menurut Smart Agribusiness and food dalam artikelnya menyebutkan bahwa pohon kelapa sawit ini memanfaatkan teknologi pertanian didukung oleh analisis satelit di seluruh area perkebunan, sehingga pemanfaatan hasil panen dapat dihasilkan hasil panen dapat digunakan secara optimal (Smart Agribusiness and Food, 2017). Dalam pengelolaan perkebunan kelapa sawit, populasi dalam satuan hektar (ha) sangat penting dan hal ini

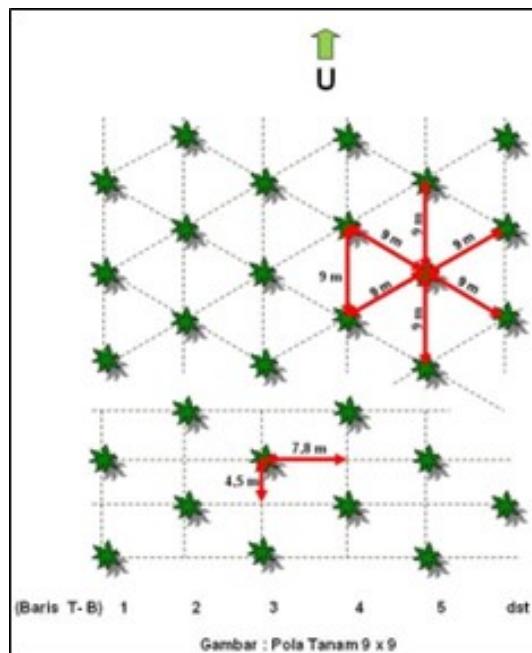
berhubungan dengan pengaruh jarak tanam. Jarak tanam merupakan faktor yang mempengaruhi pertumbuhan tanaman kelapa sawit. Pengaturan jarak tanam bertujuan untuk mendapatkan ruang tumbuh bagi pertumbuhan tanaman guna menghindari kompetisi unsur hara dan cahaya matahari dari setiap tanaman kelapa sawit pada jarak tanam yang berbeda. Jarak tanam 9x9x9 m baik untuk kelapa sawit (Hayata et al, 2020). Kepadatan pada jarak tanam normal biasanya bervariasi sesuai dengan jenis tanah yang ditanami kelapa sawit. Jumlah tanaman kelapa sawit pada pesisir dan di tanah mineral adalah antara 136 - 148 kelapa sawit/hektar seperti pada Gambar 1.1, sedangkan di tanah gambut, jarak tanamnya biasanya lebih padat, sekitar 150 kelapa sawit/hektar (J. Latif et al., 2003). Gambar 1.1 merupakan area lahan Kebun Pendidikan dan Penelitian IPB Cargil - Jonggol Blok 3 dengan tanah mineral (J. Albari et al., 2018). Tanah mineral adalah tanah yang terbentuk dan berkembang dari bahan mineral, melalui proses pelapukan, baik secara fisis maupun kimia, didominasi oleh pelapukan bebatuan (?).



Gambar 1.1: Area Lahan Kelapa Sawit IPB Cargil Jonggol Blok 3 Jarak Tanam 9x9x9

Dalam memenuhi kebutuhan minyak kelapa sawit, selain membutuhkan areal lahan, dibutuhkan pola tanam dan jarak tanam kelapa sawit yang tepat.

Pola tanam kelapa sawit yang baik dibutuhkan perhatikan yang lebih karena berkaitan dengan efektifitas penggunaan lahan. Pola tanam segitiga sama sisi merupakan pola tanam yang paling efektif di areal datar, sehingga untuk areal bergelombang atau berbukit perlu dilakukan "viol linning" untuk mempertahankan jumlah populasi per hektarnya dengan tetap memperhatikan tingkat kesuburan tanahnya (GD Morganic, 2018), ilustrasi jarak tanam kelapa sawit seperti pada Gambar 1.2.



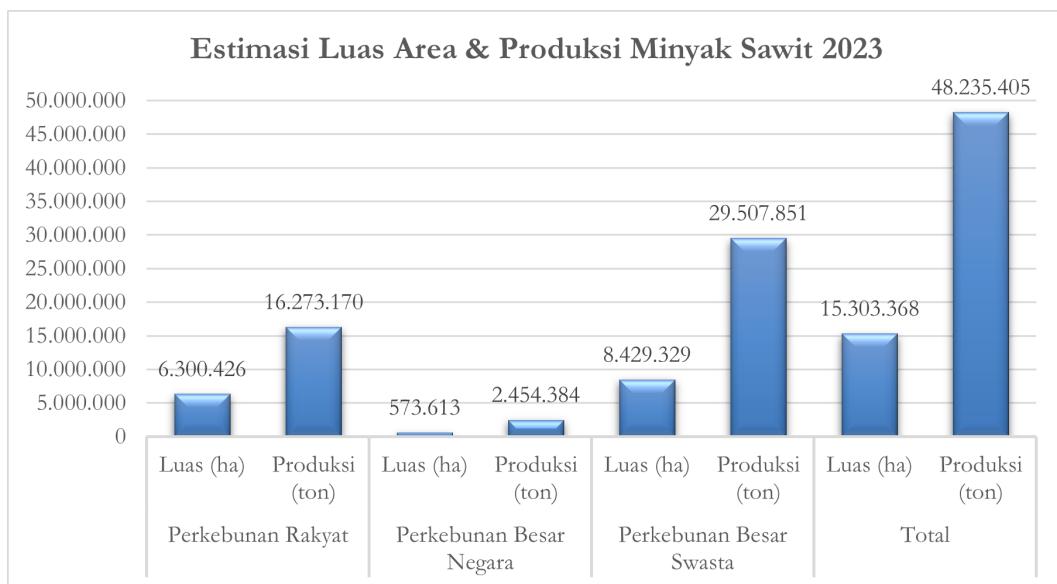
Gambar 1.2: Pola Tanam dan Jarak Tanam Kelapa Sawit yang Tepat
(Sumber: GD Morganic, 2018)

Direktorat Jenderal Perkebunan (Ditjenbun) memetakan produksi kelapa sawit menurut provinsi dan menunjukkan bahwa provinsi Riau memiliki produksi terbesar 2021 sebesar 8,96 juta ton dengan luas 3,49 juta hektar (Ditjenbun, 2023). Berikut ini Gambar 1.3. peta kelapa sawit Indonesia berdasarkan Direktorat Jenderal Perkebunan.



Gambar 1.3: Peta 10 Besar Provinsi Sentra Kelapa Sawit Tahun 2021
(Sumber: Ditjenbun, 2023)

Direktorat jenderal perkebunan juga mencatat estimasi luas area dan produksi minyak sawit Indonesia tahun 2023 mencapai 48,24 juta ton dan luas hektar mencapai 15,3 juta ton (Ditjenbun, 2023) seperti pada Gambar 1.4.



Gambar 1.4: Estimasi Luas Area dan Produksi Minyak Sawit Indonesia 2023
Sumber: (Ditjenbun, 2023)

Gambar 1.4 menyatakan bahwa perkebunan swasta (PBS) menguasasi sebanyak 61,2% dengan total produksi 29,5 juta ton, perkebunan rakyat (PR) sebesar 33,7% dengan total produksi sebanyak 16,2 juta ton, sedangkan perkebunan besar negara (PBN) hanya 5% dengan total 2,45 juta ton.

Jumlah estimasi luas area lahan pohon kelapa sawit yang besar juga menjadikan permasalahan bagi pengelolaan perkebunan kelapa sawit. Dari sisi agronomis, bahwa batasan minimum untuk perusahaan perkebunan kelapa sawit yaitu sebesar 6.000 (enam ribu) hektar yang harus dikelola (PPRI, 2021). Luas lahan untuk petani kelapa sawit yang didefinisikan sebagai petani kelapa sawit adalah tinggal di pedesaan/sekitar kebun yang dimana kelapa sawit sebagai mata pencaharian utama, dikerjakan dan dikontrol sendiri oleh keluar-ganya, dan mengalami kesulitan karena bibit yang digunakan disemai sendiri dan tidak bersertifikat, sulit mengakses area karena luas, serta produktivitas rendah menjadi permasalahan utama (SPKS, 2020). Di samping permasalahan luas area ini, lokasi perkebunan sawit juga berada pada remote area, yaitu area atau daerah yang terpencil, jauh dari peradaban (daerah pedalaman atau pelosok hutan) yang dimana karena lokasi perkebunan kelapa sawit yang luas, sehingga sulit dimonitoring tanpa teknologi informasi. Permasalahan yang ada bahwa di perkebunan kelapa sawit yang luas juga terkendala akses infrastruktur, seperti jalan yang belum memadai untuk akses kendaraan, seperti di Kebun Pendidikan dan Penelitian IPB-Cargil akses jalan bergelombang dan masih bebatuan, dan menurut dinas perkebunan provinsi Kalimantan timur bahwa infrastruktur dan aksesibilitas terhubung dengan baik, maka produktivitas dan kegiatan monitoring kawasan perkebunan kelapa sawit akan sangat membantu (Disbun, 2014). Produksi per individu tanaman pohon kelapa sawit berkontribusi besar terhadap produktivitas karena setiap pohon kelapa sawit akan menghasilkan buah kelapa sawit yang dapat diproduksi dan dipanen hasilnya untuk masyarakat, sehingga data pohon kelapa sawit sangat penting untuk dilakukan. Selain itu, salah satu upaya yang dilakukan dalam perkebunan kelapa sawit adalah masalah pemupukan. Adanya peningkatan produksi melalui intensifikasi dengan pemberian pupuk yang presisi, secara cepat, dan akurat, serta up-to-date pada suatu luasan lahan. Upaya

yang dapat dilakukan adalah dengan meningkatkan produktivitas melalui peningkatan efektivitas dan efisiensi penggunaan pupuk. Efektivitas dan efisiensi pemberian pupuk sangat penting dilakukan karena biaya pemupukan tanaman kelapa sawit sangat besar yaitu 50%-70% dari biaya pemeliharaan dan 25% dari seluruh biaya produksi (Fairhurst et al., 2006). Penambahan unsur hara dapat meningkatkan pertumbuhan tanaman, produksi tanaman, dan kualitas produk yang dihasilkan, seperti buah kelapa sawit (S. Parman, 2007).

Buah kelapa sawit yang dapat dihasilkan di Indonesia cukup besar dan harus diproduksi atau dipanen dengan baik, agar hasil dari buah kelapa sawit yang mayoritas menjadi minyak menghasilkan minyak yang berkualitas. Hal ini berkenaan dengan standar mutu kematangan buah kelapa sawit yang dikenal dengan standar kematangan panen. Buah sawit bergerombol dalam tandan yang muncul dari tiap pelapah. Minyak dihasilkan oleh buah. Kandungan minyak bertambah sesuai dengan tingkat kematangan buah. Penghitungan pohon kelapa sawit yang menghasilkan minyak dapat dilakukan secara manual di lapangan, tetapi ini menghabiskan waktu, membutuhkan pekerja yang tidak sedikit, dan mahal (N. A Mubin, et al, 2019). Disamping itu, Manajer perkebunan kelapa sawit biasanya mengukur kepadatan kelapa sawit secara manual setiap tahun. Data penting ini dapat digunakan untuk memperkirakan produktivitas kelapa sawit, jumlah pupuk yang dibutuhkan, biaya penyiraman berkala, dan jumlah pekerja yang dibutuhkan, dan terkait dengan kegiatan lain (J. W. Kiama, 2014).

Oleh karena itu, dibutuhkan suatu metode atau teknologi untuk membantu mengklasifikasikan atau mendeteksi jumlah dan estimasi usia pohon kelapa sawit untuk dapat diketahui tingkat kematangan buah kelapa sawit, maka dapat diproduksi (dipanen) dengan baik, sehingga produksi yang dihasilkan dapat mencapai optimum. Berikut ini penelitian-penelitian yang telah dilakukan penelitian sebelumnya.

Klasifikasi berbasis piksel sering digunakan untuk mengklasifikasikan kelas fitur dari gambar. Metode Object-Based Image Analysis (OBIA) telah berevolusi untuk menganalisis gambar resolusi tinggi dengan cepat (T. Blaschke, 2010), dengan semakin tersedianya resolusi tinggi dan citra skala besar, maka dihasilkan cara baru untuk memulai penelitian tentang interpretasi foto pohon berbasis komputer (P. Gong et al, 1999). OBIA dikembangkan oleh Hay dan Castilla merupakan disiplin ilmu spasial yang berfokus pada pengelompokan citra penginderaan jauh ke makna objek penuh melalui pemanfaatan spasial (G. J. Hay et al, 2008). Gagasan menganalisis gambar dalam ruang objek daripada ruang pixel dikembangkan karena kekurangan metode berbasis pixel, terutama pada gambar resolusi tinggi. Selain itu, ruang objek telah diperkuat dalam kapasitas komputasi dan ketersediaan untuk analisis gambar resolusi tinggi seperti IKONOS, GeoEye, dan WorldView (T. Blaschke, 2010).

Jusoff dan Pathan menggunakan airborne hyperspectral sensing linear analis campuran spectral, bersama dengan campuran ke konverter murni dan norma Euclidean teknik untuk memetakan masing-masing pohon kelapa sawit (K. Jusoff, 2009). Korom et all tahun 2014 mengelompokkan bentuk kanopi atau mahkota kelapa sawit menggunakan citra WorldView-2 berdasarkan segmentasi watershed dan mencapai akurasi sekitar 77% (A. Korom et al, 2014).

Wong-in et al. tahun 2015 telah mencapai akurasi 90% menggunakan gambar udara dengan beberapa langkah, seperti menghapus komponen non-pohon dari gambar, membedakan minyak sawit dari komponen lain menggunakan filter low-pass dan korelasi silang dinormalisasi, mengidentifikasi pohon kelapa sawit secara individu dan menghitung jumlah pohon kelapa sawit (Wong-in et al, 2015).

H. Santoso et al. pada tahun 2016 membangun dan mengembangkan metode yang mudah digunakan pengguna yang akan memungkinkan manager kelapa sawit untuk menghitung minyak pohon kelapa sawit menggunakan

teknik penginderaan jauh. Pohon kelapa sawit dianalisis dalam penilitian ini dengan melihat usia dan kepadatan yang berbeda. Penelitian ini menggunakan Citra QuickBird yang diaplikasikan dengan enam metode pansharpening. Hitam dan citra putih dari komposit warna palsu citra pansharpening diproses dalam tiga cara: (1) deteksi pohon kelapa sawit, (2) penggambaran area kelapa sawit, (3) penghitungan pohon kelapa sawit dan penilaian akurasi. Penelitian ini menggunakan ENVI 5.2, ERDAS Imagine 2015, dan ArcGIS 10.2.2. Hasil penelitian ini meningkat pada akurasi dari beberapa studi penelitian sebelumnya yang memiliki akurasi 90-95%. Hasil dalam penelitian ini menunjukkan (1) gabungan resolusi intensitas huesaturation (HIS) cocok untuk kelapa sawit berusia 16 tahun, pohon dan memiliki kepadatan agak tinggi dengan akurasi 100%, (2) untuk kelapa sawit berusia 21 tahun dan memiliki kepadatan rendah didapatkan hasil dengan akurasi 99,5%, (3) resolusi subtraktif penggabungan ini cocok untuk pohon kelapa sawit berusia 15-18 tahun dan memiliki kerapatan yang agak tinggi dengan akurasi 99,8%, (4) penajaman spectral PC dengan akurasi 99,3% cocok untuk pohon kelapa sawit berusia 10 tahun dan memiliki kepadatan rendah, dan (5) untuk semua kondisi objek studi, warna dinormalisasi (Brovey) dan penggabungan resolusi wavelet adalah dua metode pansharpening yang cocok untuk ekstraksi dan kelapa sawit dengan penghitungan masing-masing dengan akurasi 98,9% dan 98,4% (H. Santoso et al, 2016).

H. M. Rizeei et al. tahun 2018 meneliti bahwa pemantauan karakteristik kelapa sawit di area perkebunan sangat berharga bagi petani dan stakeholders untuk memaksimalkan produktivitas. Penelitian ini mengusulkan metode baru untuk estimasi dan penghitungan usia kelapa sawit. Algoritma Support Vector Machine (SVM) dari analisis gambar berbasis objek (OBIA) diterapkan untuk penghitungan kelapa sawit. Analisis sensitivitas dilakukan pada empat jenis kernel SVM (Sigmoid (SIG), Linear (LN), fungsi basis radial (RBF), dan

polynomial (PL)) dengan parameter terkait (Nilai ambang batas, Gamma (γ) dan faktor Penalti (c)) untuk mendapatkan pendekatan klasifikasi OBIA optimal untuk setiap blok perkebunan. Citra dengan resolusi sangat tinggi dari Worldview-3 (WV-3) digunakan untuk deteksi kelapa sawit. Hasil deteksi kelapa sawit memiliki akurasi keseluruhan 98,27%, 99,48%, 99,28%, 99,49%, dan 97,49% untuk blok A, B, C, D, dan E (H. M. Rizeei, 2018).

W Li et al tahun 2018 meneliti deteksi pohon kelapa sawit dalam skala besar dari gambar satelit beresolusi tinggi (QuickBird) menggunakan Two-Stage Convolutional *Neural Networks* (TS-CNN). TS-CNN terdiri dari satu CNN untuk klasifikasi tutupan lahan dan satu CNN untuk klasifikasi objek. Kedua CNN dilatih dan dioptimalkan secara independen berdasarkan pada 20.000 sampel yang dikumpulkan melalui interpretasi manusia. Skala area pohon kelapa sawit sebesar 55 km². Penelitian ini mengusulkan alur kerja yang efektif yang terdiri dari metode partisi yang tumpeng tindih untuk divisi gambar skala besar, metode *multi-scale sliding window* untuk memprediksi koordinat kelapa sawit, dan metode filter jarak untuk pasca-pemrosesan. Pendekatan yang diusulkan mencapai skor F1 rata-rata yang jauh lebih tinggi yaitu 94,99% di wilayah studi dibandingkan dengan metode deteksi kelapa sawit yang ada masing-masing sebesar (87,95%, 81,80%, 80,61%, dan 78,35% untuk CNN satu-tahap, *Support Vector Machine* (SVM), *Random Forest* (RF), dan *Artificial Neural Network* (JST)), dan jauh lebih sedikit kebingungan dengan vegetasi dan bangunan lain di seluruh hasil deteksi gambar (W. Li et al, 2019).

N. A. Mubin et al. tahun 2019 telah meneliti bahwa deteksi dan penghitungan kelapa sawit penting dalam pengelolaan perkebunan kelapa sawit. Dalam penelitian ini, digunakan pendekatan pembelajaran mendalam untuk memprediksi dan menghitung kelapa sawit dalam citra satelit. Deteksi kelapa sawit sebelumnya secara umum berfokus pada mendeteksi kelapa sawit

yang tidak memiliki tumpeng tindih mahkota. Selain itu, terdapat kurangnya penelitian yang membangun sistem deteksi terpisah untuk kelapa sawit muda dan dewasa, memanfaatkan pendekatan pembelajaran mendalam untuk mendeteksi kelapa sawit dan menggabungan sistem informasi geografis (SIG) dengan pendekatan pembelajaran mendalam. Penelitian ini mencoba untuk mengisi kesenjangan ini dengan memanfaatkan dua jaringan saraf convolution (CNN) yang berbeda untuk mendeteksi kelapa sawit muda dan matang secara terpisah dan menggunakan GIS selama pemrosesan data dan proses penyimpanan hasil. Arsitektur awal yang dikembangkan didasarkan pada CNN yang disebut LeNet. Proses pelatihan mengurangi kerugian dengan menggunakan algoritma gradient adaptif dengan batch mini ukuran 20 untuk semua set pelatihan yang digunakan. Kemudian, mengekspor hasil prediksi ke perangkat lunak GIS dan membuat peta prediksi kelapa sawit dewasa dan muda. Berdasarkan metode yang diusulkan, akurasi keseluruhan untuk kelapa sawit muda dan matang masing-masing adalah 95,11% dan 92,96%. Secara keseluruhan, pengklasifikasi bekerja dengan baik pada dataset yang sebelumnya tidak terlihat, dan mampu secara akurat mendeteksi kelapa sawit dari latar belakang, termasuk bayangan tanaman lainnya (N. A. Mubin, 2019).

1.2 Batasan Penelitian

Ruang lingkup yang menjadi penelitian adalah:

1. Dataset citra sebagai dataset primer berjumlah 69 citra dari Kebun Pendidikan dan Penelitian Kelapa Sawit IPB-Cargil, Jonggol, Jawa Barat dan 205 citra dari Universitas Gunadarma, PPU Campus, Kalimantan Timur yang ditanami pohon kelapa sawit.
2. Dataset sekunder digunakan melalui akses roboflow secara free sebanyak 1795 citra dataset sekunder yang sudah memiliki label atau anotasi kelas

'oil palm'.

3. Pengujian penerapan sistem menghitung dan mendeteksi pohon kelapa sawit dilakukan pada lahan pohon kelapa sawit tanah mineral Kebun Pendidikan dan Penelitian Kelapa Sawit IPB-Cargil, Jonggol Jawa Barat.

1.3 Rumusan Masalah

Adapun rumusan masalah dalam penelitian ini secara umum bahwa pengelolaan perkebunan kelapa sawit memiliki permasalahan yang utama, diantaranya luas area pengelolaan kebun kelapa sawit dalam skala besar (perusahaan perkebunan kelapa sawit memiliki luas minimum sebesar 6.000 (enam ribu) hektar dan petani 4 hektar yang harus dikelola). Perkebunan berada di *remote area*, area yang terpencil, daerah pedalaman atau pelosok hutan, yang sulit untuk dilakukan monitoring, dan akses infrastruktur terutama jalan yang terbatas dan sulit. Permasalahan yang dihadapi juga berupa pemupukan presisi yang mengalami kesulitan untuk mendapatkan secara akurat, dan cepat, serta *up-to-date* jumlah tegakkan pohon kelapa sawit pada suatu luasan lahan tertentu, sementara praktik yang ada adalah menggunakan perhitungan manual dan normatif berbasis jarak tanam pohon kelapa sawit. Hal ini menyebabkan pentingnya data jumlah pohon kelapa sawit sangat penting dibutuhkan karena produksi per individu tanaman berkontribusi besar terhadap produktivitas dengan penyelesaian teknologi informasi. Berdasarkan permasalahan tersebut, maka rumusan masalah secara khusus:

1. Bagaimana membuat pengembangan metode *object-based image analysis* (OBIA) untuk membuat dataset secara otomatis dengan memberikan anotasi atau label suatu kelas pada data citra?
2. Bagaimana mendapatkan model CNN yang baik dari algoritma YOLOv5,

YOLOv6, dan YOLOv7 untuk mendekripsi dan menghitung pohon kelapa sawit pada suatu area tertentu?

3. Bagaimana mengembangkan sistem berbasis web dengan menerapkan model CNN dari algoritma YOLO agar dapat mendekripsi, menghitung, dan mendapatkan letak koordinat latitude-longitude setiap pohon kelapa sawit dari citra satelit agar dapat dilakukan pemantauan, pengelolaan, dan estimasi produktivitas secara otomatis?

1.4 Tujuan Penelitian

Adapun tujuan dari penelitian ini secara umum adalah menghasilkan purwarupa sistem berbasis web dengan menerapkan model OBIA dan *Deep Learning* (DL) untuk mendekripsi, menghitung jumlah pohon dalam suatu luasan tertentu dan mendapatkan nilai koordinat latitude-longitude setiap pohon kelapa sawit dari citra satelit. Tujuan penelitian secara khusus sebagai berikut:

1. Menghasilkan metode object-based image analysis (OBIA) untuk membuat dataset secara otomatis dengan memberikan anotasi atau label suatu kelas, kelas 'oil palm' pada data citra.
2. Menghasilkan model CNN dari algoritma YOLOv5, YOLOv6, dan YOLOv7 yang dapat mendekripsi dan menghitung pohon kelapa sawit pada suatu area tertentu.
3. Menghasilkan purwarupa perangkat lunak aplikasi dengan model convolutional neural network dari algoritma YOLO yang terintegrasi dengan citra satelit dengan menggunakan Google Maps API yang dapat mendekripsi, menghitung, dan mendapatkan letak koordinat latitude-longitude setiap pohon kelapa sawit pada suatu area tertentu.

1.5 Kontribusi Hasil Penelitian

Kontribusi penting dari penelitian yang dilakukan dalam penelitian ini adalah:

1. Dari sisi keilmuan, berupa pengembangan metode *object-based image analysis* (OBIA) untuk membuat dataset secara otomatis dengan memberikan anotasi atau label suatu kelas pada data citra dengan menggunakan algoritma *template matching* dan BIRCH.
2. Dari sisi teknologi, penelitian ini menghasilkan dataset primer dengan kelas pohon kelapa sawit (*oil palm*) dan purwarupa perangkat lunak aplikasi dengan model *convolutional neural network* dari algoritma YOLO yang terintegrasi dengan citra satelit dengan menggunakan Google Maps API yang dapat mendeteksi, menghitung, dan mendapatkan letak koordinat latitude-longitude setiap pohon kelapa sawit yang berhasil didekksi.
3. Dari sisi pemanfaatan, menawarkan suatu cara alternatif untuk proses persiapan data citra untuk mendapatkan dataset dengan proses anotasi yang lebih cepat, serta bagi *stakeholder* dapat memantau, mengelola, dan melakukan estimasi produktivitas secara otomatis pohon kelapa sawit pada suatu area tertentu dengan penerapan teknologi informasi.

BAB II

TELAAH PUSTAKA

2.1 *Computer Vision*

Computer Vision atau visi komputer adalah bidang kecerdasan buatan (AI) yang memungkinkan komputer dan sistem memperoleh informasi bermakna dari gambar digital, video, dan input visual lainnya, dan mengambil tindakan atau membuat rekomendasi berdasarkan informasi tersebut. Jika kecerdasan buatan memungkinkan komputer untuk berpikir, visi komputer memungkinkan untuk melihat, mengamati, dan memahami (IBM, 2020).

Computer vision bekerja hamper sama dengan visi manusia, kecuali manusia memiliki permulaan. Penglihatan manusia memiliki keunggulan konteks seumur hidup untuk melatih cara membedakan objek, seberapa jauh jaraknya, apakah bergerak, dan apakah ada yang salah dalam sebuah gambar.

Computer vision melatih mesin untuk melakukan fungsi-fungsi melatih mesin untuk melakukan fungsi-fungsi ini, tetapi ia harus melakukannya dalam waktu yang jauh lebih singkat dengan kamera, data, dan algoritma daripada retina, saraf, optic, dan korteks visual karena sistem yang dilatih untuk memeriksa produk atau mengamati asset produksi dapat menganalisis ribuan produk atau proses dalam satu menit, memperhatikan cacat atau masalah yang tidak terlihat, sistem tersebut dapat dengan cepat melampaui kemampuan manusia.

Memahami dan menentukan tugas visi komputer tertentu dapat memfokuskan dan memvalidasi proyek dan aplikasi, serta mempermudah untuk memulai. Hal dasar untuk *computer vision* adalah *object detection*, beberapa tugas visi komputer lainnya, seperti *image classification*, *object detection*,

object tracking, dan *content-based image retrieval* (U. Arshad, 2021).

2.2 Pengertian Citra

Citra didefinisikan sebagai fungsi dari dua variabel misalnya $a(x,y)$ dimana a sendiri sebagai amplitudo (misalnya kecerahan) citra pada koordinat (x,y) (I. T. Young et al, 1995). Citra digital $a[m,n]$ merupakan citra dalam ruang diskrit 2D yang berasal dari citra analog $a(x,y)$ di ruang kontinyu 2D melalui proses sampling yaitu yang biasa disebut sebagai digitalisasi. Sedangkan, menurut Maria citra digital adalah citra $f(x,y)$ yang telah didiskritkan oleh pada koordinat spasial dan kecerahan. Citra digital direpresentasikan oleh *array* dua dimensi atau sekumpulan *array* dua dimensi dimana setiap *array* merepresentasikan satu kanal warna. Nilai kecerahan yang didigitalkan dinamakan nilai tingkat keabuan (A. McAndrew, 2004).

Setiap elemen *array* tersebut dinamakan piksel yang diambil dari istilah *picture element*. Dimensi citra biasanya ditulis dengan format panjang x tinggi (misalnya 640 x 480 piksel). Namun, perlu diperhatikan dengan seksama bahwa secara matematis, definisi citra terlihat seperti di bawah ini, dimana x menunjukkan baris dan y menunjukkan kolom:

$$f(x,y) = \begin{bmatrix} f(0,0) & f(0,1) & \dots & f(0, N-1) \\ f(1,0) & f(1,1) & \dots & f(1, N-1) \\ \dots & \dots & \dots & \dots \\ f(M-1,0) & f(M-1,1) & \dots & f(M-1, N-1) \end{bmatrix}$$

Seperti pada layar monitor, koordinat citra dimulai dari pojok kiri atas. Secara matematis dimulai dari $(0,0)$ dan berakhir di $(M-1, N-1)$, dimana M

menunjukkan tinggi, dan N menunjukkan panjang.

2.3 Pengolahan Citra

Pengolahan citra adalah pemrosesan citra, khususnya menggunakan komputer menjadi citra yang kualitasnya lebih baik. Pengolahan citra adikembangkan bertujuan untuk (M. Petrou, 1999):

1. Untuk memperbaiki tampilan citra (*image enhancement*).
2. Untuk mengurangi ukuran file citra dengan tetap mempertahankan kualitas citra (*image compression*).
3. Untuk memulihkan citra ke kondisi semula (*image restoration*).
4. Untuk menyoroti ciri tertentu dari citra agar lebih mudah untuk di analisis.

Pengolahan citra adalah cabang ilmu informatika untuk memperbaiki kualitas citra agar kualitasnya lebih baik atau lebih mudah diinterpretasi oleh manusia maupun komputer. Input dari program pengolahan citra adalah citra dan outputnya pun citra pula.

Pengolahan citra digital digunakan dalam berbagai bidang untuk mempermudah manusia dalam melakukan analisis dan pekerjaan. Bentuk aplikasi pengolahan citra digital yang digunakan bidang militer, industry, medis, transportasi, hukum dan keamanan, pemetaan, robotika, fotografi, film, pencarian gambar berdasarkan kandungan citra, dan pemahaman kandungan citra. Salah satu pemanfaatan teknologi pengolahan citra digital yaitu bisa memahami maksud dari sebuah citra. Apabila aplikasi diberikan input berupa gambar yang mampu mendefinisikan bahwa dalam gambar tersebut terdapat gambar yang mendapatkan objek, seperti kendaraan, jalan, buah, dan lainnya.

2.4 Artificial Intelligence (AI)

Artificial intelligence atau kecerdasan buatan adalah studi tentang teori dan pengembangan sistem komputer agar mampu melakukan tugas-tugas yang dahulu hanya dapat dilakukan oleh manusia. Seperti membandakan berbagai gambar, menjawab pertanyaan, mengenali dan menerjemahkan bahasa, dan sebagainya (R. Primartha, 2018).

Komputer atau mesin cukup bagus untuk melakukan hal-hal berikut: menyelesaikan perhitungan aritmatika dengan cepat, mengerjakan secara akurat apa-apa yang sudah deprogram oleh komputer. Namun, komputer atau mesin memiliki kelemahan, seperti sulit berinteraksi dengan *noisy data* (data yang blur/bias), sulit memahami lingkungan, kurang toleran terhadap kesalahan (*fault tolerance*), sulit beradaptasi dengan situasi dan kondisi tertentu. Untuk mengatasi hal tersebut ada lima hal yang perlu dimiliki oleh mesin, yaitu:

1. Persepsi

Terkait dengan permasalahan pengindraan. Mesin harus memiliki indra untuk dapat mengenali dunia sekitarnya.

2. Pemrosesan bahasa alami (NLP)

Kemampuan untuk mengidentifikasi kalimat dan memahami perbedaan aksesn dan maknanya.

3. Menyampaikan pengetahuan

Menyampaikan berbagai informasi di dunia luar berdasarkan pemikirannya sendiri.

4. Pengambilan keputusan

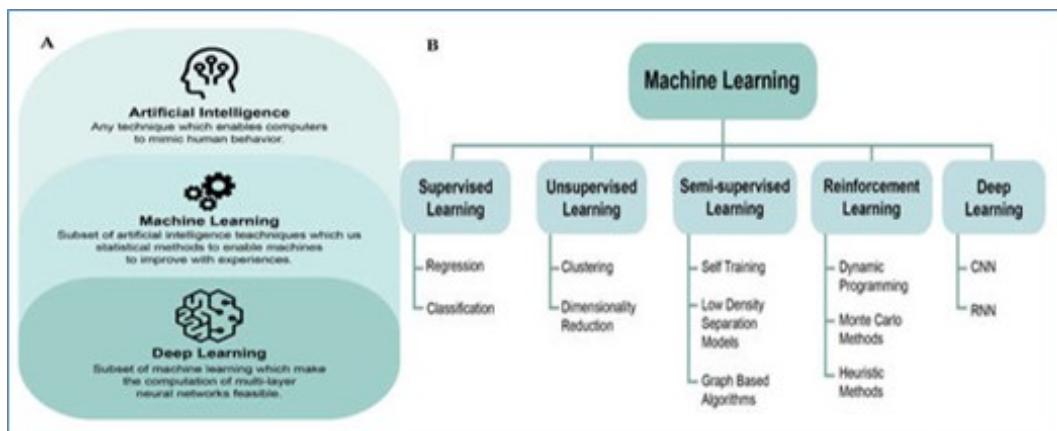
Mampu memecahkan berbagai permasalahan secara logis.

5. Perencanaan dan pemetaan

Memetakan dunia tiga dimensi dan merencanakan rute paling efektif.

Salah satu bagian penting dari AI adalah machine learning atau pembelajaran

Salah satu bagian penting dari AI adalah *machine learning* atau pembelajaran mesin, yaitu dicirikan sebagai studi tentang algoritma dan model statistik yang digunakan sistem komputer untuk belajar dari data sampel dan pengalaman sebelumnya tanpa diprogram secara eksplisit untuk mencapai tugas tertentu. Dengan kemampuan untuk mengidentifikasi pola yang tidak jelas dalam data, kita dapat menggunakan pembelajaran mesin untuk memecahkan banyak masalah, termasuk menilai hubungan dua variabel, membuat prediksi berdasarkan karakteristik dasar, mengidentifikasi objek dengan pola yang sebanding, dan menggabungkan subjek dengan kriteria tertentu. Baik *Artificial Intelligence*, *Machine Learning*, dan *Deep Learning* merupakan tiga istilah yang popular, namun banyak yang salah mengira bahwa ketiganya menggambarkan hal yang sama, padahal tiga hal yang berbeda. Bagaimana hubungan tiga istilah ini, seperti pada Gambar 2.1



Gambar 2.1: (a) Hubungan Artificial Intelligence, Machine Learning, dan Deep Learning. (b) Jenis Machine Learning
(Sumber: C. Thongprayoon et al, 2020)

Disamping hal tersebut, *artificial intelligence* (AI) memungkinkan un-

tuk berfikir, dan salah satu bagian dari *artificial intelligence* untuk dapat melihat, mengamati, dan memahami adalah komputer visi atau *computer vision*. *Computer vision* memungkinkan untuk komputer dan sistem memberikan informasi berarti dari gambar digital, dan visual input.

2.5 Object Detection

Object detection atau deteksi objek dianggap sebagai salah satu bidang penting dalam pembelajaran mendalam dan visi komputer. Deteksi objek telah ditentukan oleh banyak aplikasi dalam visi komputer, seperti pelacakan objek, pengambilan, dan pengawasan video. Deteksi objek adalah teknologi *deep learning* dimana benda, manusia, bengunan, mobil, dapat dideteksi sebagai objek dalam gambar dan video (U. Arshad, 2021).

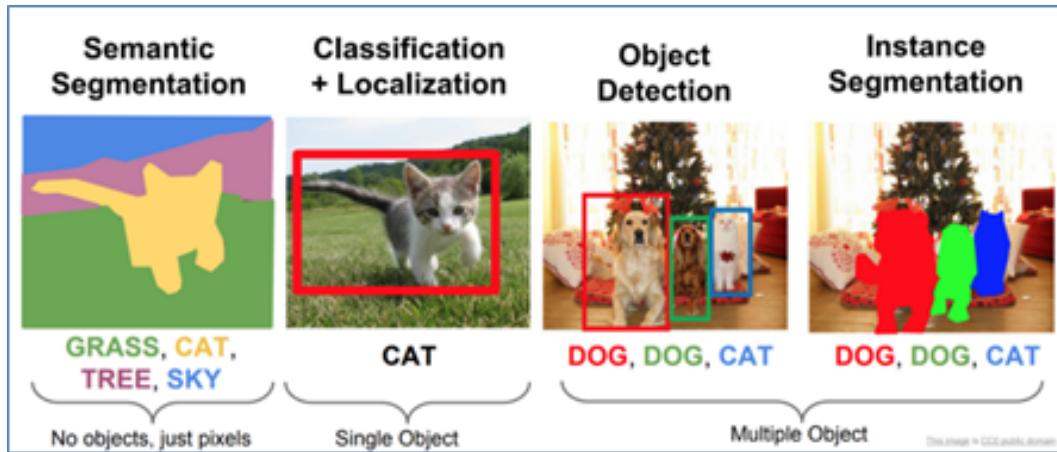
Deteksi objek untuk mengenali objek dengan kotak pembatas pada gambar, dimana dalam klasifikasi gambar, cukup mengkategorikan (mengklasifikasikan) objek pada gambar atau tidak dalam hal kemungkinan (*probability*), seperti contoh pada Gambar 2.2



Gambar 2.2: Classification, Object Detection dan Segmentation Representation
(Sumber: A. Patel, 2020)

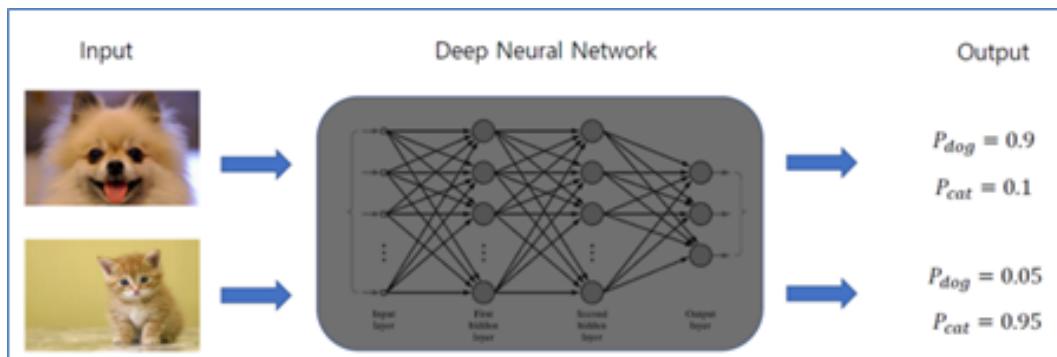
Pada Gambar 2.3, terlihat bahwa kucing (cat) dengan kotak pembatas dan tanpa kotak pembatas dapat membedakan mendasar antara klasifikasi

citra dan deteksi objek.



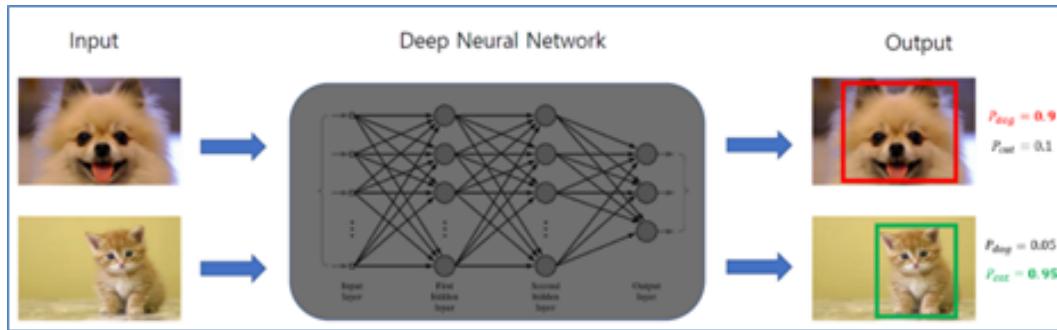
Gambar 2.3: Segmentation, Classification+Localization, Object Detection
(Sumber: A. Patel, 2020)

Dalam mempelajari deteksi objek, maka diperlukan mengetahui klasifikasi citra (*image classification*). Ketika gambar adalah input ke CNN, masalah mengklasifikasikan kelas yang sesuai dengan gambar dikenal sebagai klasifikasi gambar, dan seperti yang ditunjukkan pada Gambar 2.4, nilai probabilitas untuk semua kelas yang ditargetkan adalah keluaran.



Gambar 2.4: Image Classification
(Sumber: A. Patel, 2020)

Dapat juga dianggap bahwa deteksi objek sebagai masalah dimana tugas klasifikasi gambar memiliki tugas regresi yang memprediksi posisi objek menggunakan *bounding box* (kotak pembatas) pada Gambar 2.5.



Gambar 2.5: Object Detection dengan Bounding Box

(Sumber: A. Patel, 2020)

Masalah deteksi objek mengasumsikan bahwa beberapa kelas objek mungkin ada dalam gambar pada waktu yang sama. Dapat memvisualisasikan seperti dua jenis masalah, 1) klasifikasi multi label (beberapa kelas dalam satu gambar), 2) Bounding Box (masalah regresi) dimana harus memprediksi nilai koordinat kotak pembatas (*bounding box*) dalam bentuk x, y, w, h.

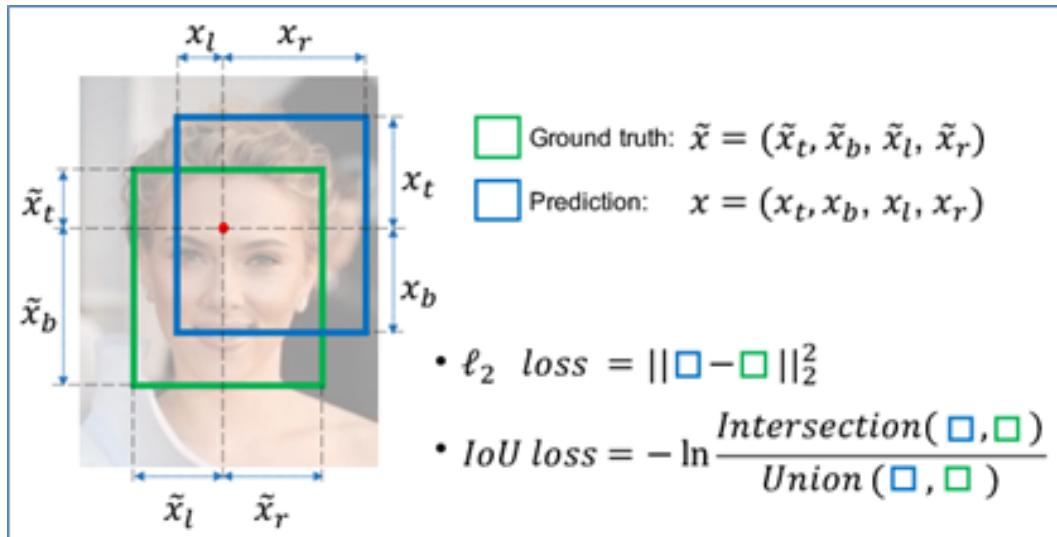
Dalam deteksi objek, terdapat *object localization* atau lokalisasi objek yang merupakan untuk memprediksi objek dalam sebuah citra serta batas-batasnya. Perbedaan antara lokalisasi objek dan deteksi objek tidak ken-tara. Sederhananya, lokalisasi objek bertujuan untuk menemukan objek utama (atau yang paling terlihat) dalam sebuah gambar, sedangkan deteksi objek mencoba untuk mengetahui semua objek dan batasannya.

Suatu klasifikasi citra atau model pengenalan citra hanya mendekripsi probabilitas suatu objek dalam suatu citra. Berbeda dengan ini, lokalisasi objek mengacu pada mengidentifikasi lokasi suatu objek dalam gambar. Algoritma lokalisasi objek akan menampilkan koordinat lokasi objek sehubungan dengan gambar. Dalam visi komputer, cara paling popular untuk melokalkan objek dalam gambar adalah dengan merepresentasikan lokasinya dengan bantuan kotak pembatas (*bounding box*).

Bounding box dapat diinisialisasi menggunakan parameter berikut:

- bx, by: koordinat pusat kotak pembatas (center of bounding box)

- bw: lebar kotak pembatas dengan lebar gambar (width)
- bh: tinggi kotak pembatas dengan tinggi gambar (height)



Gambar 2.6: Intersection Over Union (IoU)
(Sumber: A. Patel, 2020)

Dengan memprediksi ini, dapat menghitung Mean-IoU dan memprediksi kotak pembatas (*bounding box*) yang melokalkan objek di gambar.

- IoU adalah Intersection-Over-Union (IoU) disebut sebagai Indeks Jaccard (*Jaccard Index*) dianggap sebagai salah satu metrik kinerja yang paling banyak digunakan dalam deteksi objek.

$$\text{IoU} = \frac{\text{Area of Overlap}}{\text{Area of Union}}$$

Gambar 2.7: Persamaan IoU
(Sumber: A. Patel, 2020)

- IoU adalah area tumpang tindih (*overlap*) antara segmentasi yang diprediksi (*prediction*) dan kebenaran dasar (*ground truth*), seperti yang ditampilkan pada Gambar 2.7. Metrik ini bervariasi dari 0-1 (0-100%) dengan 0 menyiratkan tidak ada tumpeng tindih (sampah) dan 1 menandakan segmentasi yang tumpang tindih sempurna (*fat dub*).
- Mean IoU adalah segmentasi biner (dua kelas) atau multi-kelas, mean Io Udari gambar dihitung dengan mengambil IoU dari setiap kelas dan merata-ratakannya.

2.6 Machine Learning

Istilah *machine learning* mula-mula diperkenalkan oleh Arthur Samuel pada tahun 1959 melalui jurnalnya yang berjudul "Some Studies in Machine Learning Using the Game of Checkers". (IBM Journal of Research and Development). Samuel mencoba mengajari program komputer untuk bermain catur. Tujuannya adalah membuat agar komputer dapat bermain catur lebih baik dari dirinya. Pada tahun 1962 program buatannya dapat mengalahkan juara catur dari negara bagian Connecticut (R. Primartha, 2018).

Machine learning membutuhkan sebuah model yang didefinisikan berdasar parameter-parameter tertentu. Proses learning adalah eksekusi program komputer untuk mengoptimasi parameter-parameter dari model tersebut, dengan memanfaatkan data training atau *past experience*.

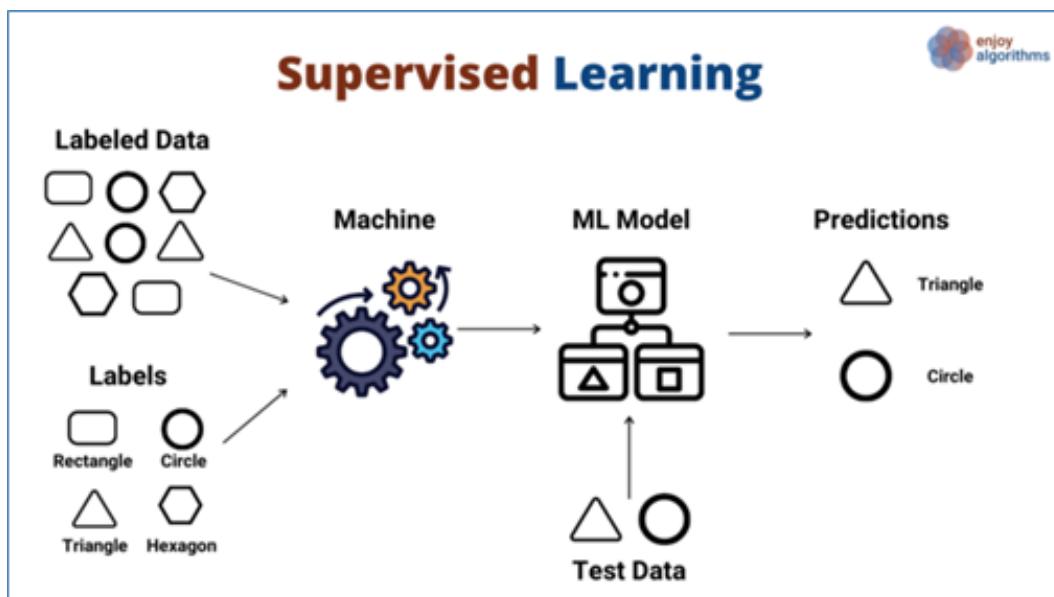
Jadi, secara sederhana dapat dijelaskan bahwa *machine learning* adalah pemrograman komputer untuk mencapai kriteria/performa tertentu dengan menggunakan sekumpulan data training atau pengalaman di masa lalu (*past experience*). *Machine learning* mempelajari teori agar komputer mampu "belajar" dari data.

Secara umum algoritma *machine learning* dapat dikelompokkan menjá-

di lima bagian, yaitu *supervised learning*, *unsupervised learning*, *semi-supervised learning*, *reinforcement learning*, dan *deep learning*.

2.6.1 Supervised Learning

Sebagian besar praktik machine learning mengandalkan algoritma *supervised learning*. Algoritmanya dinamakan seperti ini karena training dataset (sekumpulan data untuk training) akan memandu dan mengajari komputer agar menghasilkan outcome sesuai harapan. Pada *supervised learning* menggunakan sebuah algoritma untuk mempelajari mapping function antara input dengan output. Berbagai kemungkinan output sudah diketahui dan data-data yang digunakan untuk latihan (training) sudah diberi label dengan jawaban yang benar. *Supervised learning* dapat bermanfaat untuk memprediksi sesuatu dengan bantuan training dataset. Berikut skema *supervised learning* pada Gambar 2.8 (R. Primartha, 2018).



Gambar 2.8: Skema Supervised Learning
(Sumber: M. Kozan, 2021)

Supervised learning menggunakan training data yang sudah diberi label untuk mempelajari *mapping function*, dari input variables (x) ke output

variables (y).

$$y = f(x)$$

Sebagai contoh, sebuah algoritma klasifikasi akan dapat mengidentifikasi berbagai bentuk bangun setelah melalui proses belajar dari sekumpulan bangun datar yang sudah ditandai atau diberi label dengan ciri tertentu seperti pada gambar 2.12.

Permasalahan-permasalahan yang terkait dengan *supervised learning* dapat dikategorikan menjadi dua jenis:

1. *Classification*

Klasifikasi bertujuan untuk memprediksi outcome dari input (sample yang diberikan), dimana output variabel berbentuk kategori-kategori. Contoh: pria/wanita, sakit/sehat, tinggi/rendah, dan sebagainya.

2. *Regression*

Regression bertujuan untuk memprediksi outcome dari input (sample yang diberikan), dimana output, variabel berbentuk nilai aktual (*real values*). Contoh: prediksi harga rumah, tinggi badan seseorang, curah hujan, dan sebagainya.

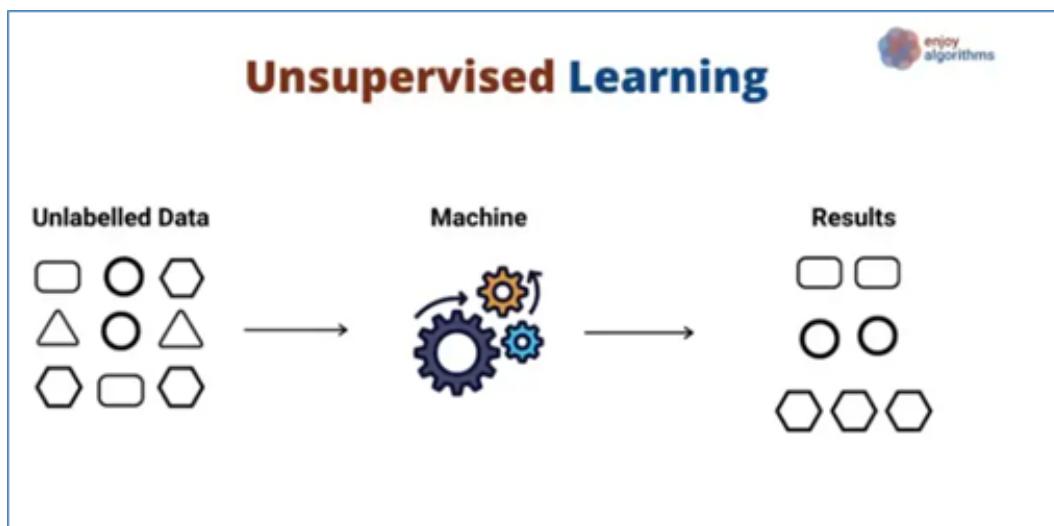
Ada beberapa algoritma yang sudah dikembangkan dan terkait dengan *supervised learning*, diantaranya:

1. Decision tree,
2. Naïve Bayes Classifier,
3. Artificial Neural Network,
4. Support Vector Machine,
5. Linear Regression,

6. Logistic Regression,
7. CART,
8. KNN (-KNearest Neighbor), dsb.

2.6.2 Unsupervised Learning

Berbeda dengan supervised learning, pada *unsupervised learning* persoalan diproses hanya mengandalkan data yang belum dilatih sebelumnya. *Unsupervised learning* menggunakan *unlabeled training dataset* untuk memodelkan struktur dari data, sehingga unsupervised learning bersifat lebih subjektif dibandingkan *supervised learning*. Berikut skema *unsupervised learning* pada Gambar 2.9 (R. Primartha, 2018).



Gambar 2.9: Skema Unsupervised Learning
(Sumber: M. Kozan, 2021)

Unsupervised learning bermanfaat untuk kasus-kasus dimana kita ingin menemukan relasi implisit (implicit relationships) dari *unlabeled dataset* yang disediakan. Jadi, pada *unsupervised learning* kita tidak memprediksi masa depan, sebab input variable (X) tidak memiliki relasi dengan output variabel

(Y).

$$f(x)$$

Untuk memudahkan memahaminya, dapat diasumsikan saat ini belum pernah membeli majalah sama sekali. Suatu ketika membeli beberapa buah majalah dan ingin membaginya menjadi beberapa kategori, dengan tujuan agar nantinya mudah dicari. Maka, dapat dimulai dengan mengidentifikasi majalah-majalah berdasarkan kemiripan. Misalnya, berdasarkan isi, penerbit, dan lain-lain yang bisa ditentukan sesuai kebutuhan.

Permasalahan seputar *unsupervised learning* dapat dikelompokkan menjadi tiga kategori, yaitu:

1. Association

Association bertujuan untuk menemukan peluang (probabilitas) berdasarkan keterkaitan (co-occurrence) dari item-item dalam sebuah kumpulan. Sebuah contoh, jika customer membeli teh celup, maka kemungkinan besar (sekitar 80

2. Clustering

Clustering bertujuan untuk mengelompokkan sample dalam cluster yang sama berdasarkan kemiripan (similarity).

3. Dimensionality Reduction

Dimensionality Reduction berarti mengurangi sejumlah variabel dari dataset namun tetap memastikan informasi yang penting masih tersedia. Dimensionality Reduction dapat diwujudkan menggunakan metode:

(a) Feature Extraction

Melakukan transformasi data dari dimensi tinggi (a high-dimensional space) ke dimensi yang lebih rendah (a low-dimensional space).

(b) Feature Selection

Memilih Sebagian saja (subset) dari variabel asal (original variabel).

Beberapa algoritma yang dikelompokkan dalam *unsupervised learning*, antara lain:

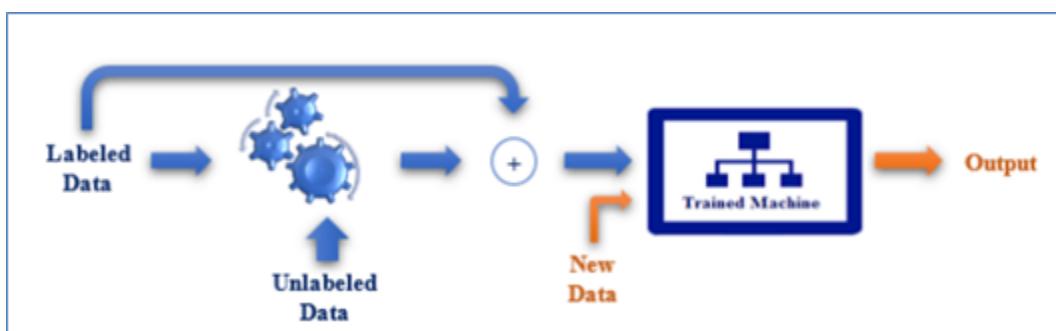
1. K-Means
2. Hierarchical Clustering
3. DBSCAN
4. Fuzzy C-Means
5. Self-Organizing Map, dan sebagainya.

2.6.3 Semi-supervised learning

Metode ini berada diantara yang *supervised* dan *unsupervised learning* di mana memiliki sejumlah besar masukan data, beberapa di antaranya diberi label dan sisanya tidak. Banyak masalah pembelajaran kehidupan nyata termasuk dalam bidang pembelajaran mesin ini. Alasannya adalah *semi-supervised* membutuhkan lebih sedikit intervensi manusia karena menggunakan data berlabel dalam jumlah yang sangat kecil dan data yang tidak berlabel dalam jumlah besar. Memanfaatkan kumpulan data yang kurang berlabel lebih menarik karena kumpulan data tersebut sangat sulit untuk dikumpulkan serta mahal dan mungkin memerlukan akses ke pakar domain. Dataset yang tidak berlabel di sisi lain lebih murah dan lebih mudah diakses (X. Zhu, 2018).

Kedua teknik pembelajaran *supervised* dan *unsupervised learning* bisa digunakan untuk melatih algoritma pembelajaran dalam pembelajaran *semi-supervised*. Teknik *unsupervised learning* dapat digunakan untuk mengungkap struktur dan pola tersembunyi dalam kumpulan data input. Sedangkan teknik

supervised learning dapat digunakan untuk membuat prediksi tebakan pada data yang tidak berlabel, memasukkan data kembali ke algoritma pembelajaran sebagai data pelatihan, dan menggunakan pengetahuan yang diperoleh untuk membuat prediksi pada kumpulan data baru. Dengan demikian, dapat mengatakan bahwa data yang tidak berlabel digunakan untuk memodifikasi atau memprioritaskan kembali prediksi atau hipotesis yang diperoleh dari data yang berlabel. Gambar 2.10 mengilustrasikan berbagai tahapan metode *semi-supervised learning*.



Gambar 2.10: Skema Semi-Supervised Learning
(Sumber: (?)

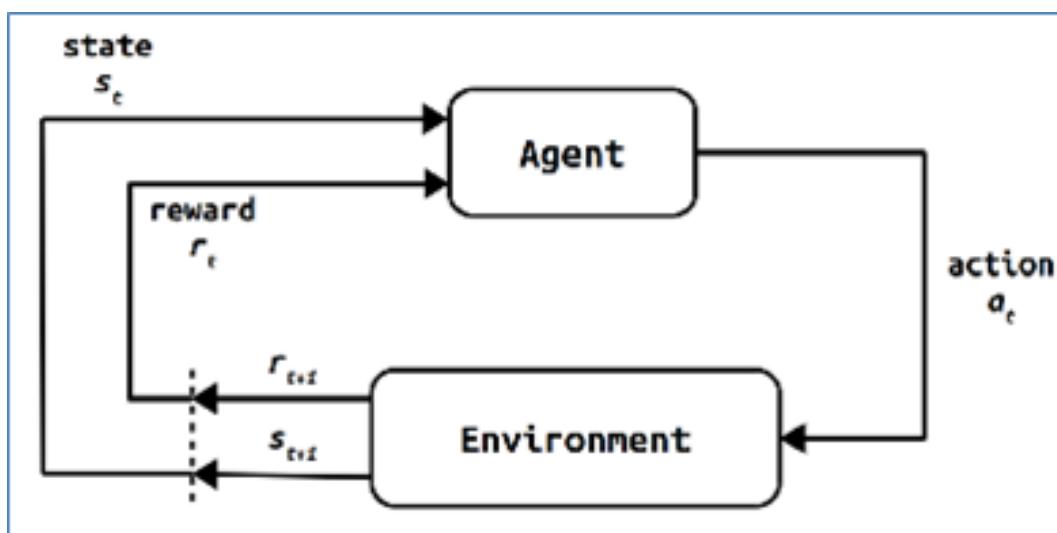
Untuk memanfaatkan data pelatihan yang tidak berlabel, semua algoritma *semi-supervised learning* melakukan setidaknya satu dari asumsi berikut asumsi kehalusan, asumsi cluster, dan asumsi manifold.

2.6.4 Reinforcement Learning

Reinforcement learning merupakan metode pembelajaran yang dipengaruhi oleh feedback dari lingkungan dengan Teknik pembelajaran yang iterative (berulang-ulang) dan adaptive (menyesuaikan). *Reinforcement learning* dipercaya mendekati cara manusia belajar (R. Primartha, 2018).

Reinforcement learning (RL) diinspirasi oleh kebiasaan makhluk hidup dalam belajar dan bertindak, khususnya manusia. Pada RL tidak ada dataset. Data-data diperoleh berdasarkan pengalaman. Algoritma *reinforcement learning*

ing mengijinkan agent untuk memutuskan aksi selanjutnya berdasarkan kondisi saat ini (*current state*). *Reinforcement learning* kadang disebut juga *credit assessment learning*, sebab learning difokuskan untuk memaksimalkan perolehan *rewards*. *Reinforcement learning* tergantung pada proses coba-coba untuk mengungkap rangkaian tindakan yang memaksimalkan metrik imbalan kumulatif, yang digunakan untuk membuat algoritma memahami apakah itu mengearah ke arah yang benar atau tidak. Berikut skema *reinforcement learning* pada Gambar 2.11.



Gambar 2.11: Skema Reinforcement Learning

(Sumber: R. Sutton, 1998)

Menurut R. Sutton, proses *reinforcement learning* dapat dipresentasikan dalam matematika sebagai *Markov Decision Process* (MDP), memperkenalkan 4 set S, A, P, R, dengan:

S - Kumpulan status tempat agen dapat berada, pada saat tertentu;

A - Serangkaian kemungkinan Tindakan yang dapat dilakukan agen dalam waktu tertentu;

P - Himpunan probabilitas, bahwa sebuah agen, yang berada dalam keadaan s, bertransisi ke keadaan s' dengan melakukan Tindakan A dalam waktu t+1;

Representasi *reinforcement learning* mirip dengan supervised learning.

Yang membedakan adalah pada reinforcement learning tidak hanya x, namun x dan z.

$$y = f(x) \text{ given } z$$

Tidak seperti *supervised* dan *unsupervised learning* dimana algoritma sudah memiliki tujuan (goal). Algoritma *reinforcement learning* tidak memiliki tujuan eksplisit, sebagai gantinya algoritma dipaksa untuk belajar menemukan nilai optimal melalui kegiatan trial dan error.

Reinforcement learning banyak diimplementasikan pada *game theory*, *control theory*, *operation research*, *information theory*, *simulation-based optimization*, *multi-agent systems*, *swarm intelligence*, *statistics*, dan *genetic algorithm*.

Contoh penerapan *reinforcement learning* yaitu pada bidang robotic. Sebuah robot dapat belajar untuk menghindari tabrakan dengan cara menerima feedback negative manakala robot tersebut menabrak halangan tertentu. Robot akan dibiarkan berjalan tanpa dipandu. Robot akan belajar dari pengalaman sebelumnya untuk menemukan rute paling optimal.

Beberapa algoritma yang dikelompokkan dalam *reinforcement learning* antara lain:

1. Genetic Algorithm (GA)
2. Dynamic Programming (DP)
3. Generalized Policy Iteration (GPI)
4. Monte Carlo Methods

2.6.5 Deep Learning

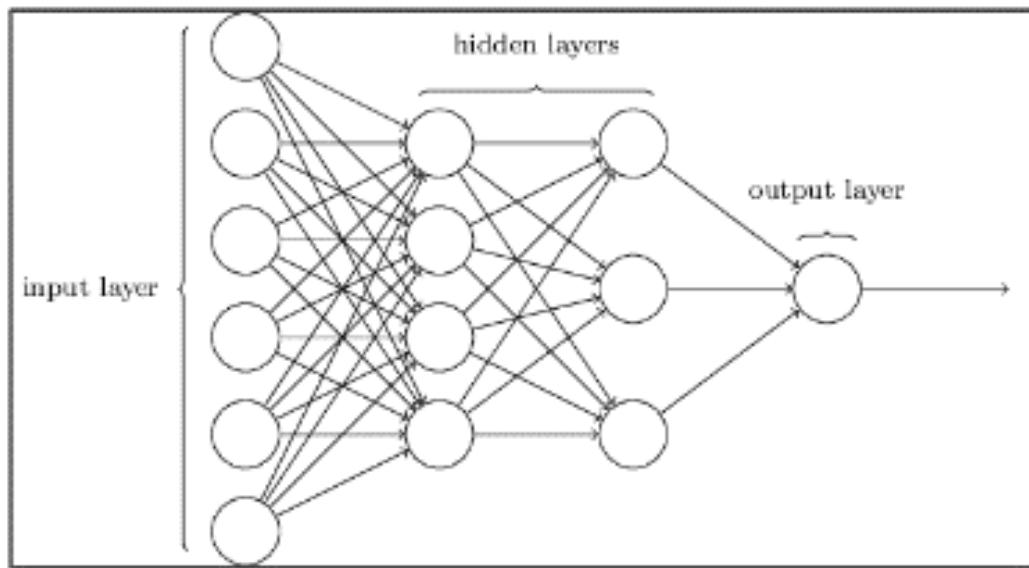
Deep learning merupakan metode pembelajaran yang memanfaatkan *artificial neural networks* yang berlapis-lapis (multi layer). *Artificial neu-*

ral networks ini dibuat mirip dengan otak manusia, di mana neuron-neuron terkoneksi satu sama lain, sehingga membentuk sebuah jaringan neuron yang sangat rumit (R. Primartha, 2018).

Deep learning atau *deep structured leaning* atau *hierarchical learning* atau *deep neural* merupakan metode pembelajaran yang memanfaatkan multiple non-linear transformation. *Deep learning* dapat dipandang sebagai gabungan machine learning dengan *artificial intelligence* (AI). *Deep learning* pada hakekatnya merupakan perluasan atau pengembangan dari *neural network* atau jaringan saraf tiruan (JST).

Jika dikembalikan kepada tujuan *machine learning* semula, yaitu komputer yang dapat belajar (dari data atau pengalaman), maka *deep learning* adalah apa yang selama ini dicari. *Deep learning* menirukan cara berpikir manusia. Pada *deep learning*, komputer harus memproses data yang sangat banyak, berlapis-lapis, dan output dari layer sebelumnya akan menjadi input bagi layer sesudahnya.

Struktur umum dan dasar dari skema *deep learning* ditunjukkan pada Gambar 2.12. Ini terdiri dari lapisan masukan, yang merupakan data masukan ke algoritma; lapisan tersembunyi, di mana algoritma membuat banyak perhitungan matematis, dan lapisan output, yang merupakan hasil dari perhitungan algoritma.

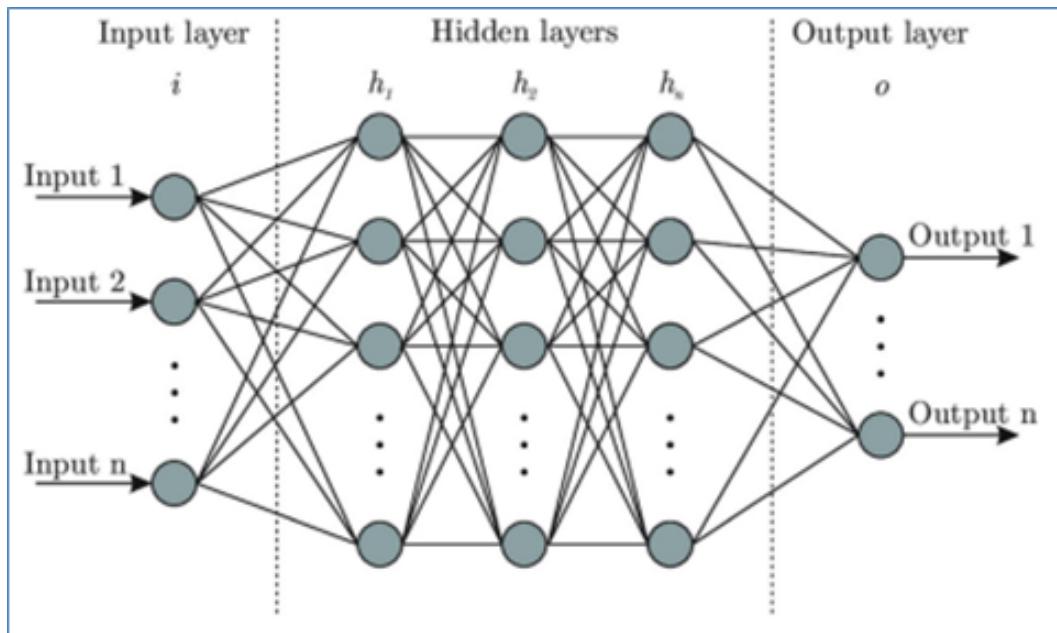


Gambar 2.12: Skema Umum *Deep Learning*
(Sumber: A. Feizollah et al, 2022)

Sejarah *deep learning* dimulai pada tahun 2006, yaitu setelah Geoffrey Hinton mempublikasikan paper yang memperkenalkan salah satu varian *neural networks* yang disebut *deep belief nets*. Paper ini merupakan awal kemunculan istilah *deep learning*, untuk membedakan arsitektur *neural network* konvensional (*single layer*) dengan arsitektur neural network multi/banyak layer. Dengan kata lain, *deep learning* adalah salah satu cabang machine learning yang menggunakan *deep neural network* untuk menyelesaikan permasalahan pada domain *machine learning*.

Pada tahun 2009, Andrew memperkenalkan penggunaan GPU untuk *deep learning* melalui paper yang berjudul *large-scale deep unsupervised learning using graphics processors*. Dengan menggunakan GPU, algoritma *deep learning* dapat dijalankan lebih cepat dibanding dengan tanpa GPU (hanya menggunakan CPU). Perkembangan *deep learning* maju pesat berkat keberadaan *hardware* yang memadai. Dan saat ini, *deep learning* sudah banyak diaplikasikan di berbagai area, seperti pengenal wajah, *self-driving car*, pengenal suara, dan sebagainya.

Deep learning merupakan jalan untuk mencapai apa yang sudah dicita-citakan sebelumnya oleh manusia, yaitu kecerdasan buatan bagi mesin. Bentuk diagram *network model deep learning* seperti pada Gambar 2.13. Perhatikan bahwa *hidden layer* hanya digambarkan tiga lapis saja, padahal kenyataannya bisa berjumlah sangat banyak, dapat diasumsikan seperti Gambar 2.13.



Gambar 2.13: Skema *Deep Learning* dengan Penambahan beberapa *hidden layer*

(Sumber: H. Kaur et al, 2021)

Pada Gambar 2.13 mengandung 3 layer, yaitu input, *hidden* dan output layer. Penambahan *layer* ini terjadi pada *hidden layer*. *Hidden layer* pada skema *deep learning* yang disebut dengan *Multi Layer Perceptron* (MLP) disebabkan jumlah neuron semakin banyak dan itu artinya semakin banyak juga perhitungan yang harus dikerjakan pada setiap *layer*. MLP merupakan pengembangan dari *Single Layer Perceptron* (SLP) yang merupakan model paling sederhana dari neural network dan sekaligus merupakan dasar bagi model-model tingkat lanjut yang digunakan pada *deep learning*. MLP kemudian menjadi cikal bakal metode *deep learning* atau *deep neural network* (DNN).

Deep learning sudah dikembangkan ke berbagai model atau arsitektur yang berbeda-beda. Berikut daftar beberapa model atau arsitektur untuk *deep learning*.

1. *Recurrent Neural Networks* (RNN)
2. *Long Short-Term Memory* (LSTM)
3. *Convolutional Neural Network* (CNN)
4. *Deep Believe Networks* (DBN)
5. *Deep Stacking Networks* (DSN)

Contoh penerapan masing-masing arsitektur deep learning dapat dipelajari pada Tabel 2.1

Tabel 2.1: Penerapan Arsitektur Deep Learning

No.	Arsitektur	Penerapan
1	RNN	<i>Speech recognition, handwriting recognition</i>
2	LSTM	<i>Natural language text compression, handwriting recognition, speech recognition, gesture recognition, image captioning</i>
3	CNN	<i>Image recognition, video analysis, natural language processing</i>
4	DBN	<i>Image recognition, information retrieval, natural language understanding, failure prediction</i>
5	DSN	<i>Information retrieval, continuous speech recognition</i>

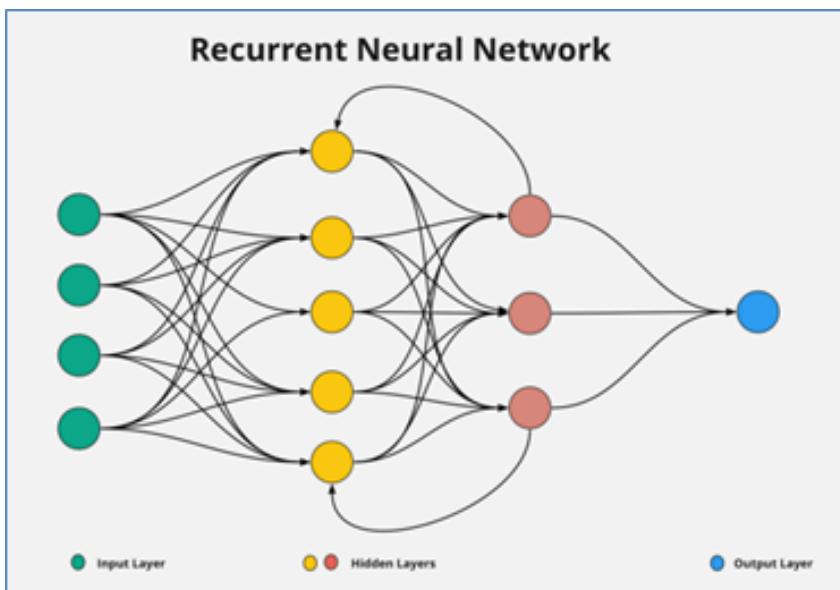
Masing-masing arsitektur pada Tabel 2.1 memiliki perbedaan, berikut penjelasan dan diagram network beberapa arsitektur deep learning yang umum.

2.6.5.1 Recurrent Neural Networks (RNN)

Recurrent Neural Network (RNN) merupakan arsitektur *deep learning* yang popular serta sangat menjanjikan untuk menyelesaikan berbagai persoalan yang terkait dengan *Natural Language Processing* (NLP). Model RNN digunakan agar mesin dapat memahami bahasa manusia. Mulai dari cara berkomunikasi, mendengarkan, mengenali percakapan, hingga memahami tata

bahasa dan aksen. RNN juga dapat diimplementasikan untuk mengenali gambar-gambar atau objek (R. Primartha, 2018).

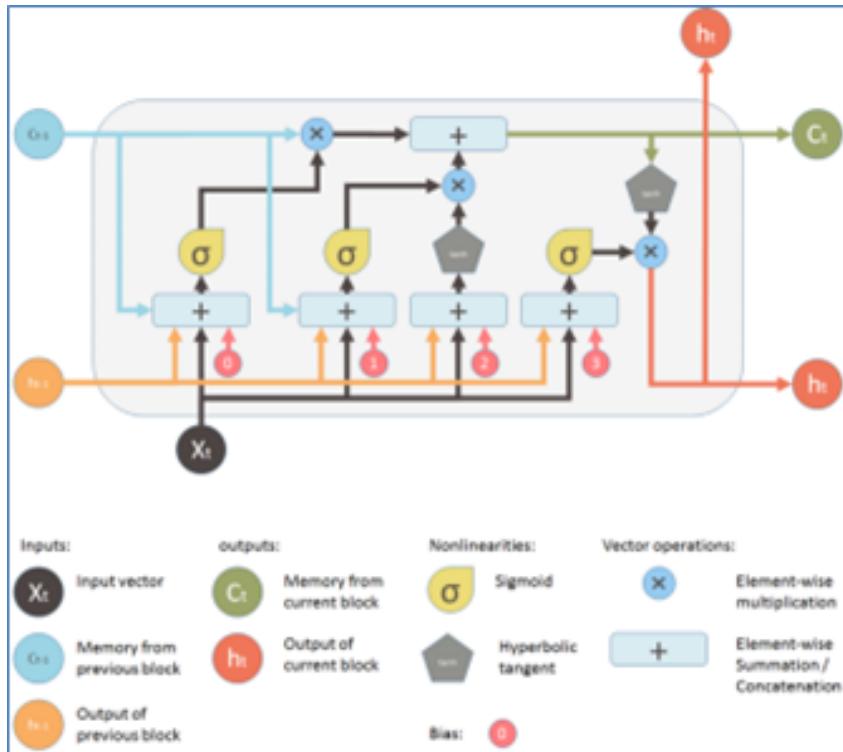
Diagram network RNN seperti pada Gambar 2.14 berikut.



Gambar 2.14: Diagram *Recurrent Neural Network* (RNN)
(Sumber: K. Dass, 2020)

2.6.5.2 Long Short Term Memory (LSTM)

Long Short Term Memory (LSTM) merupakan building unit untuk *layer-layer* pada recurrent neural network (RNN). LSTM mula-mula diusulkan oleh Sepp Hochreiter dan Jurgen Schmidhuber pada tahun 1997. LSTM juga banyak diimplementasikan pada bidang NLP. Boleh dibilang LSTM merupakan pengembangan dari RNN. Secara teoritis, jaringan saraf yang terhubung secara naif, yang disebut jaringan saraf berulang, dapat bekerja. Namun dalam praktiknya, mengalami dua masalah: gradien menghilang dan gradien meledak, yang membuatnya tidak dapat digunakan (R. Primartha, 2018). Kemudian, LSTM ditemukan untuk mengatasi masalah ini dengan secara eksplisit memasukkan unit memori, yang disebut sel ke dalam jaringan. Ini adalah diagram blok bangunan LSTM pada Gambar 2.15.



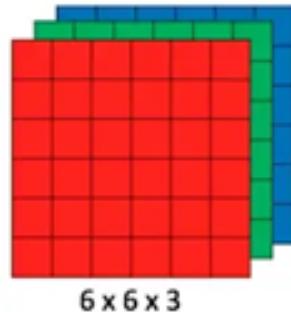
Gambar 2.15: Diagram *Long Short Term Memory* (LSTM)
(Sumber: S. Yan, 2016)

2.6.5.3 Convolutional Neural Network (CNN)

Convolutional Neural Networks (CNN atau ConvNet) merupakan salah satu model deep learning yang banyak digunakan untuk keperluan analisis citra/visual. CNN adalah salah satu kategori utama untuk melakukan pengenalan dan klasifikasi gambar, deteksi objek, pengenalan wajah, dan sebagainya merupakan beberapa area dimana CNN banyak digunakan (R. Primartha, 2018).

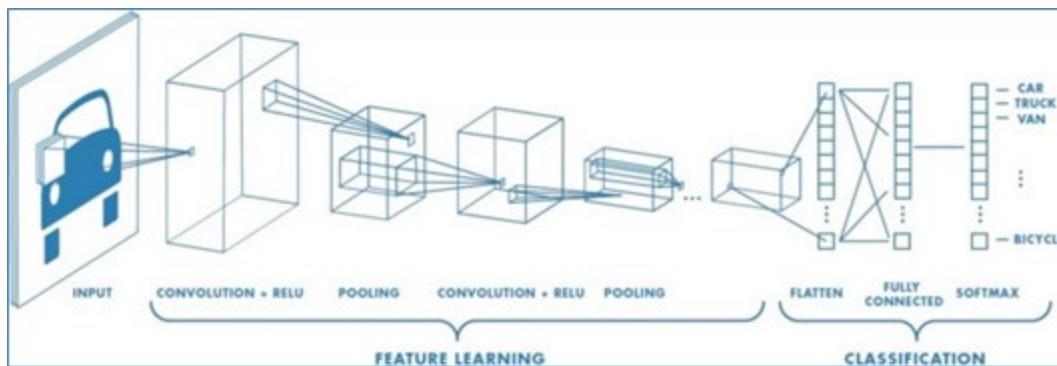
Klasifikasi gambar CNN mengambil input gambar, memproses, dan mengklasifikasikannya dalam kategori tertentu, misalnya kucing, harimau, singa. Komputer melihat gambar input sebagai susunan piksel dan itu tergantung pada resolusi gambar. Berdasarkan resolusi gambar, akan terlihat $h \times w \times d$ (h = Tinggi, w = Lebar, d = Dimensi). Misalnya, gambar array matriks RGB $6 \times 6 \times 3$ (3 mengacu pada nilai RGB) dan gambar array matriks $4 \times 4 \times 1$

dari gambar skala abu-abu, seperti pada Gambar 2.16.



Gambar 2.16: Array dari Matriks RGB
(Sumber: R. Prabhu, 2018)

Secara teknis, model *deep learning* CNN untuk latih dan uji, setiap gambar input akan melewati serangkaian lapisan konvolusi dengan filter (Kernel), Pooling, *fully connected layers* (FC) dan menerapkan fungsi softmax untuk mengklasifikasikan objek dengan nilai probabilistic antara 0 dan 1. Gambar 2.17. adalah alur dari CNN untuk memproses gambar input dan mengklasifikasikan objek berdasarkan nilai.



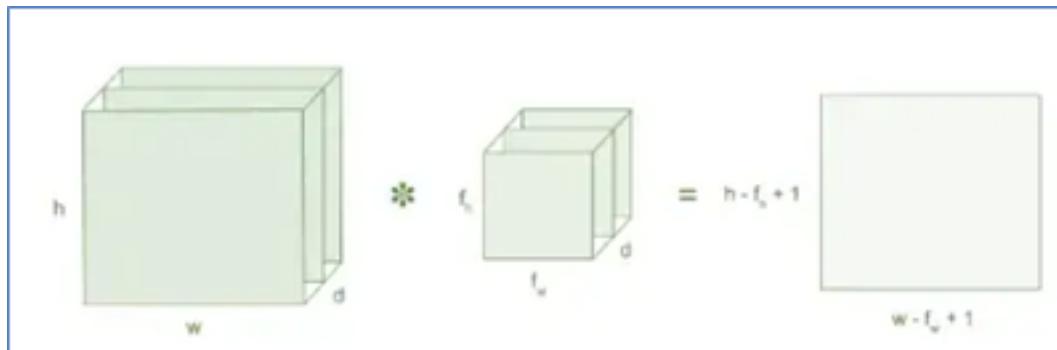
Gambar 2.17: Neural Network dengan banyak Convoluti Layer
(Sumber: R. Prabhu, 2018)

1. *Convolution Layer*

Konvolusi adalah lapisan pertama untuk mengekstraksi fitur dari gambar masukan. Konvolusi mempertahankan hubungan antara piksel dengan mempelajari fitur gambar menggunakan kotak kecil data masukan. Ini

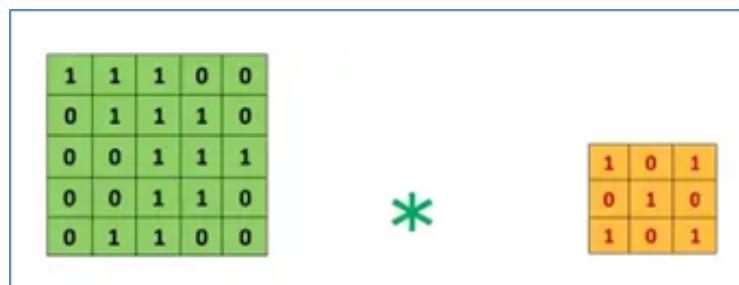
adalah operasi matematika yang mengambil dua input, seperti matriks gambar dan filter atau kernel.

- Sebuah gambar matriks (volume) dari dimensi $(h \times w \times d)$
- Sebuah filter ($f_h \times f_w \times d$)
- Output volume dimensi $(h - f_h + 1) \times (w - f_w + 1) \times 1$



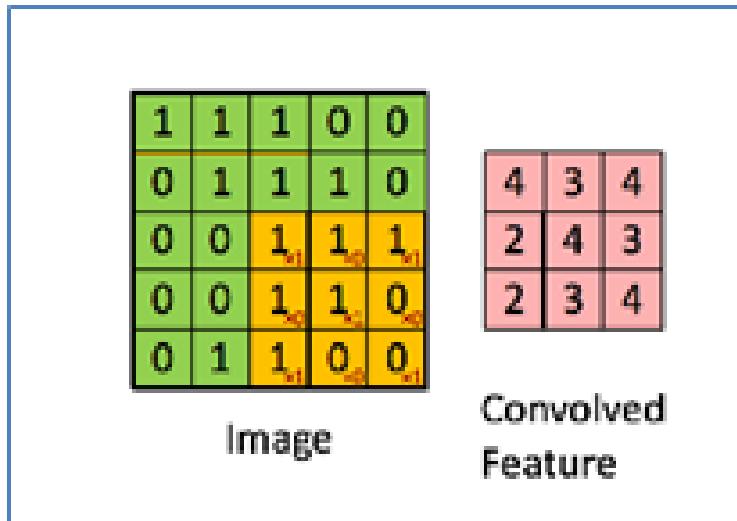
Gambar 2.18: Gambar Matriks Multiplies Kernel atau Filter Matriks
(Sumber: R. Prabhu, 2018)

Pertimbangkan gambar 5×5 yang nilai piksel gambarnya adalah 0, 1 dan matriks filter 3×3 , seperti pada Gambar 2.19.



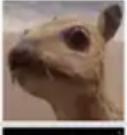
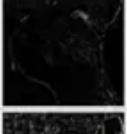
Gambar 2.19: Gambar matriks 5×5 dikalikan dengan Filter matiks 3×3

Kemudian, konvolusi matriks gambar 5×5 dikalikan dengan filter matriks 3×3 yang disebut "Feature Map" sebagai output yang ditunjukkan pada Gambar 2.20.



Gambar 2.20: Output Matriks 3 x 3

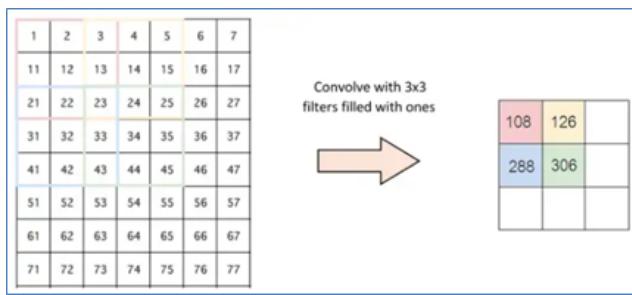
Konvolusi gambar dengan filter berbeda dapat melakukan operasi seperti deteksi tepi, mengaburkan, dan memeprtajam dengan menerapkan filter. Berikut ini contoh yang menunjukkan berbagai gambar konvolusi setelah menerapkan berbagai jenis filter (Kernel) pada Gambar 2.21.

Operation	Filter	Convolved Image
Identity	$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$	
Edge detection	$\begin{bmatrix} 1 & 0 & -1 \\ 0 & 0 & 0 \\ -1 & 0 & 1 \end{bmatrix}$	
	$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$	
	$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$	
Sharpen	$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$	
Box blur (normalized)	$\frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$	
Gaussian blur (approximation)	$\frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$	

Gambar 2.21: Beberapa Filter Umum
(Sumber: R. Prabhu, 2018)

2. Strides

Stride adalah jumlah piksel yang bergeser di atas matriks input. Saat langkahnya 1, maka memindahkan filter ke 1 piksel sekaligus. Saat langkahnya 2, maka memindahkan filter ke 2 piksel sekaligus dan seterusnya. Gambar 2.22 menunjukkan konvolusi akan bekerja dengan Langkah 2.



Gambar 2.22: Stride 2 Piksel
(Sumber: R. Prabhu, 2018)

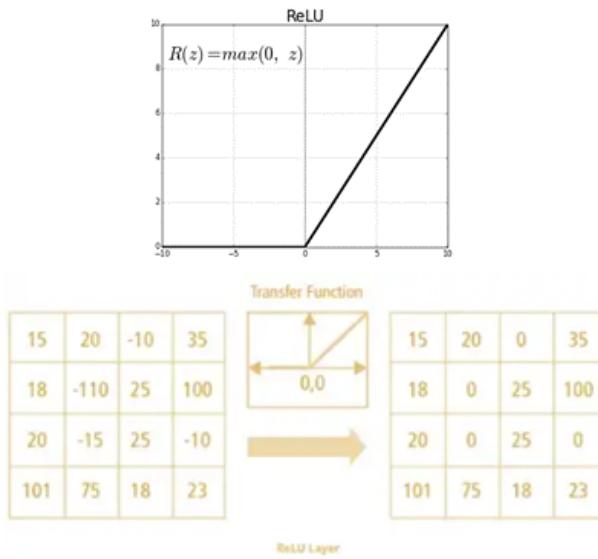
3. Padding

Pada saat penggunaan filter, terkadang filter tidak pas dengan gambar masukan. Maka, terdapat dua pilihan:

- Memadatkan gambar dengan angka nol (*zero padding*) agar pas;
- Menghilangkan bagian gambar yang tidak sesuai dengan filter. Ini disebut dengan *valid padding* yang hanya menyimpan bagian gambar yang valid.

4. Non Linearity (ReLU)

ReLU adalah singkatan dari *Rectified Linear Unit* untuk operasi non-linear. Outputnya adalah $f(x) = \max(0, x)$. ReLU penting karena tujuan ReLU adalah untuk mengenalkan non-linearitas di ConvNet, karena data dunia nyata ingin ConvNet pelajari adalah nilai linier non-negatif. Fungsi ini hanya mengembalikan nilai 0 jika nilai tersebut bernilai negatif, selain itu mengembalikan nilai yang sama dengan yang diberikan, tidak lain adalah menghilangkan keluaran negative dan mempertahankan nilai antara 0 hingga + tak terhingga, seperti pada Gambar 2.23.

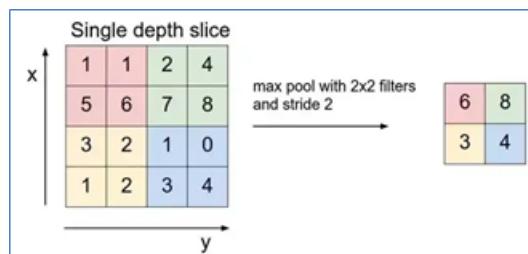


Gambar 2.23: Operasi ReLU
(Sumber: P. Ratan, 2021)

5. Pooling Layer

Bagian layer pooling akan mengurangi jumlah parameter ketika gambar terlalu besar. Penyatuan spasial juga disebut subsampling atau down-sampling yang mengurangi dimensi setiap peta tetapi tetap mempertahankan informasi penting. Penyatuan spasial dapat dari berbagai jenis, diantaranya Max Pooling, Average Pooling, dan Sum Pooling.

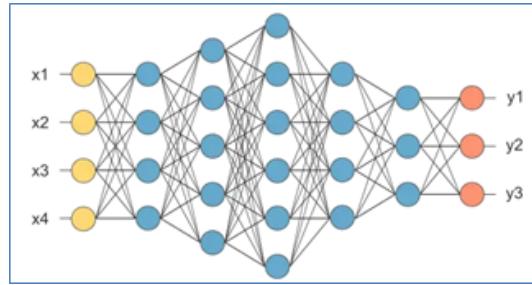
Max Pooling mengambil elemen tersbesar dari peta fitur yang diperbaiki. Mengambil elemen terbesar juga bisa mengambil pooling rata-rata. Jumlah semua elemen dalam peta fitur disebut sebagai kumpulan jumlah.



Gambar 2.24: Max Pooling

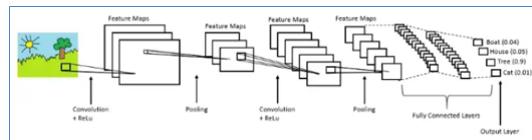
6. Fully Connected Layer

Lapisan yang disebut *Fully Connected Layer*, diratakan matriks menjadi vector dan memasukkannya ke dalam *fully connected layer*, seperti jaringan saraf (*neural network*), seperti Gambar 2.25.



Gambar 2.25: Setelah Pooling Layer Diratakan sebagai FC Layer

Pada Gambar 2.25, matriks peta fitur akan diubah menjadi vector (x_1 , x_2 , x_3 , ...). Dengan lapisan yang terhubung sepenuhnya, digabungkan fitur ini bersama untuk membuat model. Setelah itu, akhirnya memiliki fungsi aktivasi seperti softmax atau sigmoid untuk mengklasifikasikan keluaran sebagai objek, misalnya rumah, pohon, kucing, mobil, truk, dan sebagainya, seperti pada Gambar 2.26.

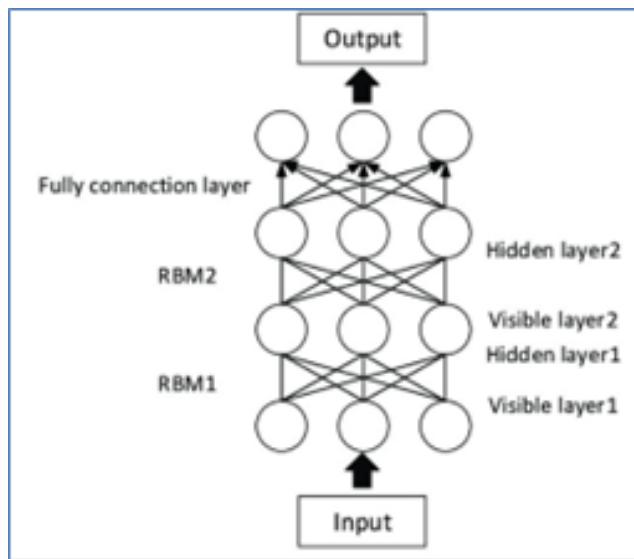


Gambar 2.26: Arsitektur CNN Lengkap

2.6.5.4 Deep Believe Networks (DBN)

Deep Belief Networks (DBN) merupakan model *deep learning* yang memanfaatkan tumpukan/*stack Restricted Boltzmann Machines* (RBM) atau kadangkala *Autoencoders*. *Autoencoders* adalah model *neural networks* yang memiliki input dan output yang sama. *Autoencoder* mempelajari data input dan

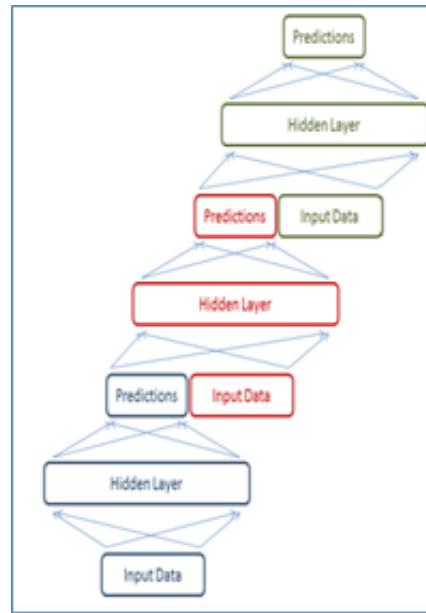
berusaha untuk melakukan rekonstruksi terhadap data input tersebut (R. Primartha, 2018). Skema diagram DBN seperti pada Gambar 2.27.



Gambar 2.27: Skema Diagram DBN
(Sumber: H. Liu & B. Lang, 2019)

2.6.5.5 Deep Stacking Networks (DSN)

Salah satu masalah pada *deep learning* adalah proses learning sangat sulit dilakukan dan memerlukan komputasi yang cukup kompleks. Pada tahun 2011 Deng Yu mengusulkan model *Deep Convex Networks* (DCN) atau *Deep Stacking Network* (DSN), yang sedikit berbeda dibandingkan model *deep learning* lain (R. Primartha, 2018). Secara umum model DSN terdiri atas ub-nets berukuran kecil dengan hanya sebuah hidden layer, seperti pada Gambar 2.28.



Gambar 2.28: Skema Diagram DSN
(Sumber: L. Deng et al, 2012)

Model *deep learning* yang popular lainnya adalah Region Based CNN, Google Net, Generative Adversarial Network (GAN), dan You Only Look Once (YOLO).

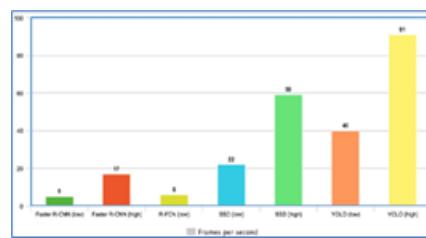
2.7 You Only Look Once (YOLO)

You Only Look Once (YOLO) adalah algoritma deteksi objek real-time yang diperkenalkan pada tahun 2015 oleh Joseph Redmon, Santosh Divvala, Ross Girshick dan Ali Farhadi dalam paper dengan judul "You Only Look Once: Unified, Real-Time Object Detection" (J. Redmon et al., 2015). Penulis membingkai masalah deteksi objek sebagai masalah regresi klasifikasi dengan memisahkan kotak pembatas (*bounding box*) secara spasial dan menghubungkan probabilitas ke masing-masing gambar yang terdeteksi menggunakan *convolutional neural network* (CNN). YOLO adalah *Convolutional Neural Network* (CNN) untuk melakukan deteksi objek secara real-time (V. Meel, 2022) (G. Boesch, 2022).

Beberapa alasan mengapa YOLO baik digunakan untuk deteksi objek real-time, diantaranya:

1. Kecepatan (*speed*)

YOLO sangat cepat karena tidak berurusan dengan jalur pipa (*pipelines*) yang rumit. YOLO dapat memproses gambar pada 45 frames per second (FPS). Selain itu, YOLO mencapai rata-rata presisi atau mean average precision (mAP) lebih dari dua kali dibandingkan dengan sistem real-time lainnya, yang menjadikannya kandidat yang bagus untuk pemrosesan real-time. Dari grafik pada Gambar 2.29 diamati bahwa YOLO jauh melampaui pendeksi objek lainnya dengan 91 FPS.



Gambar 2.29: Kecepatan YOLO dibandingkan dengan detector Objek Lainnya

(Sumber: S. A. S. Hernandez et al, 2020)

2. Akurasi deteksi tinggi (*high detection accuracy*)

YOLO jauh melampaui model *state-of-the-art* dalam akurasi dengan sedikit kesalahan latar belakang.

3. Generalisasi yang bagus (*good generalization*)

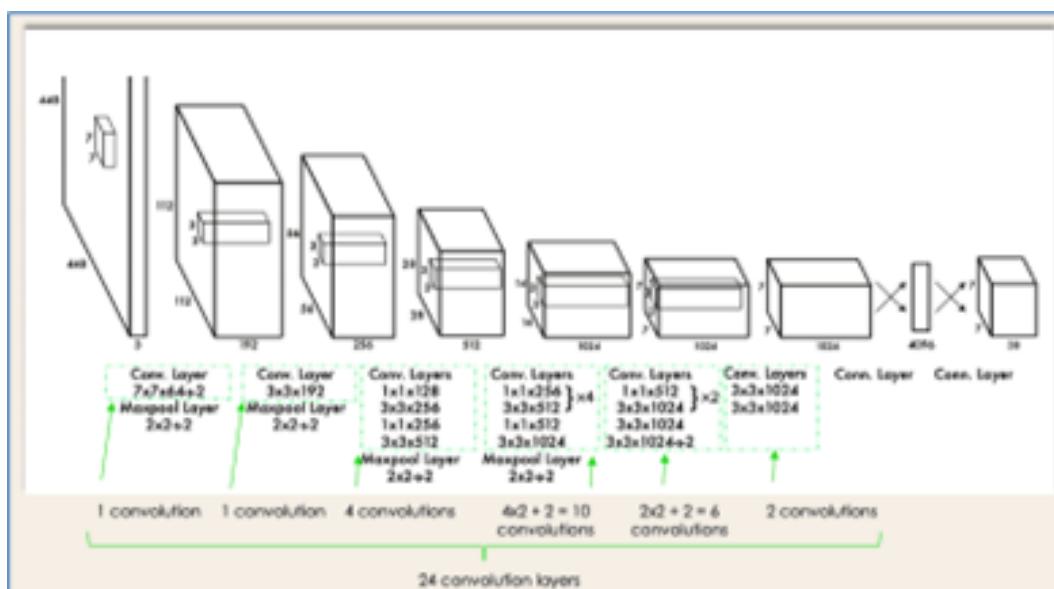
YOLO mendorong sedikit lebih jauh dengan memberikan generalisasi yang lebih baik untuk domain baru, yang menjadikannya bagus untuk aplikasi yang mengandalkan deteksi objek yang cepat dan kuat.

4. Sumber terbuka (*open-source*)

Membuat YOLO open-source membuat komunitas terus meningkatkan model. Inilah salah satu alasan mengapa YOLO telah melakukan begitu banyak perbaikan dalam waktu yang begitu terbatas.

2.7.1 Arsitektur YOLO

Arsitektur YOLO memiliki keseluruhan 24 lapisan konvolusional, empat lapisan penyatuan maksimum, dan dua lapisan yang terhubung sepenuhnya, arsitektur YOLO secara umum pada Gambar 2.30.



Gambar 2.30: Arsitektur YOLO dari *Original Paper*
(J. Redmon et al., 2015)

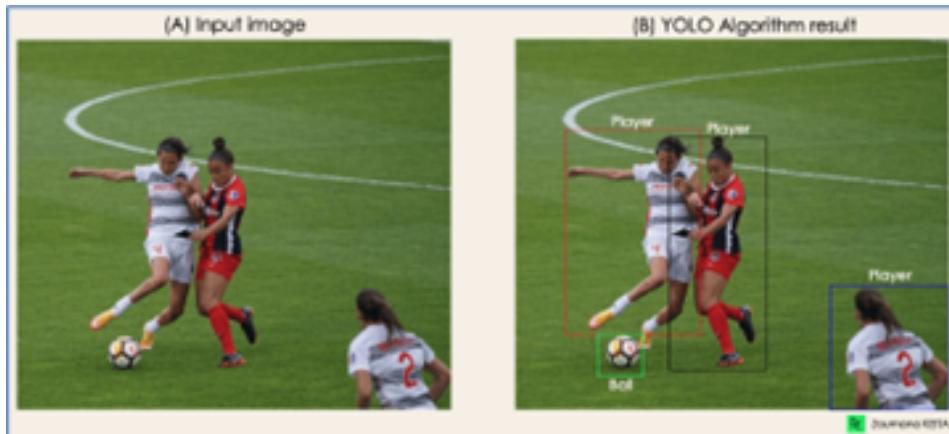
Arsitektur YOLO bekerja sebagai berikut:

- Mengubah ukuran gambar input menjadi 448x448 sebelum melalui *convolutional network*.
- Konvolusi 1x1 pertama kali diterapkan untuk mengurangi jumlah saluran, yang kemudian diikuti oleh konvolusi 3x3 untuk menghasilkan output kuboid.

- Fungsi aktivasi ReLU, kecuali lapisan terakhir, yang menggunakan fungsi aktivasi linier.
- Beberapa Teknik tambahan, seperti normalisasi batch dan dropout, masing-masing mengatur model dan mencegah overfitting.

2.7.2 Cara kerja Deteksi Objek YOLO

Berikut ini adalah proses bagaimana YOLO melakukan deteksi objek untuk mendapatkan gambar (b) dari gambar (a) pada Gambar 2.31.

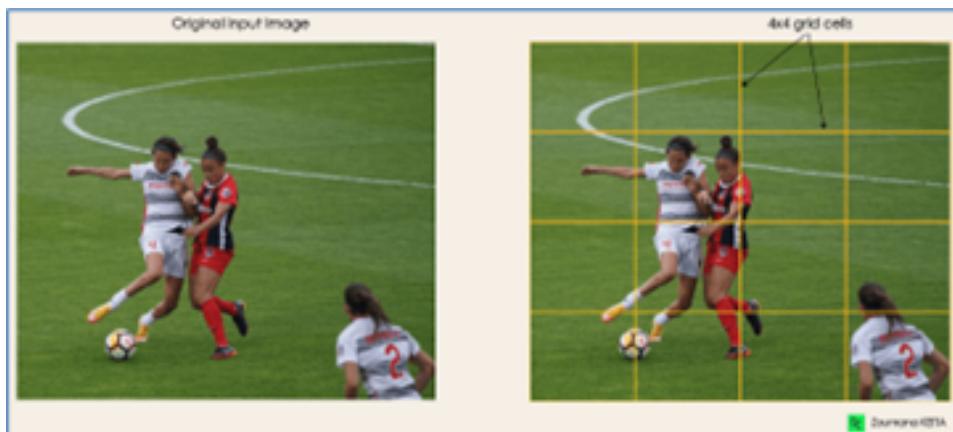


Gambar 2.31: (A) Input Image dan (B) Hasil Algoritma YOLO
(Sumber: Z. Kelta, 2022)

Algoritma YOLO bekerja berdasarkan empat pendekatan, sebagai berikut:

- (a) *Residual Blocks*(Blok Sisa)

Langkah pertama dimulai dengan membagi gambar asli (A) menjadi sel grid ($N \times N$) dengan bentuk yang sama, di mana N dalam hal ini adalah 4×4 grid sel pada Gambar 2.32. Setiap sel dalam grid bertanggung jawab untuk melokalkan dan memprediksi kelas objek yang dicakupnya, bersama dengan nilai probabilitas/kepercayaan.



Gambar 2.32: Residual Blocks
(Sumber: Z. Kelta, 2022)

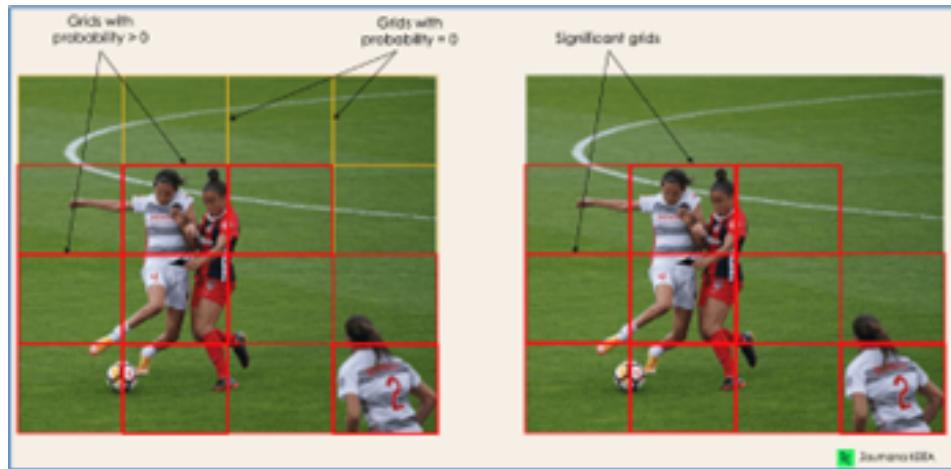
(b) *Bounding Box Regression* (Regresi Kotak Pembatas)

Langkah selanjutnya adalah menentukan kotak pembatas (*bounding box*) yang sesuai dengan persegi Panjang yang menyoroti semua objek dalam gambar. Dapat memiliki kotak pembatas sebanyak objek di dalam gambar yang diberikan. YOLO menentukan atribut kotak pembatas ini menggunakan modul regresi tunggal dalam format berikut, dimana Y adalah representasi vector terakhir untuk setiap kotak pembatas.

$$Y = [pc, bx, by, bh, bw, c1, c2]$$

Ini sangat penting selama fase pelatihan model.

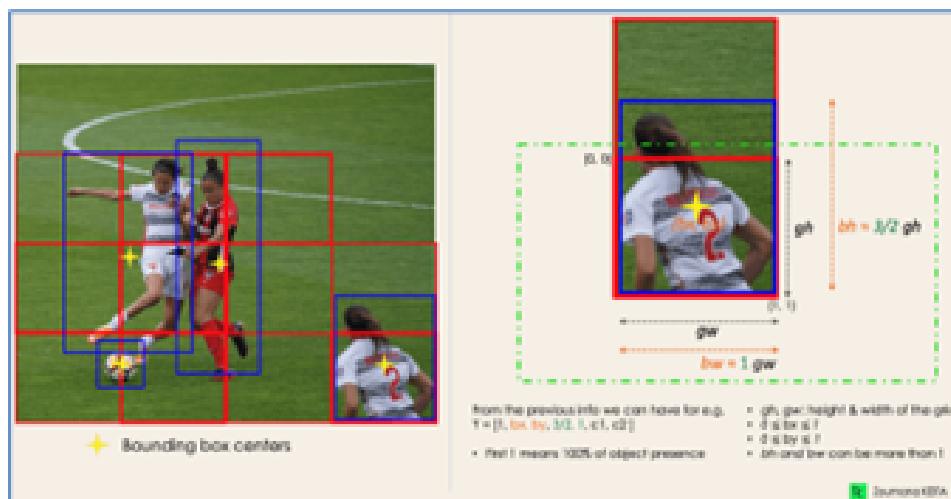
- pc sesuai dengan skor probabilitas dari grid yang berisi objek. Misalnya, semua grid yang berwarna merah akan memiliki skor probabilitas lebih tinggi dari nol. Gambar 2.33 adalah versi yang disederhanakan karena probabilitas setiap sel kuning adalah nol (tidak signifikan)



Gambar 2.33: Grid dengan Probabilitas
(Sumber: Z. Kelta, 2022)

- bx dan by adalah koordinat x dan y dari pusat kotak pembatas (*center of bounding box*) sehubungan dengan grid sel pembungkus.
- bh dan bw sesuai dengan tinggi dan lebar kotak pembatas sehubungan dengan sel grid pembungkus
- $c1$ dan $c2$ sesuai dengan dua kelas Player dan Ball, dapat memiliki kelas sebanyak yang dibutuhkan oleh pengguna.

Untuk dapat memahami dan terlihat, seperti pada Gambar 2.34 berikut.



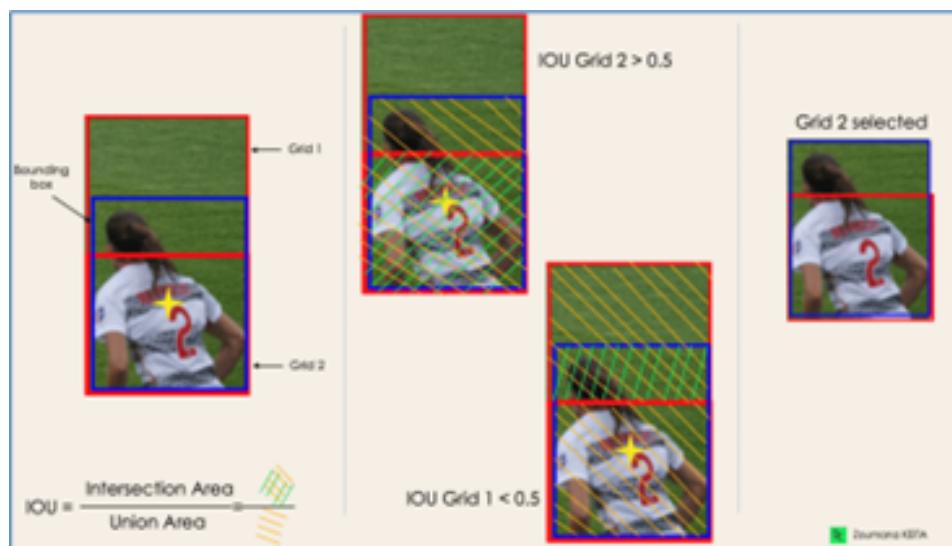
Gambar 2.34: Cara Bounding Box
(Sumber: Z. Kelta, 2022)

(c) *Intersection Over Unions* (IOU)

Sebagian besar waktu, satu objek dalam gambar dapat memiliki beberapa kandidat kotak petak untuk prediksi, meskipun tidak semuanya relevan. Tujuan dari IOU (nilai antara 0 dan 1) adalah untuk membuang kotak kisi tersebut agar hanya menyimpan yang relevan. Inilah logika dari IOU:

- Pengguna menentukan ambang pemilihan IOU-nya, misalnya, 0,5.
- Kemudian YOLO menghitung IOU dari setiap sel grid yang merupakan area persimpangan dibagi dengan Union Area.
- Terakhir, ia mengabaikan prediksi sel kisi yang memiliki $\text{IOU} \leq$ ambang batas dan mempertimbangkannya dengan $\text{IOU} > \text{ambang}$ batas.

Pada Gambar 2.35 adalah ilustrasi penerapan proses pemilihan grid pada objek kiri bawah. Dapat diamati bahwa objek awalnya memiliki dua kandidat kisi, kemudian hanya "Kisi 2" yang dipilih di bagian akhir.



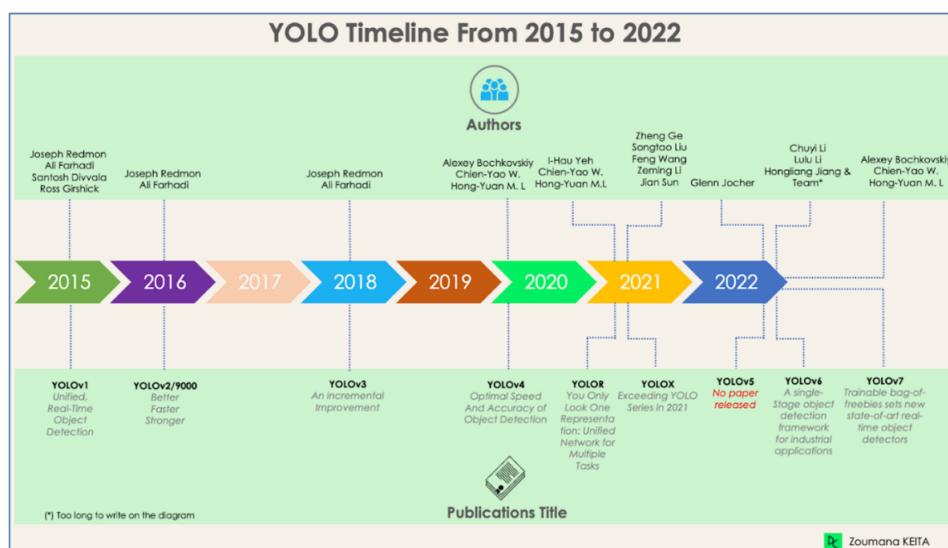
Gambar 2.35: IOU
(Sumber: Z. Kelta, 2022)

(d) Non-Maximum Supression (NMS)

Menetapkan ambang batas untuk IOU tidak selalu cukup karena sebuah objek dapat memiliki beberapa kotak dengan IOU di luar ambang batas, dan meninggalkan semua kotak tersebut mungkin termasuk kebisingan (*noise*). Di sinilah, dapat menggunakan NMS untuk menyimpan hanya kotak dengan skor probabilitas deteksi tertinggi.

2.7.3 Perkembangan YOLO

Sejak rilis pertama YOLO pada tahun 2015, YOLO telah banyak berkembang dengan versi berbeda, seperti pada Gambar 2.36.



Gambar 2.36: Perkembangan YOLO
(Sumber: Z. Kelta, 2022)

1. YOLO atau YOLOv1

Versi pertama YOLO ini adalah pengubah permainan untuk deteksi objek, karena kemampuannya mengenali objek dengan cepat dan efisien. Namun, seperti banyak solusi lainnya, versi pertama YOLO memiliki keterbatasannya sendiri:

- Kesulitan untuk mendeteksi gambar yang lebih kecil dalam sekelompok gambar, seperti sekelompok orang di stadion. Ini karena setiap kisi dalam arsitektur YOLO dirancang untuk deteksi objek tunggal.
- Kemudian, YOLO tidak berhasil mendeteksi bentuk baru atau tidak biasa.
- Terakhir, fungsi kerugian yang digunakan untuk memperkirakan kinerja pendekripsi memperlakukan kesalahan yang sama untuk kotak pembatas kecil dan besar, yang sebenarnya membuat pelokalan yang salah.

2. YOLOv2 atau YOLO9000

YOLOv2 dibuat pada tahun 2016 dengan ide membuat model YOLO lebih baik, lebih cepat, dan lebih kuat.

Peningkatan termasuk tetapi tidak terbatas pada penggunaan Darknet-19 sebagai arsitektur baru, normalisasi batch, resolusi input yang lebih tinggi, lapisan konvolusi dengan anchors, pengelompokan dimensi, dan (5) fitur-fitur halus.

- *Batch Normalization*

Menambahkan lapisan normalisasi batch meningkatkan kinerja sebesar 2% mAP. Normalisasi batch ini menyertakan efek regularisasi, mencegah overfitting.

- *Higher input resolution*

YOLOv2 secara langsung menggunakan input 448x448 beresolusi lebih tinggi daripada 224x224, yang membuat model menyesuaikan filternya untuk bekerja lebih baik pada gambar beresolusi lebih tinggi. Pendekatan ini meningkatkan akurasi sebesar 4% mAP, setelah dilatih selama 10 epochs pada data ImageNet.

3. YOLOv3 - Peningkatan Bertahap

Perubahan tersebut terutama mencakup arsitektur jaringan baru: Darknet-53. Ini adalah jaringan saraf 106, dengan jaringan upsampling dan blok residual. Jauh lebih besar, lebih cepat, dan lebih akurat dibandingkan dengan Darknet-19, yang merupakan tulang punggung YOLOv2. Arsitektur baru ini telah bermanfaat di banyak tingkatan:

- Prediksi *Bounding Box* Lebih Baik

Model regresi logistic digunakan oleh YOLOv3 untuk memprediksi skor objektivitas untuk setiap kotak pembatas (bounding box).

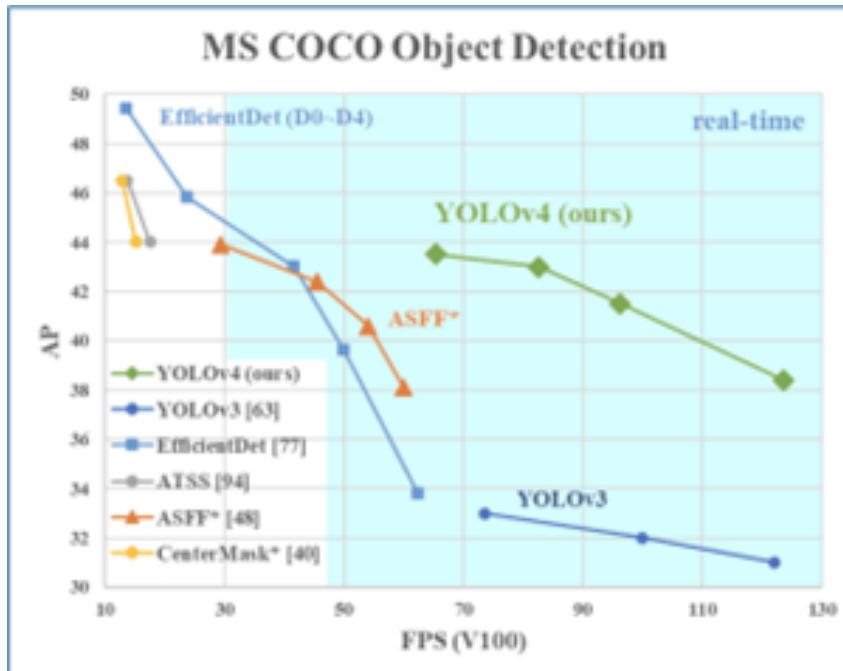
- Prediksi Kelas yang Lebih Akurat

Menggantikan penggunaan softmax seperti yang dilakukan di YOLOv2, pengklasifikasi logistik independen telah diperkenalkan untuk memprediksi kelas kotak pembatas secara akurat. Ini bahkan berguna saat menghadapi domain yang lebih kompleks dengan label yang tumpang tindih (Misalnya, → Pemain Sepak Bola). Menggunakan softmax akan membatasi setiap kotak hanya memiliki satu kelas, yang tidak selalu benar.

4. YOLOv4 - *Optimal Speed and Accuracy of Object Detection*

Versi YOLO ini memiliki Kecepatan dan Akurasi Deteksi Objek Optimal dibandingkan dengan semua versi sebelumnya dan detektor objek canggih lainnya.

Gambar 2.37 menunjukkan YOLOv4 mengungguli YOLOv3 dan FPS dalam kecepatan masing-masing sebesar 10% dan 12%.



Gambar 2.37: Komparasi YOLOv4 dengan YOLOv3 dan *state-of-the-art*

Deteksi Objek Lain

(Sumber: Z. Kelta, 2022)

YOLOv4 dirancang khusus untuk sistem produksi dan dioptimalkan untuk komputasi paralel.

Backbone arsitektur YOLOv4 adalah CSPDarknet53, jaringan yang berisi 29 lapisan konvolusi dengan filter 3 x 3 dan sekitar 27,6 juta parameter. Arsitektur ini, dibandingkan dengan YOLOv3, menambahkan informasi berikut untuk deteksi objek yang lebih baik:

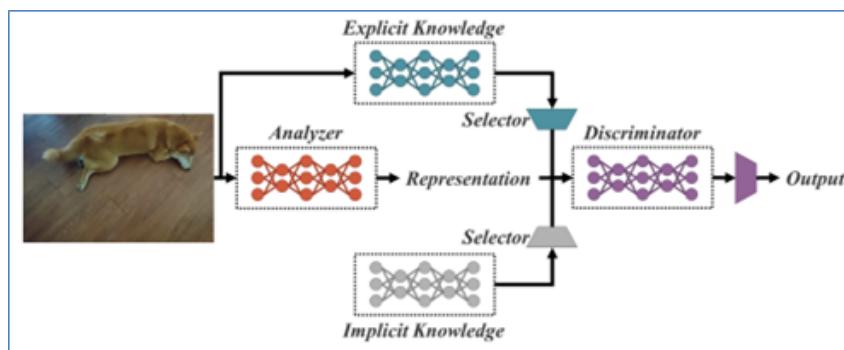
- *Spatial Pyramid Pooling* (SPP) secara signifikan meningkatkan bidang reseptif, memisahkan fitur konteks yang paling relevan, dan tidak memengaruhi kecepatan jaringan.
- Menggantikan *Feature Pyramid Network* (FPN) yang digunakan di YOLOv3, YOLOv4 menggunakan PANet untuk agregasi parameter dari tingkat deteksi yang berbeda.
- Augmentasi data menggunakan teknik mosaik yang menggabungkan

empat gambar pelatihan selain pendekatan pelatihan permusuhan diri.

- Menggunakan pemilihan hyper-parameter yang optimal menggunakan algoritma genetika.

5. YOLOR - You Only Look One Representation

Sebagai *Unified Network for Multiple Tasks*, YOLOR didasarkan pada jaringan terpadu yang merupakan kombinasi dari pendekatan pengetahuan eksplisit dan implisit.



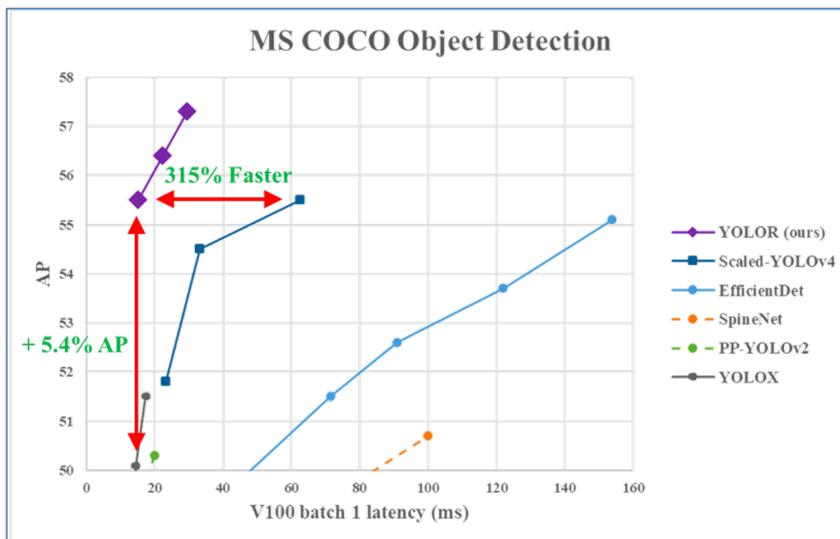
Gambar 2.38: Unified Network Architecture
 (Sumber: C. Y. Wang et al, 2021)

Pengetahuan eksplisit adalah pembelajaran normal atau conscious learning. Pembelajaran implisit di sisi lain dilakukan secara tidak sadar (dari pengalaman).

Menggabungkan kedua teknik ini, YOLOR mampu menciptakan arsitektur yang lebih kuat berdasarkan tiga proses: (1) penyelarasan fitur, (2) penyelarasan prediksi untuk deteksi objek, dan (3) representasi kanonis untuk pembelajaran multi-tugas.

Salah satu yang mengalami peningkatan adalah penajaran prediksi. Pendekatan ini memperkenalkan representasi implisit ke dalam peta fitur dari setiap jaringan piramida fitur (FPN), yang meningkatkan presisi sekitar 0,5%.

Dari grafik berikut, dapat diamati bahwa YOLOR mencapai kecepatan inferensi data MS COCO yang canggih dibandingkan dengan model lain seperti pada Gambar 2.39.



Gambar 2.39: Performance YOLOR vs YOLOv4 dan Model Lainnya

(Sumber: C. Y. Wang et al, 2021)

6. YOLOX - Exciding *YOLO Series in 2021*

Ini menggunakan baseline yang merupakan versi modifikasi dari YOLOv3, dengan Darknet-53 sebagai *backbone*.

Diterbitkan dalam makalah *Exceeding YOLO Series in 2021*, YOLOX menghadirkan empat karakteristik utama berikut untuk membuat model yang lebih baik dibandingkan dengan versi yang lebih lama.

- Kepala terpisah yang efisien: Coupled head yang digunakan pada versi YOLO sebelumnya terbukti mengurangi performa model. YOLOX menggunakan decoupled sebagai gantinya, yang memungkinkan pemisahan tugas klasifikasi dan lokalisasi, sehingga meningkatkan kinerja model.
- Augmentasi data yang kuat: Integrasi Mosaic dan MixUp ke dalam pendekatan augmentasi data sangat meningkatkan kinerja YOLOX.

- *Anchor free system*: Algoritma *anchor-based* melakukan pengelompokan di bawah tenda, yang meningkatkan waktu inferensi. Menghapus mekanisme jangkar di YOLOX mengurangi jumlah prediksi per gambar, dan meningkatkan waktu inferensi secara signifikan.
- SimOTA untuk penetapan label: Pergantian penggunaan pendekatan interseksi penyatuan (IoU), penulis memperkenalkan SimOTA, strategi penetapan label yang lebih kuat yang mencapai hasil canggih dengan tidak hanya mengurangi waktu pelatihan tetapi juga menghindari masalah hiperparameter tambahan. Selain itu, ini meningkatkan peta deteksi sebesar 3%.

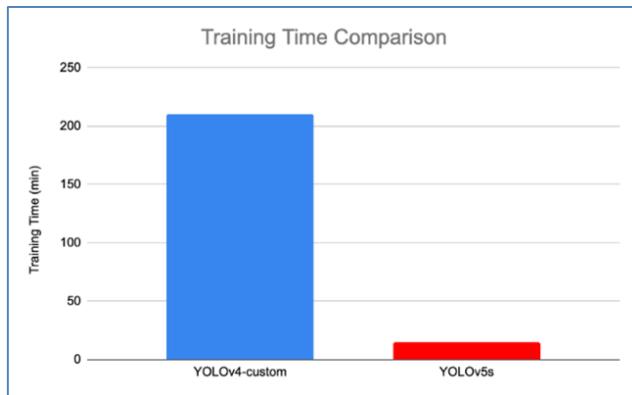
7. YOLOv5

YOLOv5, dibandingkan dengan versi lain, tidak memiliki makalah penelitian yang diterbitkan, dan ini adalah versi YOLO pertama yang diimplementasikan di Pytorch, bukan di Darknet.

Dirilis oleh Glenn Jocher pada Juni 2020, YOLOv5, mirip dengan YOLOv4, menggunakan CSPDarknet53 sebagai tulang punggung arsitekturnya. Rilis ini mencakup lima ukuran model yang berbeda: YOLOv5s (terkecil), YOLOv5m, YOLOv5l, dan YOLOv5x (terbesar).

Salah satu peningkatan besar dalam arsitektur YOLOv5 adalah integrasi lapisan Fokus, yang diwakili oleh satu lapisan, yang dibuat dengan mengganti tiga lapisan pertama YOLOv3. Integrasi ini mengurangi jumlah lapisan, dan jumlah parameter dan juga meningkatkan kecepatan maju dan mundur tanpa dampak besar pada peta.

Ilustrasi Gambar 2.40 membandingkan waktu pelatihan antara YOLOv4 dan YOLOv5.



Gambar 2.40: Perbandingan Waktu Pelatihan antara YOLOv4 dan YOLOv5
(Sumber: J. Nelson, 2020)

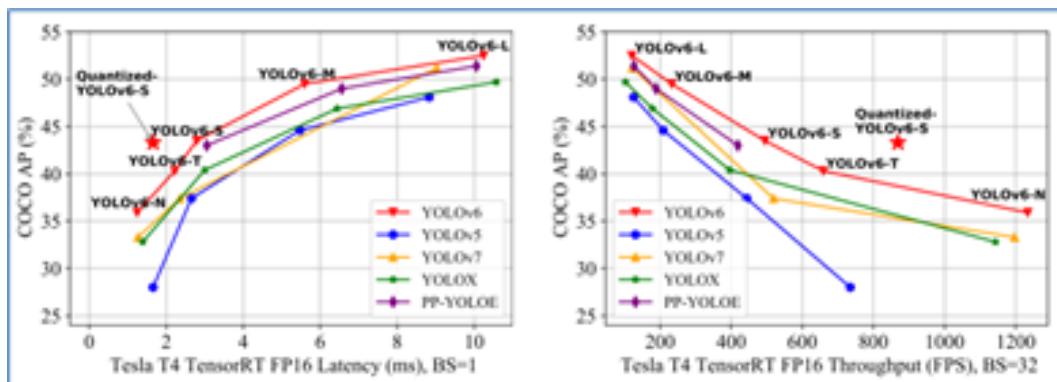
8. YOLOv6

Didedikasikan untuk aplikasi industri dengan desain efisien yang ramah perangkat keras dan kinerja tinggi, kerangka kerja YOLOv6 (MT-YOLOv6) dirilis oleh Meituan, sebuah perusahaan e-commerce Tiongkok.

Ditulis dalam Pytorch, versi baru ini bukan bagian dari YOLO resmi tetapi tetap diberi nama YOLOv6 karena tulang punggungnya terinspirasi oleh arsitektur YOLO satu tahap yang asli.

YOLOv6 memperkenalkan tiga peningkatan signifikan pada YOLOv5 sebelumnya: desain tulang punggung dan leher yang ramah perangkat keras, kepala terpisah yang efisien, dan strategi pelatihan yang lebih efektif.

YOLOv6 memberikan hasil yang luar biasa dibandingkan dengan versi YOLO sebelumnya dalam hal akurasi dan kecepatan pada dataset CO-CO seperti yang diilustrasikan pada Gambar 2.41



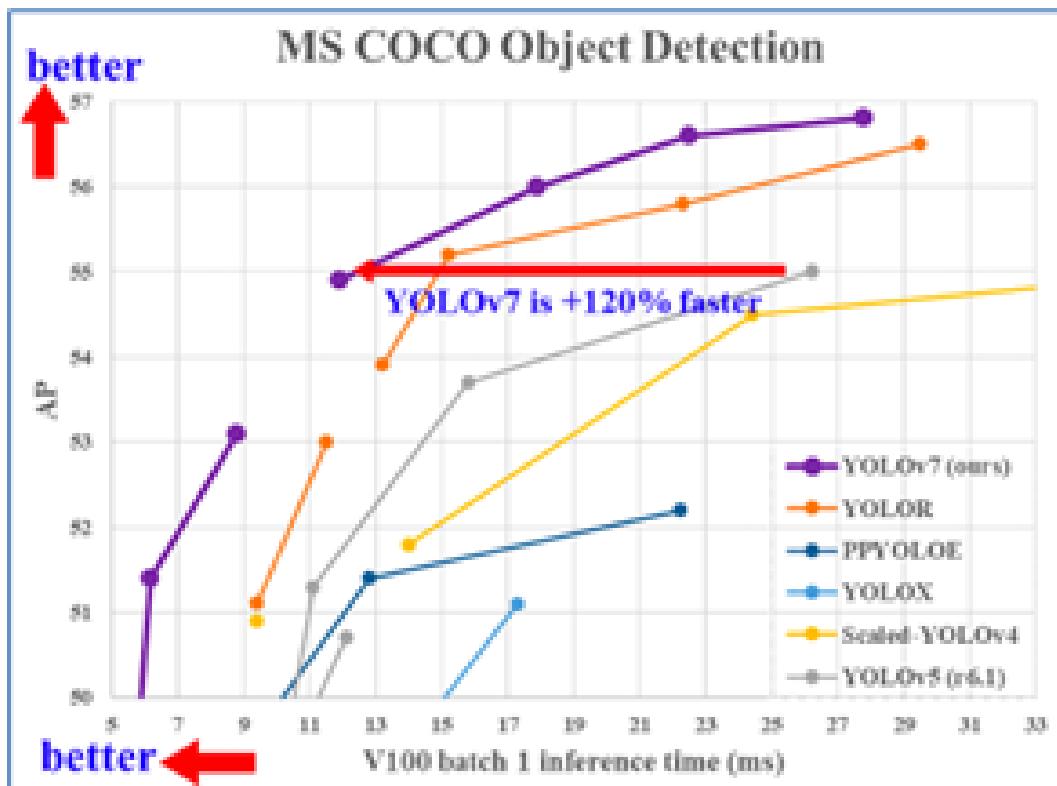
Gambar 2.41: Comparison of state-of-the-art efficient object detectors. All models are tested with TensorRT 7 except that the quantized model is with TensorRT 8

(Sumber: C. Li et al, 2022)

- YOLOv6-N mencapai 35,9% AP pada dataset COCO dengan throughput 1234 (throughput) FPS pada GPU NVIDIA Tesla T4.
- YOLOv6-S mencapai AP 43,3% yang canggih pada 869 FPS.
- YOLOv6-M dan YOLOv6-L juga mencapai kinerja akurasi yang lebih baik masing-masing sebesar 49,5% dan 52,3% dengan kecepatan inferensi yang sama.

9. YOLOv7

YOLOv7 dirilis pada Juli 2022 di Paper *Trained bag-of-freebies set state-of-the-art real-time object detector*. Versi ini membuat langkah signifikan di bidang deteksi objek, dan melampaui semua model sebelumnya dalam hal akurasi dan kecepatan.



Gambar 2.42: Perbandingan YOLOv7 inference time dengan real-time object detector lainnya

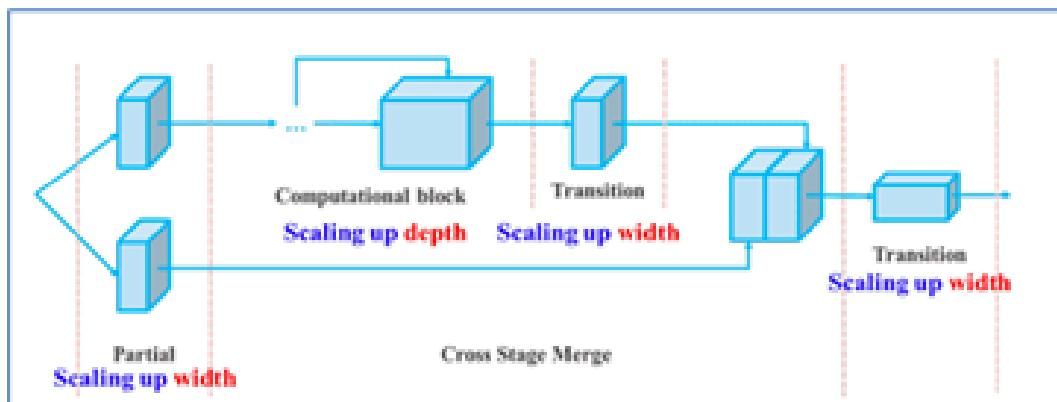
(Sumber: C. Y. Wang et al, 2022)

YOLOv7 telah membuat perubahan besar dalam (1) arsitekturnya dan (2) pada tingkat bag-of-freebies yang dapat dilatih:

- Level Arsitektur

YOLOv7 mereformasi arsitekturnya dengan mengintegrasikan Extended Efficient Layer Aggregation Network (E-ELAN) yang memungkinkan model mempelajari fitur yang lebih beragam untuk pembelajaran yang lebih baik.

Selain itu, YOLOv7 menskalakan arsitekturnya dengan menggabungkan arsitektur model asalnya seperti YOLOv4, Scaled YOLOv4, dan YOLO-R. Hal ini memungkinkan model untuk memenuhi kebutuhan kecepatan inferensi yang berbeda.



Gambar 2.43: *Compound scaling up depth and width for concatenation-based model*

(Sumber: C. Y. Wang et al, 2022)

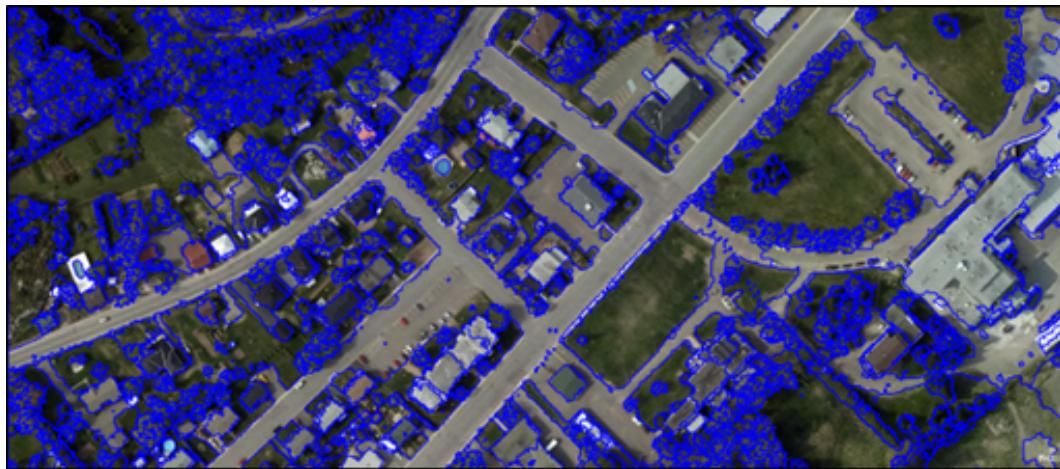
- Trainable bag-of-freebies

Istilah bag-of-freebies mengacu pada peningkatan akurasi model tanpa meningkatkan biaya pelatihan, dan inilah alasan mengapa YOLOv7 tidak hanya meningkatkan kecepatan inferensi tetapi juga akurasi deteksi.

2.8 Object Based Image Analysis (OBIA)

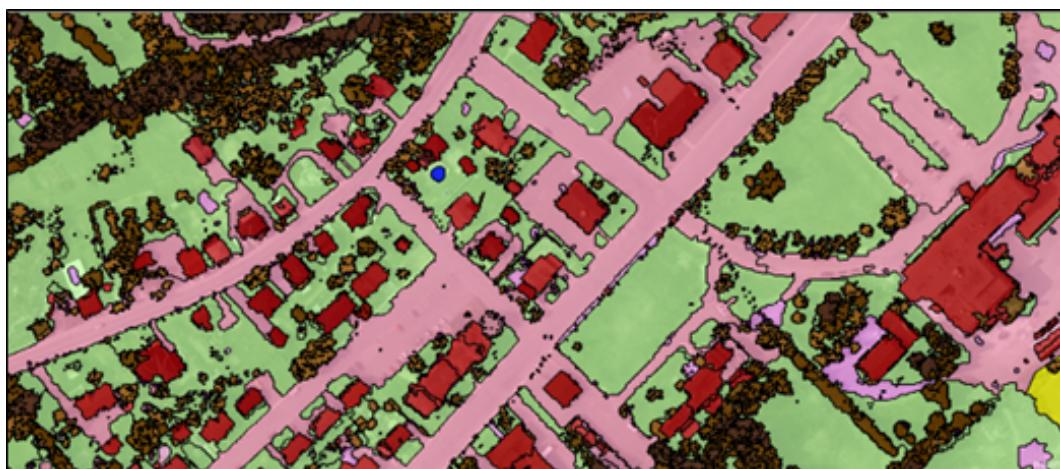
OBIA merupakan teknik klasifikasi yang tidak hanya memandang rona dan tekstur piksel namun berdasarkan dari kesatuan objek, atau dapat dikatakan OBIA adalah pendekatan yang proses klasifikasinya tidak hanya mempertimbangkan aspek spectral namun aspek spasial objek. Data citra penginderaan jauh yang digunakan untuk klasifikasi ini biasanya menggunakan data citra penginderaan jauh resolusi tinggi seperti Quickbird, Ikonos, World View, dll. Klasifikasi ini hamper mirip dengan klasifikasi unsupervised, akan tetapi basis dari klasifikasi OBIA yaitu dengan segmentasi (A. Bakar, 2014).

OBIA mengelompokkan gambar yang mengelompokkan piksel kecil menjadi objek vektor. Alih-alih berbasis per-pixel, segmentasi secara otomatis mendigitalkan gambar untuk pengguna (GIS Geography, 2020).



Gambar 2.44: OBIA Segmentasi Proses Pengelompokkan Pixel yang hampir sama ke dalam Objek
(Sumber: GISGeography, 2020)

Apa yang dilakukan segmentasi adalah meniru apa yang dilakukan mata pengguna. Tetapi dengan objek tersegmentasi ini, pengguna menggunakan properti spektral, geometris, dan spasialnya untuk mengklasifikasikan ke dalam tutupan lahan.



Gambar 2.45: OBIA Klasifikasi menggunakan shape, size, dan spectral properties objek untuk klasifikasi setiap objek

Dua prinsip dasar OBIA adalah klasifikasi dan segmentasi. Segmentasi adalah memecah gambar menjadi objek yang mewakili fitur berbasis darat, sedangkan Klasifikasi adalah mengklasifikasi objek-objek tersebut menggu-

nakan bentuk, ukuran, sifat spasial dan spektralnya. Analis sering menggunakan statistik ini untuk mengklasifikasikan tutupan lahan menggunakan OBIA, yaitu *trees* (pohon) memiliki ketinggian yang bervariasi (standar deviasi *normalized Digital Surface Model* (nDSM)) yang merupakan model elevasi yang menangkap fitur alami dan fitur buatan, seperti gedung, pohon, kabel listrik, dan objek lainnya. Di samping itu, memiliki reflektansi inframerah-dekat yang tinggi (*normalized difference vegetation index* (NDVI) tinggi), sedangkan *grass* memiliki pendek (nDSM rendah), datar (deviasi standar nDSM rendah) dan memiliki reflektansi inframerah-dekat sedang (NDVI sedang). NDVI digunakan untuk mengukur indeks yang menggambarkan tingkat kepadatan, kehijauan suatu tanaman dan kondisi dari suatu vegetasi.

2.9 Sentinel-2

Data yang digunakan dalam penelitian ini diambil dari gambar citra satelit Sentinel-2. Sentinel-2 diluncurkan sebagai bagian dari program Copernicus Komisi Eropa pada tanggal 23 Juni 2015 yang dirancang khusus untuk memberikan banyak data dan citra. Satelit dilengkapi dengan sensor multispectral opto-elektronik untuk survei dengan resolusi Sentinel-2 yaitu 10 hingga 60m di zona spektral tampak, VNIR, SWIR, termasuk 13 saluran spectral yang memastikan penangkapan perbedaan dalam keadaan vegetasi, termasuk perubahan temporal, dna juga meminimalkan dampkan kualitas fotografi atmosfer. Sentinel-2 memiliki dua satelit dalam misi memungkinkan survei berulang setiap 5 hari di ekuator dan 2-3 hari di garis lintang tengah (EOS, 2020).

Misi Sentinel-2 terdiri dari dua satelit yang dikembangkan untuk mendukung vegetasi, tutupan lahan, dan pemantauan lingkungan. Satelit Sentinel-2A diluncurkan oleh European Space Agency (ESA) pada 23 Juni 2015, dan beroperasi di orbit sinkron matahari dengan siklus berulang 10 hari. Setelit

kedua yaitu Sentinel-2B diluncurkan pada 7 Maret 2017 dan beroperasi dengan akuisisi data yang tersedia di EarthExplorer (USGS EROS, 2020).

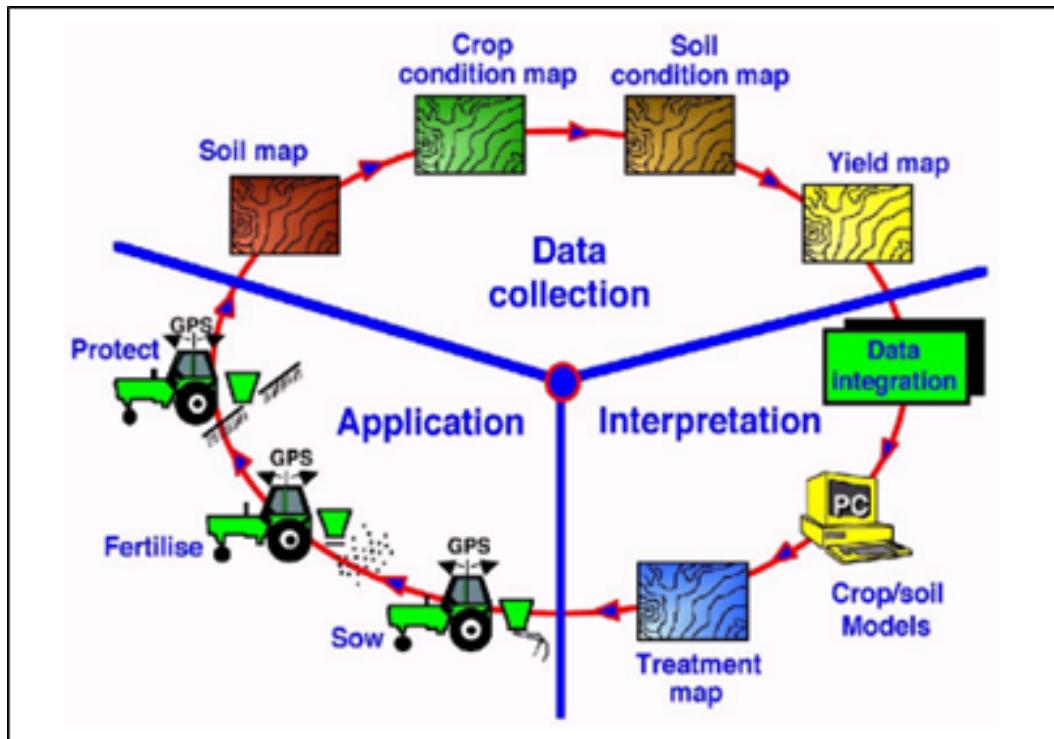
Terdapat Perbandingan antara Sentinel-2 dengan UAV Drone, seperti pada Tabel 2.2.

2.10 Pertanian Presisi

Pertanian presisi atau precision agriculture merupakan suatu sistem pertanian yang mengintegrasikan penggunaan teknologi dalam mengumpulkan informasi, sehingga dapat melakukan proses pertanian secara presisi atau dengan input, tempat, dan waktu yang tepat. Istilah presisi berarti akurat, terakar, dan terukur. Pertanian presisi disebut juga menggunakan input pertanian yang tepat dengan teknik, jumlah, tempat, dan waktu yang tepat untuk menghasilkan produksi panen secara maksimal. Input pertanian meliputi seperti pemupukan, herbisida, insektisida, benih, dan lainnya. Meskipun untuk melakukan pertanian secara akurat ini membutuhkan banyak informasi, cenderung kompleks untuk kebanyakan petani, dan membutuhkan kerjasama dari berbagai multidisiplin ilmu, namun sistem pertanian presisi mampu meningkatkan laba, mengurangi limbah, mengurangi biaya produksi, dan menjaga kualitas lingkungan (J. Taylor et al., 2016).

Pertanian presisi (PA) merupakan ilmu untuk meningkatkan hasil panen dan membantu keputusan manajemen menggunakan sensor dan alat analisis teknologi tinggi. Pertanian presisi adalah konsep yang digunakan untuk meningkatkan produksi, mengurangi waktu kerja, dan memastikan pengelolaan seperti pupuk, irigasi, dan lainnya berjalan efektif. Perkembangan teknologi, khususnya ketersediaan citra satelit beresolusi tinggi, perkembangan teknologi kendaraan udara tak berawak (UAV), menunjukkan bahwa adopsi sumber data penginderaan jauh dalam pertanian presisi mengalami peningkatan (P. Singh et al., 2020).

Siklus pertanian presisi terdiri dari tahapan seperti Gambar 2.46.



Gambar 2.46: Siklus Pertanian Presisi
(Sumber: A. Comparetti, 2011)

- Pengukuran parameter tanah dan tanaman dengan bantuan pemetaan spasial dan pemantauan kondisi cuaca setempat (pengumpulan data);
- pemetaan parameter tanah dan tanaman dalam lapangan (pengumpulan data);
- pemetaan dan integrasi data dengan bantuan aplikasi (interpretasi);
- pemantauan kesuburan dan penaburan (aplikasi);

Dalam menerapkan pertanian presisi diperlukan instrumen berikut (A. Comparetti, 2011):

- Sistem penentuan posisi satelit, untuk penginderaan posisi di mana setiap parameter lapangan yang diukur harus direferensikan secara geografis dan, kemudian, juga posisi di mana input (seperti, tanaman, lahan) yang

terdeteksi oleh mesin dapat diterapkan kebutuhannya di setiap area lahan (zona pengelolaan).

- Sensor, untuk mengukur parameter tanah dan tanaman di lapangan.
- Perangkat (devices), untuk menyiapkan dan mengontrol aplikasi input tanaman tingkat variabel spasial.
- Perangkat lunak, untuk membuat peta parameter tanah dan tanaman dalam lahan dan aplikasi input tanaman dengan data dari citra pengim-deraan, dan juga untuk menginterpretasikan data terukur;

2.11 Kelapa Sawit

Kelapa sawit adalah tanaman sejenis palma berakar serabut atau monokotil.

Bagian tanaman yang bernilai ekonomis adalah buah. Masing-masing dari seribu atau lebih buah dalam tandan buah segar itu terdiri dari inti sawit (kernel) yang dikelilingi oleh daging buah (mesocarp). Di pabrik kelapa sawit, sangat sedikit buah yang terbuang sia-sia karena pabrik tersebut mengubah setiap buah kelapa sawit menjadi minyak kelapa sawit dan minyak inti sawit), bahkan limbah dari setiap buah sawitpun di daur ulang di perkebunan sebagai pupuk atau diolah sebagai bahan bakar biomass. Sangat sedikit sekali buah kelapa sawit terbuang karena baik inti kelapa sawit dan dagingnya sama-sama digunakan untuk menghasilkan minyak.



Gambar 2.47: Buah Kelapa Sawit(Inti dan Daging Sawit)

Satu tandan tanaman dewasa beratnya mencapai 20 - 35 kg, bahkan ada yang mencapai di atas 40 kg, tergantung pada perawatan. Tandan tersusun dari 200 - 600 buah masing-masing sekitar 20-35 gram. Buah diambil minyaknya dengan hasil sabut (daging buah/mesocarp) menghasilkan minyak kasar (CPO) 20-26%, inti sawit sebanyak 6% yang menghasilkan minyak inti (PKO) 3-4%. Kadar % dihitung dari berat tandan buah segar (Smart Agribusiness and Food, 2017).

(a) Usia Tanam

Umur atau usia ekonomis tanaman kelapa sawit yang dibudidayakan bisa mencapai usia hingga 25 tahun. Pada usia tanam sudah tinggi, sehingga sulit dipanen, tandanya sudah jarang sehingga secara perhitungan tidak ekonomis lagi. Pengelompokkan berdasarkan masa berbuah (PTPN, 2018), seperti Tabel 2.2 berikut ini:

Tabel 2.2: Pengelompokkan Berdasarkan Masa Berbuah

Kelompok	Masa Berbuah (Tahun)
Tanaman Belum Menghasilkan (TBM)	0-3
Tanaman Menghasilkan	>3

(b) Produktivitas Tanaman Kelapa Sawit Menurut Umur Tanaman

Produktivitas tanaman kelapa sawit (tandan buah segar (TBS)) menurut umur tanaman dalam Kondisi Kebun Percobaan Balit Marihat berdasarkan PTPN. VII tahun 1993 dalam *lecture note* "Budi Daya Kelapa Sawit" (S. Yahya & Suwarto, 2021), seperti Tabel 2.4. berikut ini:

Tabel 2.3: Produktivitas Tanaman Kelapa Sawit Menurut Umur Tanaman dalam Kondisi Kebun Percobaan Balit Marihat

Umur Di Lapangan (Tahun)	Produksi Produksi TBS (ton/ha/Thn)
4	8
5	15
6	17
7	18
8	20
9	21
10	23
11	25
12	26
13	30
14	30
15	30
16	30
17	29
18	28
19	28
20	25
21	23
22	20
23	18
24	18
25	18

(c) Jarak Tanam Pohon Kelapa Sawit

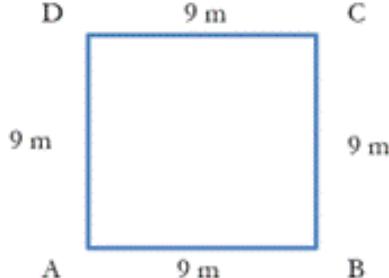
Penanaman kelapa sawit yang baik di lapangan akan menghasilkan tanaman yang sehat (tidak ada yang abnormal, non produktif, mati) dan serbaaam, sehingga tanaman akan cepat berproduksi (kurang dari 30 bulan setelah tanam) dengan hasil awal yang tinggi. Penanaman kelapa sawit

perlu diatur dengan jarak tanam yang sesuai. Jumlah populasi tanaman persatuan luas ditentukan oleh beberapa faktor yaitu jarak tanam yang digunakan dan model jarak tanam yang digunakan. Misalnya, pada penanaman kelapa sawit dengan jarak tanam 9 m x 9 m, akan memiliki jumlah populasi tanaman yang berbeda bila model jarak tanam yang digunakan berbeda (segitiga atau segiempat) (S. Nora & C D. Mual, 2018).

- (a) Penentuan jarak tanam di lapangan harus disesuaikan dengan karakter tanaman, tingkat kesuburan, topografi, dan kondisi setempat;
- (b) Jarak yang teratur hanya dapat dicapai bila dilakukan pemancangan yang baik;
- (c) Sistem jarak tanam pada kelapa sawit berkaitan erat dengan populasi per ha (kerapatan pohon/ha) dan produksi tandan setiap pohon;
- (d) Kerapatan tanaman (jumlah pohon/ha) yang lebih banyak akan mempengaruhi ruang tumbuh tanaman. Terdapat 2 cara dalam menghitung jarak tanam kelapa sawit, yaitu bujur sangkar dan segitiga.

i. Perhitungan Bujur Sangkar

Perhitungannya sama dengan tanaman yang mempunyai sistem pertanaman segiempat seperti Gambar 2.48, yaitu:



Gambar 2.48: Pola Jarak Tanam Bujur Sangkar atau Segiempat

Persamaan:

$$\text{Jumlah Populasi} = \frac{\text{Luas Area (Ha)}}{\text{Jarak Tanam}}$$

Perhitungan:

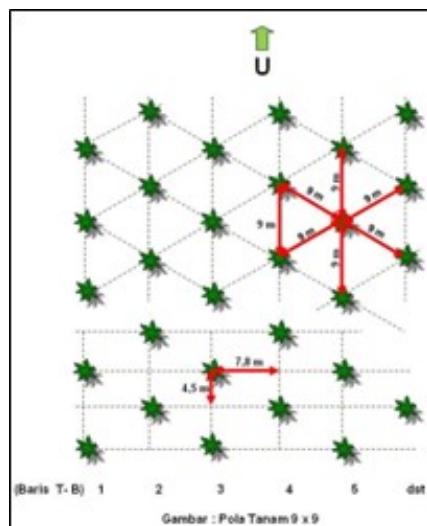
Luas area : 1 ha = 10000 m²

Jarak Tanam : 9m x 9m

$$\text{Jumlah Populasi} = \frac{10000}{9 \times 9} = 123 \text{ Tanaman}$$

ii. Perhitungan Segitiga

Dalam memahami perhitungan jumlah pohon kelapa sawit, maka seperti Gambar 2.49 segitiga sama sisi yang mewakili jarak antar tanaman kelapa sawit.



Gambar 2.49: Pola Jarak Tanam Segitiga
(Sumber: GD Morganic, 2018)

Dimana:

a: Jarak tanam

b: Jarak antar baris yang akan dicari

Persamaan:

$$Jumlah Populasi = \frac{Luas Area (Ha)}{a \times b}$$

Perhitungan:

Luas area : 1 ha = 10000 m²

Jarak Tanam : 9m x 9m x 9m

$$Jumlah Populasi = \frac{10000}{9 \times \sqrt{9^2 - 4,5^2}}$$

$$Jumlah Populasi = \frac{10000}{9 \times 7,79} = 143 \text{ Tanaman}$$

Dari kedua model tanam bujur sangkar dan segitiga dengan jarak tanam yang berbeda, maka populasi tanaman kelapa sawit segitiga lebih banyak dalam satuan luas hektar. Berdasarkan hubungan jarak taman, pola tanam, dan populasi hektar untuk tanaman kelapa sawit dapat diperkirakan seperti pada Tabel 2.4.

Tabel 2.4: Hubungan Jarak Tanam dan Populasi per Hektar Kelapa Sawit

Jarak Tanam (meter)	Bujur sangkar	Segitiga
6	278	320
7	204	236
8	156	180
9	123	143

Pada Tabel 2.4 dapat diketahui bahwa jarak tanam dengan model segitiga terbukti populasinya lebih banyak, hal ini disebabkan karena jarak tanam segitiga dapat memaksimalkan ruang yang ada dalam menangkap sinar matahari, nutrisi, tanah, dan air dengan jalan mengurangi adanya ruang kosong, serta tajuk tiak saling menutupi.

2.12 *State of The Art* (Penelitian Terdahulu)

Beberapa penelitian yang telah dilakukan berkenaan dengan kelapa sawit dirangkum dalam Tabel 2.6. berikut ini:

Tabel 2.5: *State of the Art* (Penelitian Terdahulu)h

Tahun	Penulis	Judul	Metode	Hasil
2015	Wong-in, T. Kaewkong-ka, T., Nagul C., Rajalida L.	Automatic Oil Palm Detection and Identification from Multi-Scale Clustering and Normalized Cross Correlation	Noise filtering dengan spatial domain dan Gaussian smoothing. Untuk Eliminasi Non-Oil Palm dengan Gaussian Blurre image (Black-White), dan Identifikasi Oil Palm dengan K-Means	Wilayah yang diambil dari 21 gambar udara oil palm plantation berbeda region di Thailand dengan Camera Digital pada remote airplane. Penelitian ini dapat mendekripsi dan mengidentifikasi oil palm dengan akurasi di atas 90%.

Tahun	Penulis	Judul	Metode	Hasil
2016	H. Santoso, H. Tani, X. Wang	A simple method for detection and counting of oil palm trees using high-resolution multi-spectral satellite imagery	BW, Pansharpening, Hue Saturation, ArcGIS	Dataset yang digunakan dari data gambar Satelit Quickbird dengan Pansharpening, Sobel Edge Detector, dan ArcGIS 10.2.2. Hasil penelitian ini mencapai akurasi 98%.
2018	H. M. Rizeei, H. Z. M. Shafri, M. A. Mohamoud, B. Pradhan, B. Kalantar	Oil Palm Counting and Age Estimation from WorldView-3 Imagery and LiDAR Data Using an Integrated OBIA Height Model and Regression Analysis.	SVM dan Regressi Analysis	Dataset menggunakan gambar satelit WV-3 dan Light Detection and Range (LiDAR). Hasil penelitian ini mendekati 99%.
2019	N. A. Mubin, E. Nadarajoo, H. Z. M. Shafri, and A. Hamedianfar	Young and Mature Oil Palm Tree Detection and Counting Using Convolutional Neural Network Deep Learning	CNN, GIS	Muda: 95,11% Matang: 92,96%

Tahun	Penulis	Judul	Metode	Hasil
2020	J. Zheng, H. Fu, W. Lo, W. Wu, Y. Zhao, R. Dong, L. Yu..	Cross-Regional Oil Palm Tree Counting and Detection via Multi-Level Attention Domain Adaptation Network	Multi-Level Attention Domain Adaption Network (MADAN)	Dengan cross lokasi. Eksperimen dengan 3 large scale gambar satelit dari kelapa sawit di Peninsular Malaysia dengan berbeda lokasi. Dataset yang digunakan dengan annonated (label) yang sudah diberikan pada proses labeling. MADAN mendapatkan hasil rata-rata F1-Score 84.81% tanpa target domain annotation.

Tahun	Penulis	Judul	Metode	Hasil
2021	X. Liu, K. H. Ghazali, F. Han, I. I. Mohamed	Automatic Detection of Oil Palm Tree from UAV Images Based on the Deep Learning Method	Faster RC-NN	Dataset yang digunakan dari UAV dengan gambar mencakup 22 hektar oil palm tree plantation. Hasil menunjukkan bahwa OA (overall accuracy sebesar 96%).
2021	Y. Nurmasari, A. W. Wijayanto	Oil Palm Plantation Detection in Indonesia Using Sentinel-2 and Landsat-8 Optical Setllite Imagery (Case Study: Rokan Hulu Regency, Riau Province)	Supervised ML (Decision tree, Random Forest, Support Vector Machine, dan NaÃ¢ve Bayes).	Dataset diambil dari Google Earth Engine (GEE) untuk Google Earth, Sentinel-2 dan Landsat-8. Sentinel-2 tercatat 92% tertinggi dengan Random Forest, dan Landsat-8 tercatat 66%.

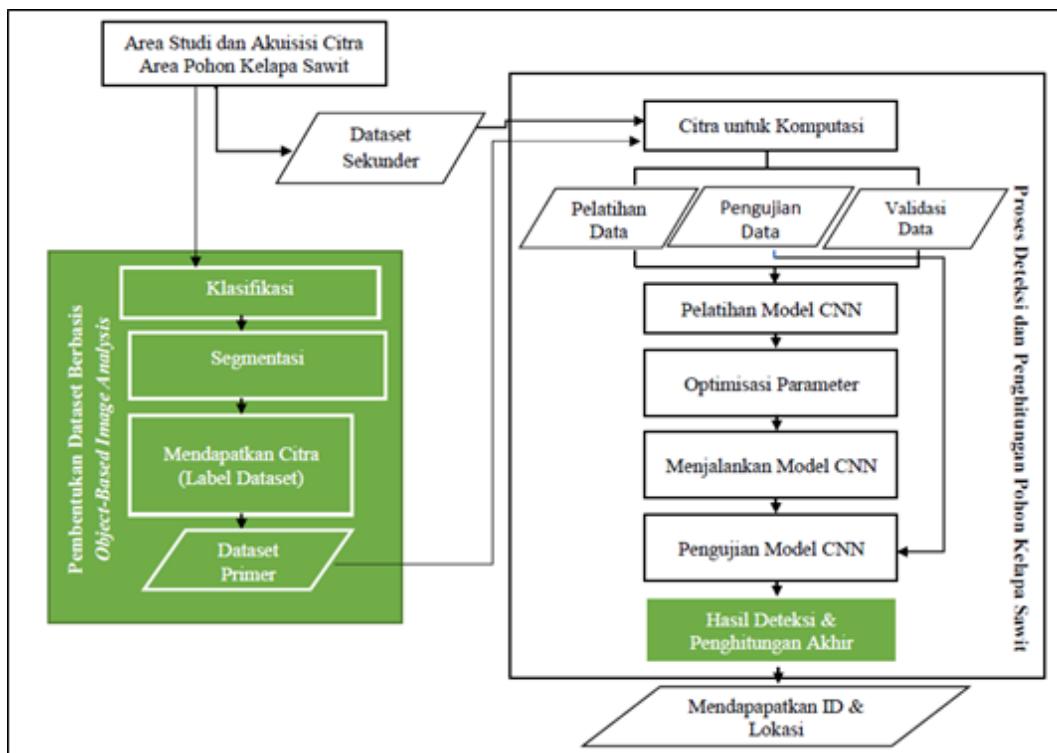
Tahun	Penulis	Judul	Metode	Hasil
2022	H. Wibowo, I. S. Sitanggang, M. Mushthofa, H. A. Adrianto. 2022.	Large-Scale Oil Palm Trees Detection from High-Resloution Remote Sensing Images Using Deep Learning	YOLOv3, YOLOv4, YOLOv5	Akurasi yang dicapai pada penelitian ini YOLOv4: 97.28%, YOLOv4: 97.74%, dan YOLOv5: 94.94%
2023	S. Munir, K. B. Seminar, Sudradjat, H. Sukoco, A. Buono	The Use of Random Forest Regression for Estimating Leaf Nitrogen Content of Oil Palm Based on Sentinel 1-A Imagery	Random Forest Regression	Penilaian kinerja model memperoleh rata-rata MAPE, koreksi, dan MSE 9,68%, 90.32% dan 11.03%. Peta spasial dari distribusi nilai nitrogen di daerah dapat diproduksi dan divisualisasikan di web sehingga dapat diakses dengan mudah dan cepat untuk pengolahan.

BAB III

METODE PENELITIAN

3.1 Tahapan Penelitian

Berdasarkan rumusan masalah, pada dasarnya kegiatan penelitian ilmiah ini terdapat beberapa tahap yang dijelaskan pada Gambar 3.1, warna hijau pada diagram adalah kebaruan (*novelty*).



Gambar 3.1: Tahapan Penelitian yang Dilakukan

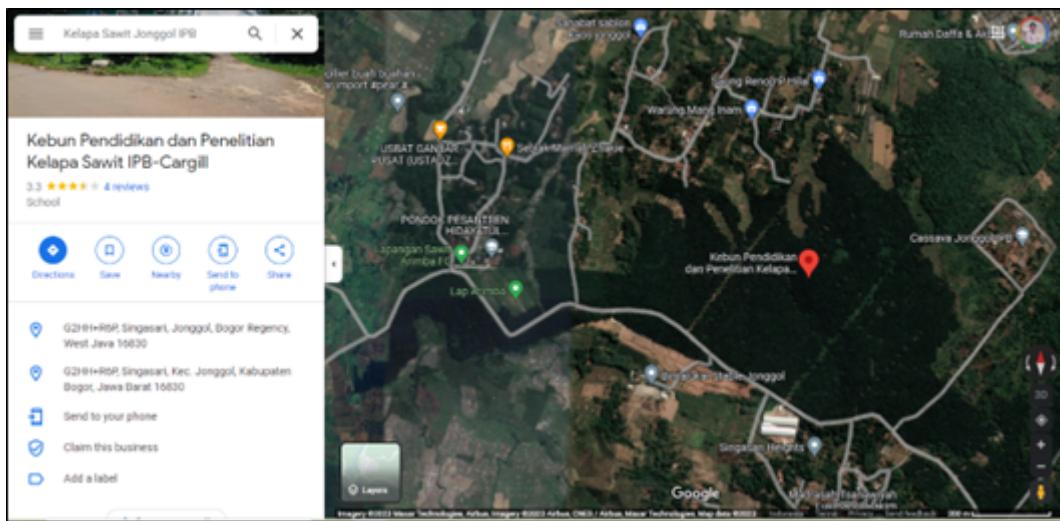
Penelitian ini secara umum digambarkan pada Gambar 3.1. Dimulai dari studi area dan akusisi citra area kelapa sawit yang merupakan gambaran dimana citra data terbentuk untuk mendapatkan dataset primer dan dataset sekunder untuk dapat digunakan sebagai citra untuk komputasi. Penelitian

ini mengusulkan data citra primer yang didapat dari citra drone dengan pendekatan *object-based image analysis* (OBIA). Pendekatan OBIA untuk dihasilkan citra dengan otomasi dataset atau data citra yang belum diberikan label atau anotasi sebuah kelas, dapat dilakukan anotasi atau pemberikan label kelas pohon kelapa sawit secara otomatis, tidak dilakukan anotasi data citra satu per satu. Pendekatan ini dengan menggunakan proses klasifikasi dan segmentasi yang merupakan kunci dari proses pendekatan OBIA. Dataset yang terbentuk digabungkan dengan dataset sekunder yang digunakan untuk menambahdata. Data tersebut merupakan gambar atau citra yang sudah memiliki label atau anotasi untuk menjadi citra yang digunakan untuk komputasi. Citra tersebut terbagi menjadi tiga, yaitu citra data untuk pelatihan, validasi, dan pengujian dengan model CNN untuk mendeteksi dan menghitung citra pohon kelapa sawit, kemudian model digunakan dalam sistem yang terintegrasi secara real-time, sehingga dapat mendeteksi dna menghitung kepala sawit, serta mengetahui dan mendapatkan identitas dan letak posisi koordinat (latitude dan longitude) setiap pohon kelapa sawit yang berhasil terdeteksi.

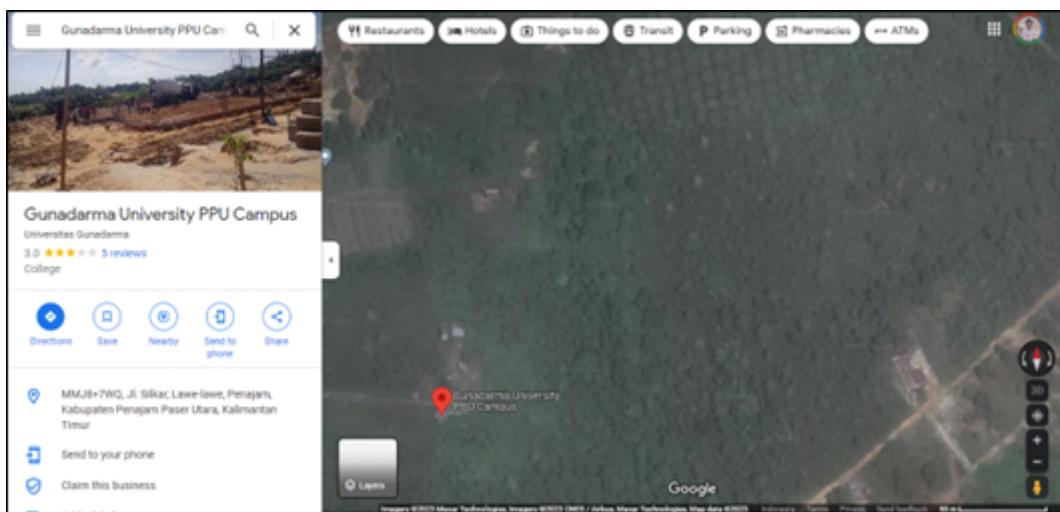
3.2 Area Studi Penelitian Kelapa Sawit

Penelitian ini dilakukan di Indonesia yang berada pada lokasi Kebun Pendidikan dan Penelitian Kelapa Sawit milik Institut Pertanian Bogor-Cargil yang berlokasi di Jonggol, Jawa Barat dan Universitas Gunadarma di Kalimantan yang memiliki lahan yang ditumbuhi pohon kelapa sawit. Hasil lahan perbekunan pohon kelapa sawit ini berupa citra yang diambil dari drone dengan jenis DJI Mavic 2 Pro dari ketinggian 100 m di atas permukaan tanah. Kebun Pendidikan dan Penelitian Kelapa Sawit IPB-Cargil juga dijadikan untuk pengujian penerapan sistem yang diusulkan dalam penelitian ini. Lokasi Kebun Pendidikan dan Penelitian kelapa Sawit tersebut berada pada latitude -6.4277942 dan longitude 106.8418378, sedangkan lokasi di Uni-

versitas Gunadarma PPU Campus berada pada latitude -1.318495 dan longitude 116.6678405. Luas dari kebun pohon kelapa sawit yang berada pada IPB-Cargil berdasarkan studi lapang dan perhitungan dengan DroneDeploy ini seluas 63,48 hektar, dan untuk luas wilayah yang ditanami pohon kelapa sawit pada lahan yang dimiliki Universitas Gunadarma berdasarkan hasil wawancara seluas \approx 18 hektar. Gambar 3.2. dan 3.3. menampilkan posisi studi area dari penelitian ini.



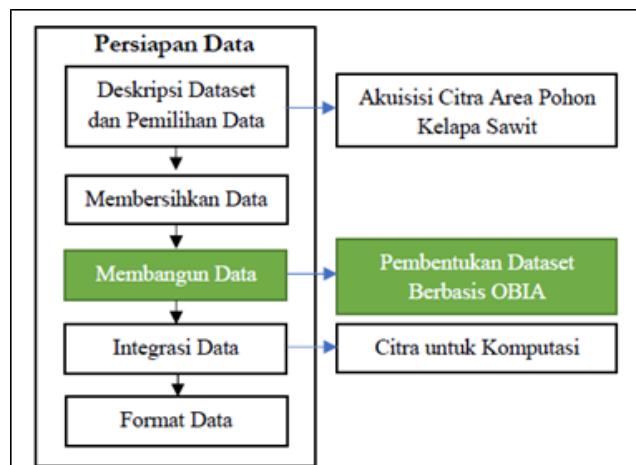
Gambar 3.2: Lokasi Kebun Pendidikan dan Penelitian Kelapa Sawit IPB-Cargil



Gambar 3.3: Lokasi Universitas Gunadarma PPU Campus yang ditanami Pohon Kelapa Sawit

3.3 Persiapan Data

Persiapan data atau yang dikenal dengan *data preprocessing* merupakan tahap untuk menyiapkan atau mengubah data awal dan siap untuk dapat digunakan dalam menjalankan proses modeling pada *deep learning*. Tahap ini digunakan berdasarkan *data mining life cycle* atau siklus hidup data mining yang dikemukakan oleh Cross-Industry Standard Process for Data Mining (CRISP-DM) Nvidia. Salah satu proses tersebut untuk membangun data dengan pembangunan dataset berbasis OBIA, sehingga proses anotasi atau pelabelan secara otomatis menjadi lebih cepat yang menjadi salah satu kebaruan dari penelitian ini. Adapun diagram persiapan data pada penelitian ini pada Gambar 3.4.



Gambar 3.4: Diagram Proses Persiapan Data

3.3.1 Deskripsi Dataset dan Pemilihan Data

Pada bagian ini merupakan deskripsi dataset dan pemilihan data yang berisi akuisisi citra area pohon kelapa sawit yang digunakan sebagai dataset untuk citra komputasi. Akuisisi citra ini merupakan proses menangkap atau memindai dari alat yang digunakan untuk menjadi suatu citra atau gambar. Citra area pohon kelapa sawit dibagi menjadi dua, yaitu citra dataset primer

dan sekunder. Setiap gambar di dataset memiliki format *.jpg.

1. Dataset Primer

Dataset primer yang digunakan data yang dikumpulkan langsung dari sumber data langsung tanpa melalui sumber yang sudah ada atau disediakan. Dataset primer ditangkap dengan menggunakan sarana penelitian drone DJI Mavic 2 Pro dari 100 m di atas permukaan tanah. Dataset ini dijadikan sebagai dataset yang diambil secara langsung untuk digunakan sebagai dataset primer yang belum memiliki label atau anotasi objek kelapa sawit di dalam suatu citra yang ditangkap oleh drone. Citra yang dihasilkan dari area kampus Universitas Gunadarma - PPU dihasilkan 205 gambar dengan dimensi masing-masing sebesar 5472 x 3078 dengan format *.jpg, dan gambar citra pohon kelapa sawit di area tersebut seperti pada Gambar 3.5.



Gambar 3.5: Hasil Citra Area Pohon Kelapa Sawit PPU untuk Dataset Primer

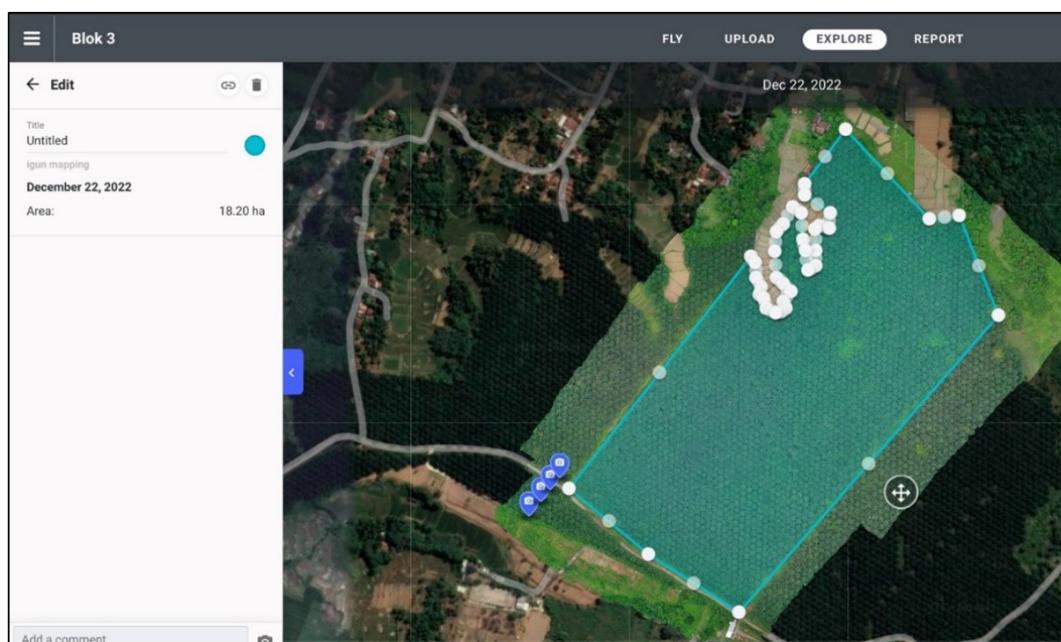
Citra yang dihasilkan dari area Kebun Pendidikan dan Penelitian Kelapa Sawit IPB-Cargil diambil dari area blok 3 dan blok 4 dengan drone DJI

Mavic 2 Pro dari ketinggian 100 m di atas permukaan tanah. Luas area dari blok 3 dan blok 4 berdasarkan perhitungan dengan bantuan layanan DroneDeploy, seperti pada Tabel 3.1.

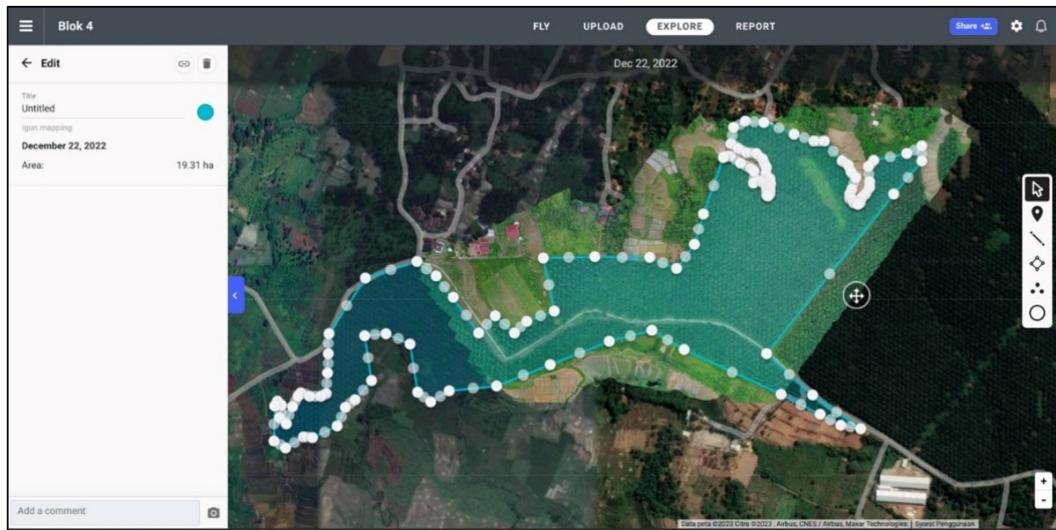
Tabel 3.1: Luas Area Dari Gambar Citra yang Ditangkap oleh drone di Kebun Pendidikan dan Penelitian Kelapa Sawit IPB-Cargil

No.	Area	Luas Area (ha)
1	Blok 3	18,20
2	Blok 4	19,31
	Total	37,51

Studi lapang dengan mengambil gambar bisa mencakup area 37,51 hektar dari blok 3 dan 4 blok dari total 4 blok yang tersedia. Berikut ini hasil tangkapan citra dari Kebun Pendidikan dan Penelitian Kelapa Sawit IPB-Cargil pada Gambar 3.6 untuk blok 3 dengan dimensi gambar 32356 x 31214 dan Gambar 3.7. untuk blok 4, dengan dimensi gambar 43684 x 23452 dengan format *.jpg.



Gambar 3.6: Hasil Citra Blok 3 untuk Dataset Primer (luas area dalam ha)



Gambar 3.7: Hasil Citra Blok 4 untuk Dataset Primer (luas area dalam ha)

Dataset primer ini digunakan untuk pembentukan dataset yang dibentuk secara otomatis dalam proses anotasi atau memberikan label kepada objek kelapa sawit pada citra file yang terdeteksi tanpa harus melakukan proses anotasi atau pemberian label kelapa sawit satu persatu. Proses ini dilakukan pada bagian sub bab 3.3.3. sebagai salah satu kebaruan atau *novelty* yang diusulkan.

2. Dataset Sekunder

Data sekunder yang digunakan merupakan data citra yang sudah memiliki label atau anotasi sebagai pohon kelapa sawit (oil palm), dan dikatakan sebagai dataset. Data yang digunakan dalam penelitian ini diambil dari 1795 citra satelit sentinel 2 dari roboflow (JS, 2022). Dataset yang sudah diberikan label dengan memiliki kotak atau bounding box. Dataset ini masing-masing memiliki dimensi citra gambar 1024 x 1024 dengan format file *.jpg. Dataset sekunder yang sudah diberikan anotasi atau label oil palm diakses melalui roboflow, seperti pada Gambar 3.8.



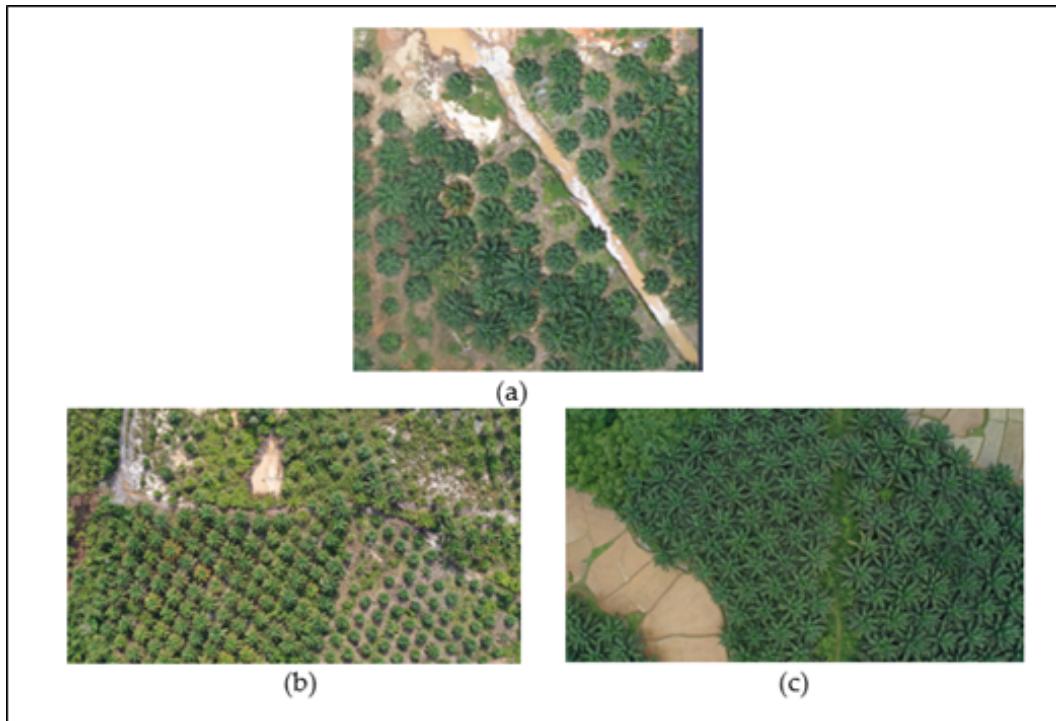
Gambar 3.8: Data sekunder yang sudah Diberikan Label atau Anotasi Oil Palm
(Sumber: JS, 2022)

Pada data sekunder ditambah untuk mendapatkan lebih banyak citra dan representasi dengan menambahkan citra menjadi 3987 citra dengan melakukan augmentasi seperti pada Tabel 3.2.

Tabel 3.2: Augmentasi Dataset Sekunder untuk Menambahkan Data

Augmentasi	Deskripsi
Crop	0% Minimum Zoom, 40% Maximum Zoom
Rotation	Between -39° and +39°
Hue	Between -30° and +3°
Exposure	Between -17° and +17°
Blur	Up to 2.5px
Noise	Up to 2% of pixels
Mosaic	Applied

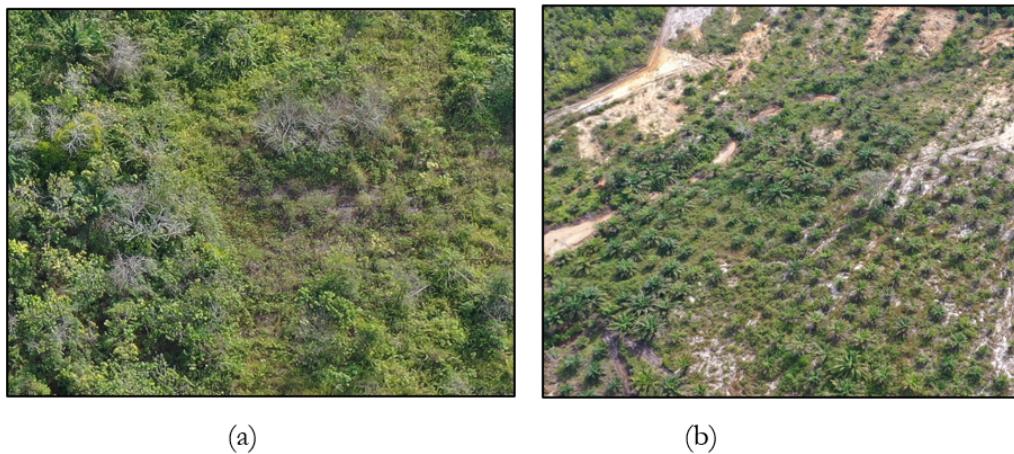
Proses Tabel 3.2. dilakukan pada roboflow. Roboflow merupakan sebuah platform yang memfasilitasi proses anotasi dataset dan juga menyimpan data citra. Untuk data citra primer yang digunakan menggunakan drone DJI Mavic 2 Pro yang diambil dari Perkebunan Kelapa Sawit Institut Pertanian Bogor yang terletak di Jonggol dan di Universitas Gunadarma Kalimantan. Dataset primer yang diambil di Universitas Gunadarma di Kalimantan menghasilkan 205 citra, sedangkan data citra primer yang diambil dari Jonggol, Jawa Barat pada blok 3 dan 4 dipecah, sehingga menghasilkan 69 citra seperti yang ditunjukkan pada Gambar 3.9.



Gambar 3.9: (a) Gambar Dataset Sekunder; (b) Gambar Dataset Primer di Kalimantan; (c) Gambar Dataset Primer di Jonggol;

3.3.2 Membersihkan Data

Bagian ini dikenal dengan *clean data* atau membersihkan data. Pada penelitian ini yang dilakukan pada tahap membersihkan data adalah memvalidasi citra yang ditangkap oleh drone berada pada tangkapan citra yang tegak, tidak diambil dari sisi samping dari kamera drone untuk menjaga keseragaman data yang dimiliki pada dataset, serta dilihat juga apakah gambar tersebut memiliki tangkapan citra pohon kelapa sawit atau tidak, seperti pada Gambar 3.10.



(a)

(b)

Gambar 3.10: Citra yang tidak digunakan sebagai dataset (a) tidak ada pohon citra kelapa sawit; (b) tangkapan citra dari drone tidak dari sisi atas yang tegak

3.3.3 Membangun Data

Persiapan dan pemahaman data dari data primer dan sekunder merupakan tugas paling penting. Tugas ini membutuhkan dan menghabiskan waktu hampir 70% waktunya untuk menganalisis kumpulan data. Tugas yang dilakukan dalam dataset diantaranya persiapan dan pemahaman data, seperti memberi anotasi atau label pada data atau dengan istilah lain membangun data (*construct data*). Selama ini, dalam melakukan anotasi atau pemberian label dilakukan masih manual atau memberikan tanda kotak (*bounding box*) untuk setiap data atau objek pada gambar yang ditandai sebagai dataset. Maka, dibutuhkan suatu pendekatan untuk melakukan anotasi atau labelisasi secara otomatis, yaitu pembentukan dataset berbasis *object-based image analysis* (OBIA). Pendekatan dalam penelitian ini menggunakan OBIA. Secara umum OBIA menawarkan sebuah alternatif untuk pemrosesan citra berbasis objek. Mengekstrak informasi tutupan lahan/penggunaan lahan yang merupakan tantangan baru, tetapi dengan prinsip dasar klasifikasi dan segmentasi dapat digunakan untuk membuat dataset secara otomatis. Dataset yang digunakan berupa gambar primer yang sudah dimiliki berasal data tangkap citra

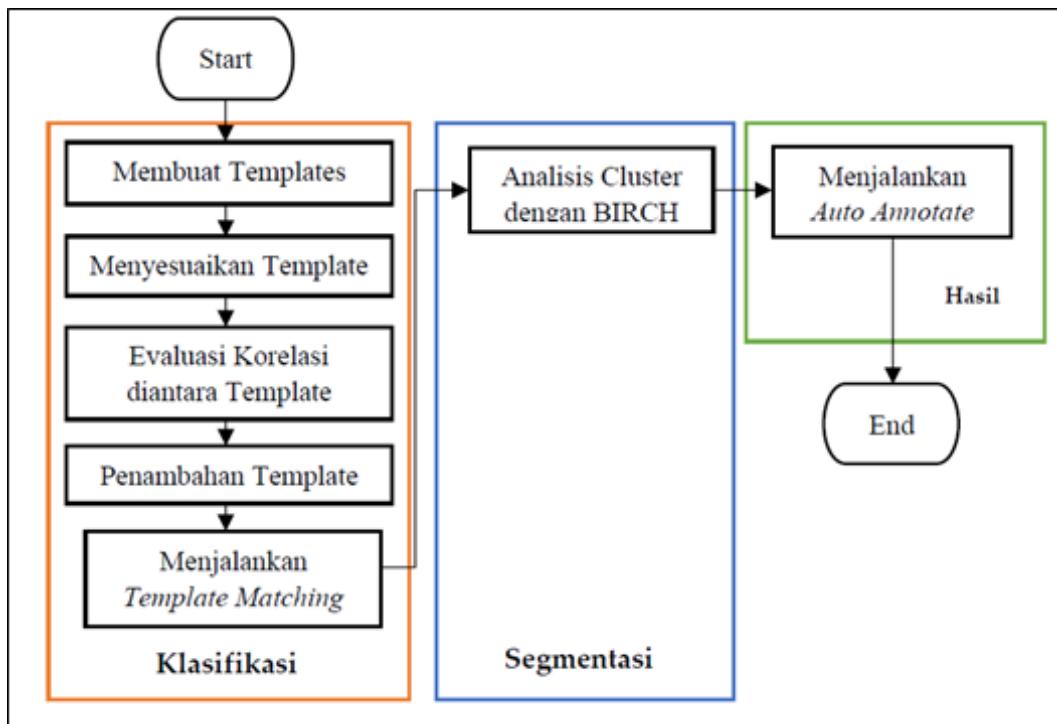
drone. Algoritma *template matching* dan *balanced iterative reducing and clustering using hierarchies* (BIRCH) dapat digunakan untuk pengenalan gambar, yaitu membandingkan dua atau lebih gambar yang identik pada waktu dan kondisi yang berbeda, sehingga dapat menyaring objek dengan nilai ambang batas atau (thereshold) dan membentuk *fine tuning* atau nilai yang lebih baik untuk dapat dideteksi secara otomatis. Pembentukan atau pemberian label dengan pendekatan OBIA inilah menjadi kebaruan atau *novelty* untuk mendukung proses tersebut menjadi lebih cepat dan signifikan dalam pemberian label atau anotasi pada dataset citra.

Pengembangan sistem untuk membangun dataset dengan anotasi atau label otomatis dengan sarana penelitian seperti pada Tabel 3.3.

Tabel 3.3: Sarana Penelitian Pengembangan Sistem untuk Membangun Dataset dengan Anotasi atau Label Otomatis

Alat	Keterangan
Drone	DJI Mavic 2 Pro
Jupyter Notebook	V 6.4.11
Python	Python 3
OS Name	Microsoft Windows 10 Pro
Processor	Intel Core i5-6300U CPU @2.40 GHz
Physical Memory (RAM)	8,00 GB

Diagram alur untuk membuat dataset otomatis seperti pada Gambar 3.11



Gambar 3.11: Diagram Alir Pembangunan Dataset Berdasarkan Pendekatan OBIA

3.3.3.1 Membuat Templates

Pada bagian ini, pembuatan template awal dilakukan dalam beberapa tahap, antara lain:

1. Input Image

Proses input pertama dilakukan sebagai proses input gambar yang nantinya dilakukan pengenalan objek pohon kelapa sawit. File input tersebut berformat *.jpg.

2. Preprocessing

Proses preprocessing pada gambar digunakan untuk menyamakan ukuran matriks yang akan dicocokkan dengan algoritma *Template Matching Correlation* dan menghilangkan noise pada gambar. Pertama, dipersiapkan dimensi citra berukuran 5472 x 3078, kemudian dibuat fungsi untuk menampilkan citra dan fungsi *click view* agar pengguna dapat menekan

sebagai simbol untuk menentukan citra dengan titik lokasi dalam koordinat (x, y) mana yang digunakan sebagai citra template dasar.

3.3.3.2 Menyesuaikan Template

Menyesuaikan template digunakan sebagai dasar untuk dapat menentukan dan mengoreksi titik tengah (x, y) gambar dasar yang digunakan, dan memastikan bahwa titik tengah gambar berada di tengah. Mengubah posisi ini dengan menggunakan titik geser sebanyak 2 piksel, bisa ke kiri, kanan, atas, atau bawah. Dalam menerapkan ukuran gambar yang dipilih sebagai template gambar dasar yaitu 12 x 12. Adapun persamaan yang diterapkan untuk menyesuaikan template sebagai berikut.

function down (no_id, MPixel = a) : pixel [no_id] [1] + = MPixelNow (1)

function up (no_id, MPixel = a) : pixel [no_id] [1] - = MPixelNow (2)

function left (no_id, MPixel = a) : pixel [no_id] [0] - = MPixelNow (3)

function right (no_id, MPixel = a) : pixel [no_id] [0] + = MPixelNow (4)

function delete (no_id) : pixel [no_id_{new}] = list(pixel[[no_id_{new}]]) (5)

Keterangan:

- no_id = nomor identifikasi gambar yang digunakan sebagai template.
- MPixel = variabel seberapa jauh piksel bergerak.
- a = angka seberapa jauh piksel bergerak.

1 = matriks id yang digunakan untuk referensi arah y.

0 = matriks id yang digunakan untuk referensi arah x.

- MPixelNow = nilai pembaruan setelah bergeser sebanyak piksel tertentu.

3.3.3.3 Evaluasi Korelasi diantara Template

Pada bagian ini, setelah gambar template awal diambil, langkah berikutnya adalah mengevaluasi korelasi di antara template yang ada. Hal ini digunakan dalam mengambil template untuk melihat gambar yang sesuai. Untuk mencocokkan gambar, gunakan korelasi silang dan normalisasi untuk menemukan contoh template dalam gambar. Dalam mengevaluasi korelasi antara template, menggunakan cara kerja pencocokan template dan *normalized cross correlation* (NCC).

1. *Template Matching*

Metode gambar yang besar untuk menemukan bagian kecil yang cocok dengan gambar template disebut pencocokan template. Metode ini membandingkan template yang diberikan dengan ukuran jendela yang sama dan gambar yang paling mirip dengan template tersebut. Dalam template matching, proses pencarian dimulai dengan mencari lokasi pusat dari template gambar dan mengisi nilai nol dari referensi gambar. Gambar 3.12. template dan referensi matriks.

Template		
8	8	6
5	4	3
4	9	3

Reference							
2	2	2	2	2	2	2	2
2	6	8	8	6	2	2	2
2	3	5	4	3	2	2	2
2	3	4	8	3	2	2	2
2	3	3	3	2	2	2	2
2	2	2	2	2	2	2	2
2	2	2	2	2	2	2	2

(a)

(b)

Gambar 3.12: Template and Referensi Matriks

Gambar 3.12 (a) menunjukkan dua matriks yang digambarkan sebagai template dan referensi gambar sebelum proses pencocokan gambar. Per-

tama kali proses dimulai, lokasi di tengah template harus diidentifikasi terlebih dahulu, sehingga titik tengah akan jatuh pada piksel referensi pertama seperti yang ditunjukkan pada Gambar 3.12 (a). Dalam menemukan titik tengah lokasi template, akan lebih mudah untuk membagi gambar template menjadi dua. Pada contoh template ini, titik tengah lokasi berada pada koordinat (2,2) dengan nilai 4. Pada gambar referensi ditambahkan baris dan kolom dengan nilai nol pada setiap sisi gambar. Jumlah baris dan kolom yang diisi dengan angka nol sama dengan panjang sumbu x dan sumbu y pada template dibagi dua. Gambar 3.12 (b) menunjukkan bagaimana gambar template mengisi piksel referensi ketika proses dimulai. Setiap sisi dari gambar referensi berhasil ditambahkan dengan baris dan kolom dengan nilai 0. Dengan menambahkan gambar dengan nilai nol, template tengah dengan mudah ditempatkan pada piksel pertama dari gambar referensi. Untuk setiap lokasi, nilai gambar ditambahkan dan template dijumlahkan dengan menggunakan metode ukuran kemiripan atau *normalized cross correlation* (NCC) dan hasil komputasi disimpan dalam matriks tertentu yang memiliki jumlah gambar referensi yang sama. Akhirnya, matriks dihasilkan setelah semua area gambar diisi. Komputasi gambar digunakan untuk menentukan seberapa tepat template dari area referensi gambar.

2. Normalized Cross Correlation (NCC)

Korelasi Silang Ternormalisasi (*Normalized Cross Correlation*, NCC) selalu dipilih karena memiliki keuntungan sebagai ukuran kemiripan yang kuat. NCC untuk nilai korelasi dengan persamaan sebagai berikut:

$$r = \frac{\sum_{k=1}^n (x_{ik} - \bar{x}_i) \cdot (x_{jk} - \bar{x}_j)}{\sqrt{\left[\sum_{k=1}^n (x_{ik} - \bar{x}_i)^2 \cdot \sum_{k=1}^n (x_{jk} - \bar{x}_j)^2 \right]}} \quad (6)$$

Dengan, xi dan xj adalah rata-rata matriks i dan j yang dapat dihitung dengan:

$$\bar{x}_i = \frac{1}{n} \sum_{k=1}^n x_{ik} \quad (7)$$

$$\bar{x}_j = \frac{1}{n} \sum_{k=1}^n x_{jk} \quad (8)$$

Keterangan:

- r = nilai korelasi antara dua matriks.
- x_{ik} = nilai piksel ke-k dalam matriks i.
- x_{jk} = nilai piksel ke-k dalam matriks j.
- \bar{x}_i = nilai piksel rata-rata dari matriks i.
- \bar{x}_j = nilai rata-rata piksel dari matriks j.
- n = jumlah piksel dalam sebuah matriks.

Setelah mendapatkan nilai korelasi antar template, langkah selanjutnya adalah menentukan hasil rata-rata korelasi antar template untuk dijadikan nilai minimum dari nilai acuan yang akan digunakan dalam klasifikasi hasil pengujian pada bagian match template. Perhitungannya sebagai berikut:

$$mean_r = \frac{(\sum_{r, k=1}^n)}{(\sum_{k=1}^n)} \quad (9)$$

Keterangan:

- $(\sum_{r, k=1}^n)$ = jumlah nilai r dari k-1 hingga n
- $(\sum_{k=1}^n)$ = jumlah data dari k-1 to n

Deteksi akhir dari nilai korelasi yang diperoleh, nilai tersebut kemudian dikonversi dalam rentang 0 hingga 255 pada kanal hijau untuk digambaran dalam bentuk persegi panjang pada koordinat gambar yang terdeteksi pohon.

3.3.3.4 Penambahan Template

Pada bagian ini, untuk menambahkan gambar template asli, yaitu dengan memutar setiap gambar template sebesar 30 derajat. Hal ini dapat diatur sesuai dengan kebutuhan pengguna. Hal ini dilakukan untuk menambahkan gambar template, sehingga jika mendapatkan gambar yang ternyata dirotasi dari gambar aslinya, maka dapat menemukan kecocokan dari gambar template.

Rumusnya sebagai berikut:

$$\text{rotations} = [i * 30 \text{ for } i \text{ in range } (1, 4)] \quad (10) \text{ Di mana gambar i}$$

dirotasi 30 derajat sebanyak 4 kali dari gambar template yang ada.

3.3.3.5 Menjalankan Template Matching

Proses penentuan klasifikasi citra yang telah diuji menggunakan algoritma Template Matching Normal Cross Correlation, seperti pada persamaan (6) pada citra yang akan ditentukan anotasi citranya dan sesuai dengan nilai minimum mean r.

3.3.3.6 Analisis Kluster dengan BIRCH

Balanced Iterative Reducing and Clustering Using Hierarchies (BIRCH) sebagai algoritma data mining tanpa pengawasan yang digunakan untuk melakukan pengelompokan hirarkis pada kumpulan data yang besar. Algoritma ini menggunakan pengelompokan fitur (feature clustering/FC) dan pengelompokan pohon fitur (CF Tree), dua konsep untuk deskripsi klaster secara umum. BIRCH mampu meningkatkan algoritma dalam melakukan clustering data set yang besar pada kecepatan dan skalabilitas dan sangat cocok untuk menangani masalah pengelompokan data atribut diskrit dan kontinu, seperti piksel yang berdekatan dan menyaring hasil untuk menemukan nilai ambang batas yang lebih baik dari objek yang terdeteksi dari template matching. Objek-objek dalam data set disusun dalam bentuk sub-clustering CF. CF ini kemudian

digabungkan menjadi k-kelompok dengan menggunakan prosedur pengelompokan hirarki tradisional. CF adalah tiga informasi yang berisi $CF = (N, LS, SS)$.

Keterangan:

- N adalah jumlah titik data,
- LS adalah jumlah nilai X (nilai atribut),
- SS adalah jumlah X kuadrat.

Jika ada 2 CF yang digabungkan, maka teorema:

$$CF_{12} = (N_1 + N_2, \bar{LS}_1 + \bar{LS}_2, SS_1 + SS_2) \quad (11)$$

Selanjutnya, untuk mencari lokasi fitur cluster yang cocok untuk digunakan beserta rumus jaraknya. rumus jarak yang digunakan D2 adalah sebagai berikut:

$$D2 = \frac{\sqrt{(N_1SS_2)+(N_2SS_1)-2LS_1LS_2}}{N_1N_2} \quad (12)$$

Selanjutnya, untuk dapat menghitung jari-jari daun CF dengan menggunakan rumus:

$$R = \frac{\sqrt{SS-(LS)^2}/n}{n} \quad (13)$$

Setelah mendapatkan nilai radius dalam kisaran titik tengah gambar, kemudian digunakan threshold. Threshold digunakan pada BIRCH untuk memisahkan objek dari latar belakang pada sebuah citra dengan menggunakan nilai 0,5 sebagai nilai ambang batas atau threshold. Setelah dilakukan perhitungan dengan BIRCH untuk mereduksi atau mengurangi data yang tidak seharusnya masuk, maka data citra ditampilkan dengan ukuran 12 x 12 dengan nilai minimum dari mean r dan citra terdeteksi sebagai citra yang akan dianotasi. Hasil citra juga menampilkan nilai kecocokan dari korelasi yang ditentukan.

3.3.3.7 Menjalankan Auto-Annotate Datasets

Proses auto annotation ini merupakan proses dimana mendapatkan data citra (*data collecting*) yang dimiliki, yaitu 205 citra yang telah berhasil dideteksi untuk didapatkan sebagai dataset. Keluaran dari proses ini adalah number, koordinat x, koordinat y, *width*, dan *height* setiap objek yang terdeteksi dari gambar yang disimpan dalam sebuah file berekstensi *.txt. Hasil ini membutuhkan waktu sesuai dengan perangkat yang digunakan. Hasil anotasi dataset ini juga menjadi kebaruan atau *novelty* dalam penelitian ini karena data primer yang diambil secara langsung, serta sudah memiliki anotasi atau label 'oil palm' pada citra pohon kelapa sawit di area lahan.

3.3.3.8 Integrasi Data

Pada tahap ini dilakukan proses menggabungkan atau menyatukan dua dataset, yaitu dataset primer dan sekunder ke dalam satu sumber. Dataset sekunder digunakan layanan roboflow untuk menampung data, dan hasil dari pembangunan dataset dengan anotasi atau pelabelan otomatis yang telah dilakukan, maka diunggah data citra dan file *.txt untuk dapat disesuaikan dengan format data yang sama.

3.3.3.9 Format Data

Pada tahap format data ini, salah satu tahap yang sangat penting karena kesuksesan suatu model machine atau deep learning sangat tergantung pada kualitas data yang digunakan untuk pelatihan model tersebut. Pada tahap ini yang dilakukan adalah memeriksa dan memastikan bahwa dataset yang digunakan tidak memiliki nilai yang hilang (yaitu adanya *bounding box*) yang menyatakan bahwa objek tersebut adalah objek kelapa sawit dengan kelas 'oil palm' yang dapat terlihat pada layanan roboflow, kemudian menyesuaikan tipe data pada citra yaitu pada dataset primer dan sekunder sama-sama *.jpg, dan

mengurangi dimensi dataset citra dengan melakukan resize citra menjadi 640 x 640 untuk mengurangi *noise* pada data dan menyamakan dataset primer dan sekunder dengan menggunakan layanan roboflow.

3.4 Citra untuk Komputasi

Citra gambar untuk komputasi merupakan kumpulan dari citra dataset primer dan sekunder. Pada tahap ini dataset dikumpulkan dalam satu folder dengan menggunakan layanan Google Drive untuk dapat diakses dan dibagi dalam pelatihan, validasi, dan pengujian. Dataset mengandung 6056 data, yang terbagi ke dalam 78% data pelatihan, 20% data validasi dan 2% data pengujian.

3.5 Pelatihan Model CNN

Algoritma CNN model yang digunakan adalah YOLOv7 (sesuai hasil pelatihan dan pengujian) yang divalidasi dengan menggunakan K-Fold Cross-Validation, dimana semua bagian dataset dapat digunakan untuk training dan testing. Teknik ini digunakan untuk evaluasi performance dari model untuk mengurangi overfitting. Langkah awal dalam menggunakan metode cross validation adalah memilih nilai k, kemudian membagi dataset menjadi beberapa nilai k yang digunakan. Pada penelitian ini menggunakan k = 5, dimana proses ini diulang sebanyak k = 5 kali. K=5, berarti, diberikan dataset and dibagi ke dalam 5 folds yang digunakan untuk menjalankan data pelatihan dan data uji. Model kemudian dilatih pada dataset pelatihan dan divalidasi pada dataset pengujian. Pada penelitian ini, dipilih nilai k = 5, mengacu pada scikit learn, dengan menerapkan 5-folds.

Masing-masing setiap fold terdiri dari pelatihan dan pengujian. Data pelatihan merupakan gabungan dari data pelatihan sebanyak 4744 data dan

1187 validasi, untuk data test sebanyak 125 data citra. Setiap fold berisi data ini untuk dapat diketahui pada fold mana hasil model yang dilatih hasilnya lebih baik.

3.6 Optimisasi Parameter

Dalam meningkatkan performa dari model CNN yang digunakan, maka dilakukan optimisasi parameter. Optimisasi parameter pada penelitian ini menggunakan epoch, batch size, initial learning rate, final one cycle learning rate, momentum, dan weight decay sebagai optimizer weight decay seperti pada Tabel 3.4.

Tabel 3.4: Optimisasi Parameter

Parameter	Nilai
lr0	0.01 (1E-2)
rf	0.1
Momentum	0.937
Weight decay	0.0005 (5e-4)
Epoch	30
Batch size	16

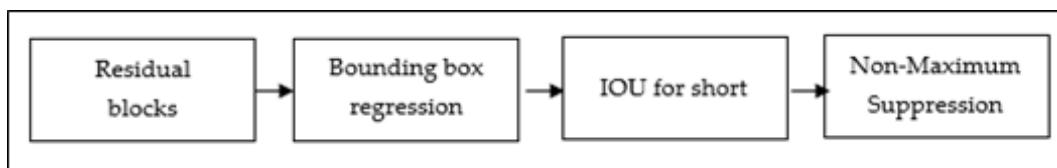
Parameter yang digunakan diantaranya adalah learning rate dengan nilai 0.01 *stochastic gradient descent* (SGD). Learning rate ini hyperparameter yang paling penting ketika mengonfigurasi neural network. Hal sangat penting untuk mengetahui bagaimana menyelidiki efek dari laju pembelajaran pada kinerja model dan untuk membangun intuisi tentang dinamika laju pembelajaran pada perilaku model. Nilai *learning rate* yang terlalu kecil dapat menyebabkan proses pelatihan yang lama dan bisa *stuck* atau berhenti, sedangkan nilai yang terlalu besar dapat menyebabkan pembelajaran set bobot yang tidak optimal menjadi terlalu cepat atau proses pelatihan yang tidak stabil. Penelitian ini menggunakan learning rate dari SGD 0.001 karena metode optimasi adaptif lainnya diketahui memiliki generalisasi yang buruk dibandingkan

dengan SGD. Metode lainnya cenderung berkinerja baik pada data pelatihan tetapi, tidak saat data pengujian. Parameter yang digunakan berikutnya adalah momentum. Momentum digunakan sebagai metode yang membantu mempercepat vektor gradien ke arah yang benar, sehingga menghasilkan konvergen yang lebih cepat, atau dengan kata lain sebagai pengoptimal yang meminimalisir dampak noise dalam konvergensi ke bobot optimal.

Pada penelitian ini menggunakan epoch sebesar 30 epoch karena sebelumnya sudah diujicoba kan dengan epoch 55 dan 50, maka tidak bisa berjalan hingga selesai karena adanya *out of memory* dan batas GPU dari Google, sehingga digunakan 30 epoch dengan batch size 16. Epoch menjadi penting karena hyperparameter yang menentukan berapa kali algoritma *deep learning* bekerja melewati seluruh dataset. Batch size yang digunakan adalah 16, ini mengacu pada jumlah contoh data pelatihan yang digunakan dalam satu iterasi.

3.7 Menjalankan Model CNN

Pada tahap ini menjalankan hasil training dan test dari dataset yang sudah disiapkan dengan menggunakan parameter tercantum pada Tabel 3.4 dengan YOLOv7 menggunakan teknik K-Fold Cross-validation untuk evaluasi model untuk setiap segmen. Pada tahap ini, model dijalankan dengan menggunakan YOLOv7, yang dimana algoritma ini bekerja berdasarkan empat pendekatan seperti pada alur diagram Gambar 3.13.



Gambar 3.13: Cara Kerja Pendekatan Algoritma YOLOv7

Langkah pertama ini dimulai dengan membagi gambar asli menjadi sel

grid $S \times S$ dengan bentuk yang sama. Setiap sel grid bertanggung jawab untuk menentukan dan memprediksi lokasi kelas dan nilai probabilitas/*confidence* objek tersebut. Langkah selanjutnya adalah menentukan kotak pembatas yang sesuai dengan persegi panjang yang menyoroti semua objek dalam gambar. YOLO menentukan sifat-sifat kotak pembatas ini menggunakan modul regresi tunggal dalam bentuk berikut, di mana Y adalah representasi vektor akhir dari setiap kotak pembatas.

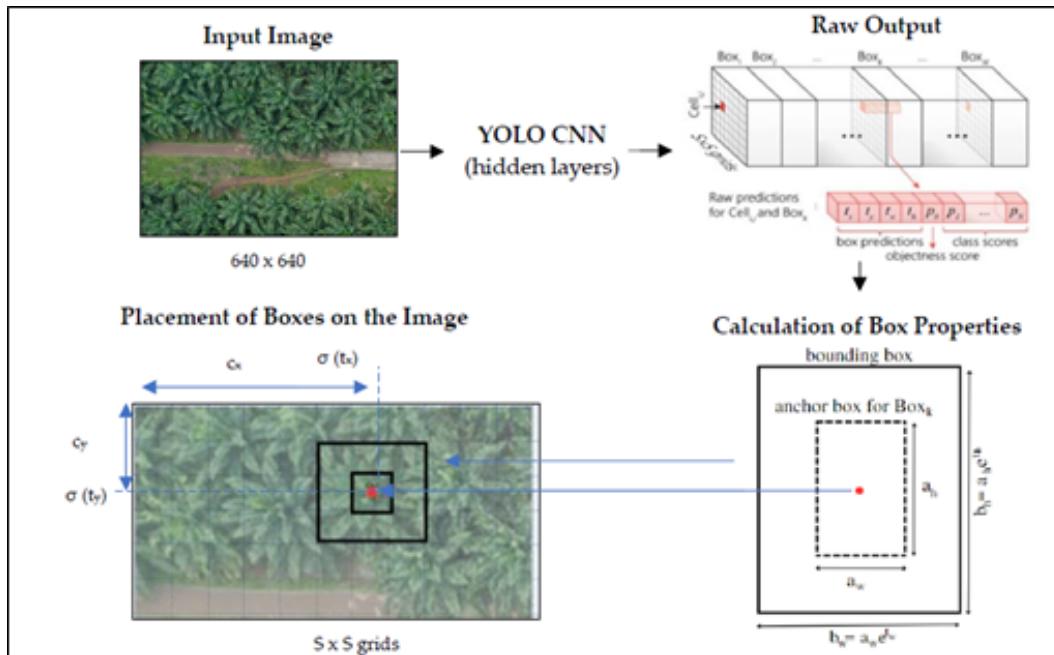
$$Y = [pc, bx, by, bh, bw, c1, c2] \quad (14)$$

Keterangan:

- pc: sesuai dengan skor probabilitas dari sisi yang berisi objek
- bx, by: koordinat tengah x dan y dari pusat kotak pembatas (bounding box)
- bh, bw: tinggi dan lebar kotak pembatas.
- c1 dan c2: berhubungan dengan kelas (dalam penelitian ini menggunakan 1 kelas, yaitu *oil palm*)

Sering kali, satu objek dalam sebuah gambar dapat memiliki beberapa kandidat kotak grid untuk prediksi, meskipun tidak semuanya relevan. Tujuan *Intersection of Union* (IoU) (nilai antara 0 dan 1) untuk membuang kotak-kotak tersebut dan hanya menyimpan kotak-kotak yang relevan, kotak ini lah yang nantinya menjadi *bounding box* yaitu klasifikasi kelas dan juga lokalisasi, yaitu titik tengah koordinat dari *bounding box* tersebut. Terkadang IoU menetapkan nilai ambang batas untuk IoU yang tidak cukup untuk sebuah objek, Di sinilah, dapat menggunakan non-maximum suppression untuk menyimpan hanya kotak dengan nilai probabilitas deteksi tertinggi, sehingga kotak tersebut dapat menentukan atau mendeteksi kelas apa dalam *bounding box* tersebut. Hasil dari menjalankan model ini, didapatkan hasil evaluasi kualitas model terbaik, dan hasil tersebut akan dipilih sebagai model terbaik.

Pada Gambar 3.14 skema diagram dari algoritma YOLOv7 yang digunakan dalam penelitian ini, bagaimana dataset dapat dideteksi sebagai oil palm.



Gambar 3.14: Skema Diagram Algoritma YOLO

Pada penelitian ini dataset yang digunakan sebagai *input image* atau data citra yang masuk ke YOLO (*neural network*) dengan resolusi 640×640 . Artinya, ukuran gambar yang lebih besar diubah ke resolusi 640×640 yang merupakan ukuran *default* dari YOLO. Hal ini dilakukan untuk mengurangi resiko kehilangan akurasi prediksi yang baik. Berdasarkan Gambar 3.13 bahwa kalkulasi box properties menjadi penting untuk mendapatkan objek yang berhasil dideteksi dengan tampilanya bounding box, dan diketahui titik tengah (x, y) atau lokasi dari setiap objek kelas oil palm yang berhasil dideteksi.

Setelah berhasil mengetahui bagaimana diagram skema kerja algoritma YOLO untuk mendeteksi objek *oil palm* pada data citra, maka dibutuhkan evaluasi model performance dari model yang dijalankan dalam penelitian ini. Evaluasi performance untuk mendapatkan model terbaik dengan evaluasi performa dari sebuah model pendekripsi objek biasanya menggunakan precision,

recall, dan F1-score serta *best Mean Average Precision* atau yang dikenal dengan *best mAP* yang umumnya digunakan untuk menganalisis kinerja sistem deteksi dan segmentasi objek.

$$\text{Precision} = \frac{(TP)}{((TP)+(FP))} \quad (15)$$

$$\text{Recall} = \frac{(TP)}{((TP)+(FN))} \quad (16)$$

$$\text{F1-Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (17)$$

$$mAP = \frac{1}{N} \sum_{i=1}^N AP_i \quad (18)$$

Keterangan:

- TP: True Positive untuk jumlah piksel yang benar diklasifikasikan sebagai pohon kelapa sawit
- FP: False Positive untuk jumlah piksel yang tidak benar diklasifikasikan sebagai pohon kelapa sawit.
- FN: False Negative untuk jumlah piksel yang tidak benar diklasifikasikan sebagai latar belakang.

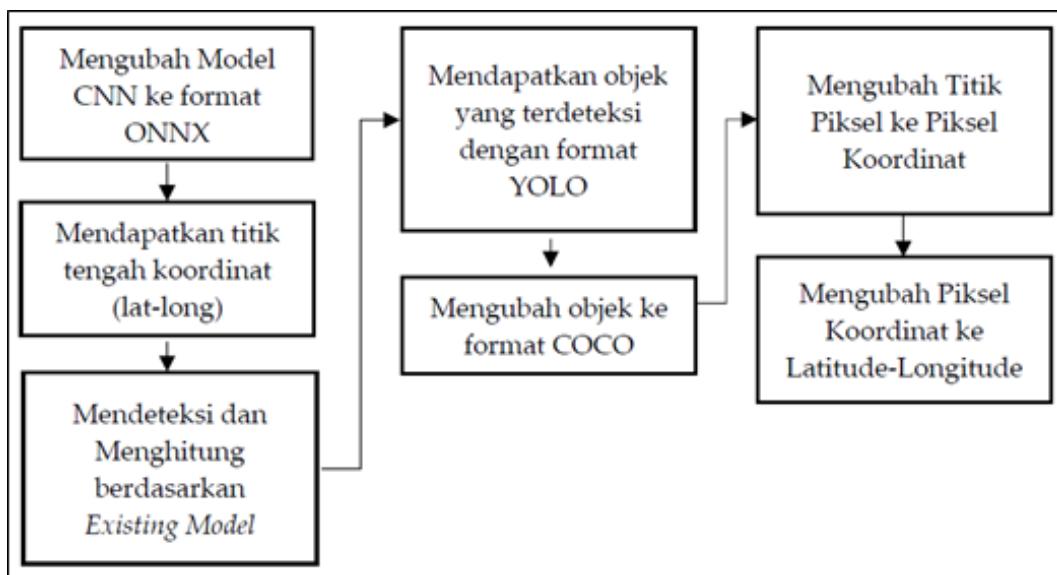
Pada tahap ini model yang dijalankan dengan menggunakan sarana penelitian dengan menggunakan mesin untuk komputasi dengan mengakses Google Colab Pro (DGX-SM4-A100, 40 GB) dan mesin (DGX-A100, 20 GB) milik Universitas Gunadarma.

3.8 Hasil Deteksi dan Penghitungan Akhir

Tahap ini menggunakan sebuah web framework react js untuk menerapkan hasil best model dari performance terbaik untuk dapat digunakan pada citra satelit realtime yang terhubung dengan Google Maps API. Untuk model terbaik dari YOLOv7 harus dikonversi yang bisa dikenali dalam backend pada pengembangan sistem. Format yang digunakan adalah *open neural network exchange* (ONNX) yang merupakan untuk merepresentasikan model pembelajaran mesin dan format file yang umum untuk memungkinkan menggunakan

model dengan berbagai kerangka kerja, alat, waktu proses, dan compiler. File onnx ini diletakkan di server dengan menggunakan FastAPI sebagai backend. Dalam mendapatkan titik tengah koordinat latitude dan longitude setiap object dari bounding box, dibutuhkan titik tengah awal sebagai titik koordinat latitude dan longitude kunci, sehingga bisa didapatkan titik tersebut yang di *request* ke Google Maps API dengan menggunakan bantuan library mapping dari leaflet js.

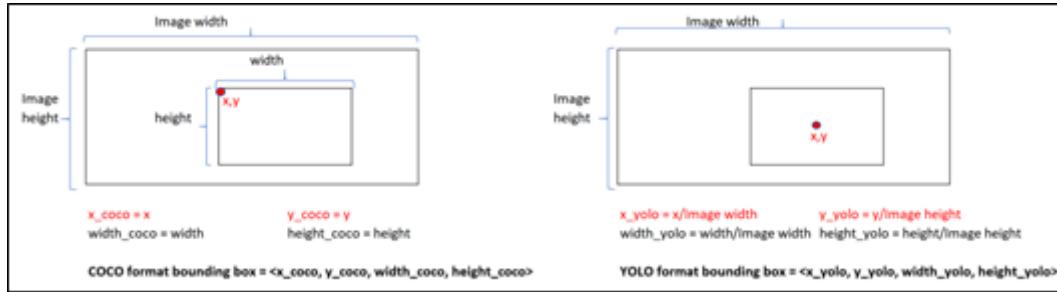
Adapun diagram alur hasil deteksi dan penghitungan akhir pada Gambar 3.15.



Gambar 3.15: Diagram Alur Hasil Deteksi dan Penghitungan Akhir

Hasil objek yang dideteksi yang ditandai dengan bounding box, masih dengan menggunakan format YOLO. *Bounding box* yang dihasilkan tersebut merepresentasikan posisi relative dari titik tengah *bounding box* (x,y) dan width, serta height dari objek yang dideteksi, sedangkan format COCO merepresentasikan piksel asli *bounding box* dari citra gambar tersebut. Terdapat perbedaan titik-titik bounding box antara format YOLO dengan COCO. Pada format COCO bounding box dengan format <top left x, top left y, width, height>, dimana lebar dan tinggi adalah dimensi dari *bounding box*. Perbedaan

format COCO dan YOLO seperti pada Gambar 3.16.



Gambar 3.16: *Bounding Box* pada COCO dan YOLO format

Berdasarkan Gambar 3.16, maka persamaan untuk mengubah format dari YOLO ke COCO, adalah:

$$x_{coco} = x_{yolo} * image\ width \quad (19)$$

$$y_{coco} = y_{yolo} * image\ height \quad (20)$$

$$w_{coco} = w_{yolo} * image\ width - x_{coco} \quad (21)$$

$$h_{coco} = h_{yolo} * image\ height - y_{coco} \quad (22)$$

$$x_{center} = x + \frac{w}{2} \quad (23)$$

$$y_{center} = y + \frac{h}{2} \quad (24)$$

Selanjutnya, setelah didapatkan titik dari format COCO, kemudian dilanjutkan dengan dikonversi untuk menghitung piksel koordinat dari latitude-longitude awal ke titik tengah ke piksel koordinat dari web mercator projection pada zoom 0 (base world map) sebesar ukuran 256 x 256 piksel. Persamaan berikut ini menempatkan sumbu x proyeksi pada khatulistiwa dan sumbu y pada bujur λ_0 , di mana λO adalah garis bujur, di mana y adalah proyeksi Mercator dari garis lintang dan φ adalah garis lintang dalam radian. λ adalah longitude dan ϕ adalah latitude.

$$x = R(\lambda - \lambda_0) y = R \ln[\tan(\frac{\pi}{4} + \frac{\phi}{2})] \quad (25)$$

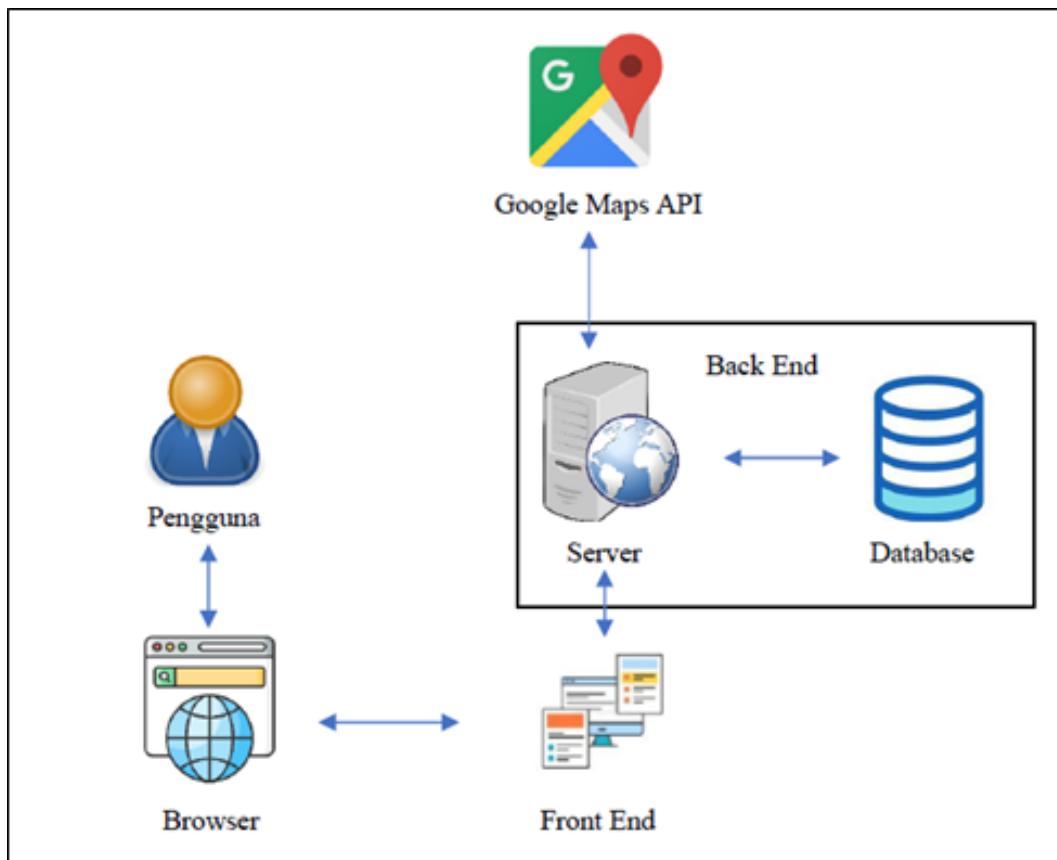
Berikutnya mengubah pixel coordinates menjadi latitude-longitude untuk setiap titik tengah dari bounding box tersebut dengan menggunakan invers transformation dari persamaan 26. Ekspresi pada kanan kedua equation

menggambarkan longitude, dan latitude pada sisi kiri, sedangkan nanti pada penulisan titik koordinat, dituliskan latitude-longitude.

$$\lambda = \lambda_0 + \frac{x}{R}, \phi = 2\tan^{-1}[\exp(yR) - \frac{1}{2}] \quad (26)$$

Dari persamaan 25 dan 26, maka dihasilkan titik koordinat latitude dan longitude dari setiap bounding box yang merepresentasikan keberadaan objek, dan untuk menghitung banyaknya pohon didasarkan pada banyaknya yang terdeteksi objek pohon kelapa sawit tersebut. Catatan bahwa yang digunakan untuk nilai π adalah 3.141592653589793.

Penerapan hasil deteksi dan penghitungan akhir, diterapkan pada diagram sistem arsitektur Gambar 3.17.



Gambar 3.17: Diagram Arsitektur Sistem

Gambar 3.17. merupakan diagram sistem arsitektur pada penerapan penelitian ini. Pengguna untuk dapat mengakses melalui browser untuk dapat

membuka laman penerapan sistem dari mendeteksi dan menghitung pohon kelapa sawit, serta mengetahui letak titik koordinat setiap pohon kelapa sawit tersebut. Pengguna dapat menggunakan perangkat lunak atau aplikasi yang dikenal dengan browser, seperti Google Chrome untuk dapat meminta laman yang dapat memberikan tampilan antarmuka.

Laman yang diakses atau diminta, memiliki tampilan antarmuka atau yang dikenal dengan *user interface*. Front end pada penerapan sistem ini menggunakan react js. React js digunakan untuk dapat membuat *user interface* pada tampilan berbasis web dan berisi library kode javascript yang memudahkan dalam web development. Tampilan antarmuka merupakan permintaan dan menampilkan ke pada server.

Server pada penelitian ini menggunakan FastAPI. FastAPI digunakan karena sebuah web framework untuk membangun sebuah *application programming interface* (API) untuk produksi atau model *machine learning*. Model CNN ini-lah yang diletakkan pada server untuk dapat mengenali objek yang akan diprediksi melalui gambar yang diminta melalui maps atau pemetaan satelit. Hasil prediksi merupakan data yang harus disimpan. Penyimpanan basis data menggunakan PostgreSQL. PostgreSQL basis data yang digunakan pada web app, aplikasi mobile, dan aplikasi analytics. Aplikasi yang membutuhkan pengolahan data yang lebih kompleks akan lebih cocok menggunakan PostgreSQL.

Tampilan Maps yang ditampilkan pada front end merupakan hasil dari permintaan yang terintegrasi terhubung dari Google Maps API. Dari Google Maps API maka dapat melihat sebuah peta datar (*mercator projection*) untuk dapat dipetakan suatu lahan pada area tertentu dan kemudian dideteksi citra kelapa sawit dan kemudian diketahui jumlah, serta letak posisi dari pohon kelapa sawit tersebut yang berguna untuk hasil pemantauan dari pohon kelapa sawit. Berikut ini komponen implementasi dalam prototype perangkat lunak

Tabel 3.5. dan perangkat keras Tabel 3.6. dalam menerapkan model terbaik di dalam prototype berbasis web.

Tabel 3.5: Komponen Implementasi Prototype Perangkat Lunak

No	Komponen	Keterangan
1	Browser	Google Chrome v. 113.0.5672.64
2	Google Maps API	Layanan Google Static Map API
3	Back End	
	a. Web Server FastAPI	v.0.85.0 (FastAPI adalah sebuah web framework untuk membangun sebuah API yang berbahasa Python).
	b. SQLAlchemy	v.1.4.41 (Library yang memfasilitasi komunikasi antara program Python dan database. Library ini digunakan sebagai alat Object Relational Mapper (ORM) yang menerjemahkan kelas-kelas Python ke tabel-tabel di database relasional dan secara otomatis mengubah panggilan fungsi menjadi pernyataan SQL).
	c. numpy	v.1.24.1 (NumPy (Numerical Python) adalah library Python yang fokus pada scientific computing. Library ini digunakan untuk melakukan perhitungan saintifik, seperti matriks, aljabar, dan statistik). Pada pengolahan gambar digunakan untuk manipulasi dan transformasi gambar, seperti memotong, merotasi, membalik dan mengubah ukuran gambar.

No	Komponen	Keterangan
4	d. onnxruntime	v.1.13.1. (Format model machine learning yang dapat dioperasikan pada berbagai platform dan dapat dijalankan pada berbagai platform seperti web)
	e. opencv-python-headless	v.4.6.0.66 (Library OpenCV untuk Python yang digunakan untuk pengolahan gambar, seperti pengolahan citra, deteksi objek, dan visi komputer)
	f. DBMS: PostgreSQL	v.14.7 (Sistem manajemen basis data yang dapat menyimpan data dalam bentuk array atau JSON dan berguna dalam Machine Learning)
4	Front End	
	a. react js	v.18.2.0 (Digunakan untuk dapat membuat user interface pada tampilan berbasis web dan berisi library kode javascript)
	b. tailwindcss	v.3.2.4 (Framework CSS yang digunakan untuk mempercepat pengembangan tampilan (UI) web)
	c. typescript	v.4.9.3 (Bahasa pemrograman yang digunakan untuk mengembangkan aplikasi Machine Learning berbasis web dengan menggunakan framework react)
	d. leaflet js	v.1.9.3 (library JavaScript yang digunakan untuk memvisualisasikan peta interaktif pada halaman web dan dapat digunakan dalam konteks Machine Learning untuk memvisualisasikan data geospasial dan membuat aplikasi peta interaktif)

Tabel 3.6: Komponen Implementasi Prototype Perangkat Keras

No	Layanan	Keterangan
1	Back End Layanan Google Compute Engine	Tipe Mesin: n2-standard-2 vCPUs: 2 Core (@ 4GB) CPU Platform: Intel Cascade Lake RAM: 8 GB Architecture: x86/64
2	Front End Layanan Google Compute Engine	Tipe Mesin: e2-medium vCPU: 2 Core (@ 4GB) CPU Platform: Intel Broadwell RAM: 4 GB Architecture: x86/64

BAB IV

HASIL DAN PEMBAHASAN

4.1 Area Studi

Terdapat dua area studi yang dilakukan pada penelitian ini, yaitu Kebun Pendidikan dan Penelitian Kelapa Sawit milik Institut Pertanian Bogor-Cargil yang berlokasi di Jonggol, Jawa Barat dan Universitas Gunadarma di Penajam Paser Utara, Kalimantan Timur yang memiliki lahan yang ditanami pohon kelapa sawit.

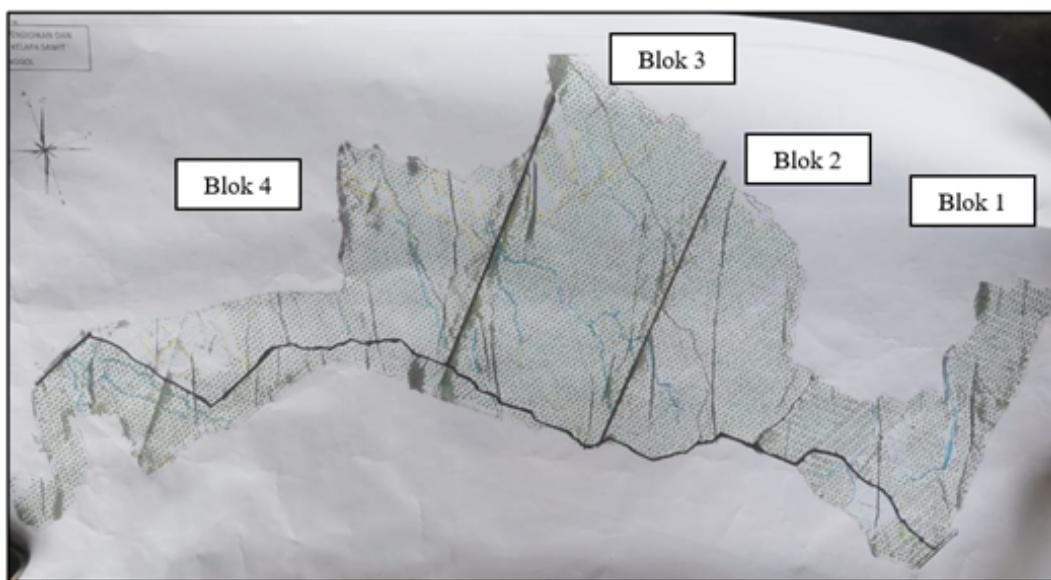
Area studi ini secara umum mencakup untuk pengambilan data citra pohon kelapa sawit sebagai dataset primer. Tabel 4.1 menampilkan lokasi koordinat dan luasan lahan yang diambil sebagai citra pohon kelapa sawit.

Tabel 4.1: Data Lokasi dan Luasan Area Studi

Area Studi	Lokasi Koordinat (Latitude, Longitude)	Luasan Lahan (ha)
Kebun Pendidikan dan Penelitian Kelapa Sawit milik Institut Pertanian Bogor-Cargil	-6.4277942, 106.8418378	63,48
Universitas Gunadarma di Penajam Paser Utara, Kalimantan Timur	-1.318495, 116.6678405	≈ 18

Pengambilan citra pada area studi diambil dengan menggunakan DJI Mavic 2 Pro dari ketinggian 100 m di atas permukaan tanah dengan pakar atau pilot drone yang sudah tersertifikasi. Pada tahap pengujian pada sistem untuk mendeteksi dan menghitung, serta titik koordinat latitude-longitude dari

setiap pohon kelapa sawit menggunakan area Kebun Pendidikan dan Penelitian Kelapa Sawit milik Institut Pertanian Bogor-Cargil Blok 1, 2, 3, dan 4. Pada proses pengambilan citra untuk dataset primer pada area studi lahan pada Kebun Pendidikan dan Penelitian Kelapa Sawit milik Institut Pertanian Bogor-Cargil berhasil ditangkap area blok 3 dan 4 dari total 4 blok. Peta sebaran area inisialisasi lahan pada kebun ini seperti pada Gambar 4.1 yang memiliki 5670 pohon tanam kelapa sawit berdasarkan hasil wawancara di lapangan pada area studi Kebun Pendidikan dan Penelitian Kelapa Sawit milik Institut Pertanian Bogor-Cargil.



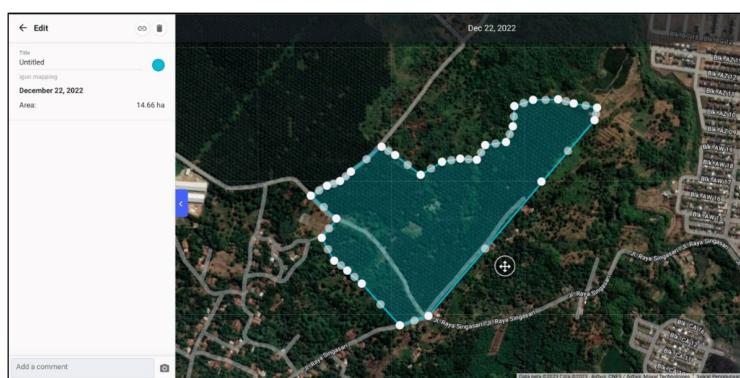
Gambar 4.1: Peta Sebaran Area Inisialisasi Lahan Kebun Pendidikan dan Penelitian Kelapa Sawit milik Institut Pertanian Bogor-Cargil

Area lahan yang berhasil ditangkap pada kebun ini berada pada area studi blok 3 dan blok 4. Area studi yang digunakan untuk dataset primer yang digunakan blok 3 pada Gambar 3.6 dan blok 4 pada Gambar 3.7. dengan total luas sebesar 37,51 hektar, dan pengujian digunakan blok 1 (Gambar 4.2), 2 (Gambar 4.3), 3, dan 4 seperti pada Tabel 4.2 dengan total luas area 63,48 hektar pada Kebun Pendidikan dan Penelitian Kelapa Sawit Institut Pertanian Bogor-Cargil dengan perhitungan luas area menggunakan bantuan layanan

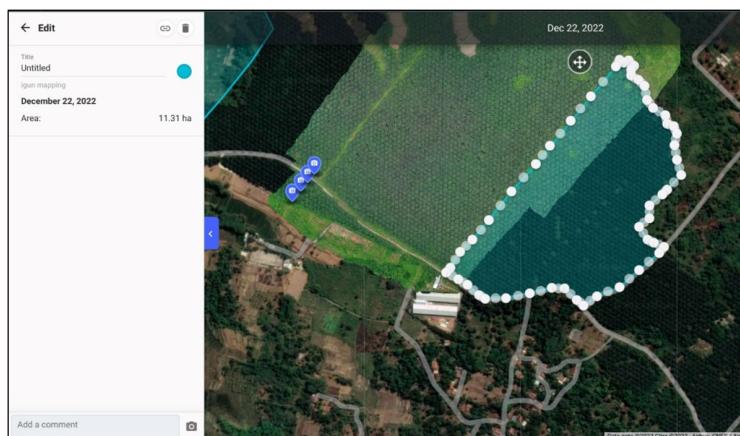
DroneDeploy.

Tabel 4.2: Area Studi yang Berhasil Ditangkap menjadi Citra Area Pohon Kelapa Sawit dengan Drone

No.	Area	Luas Area (ha)
1	Blok 1	14,66
2	Blok 2	11,31
3	Blok 3	18,20
4	Blok 4	19,31
	Total	63,48



Gambar 4.2: Luas Area Blok 1 Kebun Pendidikan dan Penelitian Kelapa Sawit Institut Pertanian Bogor-Cargil



Gambar 4.3: Luas Area Blok 2 Kebun Pendidikan dan Penelitian Kelapa Sawit Institut Pertanian Bogor-Cargil

4.2 Persiapan Data

Persiapan data, yang dikenal dengan *data preprocessing* adalah tahap yang penting untuk menyiapkan data untuk dapat digunakan dalam modeling dengan *deep learning*. Persiapan data meliputi persiapan dataset primer dan sekunder untuk dapat digunakan.

4.2.1 Deskripsi Dataset dan Pemilihan

Akuisisi citra area pohon kelapa sawit menjadi penting karena berhubungan hasil memindai, menangkap (*capture*), atau menjadi suatu foto citra yang dapat digunakan untuk dataset. Citra area pohon kelapa sawit pada penelitian ini dibagi menjadi dua, yaitu citra untuk dataset primer dan sekunder.

4.2.1.1 Dataset Primer

Dataset primer digunakan untuk membangun atau memberikan anotasi atau label oil palm pada citra gambar. Hasil gambar citra pohon kelapa sawit dari dua area studi memiliki format citra yang sama, yaitu *.jpg. Data citra dari Kebun Pendidikan dan Penelitian Kelapa Sawit milik Institut Pertanian Bogor-Cargil yang diambil dari blok 3 dan 4 terdiri dari 69 citra atau gambar, sedangkan jumlah citra yang dihasilkan dari area kelapa sawit berjumlah 205 citra dengan dimensi ukuran citra sebesar 5472 x 3078 px, seperti pada Tabel 4.3

Total citra dataset primer dari dua area studi seperti pada Tabel 4.3 berikut.

Tabel 4.3: Jumlah Citra dari Dua Area Studi

Area Studi	Jumlah Citra	Luas Area (ha)
Kebun Pendidikan dan Penelitian Kelapa Sawit milik Institut Pertanian Bogor-Cargil	69	14,66
Universitas Gunadarma di Penajam Paser Utara, Kalimantan Timur	205	11,31

Hasil citra dari area studi Universitas Gunadarma dengan jumlah 205 citra atau gambar ini digunakan untuk dataset primer dengan memberikan anotasi atau pemberian label secara otomatis yang dilakukan dalam penelitian ini. Adapun citra sampel area studi dari Universitas Gunadarma tampak pada Gambar 4.4 dan citra sampel Kebun Pendidikan dan Penelitian Kelapa Sawit milik Institut Pertanian Bogor-Cargil tampak pada Gambar 4.5



Gambar 4.4: Hasil Citra Sampel Universitas Gunadarma di Penajam Paser Utara, Kalimantan Timur



Gambar 4.5: Hasil Citra Sampel Universitas Gunadarma di Penajam Paser Utara, Kalimantan Timur

4.2.1.2 Dataset Sekunder

Data sekunder yang digunakan sudah memiliki label atau anotasi sebagai pohon kelapa sawit "oil palm". Data berjumlah 1795 citra yang sudah memiliki kotak batas atau *bounding box* dengan kelas *oil palm*. Data sekunder digunakan untuk menambah data pada sata proses pelatihan, validasi dan

pengujian. Data tersebut semakin bervariasi, maka keterwakilan data pada citra yang berupa objek pohon kelapa sawit semakin baik. Dataset sekunder dapat diakses secara daring dan dapat digunakan secara *free* atau *open source* melalui sistem berbasis web bernama roboflow.

GAMBAR

Data sample pada dataset sekunder yang sudah memiliki *bounding box* yang diberi label oil palm, seperti pada Gambar 4.6.

Dataset sekunder ini dilakukan proses augmentasi dengan menggunakan bantuan layanan roboflow. Proses augmentasi ini dilakukan untuk menambah banyaknya data citra, serta bertujuan agar mesin dapat belajar dan mengenali dari berbagai citra yang berbeda-beda. Penggunaan augmentasi ini diharapkan dapat meningkatkan performa dari model, karena mesin yang digunakan dalam penelitian ini agar dapat berhasil mengenali lebih banyak objek dari bentuk dan pola yang beragam jenis data citra.

Data sekunder ini berada pada layanan roboflow dan memiliki fasilitas layanan untuk augmentasi. Proses augmentasi dilakukan dengan menambahkan data citra dari data sekunder, seperti pada Gambar 4.7.

AUGMENTATIONS	Outputs per training example: 2
	Crop: 0% Minimum Zoom, 40% Maximum Zoom
	Rotation: Between -39° and +39°
	Hue: Between -30° and +30°
	Exposure: Between -17% and +17%
	Blur: Up to 2.5px
	Noise: Up to 2% of pixels
	Mosaic: Applied

Gambar 4.6: Proses Augmentasi Data Sekunder

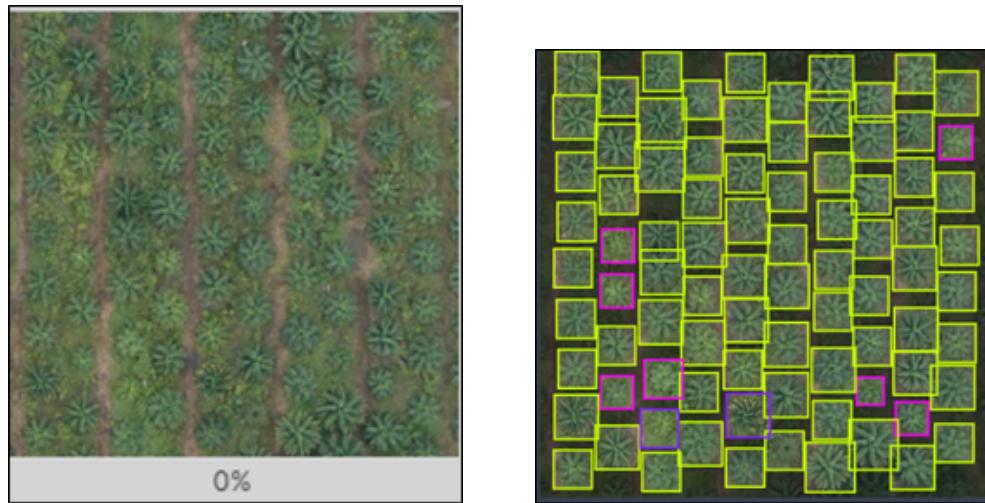
Fitur layanan augmentasi yang digunakan antara lain *crop*, *rotation*,

hue, exposure, blur, noise, dan mosaic. Fitur layanan ini memiliki kegunaan yang mewakili kondisi seperti data studi atau lapangan. Kegunaan fitur layanan yang digunakan seperti pada Tabel 4.4.

Tabel 4.4: Kegunaan Layanan Augmentasi Dataset Sekunder

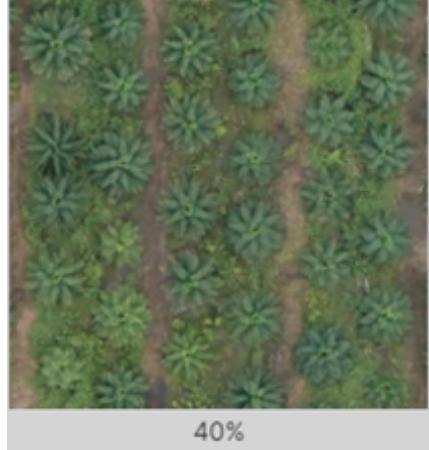
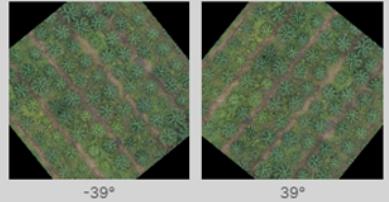
Augmentasi	Kegunaan
Crop	Menambahkan variabilitas pada pemosisan dan ukuran untuk membantu model lebih mengenali terhadap translasi subjek dan posisi kamera.
Rotation	Menambahkan variabilitas pada rotasi untuk membantu model agar dapat mendeteksi objek, bahkan saat kamera atau subjek tidak sejajar secara sempurna.
Hue	Mengubah warna secara acak untuk membuat model tidak terlalu sensitif.
Exposure	Menambahkan variabilitas pada kecerahan gambar untuk membantu model lebih dapat mengenali terhadap perubahan pencahayaan dan pengaturan kamera.
Blur	Menambahkan keburaman Gaussian secara acak untuk membantu model lebih mengenali terhadap fokus kamera, jika data citra berada pada area hutan, kebun, dan alam liar, mungkin tidak berada dalam keadaan focus tangkapan citra tersebut.
Noise	Menambahkan noise untuk membantu model lebih mengenali terhadap noise yang dapat membantu mempertahankan nilai dan menghindari overfitting dari hasil pengujian.
Mosaic	Menambahkan mosaik untuk membantu model tampil lebih baik pada objek kecil, mosaik menggabungkan beberapa foto dari rangkaian pelatihan data.

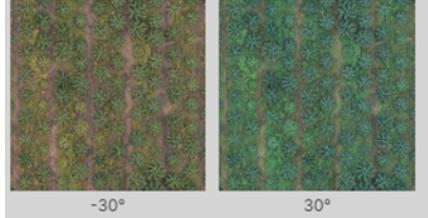
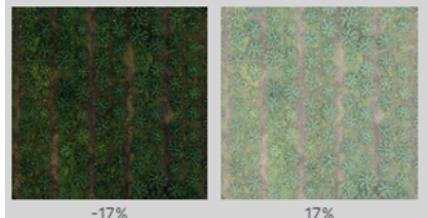
Tabel 4.5 . menampilkan hasil data citra untuk dataset yang telah dilakukan proses augmentasi sesuai dengan Tabel 4.4. dan tampilan data citra yang berbeda dengan data citra dataset sekunder asli seperti pada Gambar 4.7.

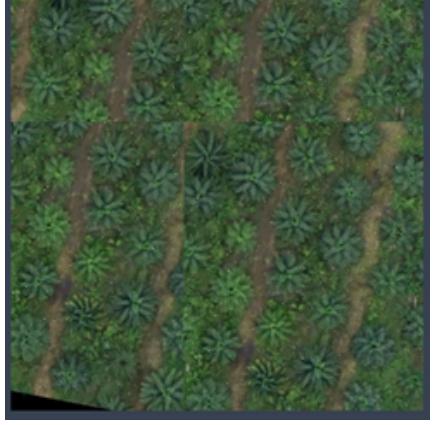


Gambar 4.7: Data Citra Asli Data Sekunder (kiri: original; kanan: tampak dengan *bounding box*)

Tabel 4.5: Hasil Proses Augmentasi Citra Dataset Sekunder

Augmentasi	Deskripsi	Citra Hasil Augmentasi
Crop	0% Minimum Zoom, 40% Maximum Zoom	 40%
Rotation	Between -39° and +39°	

Augmentasi	Deskripsi	Citra Hasil Augmentasi
Hue	0% Between -30° and $+30^\circ$	
Exposure	Between -17° and $+17^\circ$	
Blur	Up to 2.5px	
Noise	Up to 2% of pixels	

Augmentasi	Deskripsi	Citra Hasil Augmentasi
Mosaic	Applied	

Proses augmentasi dengan menambahkan 3987 citra tersimpan pada layanan roboflow yang digunakan citra untuk komputasi, yang terbagi untuk data pelatihan, validasi, dan pengujian dengan model CNN. Selanjutnya, untuk pembentukan dataset otomatis dengan memberikan anotasi atau pemberian label kelas secara otomatis pada objek di dalam citra dataset primer. Hal ini merupakan salah satu kebaruan pada penelitian ini dan proses tersebut dijelaskan pada sub bab 4.2.3.

4.2.2 Membersihkan Data

Proses pada tahap membersihkan data dilakukan dengan memvalidasi data citra yang ditangkap pada dataset primer yang tidak dapat digunakan sebagai dataset. Proses ini dilakukan dengan validasi satu persatu dataset citra yang ditangkap oleh drone. Citra yang tidak digunakan merupakan citra yang ditangkap oleh drone tidak berada posisi yang tegak dari atas, tidak diambil dari sisi samping dari kamera drone dan tangkapan citra yang tidak terdapat pohon kelapa sawit. Terdapat 26 citra yang tidak dapat digunakan pada dataset primer sebagai citra untuk proses membangun data atau melakukan proses anotasi otomatis, seperti pada Gambar 4.8.



Gambar 4.8: Data Citra Primer yang tidak Digunakan (kiri: tidak terdapat pohon kelapa sawit; kanan: tampak area yang tidak tangkapan dari atas)

4.2.3 Membangun Data

Proses anotasi otomatis merupakan pembentukan dataset berbasis OBIA. Proses anotasi ini digunakan dari data citra primer area studi Universitas Gunadarma di PPU, Kalimantan yaitu sebanyak 205 citra. Hasil dari 205 citra yang telah berhasil dideteksi sebagai objek pohon kelapa sawit ditampilkan dengan adanya *bounding box* yang digunakan sebagai dataset untuk pelatihan dan pengujian model. Keluaran atau *output* dari proses pembentukan dataset ini adalah citra yang dapat dideteksi objek di dalamnya sebagai pohon kelapa sawit yang dinyatakan dengan adanya *bounding box* pada setiap objek yang terdeteksi yang memiliki data kelas, koordinat citra (x, y), dan *width*, *height* yang tersimpan dalam file dengan ekstensi *.txt.

Berdasarkan metode yang digunakan pada pembahasan bab 3 subbab 3.3.1, hal yang dilakukan pertama adalah menyiapkan pengumpulan data dalam satu folder yang sama. Data citra yang digunakan berupa 205 citra yang diambil oleh drone dari ketinggian 100 m dari permukaan tanah. Dalam penelitian ini, digunakan *tools* atau peralatan berupa aplikasi *jupyter notebook* untuk menuliskan kode program dan berbagi file dan berjalan di local komputer yang dijalankan pada web browser. Perintah yang digunakan dengan

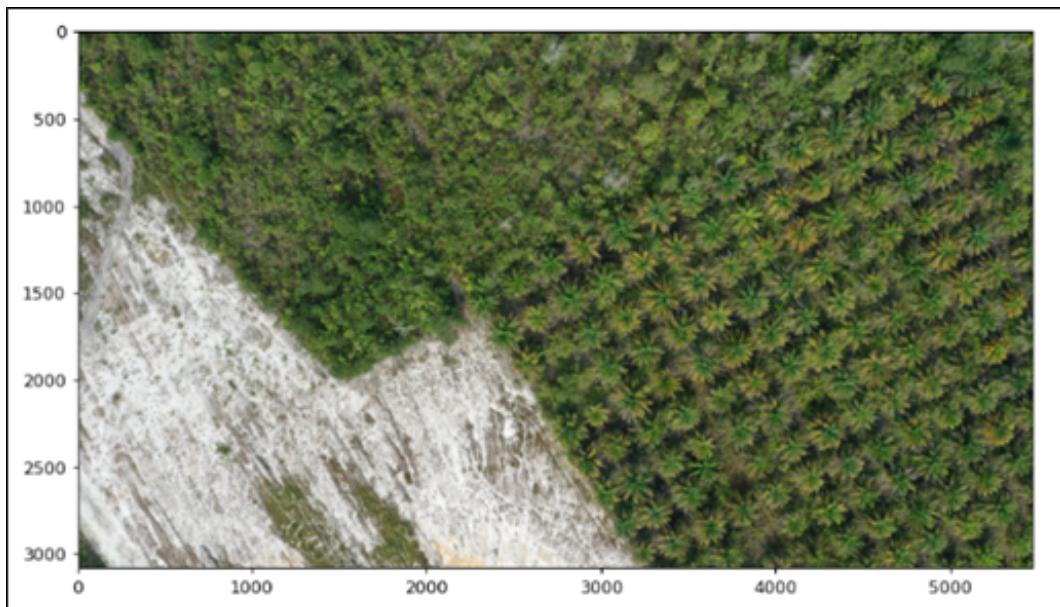
menggunakan *command prompt* untuk mengaktifkan *jupyter notebook*. Setelah berhasil aktif, maka diarahkan ke halaman homepage browser, dan dapat digunakan sebagai *tools* untuk proses anotasi otomatis, seperti pada Gambar 4.9.



Gambar 4.9: *Home Page* Jupyter Notebook

4.2.4 Membuat Template

Tahap berikutnya dalam penelitian ini adalah membuat beberapa contoh gambar atau citra yang dijadikan sebagai template. Data citra yang digunakan berdimensi 5472 x 3078, maka pada jupyter notebook ditampilkan terlebih dahulu data citra untuk dapat ditampilkan citra yang digunakan untuk membuat template awal dengan sebuah fungsi. Data citra yang digunakan sebagai template seperti pada Gambar 4.10.



Gambar 4.10: Data citra untuk membuat template (5472 x 3078)

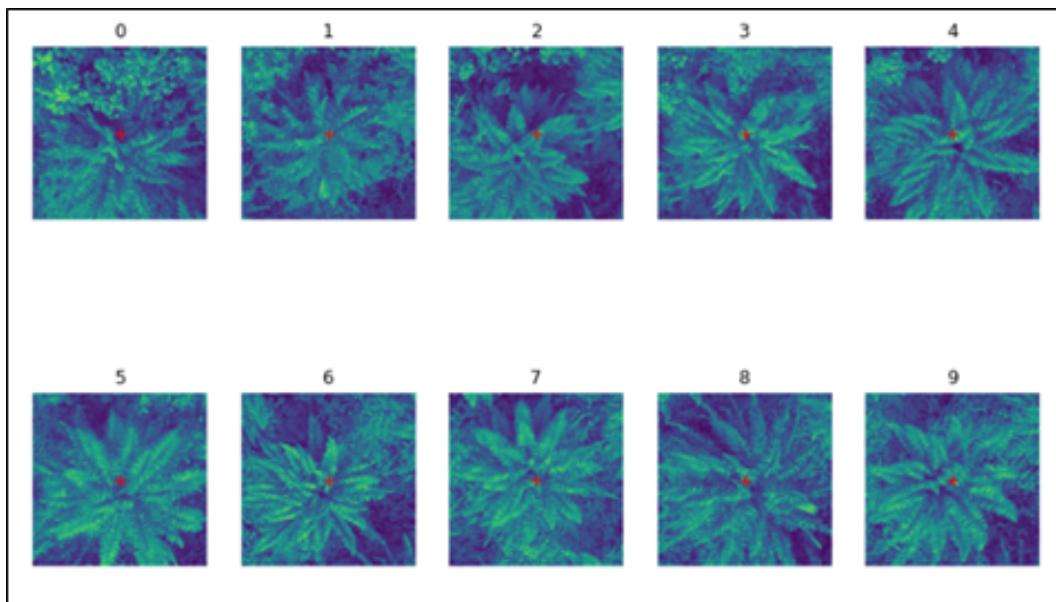
Dalam memudahkan pembuatan template gambar (objek kelapa sawit) digunakan fungsi *onClick* untuk dapat menentukan beberapa titik tengah dari lokasi objek pohon kelapa sawit pada data citra. Dalam penelitian ini, digunakan 10 titik sebagai template objek. Titik yang dipilih merupakan objek citra pohon kelapa sawit yang ditandai dengan titik tengah merah yang berukuran 12×12 berdasarkan radius dari titik tengah yang telah ditentukan, seperti pada Gambar 4.11.



Gambar 4.11: Membuat Template yang Ditandai dengan Titik Objek Pohon Kelapa Sawit

Data citra berhasil ditampilkan dan ditandai dengan titik tengah merah untuk dijadikan sebagai template. Dalam pembuatan template ini, objek pohon kelapa sawit yang telah diberi titik dengan berwarna merah menjadi citra yang berukuran atau berdimensi 12×12 dari gambar asli. Citra tersebut ditampilkan dengan memiliki id dari 0 hingga 9 karena menggunakan array yang dimulai dari 0. Hal ini digunakan untuk dapat memudahkan penyesuaian titik tengah, jika terdapat titik yang tidak berada ditengah template yang diproses pada subbab 4.2.3.2. Gambar 4.12 hasil citra yang dijadikan sebagai

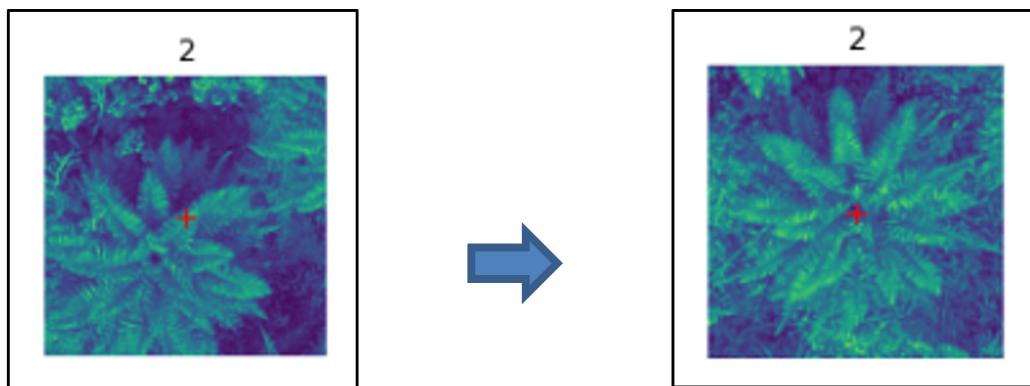
template.



Gambar 4.12: Citra Template dari Objek Pohon Kelapa Sawit

4.2.4.1 Menyesuaikan Template

Pada tahap ini, menyesuaikan template yang sudah tampak seperti pada gambar 4.10. Penyesuaian template adalah menyesuaikan titik tengah pada citra objek pohon kelapa sawit, jika titik tengah tidak berada pada tengah objek di citra tersebut. Penyesuaian ini dapat dilakukan dengan fungsi yang diterapkan, dan titik dapat dikoreksi dengan bergeser ke atas, bawah, kiri atau kanan dengan sebuah fungsi. Misalnya, pada id 2, titik tengah dari citra template tersebut yang ditandai dengan titik merah tidak berada di tengah, maka dapat digunakan fungsi turun (ke bawa) agar titik tersebut berada di tengah pohon kelapa sawit. Dalam penelitian ini dilakukan dengan 2 piksel setiap satu fungsi yang dijalankan untuk menggeser titik ke atas, bawah, kiri, atau kanan. Hasil koreksi dari menyesuaikan template untuk ID 2 ditampilkan pada Gambar 4.13.



Gambar 4.13: Menyesuaikan Template pada ID 2

4.2.4.2 Evaluasi Korelasi diantara Template

Ketika proses penyesuaian template selesai, maka pada tahap berikutnya adalah mengevaluasi korelasi antar template yang telah dibuat. Masing-masing template awal dibuat sebanyak 10 citra yang menunjukkan objek pohon kelapa sawit dengan dimensi 12×12 . Dimensi 12×12 digunakan agar citra objek pohon kelapa sawit dapat terlihat lebih jelas yang menadakan 1 (satu) objek pohon kelapa sawit yang dipilih sebagai template. Berdasarkan perhitungan yang dilakukan pada persamaan 6 - 9 pada sub bab 3.3.3.3 didapatkan hasil korelasi diantara template dan waktu yang dibutuhkan untuk mengkalkulasi korelasi template tersebut pada Tabel 4.6.

Tabel 4.6: Hasil Rerata Korelasi dan Waktu yang Dibutuhkan

Deskripsi	Nilai
Evaluasi korlasi diantara template	0.16
Waktu yang dibuthkan untuk menghitung (d)	57

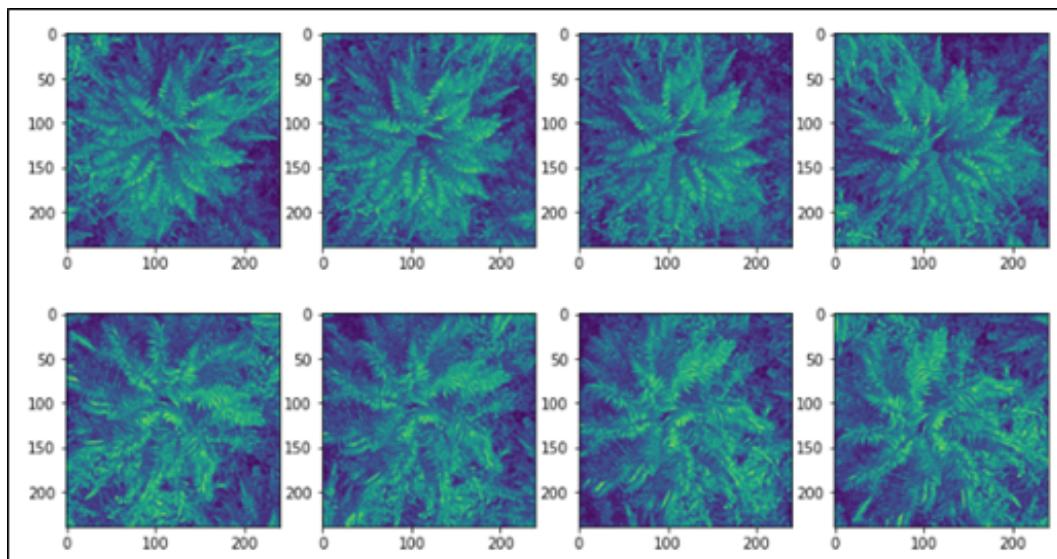
Berdasarkan Tabel 4.6. terlihat bahwa waktu yang dibutuhkan untuk menghitung nilai rata-rata korelasi antar template membutuhkan waktu sebanyak 57 detik, dan nilai korelasi citra dihasilkan sebesar 0.16. Hal ini menunjukkan nilai korelasi citra baik, karena nilai korelasi berada di rentang -1

sampai $+ 1$. Jika nilai mendekati 1 , maka nilai ini sebagai ambang batas untuk digunakan dalam metode atau algoritma *template matching* untuk mendeteksi keberadaan objek kelapa sawit pada citra yang dapat diberikan anotasi atau label pohon kelapa sawit untuk dijadikan dataset.

4.2.4.3 Penambahan Template

Nilai korelasi antar template sudah diketahui, Langkah selanjutnya dilakukan menambahkan template dari 10 citra yang sudah dimiliki sebagai template. Hal ini digunakan untuk menambahkan data citra yang diproses pada *template matching*, sehingga metode dapat digunakan dengan mengenali berbagai data citra sebagai template bahwa citra tersebut adalah objek pohon kelapa sawit.

Pada penambahan template ini digunakan rotasi sebesar 30° (tiga puluh derajat), yang dimana setiap 1 gambar sebanyak 4 (empat kali) dirotasi sebesar 30° . Hasil citra template yang sudah di rotasi seperti pada Gambar 4.14.



Gambar 4.14: Hasil Rotasi 30° pada Citra template

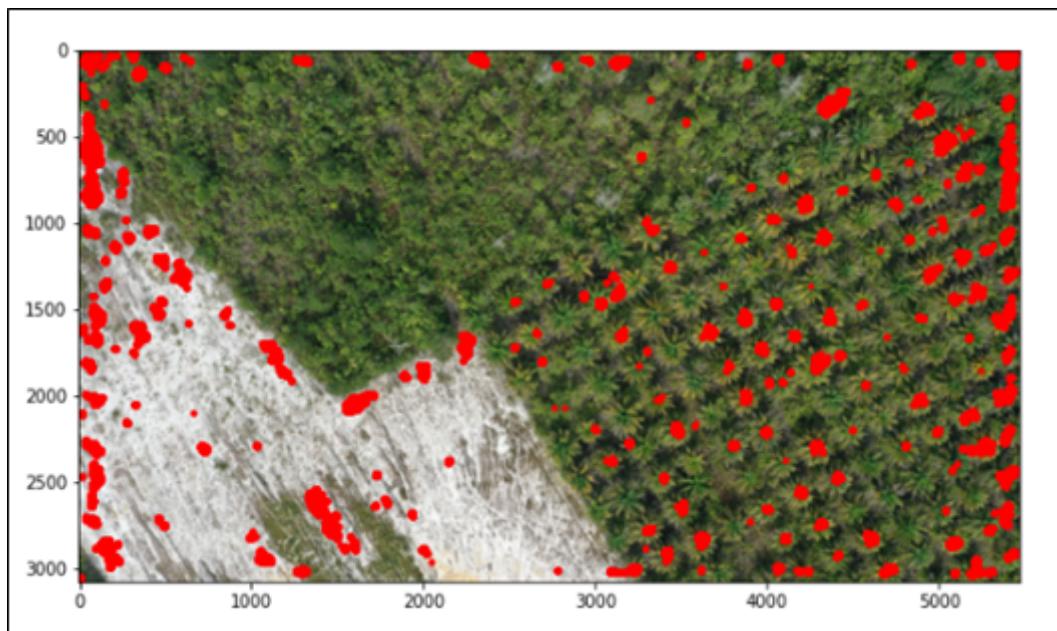
Proses untuk mendapatkan citra yang di rotasi untuk 10 (sepuluh) citra template tampak seperti pada Tabel 4.7.

Tabel 4.7: Waktu yang Dibutuhkan untuk Rotasi

Deskripsi	Nilai
Waktu yang dibutuhkan untuk rotasi (detik)	8

4.2.4.4 Menjalankan Template Matching

Pada tahap ini dilakukan match template yaitu anotasi secara otomatis dengan mendekripsi (klasifikasi) mengenali adanya objek pohon kelapa sawit pada citra. Citra utuh yang besar (gambar asli) didekripsi menjadi bagian-bagian yang terdeteksi menjadi citra yang dikenali sebagai objek yang menjadi dataset. Gambar yang terdeteksi dengan *template matching* ini didekripsi dengan lingkaran merah pada Gambar 4.15.



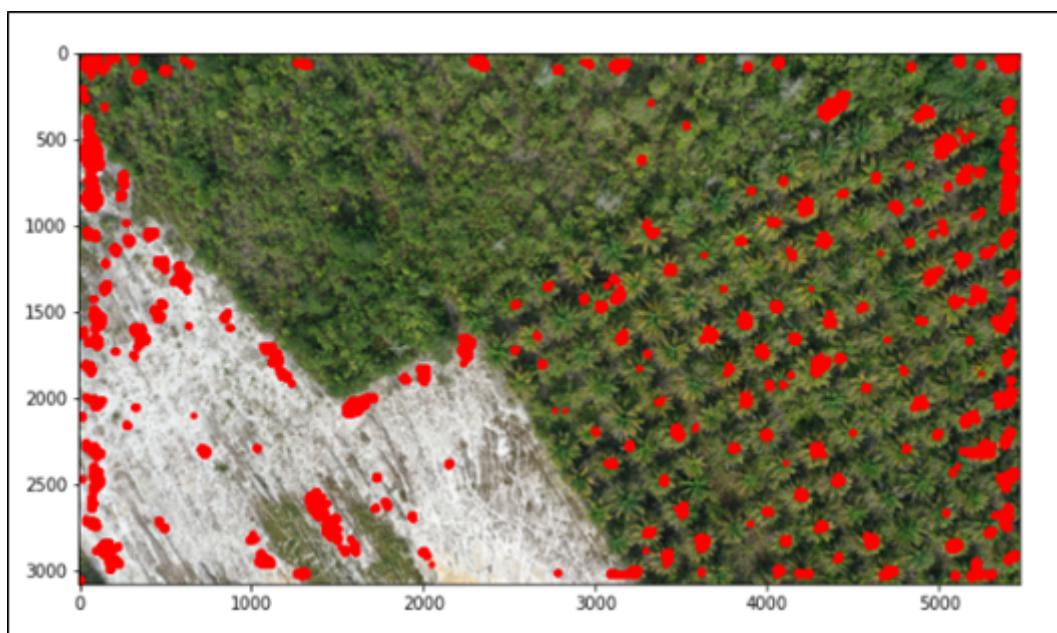
Gambar 4.15: Objek yang terdeteksi untuk Dataset dengan *Template Matching*

Gambar 4.15. yang terdeteksi terlihat seperti adanya tumpang tindih atau mungkin tidak seperti pohon kelapa sawit yang merupakan dataset yang diharapkan, karena nilai ambang batas (*threshold*) dibatasi pada nilai rata-rata nilai korelasi diantara template yang lebih rendah, sehingga ketika berada di

atas nilai korelasi, maka terdeteksi sebagai citra yang cocok dengan template.

4.2.4.5 Analisis Cluster dengan BIRCH

Tahap ini dilakukan setelah berhasil dideteksi dengan menggunakan *template matching*, selanjutnya dilakukan analisis cluster dengan algoritma BIRCH untuk mereduksi atau mencari nilai ambang batas dari citra yang tidak sesuai berdasarkan nilai mean r (korelasi). Pada BIRCH ini digunakan nilai ambang batas (*threshold*) secara acak yang digunakan sebagai nilai ambang batas minimum yang dimasukkan ke dalam citra untuk dapat mendeteksi objek yang terdeteksi pohon kelapa sawit pada citra yang sudah dideteksi dengan *template matching*. Nilai ambang batas (*threshold*) yang digunakan dalam penelitian ini adalah 0.5. Hasil objek yang terdeteksi sebagai pohon kelapa sawit pada citra untuk menjadi dataset ditampilkan dengan kotak batas (*bounding box*) dengan kotak berwarna merah. Hal ini menunjukkan bahwa kotak tersebut mendeteksi objek pohon kelapa sawit yang tampak seperti pada Gambar 4.16.



Gambar 4.16: Hasil objek yang terdeteksi untuk Dataset dengan algoritma BIRCH

Berdasarkan Gambar 4.15 diketahui template matching dari citra berhasil mendeteksi objek untuk dataset, namun masih terjadi tumpang tindih atau overlapping dan berdasarkan hasil dari algoritma BIRCH pada Gambar 4.16., dihasilkan pengurangan nilai dan terdeteksi objek pohon kelapa sawit pada citra. Hasil waktu yang digunakan untuk mendeteksi objek sebagai dataset ditunjukkan pada Tabel 4.8.

Tabel 4.8: Hasil Waktu yang Dibutuhkan untuk Deteksi Objek sebagai Dataset

Deskripsi	Waktu (detik)	Jumlah yang Terdeteksi
Waktu yang dibutuhkan pada Template Matching	334	463
Waktu yang dibutuhkan pada BIRCH	24	148

Berdasarkan hasil Tabel 4.8. diketahui bahwa *template matching* membutuhkan waktu 334 detik untuk berhasil mendeteksi 463 objek, sedangkan dengan BIRCH membutuhkan waktu 24 detik dan berhasil direduksi untuk menemukan nilai ambang batas yang baik menjadi 148 yang terdeteksi sesuai dengan nilai yang telah ditentukan.

4.2.4.6 Menjalankan Auto-Annotate Datasets

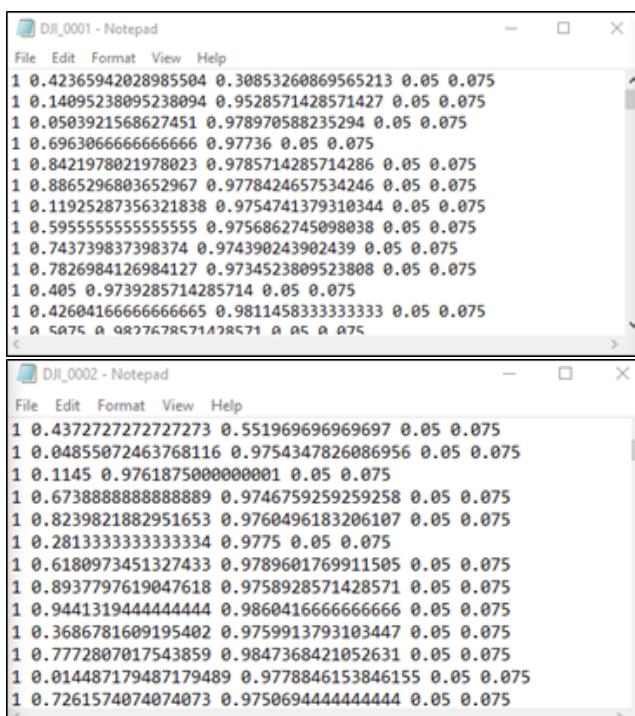
Proses dengan *template matching* dan BIRCH berhasil dideteksi objek pada citra yang ditampilkan dengan *bounding box*. Tahap selanjutnya menjalankan anotasi otomatis. Proses ini mengambil data citra sebanyak 205 citra dari data primer yang berada dalam 1 folder, kemudian dijalankan proses tersebut. Proses ini mendeteksi objek yang sesuai sebagai pohon kelapa sawit pada citra, sehingga dapat dikenali dan diperoleh kelas, koordinat x, koordinat y, *width*, *height* disimpan dalam file berekstensi *.txt (sesuai nama file citra). File merupakan hasil dari proses anotasi dataset yang berisi kelas, koordinat x, koordinat y, *width*, *height*.

Berdasarkan hasil anotasi otomatis membutuhkan waktu yang signifikan untuk membuat dataset dengan anotasi otomatis dari citra yang berjumlah 205. Waktu yang dibutuhkan proses ini seperti terlihat pada Tabel 4.9.

Tabel 4.9: Hasil Waktu Yang Dibutuhkan untuk Anotasi Otomatis

Deskripsi	Waktu (mm:dd)
Waktu yang dibutuhkan untuk Anotasi Otomatis Dataset	30:24

Berikut ini adalah contoh hasil anotasi otomatis pada file berekstensi *.txt yang berisi kelas, koordinat x, koordinat y, width, dan height terlihat pada Gambar 4.18.



The image shows two windows of the Windows Notepad application. Both windows have the title 'DJI_0001 - Notepad' and 'DJI_0002 - Notepad'. The content of both windows is identical, displaying a series of numerical values representing automatically annotated data. The data is organized into five columns separated by spaces. The first column contains the value '1' repeated five times. The second column contains a sequence of coordinates and identifiers. The third, fourth, and fifth columns contain numerical values representing width, height, and possibly confidence or scale factors. The data is as follows:

```

1 0.42365942028985504 0.30853260869565213 0.05 0.075
1 0.14095238095238094 0.9528571428571427 0.05 0.075
1 0.0503921568627451 0.978970588235294 0.05 0.075
1 0.6963066666666666 0.97736 0.05 0.075
1 0.8421978021978023 0.9785714285714286 0.05 0.075
1 0.8865296803652967 0.9778424657534246 0.05 0.075
1 0.11925287356321838 0.9754741379310344 0.05 0.075
1 0.5955555555555555 0.9756862745098038 0.05 0.075
1 0.743739837398374 0.974390243902439 0.05 0.075
1 0.7826984126984127 0.9734523809523808 0.05 0.075
1 0.405 0.9739285714285714 0.05 0.075
1 0.42604166666666665 0.9811458333333333 0.05 0.075
1 0 0 0 0 0

```

Gambar 4.17: Hasil Anotasi Otomatis dalam format file *.txt

Pada Gambar 4.17. terdapat 5 bilangan berurutan, nilai ke-1 menunjukkan bilangan yang sama, untuk digunakan sebagai kelas yang terdeteksi yaitu kelas 'oil palm', untuk barisan ke-2, ke-3, ke-4, dan ke-5 masing-masing menunjukkan koordinat x, koordinat y, lebar dan tinggi. Lebar dan tinggi da-

pat terlihat sama, karena dataset memiliki karakteristik dimensi ukuran citra yang sama.

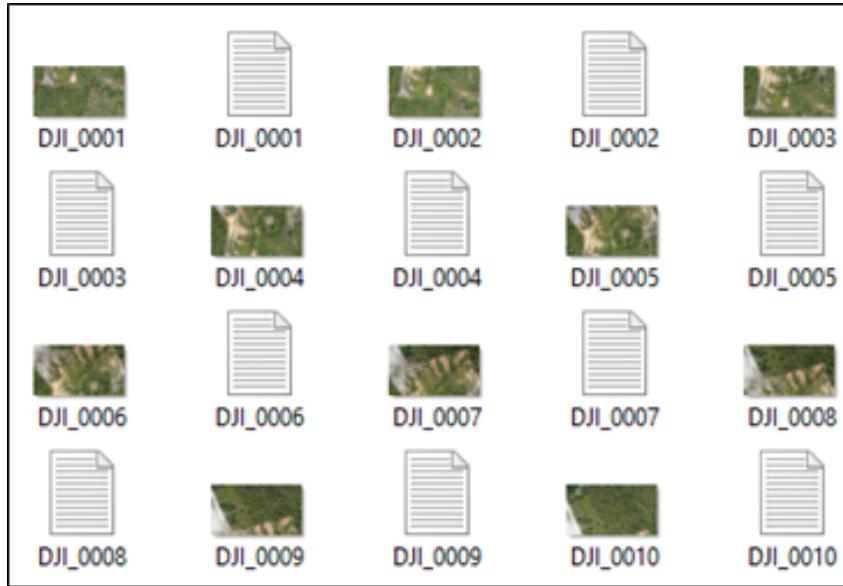
Setelah itu, pada penelitian ini dilakukan pengujian dengan 10 citra yang dapat dilihat pada Tabel 4.10.

Tabel 4.10: Pengujian dengan 10 Citra Dataset Primer

No	ID_Image	Jumlah objek Pohon yang Asli	Jumlah objek Pohon yang terdeteksi	Hasil % Terdeteksi
1	ID 1	253	233	92,1%
2	ID 2	273	250	91,6%
3	ID 3	157	142	90,4%
4	ID 4	167	148	88,6%
5	ID 5	113	101	89,4%
6	ID 6	110	97	88,2%
7	ID 7	84	75	89,3%
8	ID 8	138	121	87,7%
9	ID 9	160	144	90,0%
10	ID 10	210	186	88,6%
Rata-Rata				89,9%

Berdasarkan hasil pengujian terhadap 10 citra dari hasil pada Tabel 4.10 didapatkan rata-rata akurasi yang terdeteksi pada penelitian ini sebesar 89,90% dengan menggunakan metode *template matching* dan algoritma BIRCH. Hal ini menunjukkan hal penting yang signifikan yang harus dilakukan dalam proses anotasi otomatis yang dapat digunakan untuk memberikan deteksi gambar yang lebih akurat dan lebih cepat.

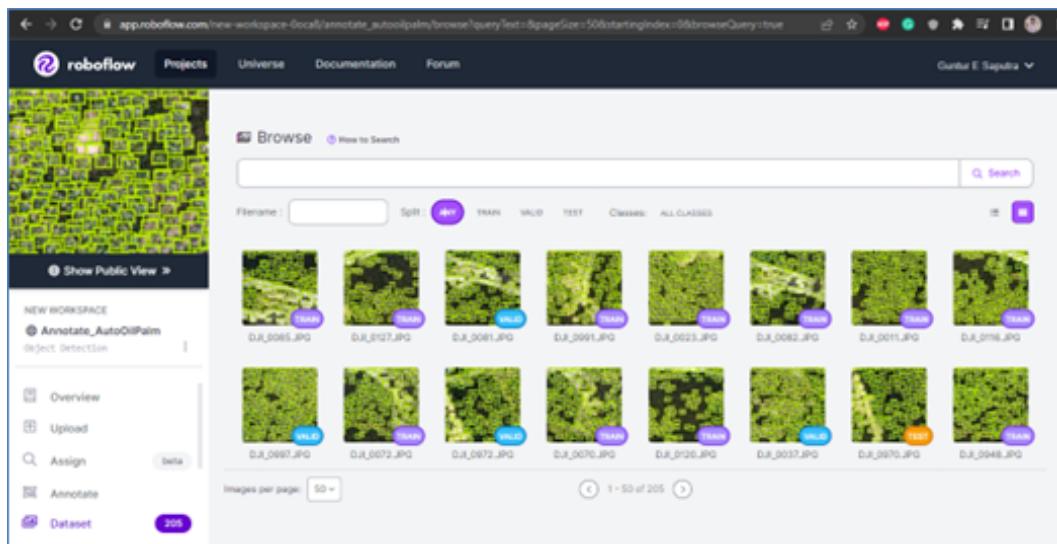
Setelah dilakukan pengujian dan juga dilakukan anotasi otomatis dari citra dan deteksi objek pohon kelapa sawit, maka citra yang tersimpan pada folder di local komputer beserta format file *.txt yang merupakan dataset. Hasil tersebut seperti pada Gambar 4.18



Gambar 4.18: Gambar dan File *.txt

4.2.4.7 Integrasi Data

Proses integrasi data dilakukan untuk menggabungkan atau menyatukan dua dataset, yaitu dataset primer dan sekunder ke dalam satu sumber. Dataset sekunder digunakan layanan roboflow untuk menampung data yang sudah tersimpan pada layanan tersebut. Selanjutnya, hasil dari pembangunan dataset dengan anotasi atau pelabelan otomatis yang telah dilakukan, maka diunggah data citra dan file *.txt.



Gambar 4.19: Dataset Data Primer Drone di Roboflow

Pada Gambar 4.19 terlihat kumpulan gambar sebanyak 205 gambar yang menandakan bahwa dataset dari citra asli sebanyak 205 berhasil terunggah, dan satu citra dapat memiliki lebih dari objek pohon kelapa sawit dengan kelas 'oil palm' pada satu citra.

Pada Gambar 4.19 dan 4.20 terlihat kotak pembatas atau yang dikenal dengan nama *bounding box* yang berwarna hijau, hal ini menandakan objek yang terdeteksi sebagai dataset. Setiap 1 citra dataset primer memiliki jumlah *bounding box* yang berbeda-beda karena tergantung hasil yang dideteksi, seperti pada Gambar 4.20 terdapat 414 objek pohon kelapa sawit yang terdeteksi dalam anotasi atau label dari 1 citra.

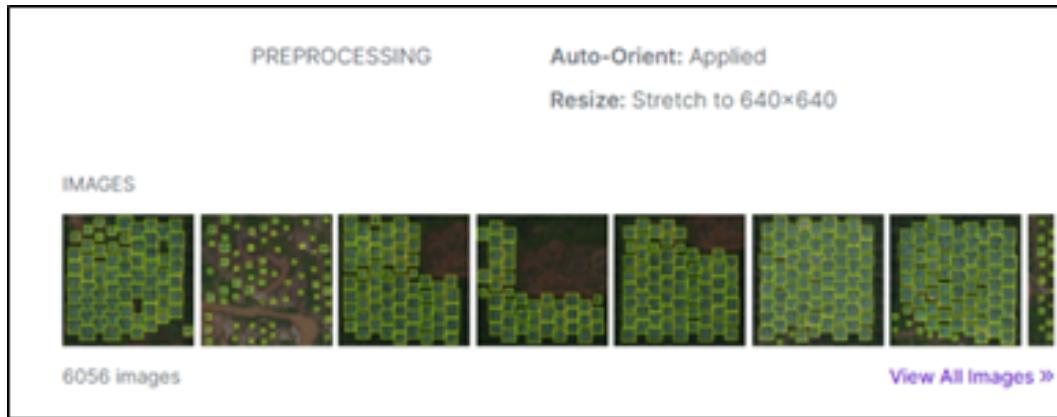


Gambar 4.20: Banyaknya Objek yang Terdeteksi sebagai Pohon Kelapa Sawit

Berdasarkan hasil yang telah dilakukan pada pembentukan dataset secara otomatis dengan pendekatan OBIA berhasil dilakukan, dan citra yang menjadi dataset primer dapat digunakan dan di integrasikan (digabungkan) dengan dataset sekunder untuk komputasi.

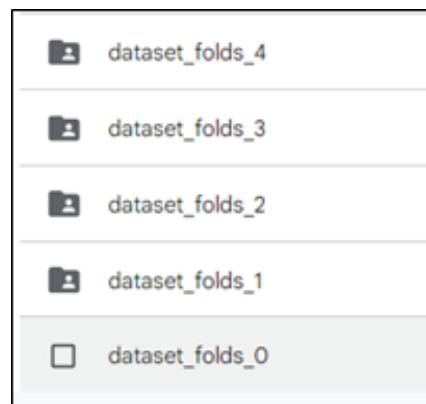
4.2.4.8 Format Data

Pada proses ini dilakukan format data primer dan sekunder dengan menggunakan bantuan layanan roboflow sebelum dapat digunakan untuk citra komputasi yang dibagi menjadi dataset pelatihan, validasi, dan test. Pada tahap ini menyesuaikan tipe data pada citra yaitu pada dataset primer dan sekunder sama-sama *.jpg, dan mengurangi dimensi dataset citra dengan melakukan resize citra menjadi 640 x 640, seperti pada Gambar 4.21.



Gambar 4.21: Format Data

Ketika proses pada Gambar 4.22 sudah dilakukan, maka selanjutnya dataset yang telah sesuai dengan format data yang ditentukan dapat digunakan untuk citra komputasi. Dataset dipindahkan ke layanan Google Drive untuk kemudian dilakukan citra komputasi yang membagi data untuk data pelatihan, validasi dan pengujian sesuai dengan fold yang digunakan, yaitu sebanyak 5 fold (dengan nama dataset folds 0 sampai 4) seperti Gambar 4.22.



Gambar 4.22: Dataset pada layanan Google Drive

4.3 Citra untuk Komputasi

Pada tahap ini citra gambar untuk komputasi merupakan kumpulan dari citra dataset primer dan sekunder. Jumlah dataset sebanyak 6056 data, yang terbagi ke dalam 78% data pelatihan, 20% data validasi dan 2% data

pengujian, seperti pada Tabel 4.11.

Tabel 4.11: Dataset untuk Komputasi

Dataset	Jumlah
Pelatihan	4744
Validasi	1187
Pengujian	125

Dataset tersebut dikumpulkan atau digabungkan, serta dibagi menjadi 5 bagian. Karena pada proses pelatihan dan pengujian menggunakan teknik evaluasi K-Fold Cross-Validation. Berikut ini contoh citra yang digunakan dengan file *.txt yang berisi anotasi objek yang digunakan untuk dataset pelatihan, dataset validasi, dan dataset pengujian seperti tampak pada Gambar 4.23.



(a)



(b)



(b)

Gambar 4.23: Contoh Citra yang digunakan untuk komputasi (a) Citra dataset pelatihan; (b) Citra dataset validasi; (c) Citra dataset pengujian;

4.4 Pelatihan Model CNN

Pada proses pelatihan ini dengan menggunakan model CNN pada YOLOv5, YOLOv6, dan YOLOv7 yang divalidasi dengan menggunakan K-Fold Cross-Validation, dimana semua bagian dataset dapat digunakan untuk pelatihan dan pengujian. Teknik ini digunakan untuk evaluasi performance dari model untuk mengurangi overfitting. Pada penelitian ini menggunakan $k = 5$, dimana proses ini diulang sebanyak $k = 5$ kali. Setiap fold masing-masing terdiri dari data pelatihan yang merupakan gabungan dari data pelatihan sebanyak 4744 data dan 1187 validasi, untuk data test sebanyak 125 data citra. Setiap fold berisi data ini untuk dapat diketahui pada fold mana hasil model yang dilatih hasilnya lebih baik. Pada proses pelatihan ini data citr a untuk dataset dikonversi untuk data masukkan gambar pada proses pelatihan dengan menggunakan ukuran gambar yang diatur oleh YOLO, yaitu secara *default* 640 x 640. Pada pelatihan model dan pengujian menggunakan parameter optimasi seperti pada Tabel 4.12.

Tabel 4.12: Optimisasi Parameter

Parameter	Nilai
lr0	0.01 (1E-2)
lrf	0.1
Momentum	0.937
Weight decay	0.0005 (5e-4)
Epoch	30
Batch size	16

Pada pelatihan menggunakan mesin pada Google Colab Pro dan DGX-A-100 yang dimiliki oleh Universitas Gunadarma. Mesin NVIDIA yang digunakan, seperti pada Tabel 4.13.

Tabel 4.13: Sarana Penelitian Mesin NVIDIA (Super Komputer)

Mesin NVIDIA	Keterangan
Google Colab Pro	A-100-SXM4-40GB, 40536.1875 MB
DGX-A-100 Universitas Gunadarma	A-100-SXM4-40GB, 20096 MB

Pelatihan model CNN pada penelitian ini menggunakan validasi performance yang menghasilkan 4 hasil yang berbeda dengan menggunakan *Precision*, *Recall*, *F1-Score*, dan *best mean Average Precision* atau best mAP@0.5.

4.4.1 Pelatihan Model CNN YOLOv5

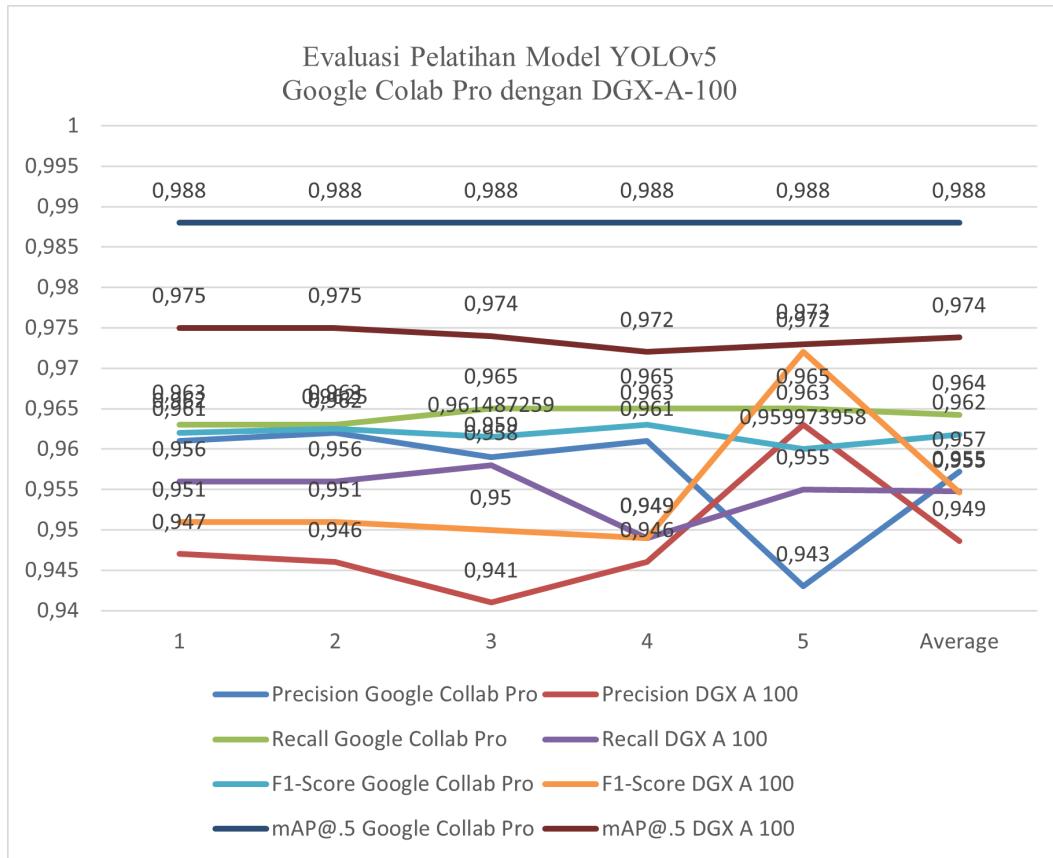
Pelatihan model pertama dilakukan dengan menggunakan YOLOv5. Pada Tabel 4.13 hasil pelatihan model CNN dengan YOLOv5 dengan 30 epoch dan batch size 16 pada Google Colab Pro. Pada mesin DGX-A-100 milik Universitas Gunadarma dengan menggunakan batch size sebesar 16 tidak dapat dijalankan karena permasalahan *sufficient memory* (runs out memory) atau kehabisan shared virtual memori untuk dapat digunakan, sehingga menggunakan batch size dibawahnya dalam 1 iterasi, yaitu 10. Hasil evaluasi model menunjukkan hasil yang cukup signifikan bahwa rata-rata dengan hasil pelatihan dengan Google Colab Pro lebih baik dibandingkan dengan hasil DGX A 100, berturut turut *Precision*, *Recall*, *F1-Score*, best mAP@0.5. Adalah 0,957, 0,955, 0,955 dan 0,974, sedangkan dengan menggunakan DGX-A100 sedikit menurun yaitu 0,949, 0,955, 0,955, dan 0,974, hal ini karena memory pemrosesan dan penggunaan GFlops pada saat pelatihan model. Hasil Fold evaluasi pelatihan model dengan YOLOv5 tampak pada Tabel 4.14.

Hasil pelatihan terbaik ditampilkan pada fold ke-4 dengan menggunakan mesin Goole Collab Pro.

Tabel 4.14: Evaluasi Pelatihan Model dengan YOLOv5 pada Google Colab Pro dan DGX-A-100

Fold	Precision		Recall		F1-Score		mAP@.5	
	Google Co-lab	DGX A						
1	0,961	0,947	0,963	0,956	0,962	0,951	0,988	0,975
2	0,962	0,946	0,963	0,956	0,962	0,951	0,988	0,975
3	0,959	0,941	0,965	0,958	0,962	0,950	0,988	0,974
4	0,961	0,946	0,965	0,949	0,963	0,949	0,988	0,972
5	0,943	0,963	0,965	0,955	0,959	0,972	0,988	0,973
Rata-Rata	0,957	0,949	0,964	0,955	0,962	0,955	0,988	0,974

Evaluasi Pelatihan model YOLOv5 dalam bentuk bagan, untuk dapat melihat perbedaan yang signifikan dalam evaluasi model seperti tampak pada Gambar 4.14



Gambar 4.24: Hasil Evaluasi Pelatihan Model YOLOv5 antara Google Colab Pro dengan DGX-A-100

Hasil dari proses pelatihan ini juga membutuhkan waktu proses yang digunakan antara Google Colab Pro dengan DGX-A-100 milik Universitas Guna darma yang tercantum pada Tabel 4.15.

Tabel 4.15: Evaluasi Waktu Proses Pelatihan Model CNN YOLOv5

Fold	Waktu yang Dibutuhkan Pelatihan	
	Google Colab Pro	DGX A 100
1	4727	922
2	4724	918
3	4695	926
4	4655	900
5	4708	936
Rata-Rata	4702	920

Waktu proses yang dibutuhkan oleh Google Colab Pro rata-rata membutuhkan waktu 4702 detik untuk setiap fold yang harus dijalankan dengan 30

epoch dan 1 iterasi dengan 16 batch size, sedangkan waktu pada DGX-A-100 dibutuhkan 920 waktu DGX-A-100 dengan batch size 10. Perbedaan waktu ini, lebih baik waktu yang diproses pada DGX-A-100 milik Universitas Gu nadarma. Berdasarkan data pada Tabel 4.15, model k-fold terbaik pada fold ke 2 untuk pelatihan YOLOv5.

4.4.2 Pelatihan Model CNN YOLOv6

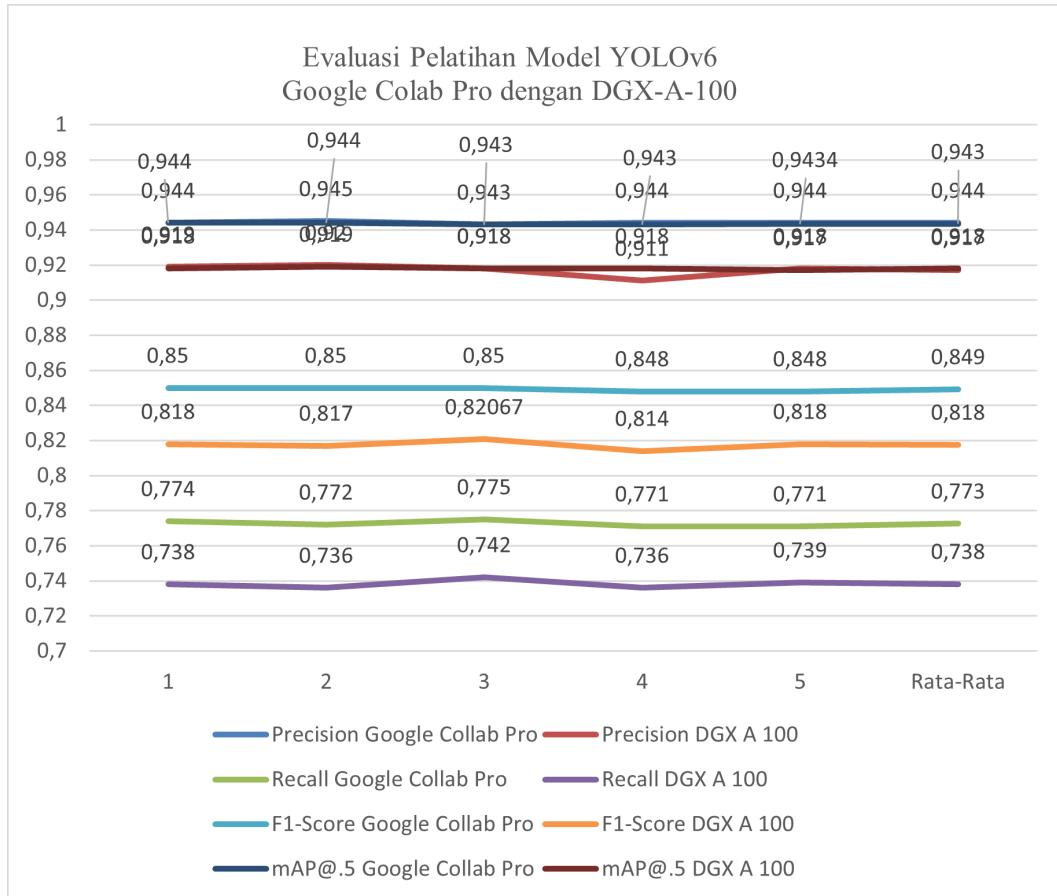
Pelatihan model kedua yaitu dengan model CNN YOLOv6 dengan menggunakan Google Colab Pro dan DGX-A-100. Batch size yang digunakan dalam satu iterasi sama-sama menggunakan 16 dan 30 epoch. Hasil evaluasi pelatihan model CNN dengan YOLOv6 tampak pada Tabel 4.16

Tabel 4.16: Evaluasi Pelatihan Model dengan YOLOv6 pada Google Colab Pro dan DGX-A-100

Fold	Precision		Recall		F1-Score		mAP@.5	
	Google Co-lab	DGX A 100						
1	0,944	0,919	0,774	0,738	0,85	0,818	0,944	0,918
2	0,945	0,920	0,772	0,736	0,85	0,817	0,944	0,919
3	0,943	0,918	0,775	0,742	0,85	0,821	0,943	0,918
4	0,944	0,911	0,771	0,736	0,848	0,814	0,943	0,918
5	0,944	0,918	0,771	0,739	0,848	0,818	0,943	0,917
Rata-Rata	0,944	0,917	0,773	0,738	0,849	0,818	0,943	0,918

Hasil pada Tabel `tbl:Evaluasi-Pelatihan-Model-Dengan-YOLOv6` menunjukkan bahwa hasil evaluasi model dengan menggunakan mesin Google Colab Pro masih lebih tinggi dibandingkan dengan DGX-A-100, yaitu berurut-turut sebesar 0,944, 0,773, 0,849, dan 0,943 dengan 4 evaluasi model. Nilai rata-rata Precision cukup tinggi berada di atas 0,90, tetapi recall di bawah 0,80, hal ini menunjukkan bahwa pada YOLOv6 rasio prediksi benar positif

dibandingkan dengan keseluruhan data yang benar positif yaitu pohon kelapa sawit masih dibawah nilai dari proses pelatihan YOLOv5. Berikut ini hasil evaluasi performance dalam bentuk diagram tampak seperti pada Gambar 4.25.



Gambar 4.25: Hasil Evaluasi Pelatihan Model YOLOv6 antara Google Colab Pro dengan DGX-A-100

Sementara itu, Proses pelatihan ini juga membutuhkan waktu untuk pemrosesan pelatihan yang tampak pada Tabel 4.17.

Tabel 4.17: Evaluasi Waktu Proses Pelatihan Model CNN YOLOv6

Fold	Waktu yang Dibutuhkan Pelatihan Google Colab Pro	DGX A 100
1	7359	2668
2	6696	2618
3	6689	2675
4	6707	2657
5	8957	2654
Rata-Rata	7282	2654

Waktu proses yang dibutuhkan oleh Google Colab Pro rata-rata membutuhkan waktu 7282 detik untuk setiap fold yang harus dijalankan dengan 30 epoch dan 1 iterasi dengan 16 batch size, sedangkan waktu pada DGX-A-100 dibutuhkan 2654 waktu DGX-A-100. Perbedaan waktu ini, lebih baik waktu yang diproses pada DGX-A-100 milik Universitas Gunadarma. Berdasarkan data pada Tabel 4.17, model k-fold terbaik pada fold ke 2 untuk pelatihan YOLOv6.

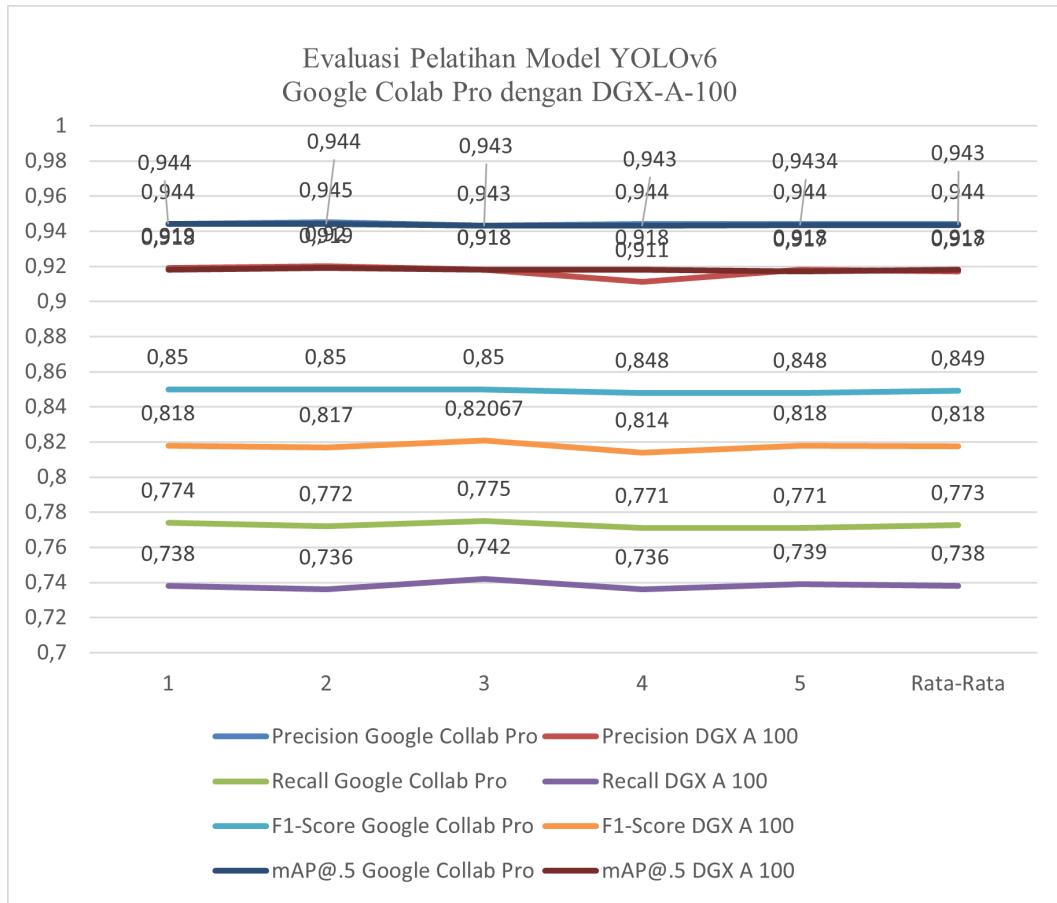
4.4.3 Pelatihan Model CNN YOLOv7

Pelatihan model ketiga yaitu dengan model CNN YOLOv7 dengan menggunakan Google Colab Pro dan DGX-A-100. Batch size yang digunakan dalam pelatihan model ini dalam satu iterasi sama-sama menggunakan 16 batch size dan 30 epoch. Hasil evaluasi pelatihan model CNN dengan YOLOv7 tampak pada Tabel 4.18.

Tabel 4.18: Evaluasi Pelatihan Model dengan YOLOv7 pada Google Colab Pro dan DGX-A-100

Fold	Precision		Recall		F1-Score		mAP@.5	
	Google Co-lab	DGX A						
1	0,967	0,961	0,977	0,970	0,972	0,965	0,993	0,989
2	0,970	0,965	0,976	0,966	0,973	0,965	0,993	0,989
3	0,967	0,961	0,979	0,970	0,973	0,965	0,993	0,989
4	0,972	0,965	0,973	0,966	0,973	0,965	0,993	0,989
5	0,971	0,963	0,974	0,964	0,972	0,963	0,993	0,988
Rata-Rata	0,969	0,963	0,976	0,967	0,973	0,965	0,993	0,989

Hasil pada Tabel 4.18 menunjukkan bahwa hasil evaluasi model dengan menggunakan mesin Google Colab Pro masih lebih tinggi dibandingkan dengan DGX-A-100, yaitu beruturut-turut sebesar 0,969, 0,976, 0,973, dan 0,993 dari 4 evaluasi model yang digunakan. Nilai rata-rata best mAP@.5 berada pada posisi nilai 0,993 yang dimana dari fold 1 hingga fold 5 berada pada nilai yang konsisten 0,993, hal ini membuktikan bahwa algoritma YOLOv7 mencapai akurasi tertinggi diantara model pendektsian objek secara real-time lainnya, dan telah ditetapkan berdasarkan artikel secara resmi yang ditulis mengenai YOLOv7. Berikut ini hasil evaluasi performance dalam bentuk diagram tampak seperti pada Gambar 4.26.



Gambar 4.26: Hasil Evaluasi Pelatihan Model YOLOv7 antara Google Colab Pro dengan DGX-A-100

Selanjutnya, waktu proses pelatihan model pada YOLOv7 tampak pada Tabel 4.19.

Tabel 4.19: Evaluasi Waktu Proses Pelatihan Model CNN YOLOv7

Fold	Waktu yang Dibutuhkan Pelatihan	
	Google Colab Pro	DGX A 100
1	4857	2988
2	4794	2866
3	4820	2966
4	4888	2952
5	4874	2968
Rata-Rata	4847	2948

Waktu proses yang dibutuhkan untuk setiap fold yang harus dijalankan dengan 30 epoch dan 1 iterasi dengan 16 batch size oleh Google Colab Pro

rata-rata membutuhkan waktu 4847 detik, sedangkan waktu pada DGX-A-100 dibutuhkan 2948 waktu. Perbedaan waktu ini, lebih baik waktu yang diproses pada DGX-A-100 milik Universitas Gunadarma. Berdasarkan data pada Tabel 4.19, model k-fold terbaik pada fold ke 2 untuk pelatihan YOLOv7.

4.5 Pengujian Model CNN

Proses pelatihan dengan dataset pelatihan telah dilakukan, selanjutnya dilakukan proses pengujian dengan dataset uji. Dataset uji yang digunakan pada setiap k-fold yaitu 125 citra. Pada tahap ini dilakukan pengujian yang dilakukan pada model CNN YOLOv5, YOLOv6, dan YOLOv7. Pada proses pengujian ini juga menggunakan mesin super komputer DGX-A-100 milik Universitas Gunadarma dan Google Colab Pro.

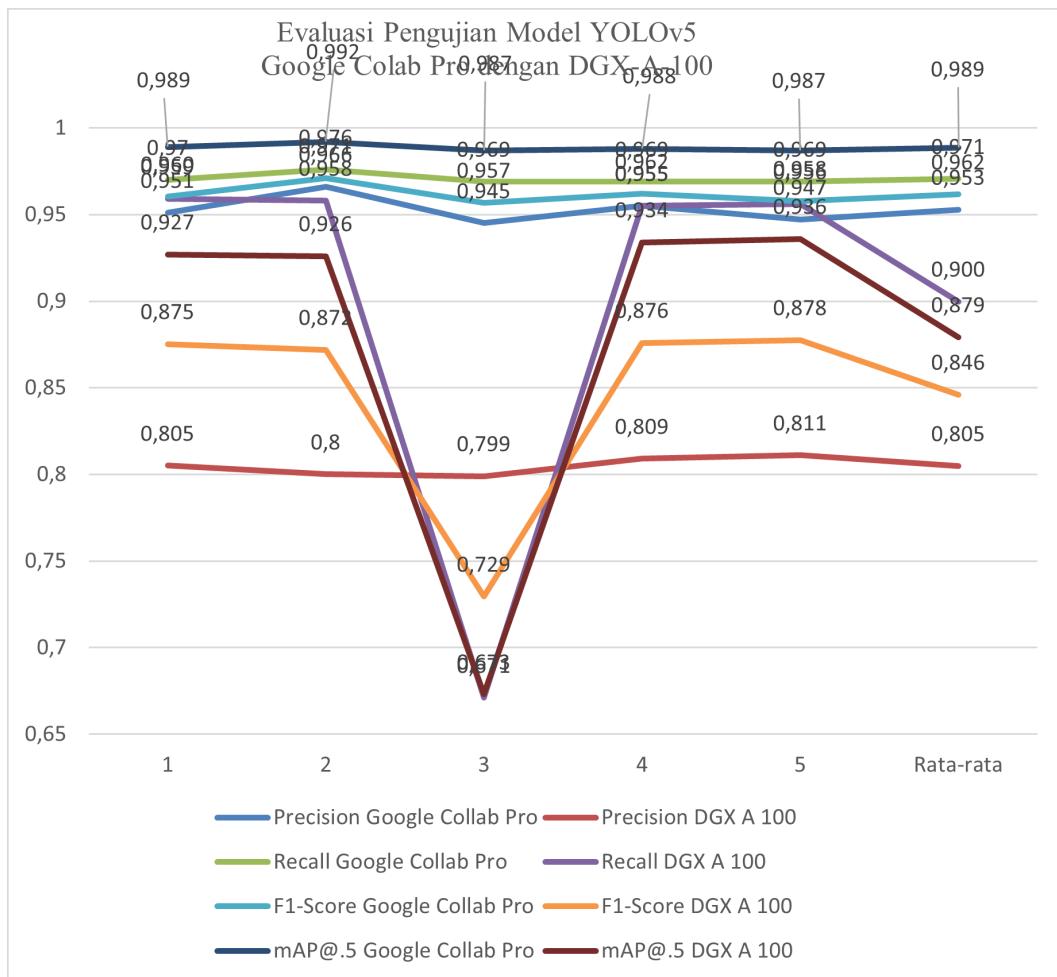
4.5.1 Pengujian Model CNN YOLOv5

Pengujian model CNN yang pertama dengan menggunakan YOLOv5 dengan data citra pengujian sebanyak 125 citra pada 5 k-fold cross-validation. Pada proses pengujian ini batch size yang digunakan adalah 32 batch size yang berarti 32 jumlah sampel data yang melewati jaringan untuk setiap 1 iterasi atau putaran. Hasil dari pengujian model YOLOv5 dengan menggunakan mesin Google Colab Pro dan DGX-A-100 tampak pada Tabel 4.20.

Tabel 4.20: Evaluasi Pengujian Model dengan YOLOv5 pada Google Colab Pro dan DGX-A-100

Fold	Precision		Recall		F1-Score		mAP@.5	
	Google Co-lab	DGX A						
1	0,951	0,805	0,970	0,959	0,960	0,875	0,989	0,927
2	0,966	0,800	0,976	0,958	0,971	0,872	0,992	0,926
3	0,945	0,799	0,969	0,671	0,957	0,729	0,987	0,673
4	0,955	0,809	0,969	0,955	0,962	0,876	0,988	0,934
5	0,947	0,811	0,969	0,956	0,958	0,878	0,987	0,936
Rata-rata	0,953	0,805	0,971	0,900	0,962	0,846	0,989	0,879

Hasil pengujian menunjukkan bahwa hasil best *mean average* (best mAP) untuk YOLOv5 dengan mesin Google Colab Pro lebih tinggi yaitu sebesar 0,989 dibandingkan dengan DGX-A-100 sebesar 0,879. Hasil ini menampilkan bahwa rata-rata presisi pada kelas pohon kelapa sawit di pengujian signifikan. Sebaran hasil pengujian model dengan YOLOv5 tampak seperti pada Gambar 4.27.



Gambar 4.27: Hasil Evaluasi Pengujian Model YOLOv5 antara Google Colab Pro dengan DGX-A-100

Dari hasil gambar menunjukkan bahwa best mAP terendah bernilai di bawah 0,70 pada mesin DGX-A-100 pada fold ke-3. Hal ini disebabkan bahwa pada fold 3 nilai rata-rata presisi cukup rendah untuk mendekripsi dengan benar bahwa itu adalah objek pohon kelapa sawit, kemudian hasil naik kembali pada fold ke 4 dan 5.

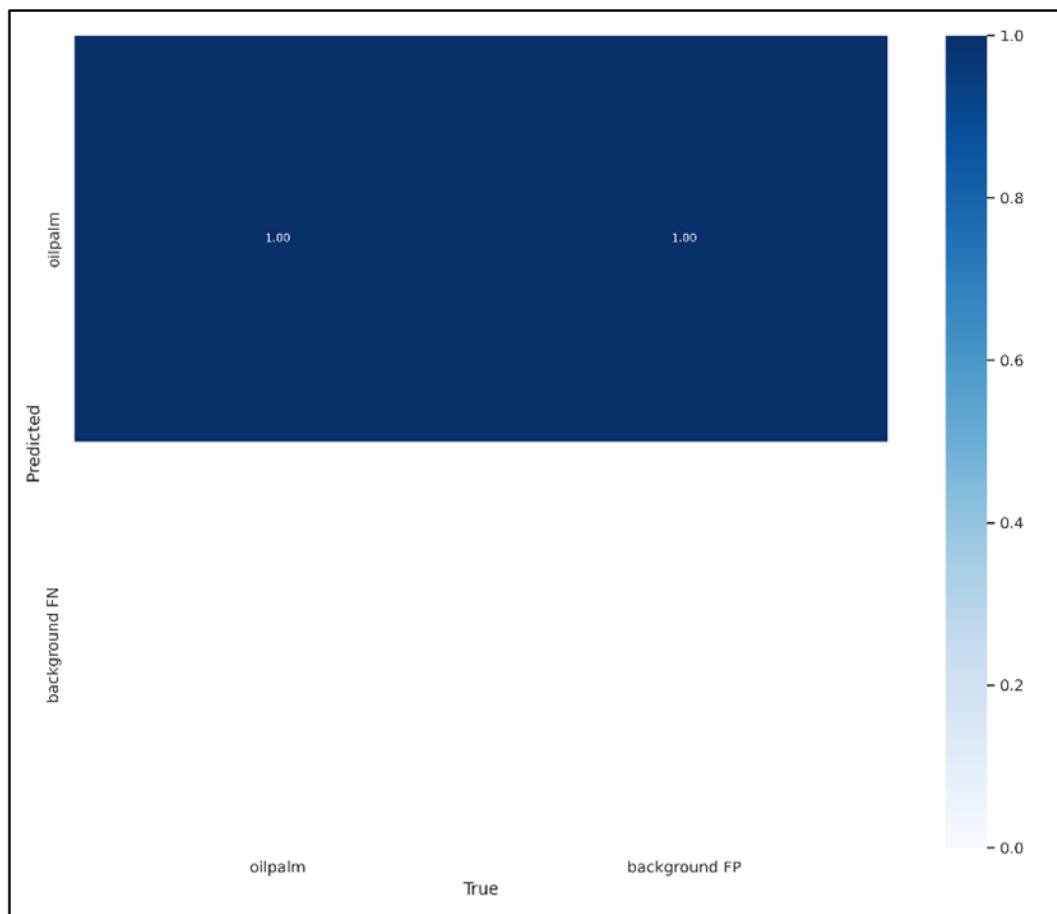
Waktu yang dibutuhkan untuk proses pengujian ini adalah tampak pada Tabel 4.21.

Tabel 4.21: Evaluasi Waktu Proses Pengujian Model CNN YOLOv5

Fold	Waktu (detik) Google Colab Pro	DGX A 100
1	122	54
2	119	50
3	118	54
4	122	52
5	123	54
Rata-Rata	121	53

Hasil dari waktu proses pengujian model dengan 125 data citra pada YOLOv5 oleh Google Colab Pro rata-rata membutuhkan waktu 121 detik, sedangkan waktu pada DGX-A-100 dibutuhkan 53 detik. Perbedaan waktu ini, lebih baik waktu yang diproses pada DGX-A-100 milik Universitas Gu nadarma. Waktu komputasi pengujian lebih baik pada penggunaan DGX-A-100.

Berdasarkan hasil pengujian tersebut, berikut ini hasil confusion matrix dan hasil keluaran pada dataset pengujian pada Gambar 4.28 (a) dan (b).



(a)



(b)

Gambar 4.28: (a) Confusion Matrix dan (b) Citra Sampel Prediksi Pohon Kelapa Sawit dari Citra Dataset Pengujian

Gambar 4.28. menunjukkan confusion matrix yang bernilai positif, bahwa hasil menunjukkan nilai 1.00 pada oil palm, hal ini berarti model dapat berhasil memprediksi dengan benar bahwa objek tersebut merupakan oil palm (pohon kelapa sawit) dataset citra pengujian, dan yang gambar yang sebelah kanan merupakan hasil pengujian dengan menampilkan kotak batas (*bounding box*) yang disertai nilai *confidence* yang menunjukkan bahwa nilai tersebut termasuk ke dalam kelas oil palm. Berdasarkan data pada Tabel 4.21, k-fold terbaik pada fold ke 2 untuk pengujian YOLOv5.

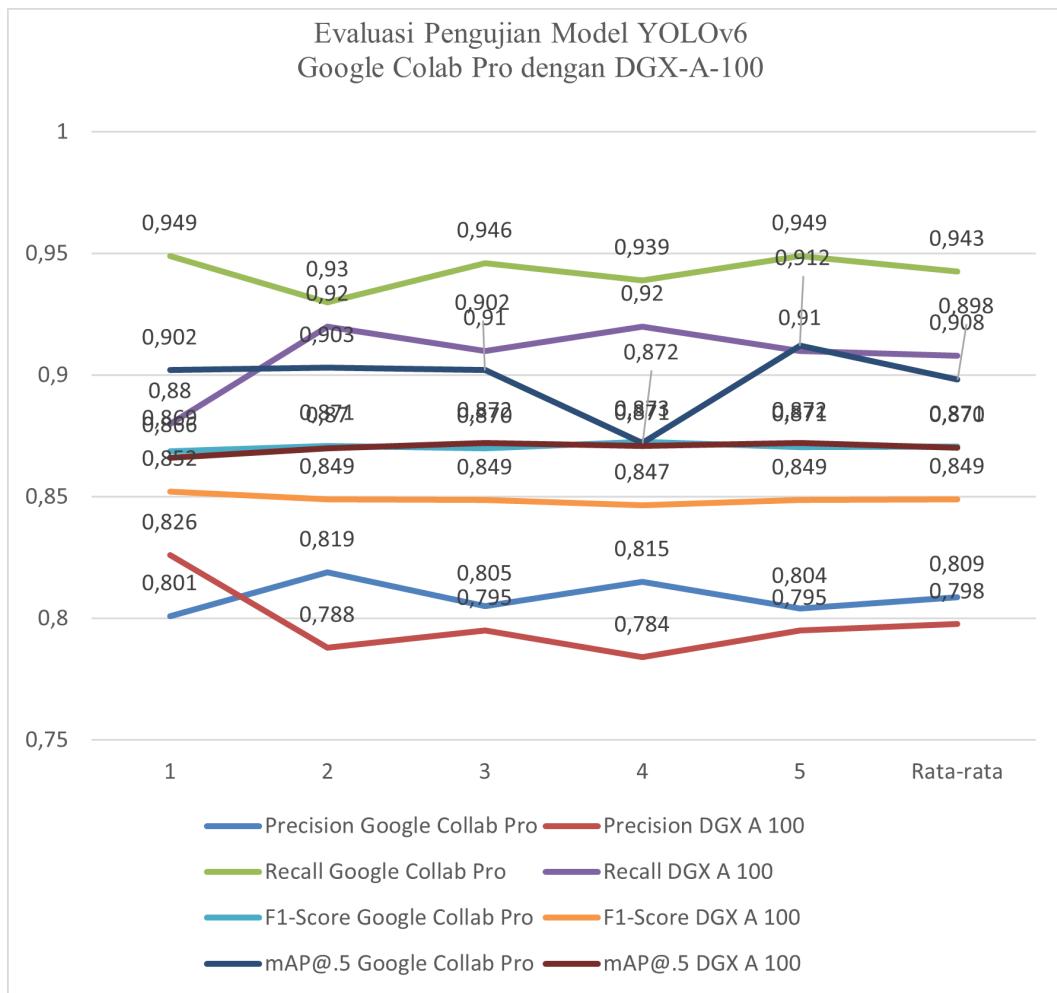
4.5.2 Pengujian Model CNN YOLOv6

Pengujian model yang kedua menggunakan model YOLOv6. Input image dengan menggunakan 640 x 640 dan batch size yang digunakan 32 batch size untuk satu iterasi. Hasil pengujian model dengan dua mesin yang digunakan, tampak seperti pada Tabel 4.22.

Tabel 4.22: Evaluasi Pengujian Model dengan YOLOv6 pada Google Colab Pro dan DGX-A-100

Fold	Precision		Recall		F1-Score		mAP@.5	
	Google	DGX	Google	DGX	Google	DGX	Google	DGX
Co-lab	A	100	Co-lab	A	Co-lab	A	Co-lab	A
Pro	Pro	Pro	Pro	Pro	Pro	Pro	Pro	Pro
1	0,801	0,826	0,949	0,880	0,869	0,852	0,902	0,866
2	0,819	0,788	0,930	0,920	0,871	0,849	0,903	0,870
3	0,805	0,795	0,946	0,910	0,870	0,849	0,902	0,872
4	0,815	0,784	0,939	0,920	0,873	0,847	0,902	0,871
5	0,804	0,795	0,949	0,910	0,871	0,849	0,912	0,872
Rata-rata	0,809	0,798	0,943	0,908	0,871	0,849	0,898	0,870

Berdasarkan Tabel 4.22 hasil terbaik pada best mAP fold ke 5, dengan nilai rata-rata terbaik presisi sebesari 0,912 dan berdasarkan hasil rata-rata best mAP dengan Google Colab Pro juga lebih tinggi dibandingkan dengan DGX-A-100. Hasil sebaran evaluasi pengujian model dengan YOLOv6 seperti tampak pada Gambar 4.29.



Gambar 4.29: Hasil Evaluasi Pengujian Model YOLOv6 antara Google Colab Pro dengan DGX-A-100

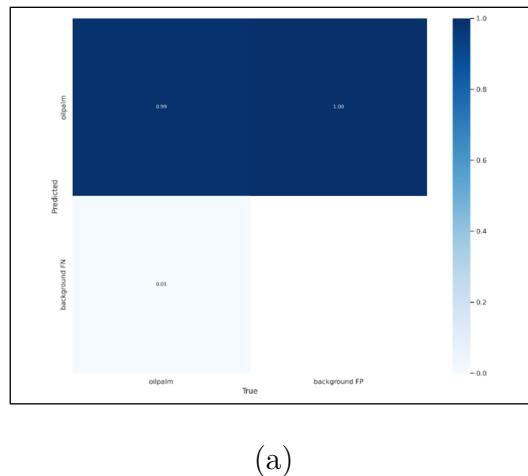
Berdasarkan dari waktu yang dibutuhkan pada proses pengujian model pada YOLOv6 dengan DGX-A-100 lebih cepat dibandingkan dengan Google Colab Pro. Hasil tampak pada Tabel 4.23.

Tabel 4.23: Evaluasi Waktu Proses Pengujian Model CNN YOLOv6

Fold	Waktu (detik) Google Colab Pro	DGX A 100
1	54	53
2	56	53
3	59	50
4	57	52
5	59	50
Rata-Rata	57	54

Hasil dari waktu proses pengujian model pada YOLOv6 dengan data pengujian 125 citra, dihasilkan waktu yang tidak terlalu berbeda. Waktu yang dibutuhkan oleh Google Colab Pro rata-rata membutuhkan waktu 57 detik, sedangkan waktu pada DGX-A-100 dibutuhkan 54 detik. Perbedaan waktu ini, waktu DGX-A-100 lebih baik dibandingkan dengan DGX-A-100.

Berdasarkan hasil dari pengujian YOLOv6 pada Tabel 4.23, pada Gambar 4.31. merupakan *confusion matriks* dari pengujian YOLOv6.



(a)



(b)

Gambar 4.30: (a) Confusion Matrix Pengujian YOLOv6 (b) Citra Sampel Pengujian

Gambar 4.30. menampilkan bahwa interpretasi dari *false positive* berni-

lai 0.0 dan benar memprediksi objek kelas oil palm, sedangkan pada *false negative* bernilai 0.01, ini menandakan bahwa model mendeteksi bahwa tersebut bukan 'oil palm' dan ternyata bernilai negative atau salah, tetapi model masih memiliki akurasi yang baik karena nilai oil palm berada pada 0,99.

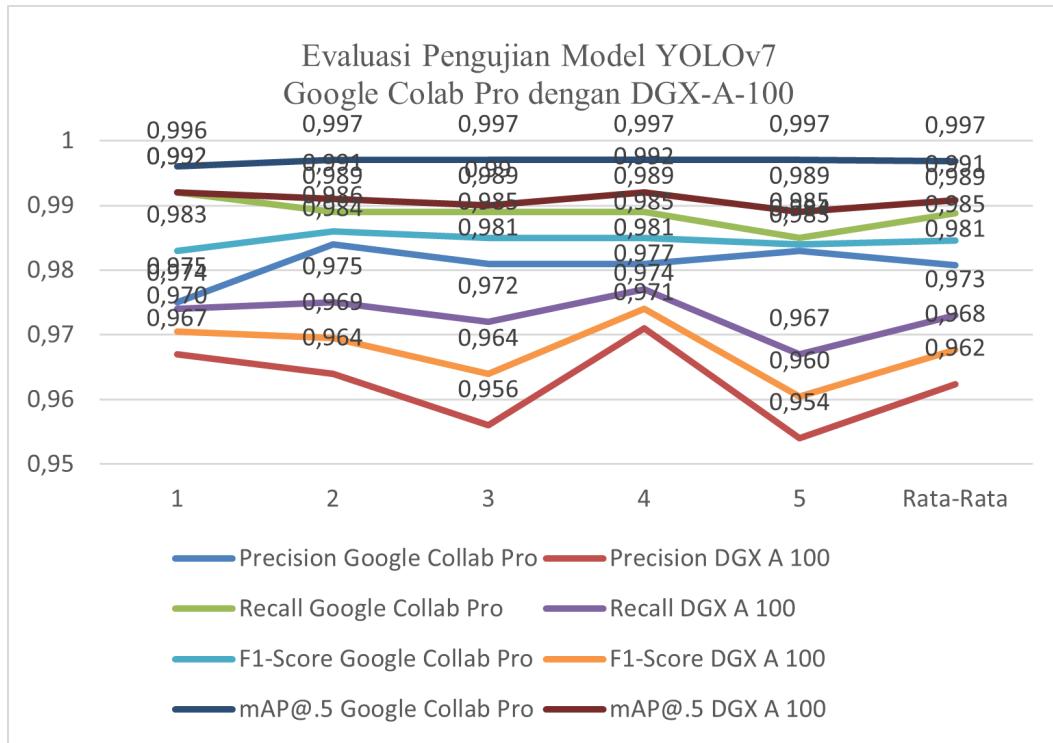
4.5.3 Pengujian Model CNN YOLOv7

Pengujian ketiga adalah pengujian YOLOv7, hasil dari pengujian dengan data test sebanyak 125 citra seperti pada Tabel 4.24.

Tabel 4.24: Evaluasi Pengujian Model dengan YOLOv7 pada Google Colab Pro dan DGX-A-100

Fold	Precision		Recall		F1-Score		mAP@.5	
	Google Co-lab	DGX A						
1	0,975	0,967	0,992	0,974	0,983	0,970	0,996	0,992
2	0,984	0,964	0,989	0,975	0,986	0,969	0,997	0,991
3	0,981	0,956	0,989	0,972	0,985	0,964	0,997	0,990
4	0,981	0,971	0,989	0,977	0,985	0,974	0,997	0,992
5	0,983	0,954	0,985	0,967	0,984	0,960	0,997	0,989
Rata-rata	0,981	0,962	0,989	0,973	0,985	0,968	0,997	0,991

Berdasarkan hasil pengujian model CNN YOLOv7 pada Tabel 4.24. dinyatakan bahwa hasil best MAP terbaik dengan rata-rata sebesar 0,997 pada Google Colab Pro, dibanding dengan DGX-A-100. Hasil sebaran nilai pengujian YOLOv7 pada Gambar 4.31.



Gambar 4.31: Hasil Evaluasi Pengujian Model YOLOv7 antara Google Colab Pro dengan DGX-A-100

Pada gambar 4.32 terlihat perbedaan dari evaluasi performance antara DGX-A-100 dengan Google Colab Pro, best mAP pada dengan mesin Google Colab Pro lebih tinggi, dan nilai *precision* dari DGX-A-100 di bawah dari nilai lainnya.

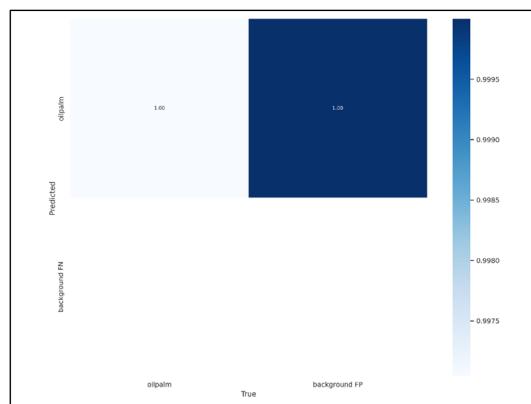
Di samping itu, pada proses waktu yang dibutuhkan dalam proses pengujian, DGX-A-100 milik Universitas Gunadarma lebih cepat dibandingkan Google Colab Pro, tampak seperti pada Tabel 4.25.

Tabel 4.25: Evaluasi Waktu Proses Pengujian Model CNN YOLOv7

Fold	Waktu (detik)	
	Google Colab Pro	DGX A 100
1	55	55
2	55	55
3	52	56
4	52	52
5	52	52
Rata-Rata	53	54

Hasil dari waktu proses pengujian model pada YOLOv7 dengan data pengujian 125 citra, dihasilkan waktu yang tidak terlalu berbeda. Waktu yang dibutuhkan oleh Google Colab Pro rata-rata membutuhkan waktu 53 detik, sedangkan waktu pada DGX-A-100 dibutuhkan 54 detik. Hasil ini menyatakan bahwa waktu proses pada Google Colab Pro lebih baik 1 detik dibanding DGX-A-100.

Berdasarkan hasil dari pengujian YOLOv7 pada Gambar 4.32. merupakan *confusion matriks* hasil gambar pengujian dari pengujian YOLOv7.



(a)



(b)

Gambar 4.32: (a) Confusion Matrix Pengujian YOLOv7 (b) Citra Sampel Pengujian;

Hasil YOLOv7 lebih tinggi dibandingkan dengan pengujian pada model CNN YOLOv5 dan YOLOv6. Hasil rata-rata presisi dari kelas oil palm tinggi, dan best model pada fold pada pengujian pada YOLOv7, pada fold ke 2 dengan evaluasi performance berurutan *precision*, *recall*, *f1-score*, dan mAP@.5 sebesar 0,984, 0,989, 0,986, dan 0,997. Best mAP ini mendekati nilai sempurna yaitu 1.0 atau senilai 100% model sangat baik dalam mendekripsi pohon kelapa sawit pada citra. Disamping itu, pada hasil pelatihan dengan dataset pelatihan bahwa fold terbaik pada proses pelatihan dengan YOLOv7 juga terletak pada fold ke 2, dengan nilai sebesar 0,970, 0,976, 0,973, dan 0,993. Terjadi peningkatan pada pelatihan model dengan pengujian model, dari 0,993 ke 0,997 meningkat senilai 0,004. Berdasarkan hasil akurasi yang dicapai pengujian sebesar 0,997 atau 99,70% bahwa penelitian ini memberikan peningkatan dari penelitian sebelumnya pada H. M. Rizeei et al. tahun 2018 integrasi OBIA dan SVM dengan mencapai akurasi 98% (H. M. Rizeei et al, 2018), tahun 2021 Y. Nurmasari & Wijayanto menggunakan SVM dan NaÃ¢ve Bayes mencapai akurasi 92% (Y. Nurmasari, 2021), dan tahun 2022 H. Wibowo et al. dengan menggunakan YOLO dengan akurasi 97,74% (H. Wibowo, 2022), sehingga ini menjadi nilai kebaruan bahwa hasil akurasi dari penelitian ini lebih baik dibandingkan dengan penelitian yang telah dilakukan sebelumnya.

Berdasarkan hasil pelatihan dan pengujian dari kecepatan waktu proses, secara umum DGX-A-100 Universitas Gunadarma lebih baik dibandingkan dengan Google Colab Pro. Hal ini disebabkan beberapa faktor, diantaranya:

1. Keterbatasan Jaringan: Waktu proses pada Google Colab Pro dipengaruhi oleh keterbatasan jaringan dan memiliki *idle time* (waktu tunggu) untuk respon. Disamping itu, pada Google Colab Pro, data dan file disimpan di cloud (Google Drive), sehingga waktu akses dan transfer data dapat lebih lambat dibandingkan dengan akses dengan DGX-A-100

Universitas Gunadarma, data dan file disimpan pada lingkungan yang sama dengan tools yang digunakan.

2. Beban Server: Jumlah pengguna yang menggunakan Google Colab Pro pada saat bersamaan dapat mempengaruhi waktu proses. Jika terdapat banyak pengguna yang menggunakan sumber daya yang sama pada saat bersamaan, waktu proses dapat lebih lama karena Google Colab Pro harus menangani permintaan dari banyak pengguna, dibandingkan DGX-A-100 Universitas Gunadarma.
3. Konfigurasi Perangkat Keras dan Lunak: Konfigurasi ini mungkin dapat berbeda, sehingga mempengaruhi waktu proses yang digunakan oleh Google Colab Pro karena banyak pengguna yang mengakses (shared sources), dibandingkan dengan DGX-A-100 yang terbatas akses pengguna tidak sebanyak dengan Google Colab Pro.

4.6 Hasil Model YOLOv7 Pelatihan dengan Inferensi

Pada fase pelatihan dan pengujian dihasilkan bahwa model terbaik pada YOLOv7, yaitu pada saat pelatihan dihasilkan akurasi matrik pada best MAP sebesar 0,993 pada fold ke 2 dengan waktu terbaik pada fold ke 2 dengan DGX-A-100 milik Universitas Gunadarma selama 2866 detik. Pada fase pelatihan ini, model dengan dataset telah dikurasi atau diolah, sehingga model tersebut dapat mempelajari semua yang diperlukan mengenai objek pohon kelapa sawit 'oil palm'. Kemudian, pada fase inferensi pada *machine learning* dibutuhkan untuk menilai kemampuan sistem membuat prediksi berdasarkan data baru dan menghasilkan hasil yang dapat ditindaklanjuti untuk diterapkan dalam sistem.

Model yang digunakan adalah model terbaik pada fold ke 2 pada pelatihan dengan YOLOv7 dan diuji dengan dataset pada fold ke 1 untuk menilai

kemampuan dari kecepatan waktu yang tercatat pada tools seperti pada Tabel 4.26.

Tabel 4.26: Hasil Kecepatan Waktu Proses Dataset dengan Model YOLOv7 Pelatihan dengan Inferensi

Hasil dari	Waktu Proses (detik)
Pelatihan	2866
Inferensi	2511

Berdasarkan Tabel 4.26 kecepatan waktu proses inferensi lebih baik dan ini menjadi salah satu hal yang penting dalam *machine learning*, terutama untuk model yang dibuat ini digunakan dalam sistem yang memerlukan pemrosesan secara real-time untuk penelitian ini. Disamping itu, waktu inferensi yang lebih cepat dapat memastikan bahwa model dapat memberikan hasil dalam waktu yang sesingkat mungkin dan tidak mengganggu respons sistem secara keseluruhan.

4.7 Hasil Deteksi dan Penghitungan Akhir

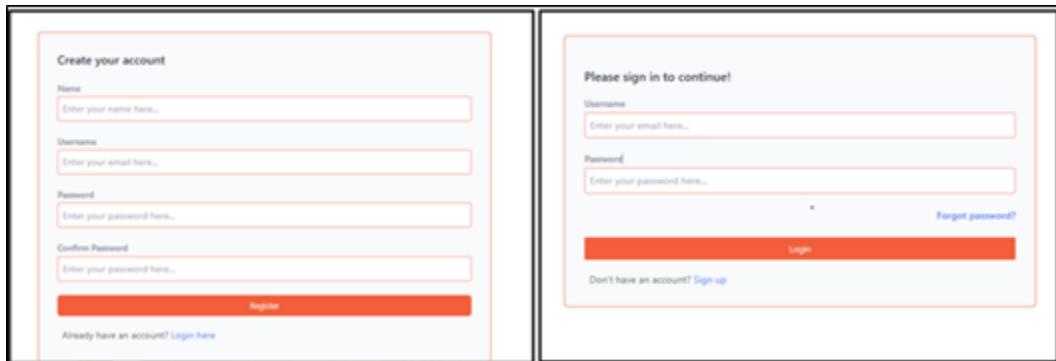
Pelatihan dan pengujian model telah dilakukan, berdasarkan hasil pelatihan dan pengujian dengan model CNN YOLOv5, YOLOv6, dan YOLOv7 bahwa model CNN dengan YOLOv7 lebih baik dibandingkan dengan model lainnya yang telah dilakukan pada pelatihan dan pengujian model dalam penelitian ini. Dalam penelitian ini, model YOLOv7 diimplementasikan untuk mendeteksi dan menghitung pohon kelapa sawit melalui citra satelit yang diambil melalui Google Maps API yang terintegrasi dengan citra dari Maxar Technologies yang memberikan citra satelit dari Sentinel-2. Penerapan deteksi dan penghitungan memiliki beberapa tahap.

4.7.1 Mengubah Model CNN ke Format ONNX

Model CNN yang digunakan pada penelitian ini dari hasil pelatihan dan pengujian adalah YOLOv7. Mengkonversi format ini dengan menggunakan repository resmi model dari YOLOv7 dengan menggunakan git dan menggunakan kode program atau script *export.py*. Model ini harus dikonversi atau diubah ke ONNX atau *open neural network exchange* karena ONNX merupakan format terbuka untuk merepresentasikan dan berbagi model deep learning. ONNX menyediakan format standar yang memungkinkan interoperabilitas antara kerangka kerja deep learning yang berbeda, sehingga lebih mudah untuk berbagi model di berbagai platform dan perangkat. Hasil konversi model ONNX ini berisi dua komponen utama, yaitu kombinasi grafik komputasi dan seperangkat parameter terlatih dari hasil pelatihan dan pengujian pada model CNN. ONNX diletakkan pada sisi backend server yaitu tepatnya di FastAPI.

4.7.2 Mendapatkan titik tengah koordinat (latitude - longitude)

Tahap berikutnya sesuai pada pembahasan bab 3 subbab mengenai diagram arsitektur, maka untuk mendapatkan titik tengah koordinat latitude - longitude dari sistem yang diterapkan untuk mendeteksi dan menghitung kelapa sawit, serta diketahui titik koordinat tersebut, menggunakan sistem yang terintegrasi dengan Google Maps API. Pengguna register terlebih dahulu untuk mendapatkan akun dan menggunakan sistem, setelah itu dapat login dan menggunakan sistem tersebut, seperti pada Gambar 4.33.



Gambar 4.33: Halaman Register dan Login

Setelah berhasil mengaktifkan akun, maka berikutnya ditampilkan halaman yang menampilkan peta maps yang terintegrasi Google Maps API.



Gambar 4.34: Tampilan Citra Maps Satelit

Selanjutnya, untuk dapat mendekripsi dan menghitung pohon kelapa sawit pada suatu area lahan dari citra maps satelit. Untuk mendapatkan titik tengah koordinat tersebut menggunakan polygon. Poligon ini mencakup area 100 x 100 m yang merupakan kesesuaian yang dimiliki atau yang diberikan dari layanan Google Maps API ini, seperti pada Gambar 4.35.



Gambar 4.35: Polygon untuk Mendeteksi dan Menghitung Pohon Kelapa Sawit

Dapat diperhatikan bahwa Gambar 4.35 ini tampilan dengan sebuah polygon sebesar 100 x 100 m yang digunakan untuk memetakan suatu area lahan yang akan dideteksi dan dihitung jumlah pohon kelapa sawit, untuk dapat diketahui setiap letak posisi koordinat, maka ditampilkan dan digunakan terlebih dahulu titik tengah berupa titik latitude dan longitude dengan menggunakan library bantuan yaitu *leaflet js* yang merupakan *mapping library*. Titik koordinat dari gambar ini sebagai kunci untuk dapat bisa dihitung dan ditampilkan titik koordinat dari setiap objek yang berhasil dideteksi dan diketahui lokasi setiap titik koordinat tersebut. Titik koordinat terdeteksi yaitu latitude: -6.47484745212241 dan longitude: 107.03319672664993.

4.7.3 Mendeteksi dan Menghitung berdasarkan Existing Model

Pada tahap ini untuk mendeteksi pohon kelapa sawit dengan menggunakan model yang tersimpan pada backend server dengan menggunakan FastAPI. Konsep mendeteksi pohon kelapa sawit sama seperti pada penjelasan subbab 3.7 pada cara kerja algoritma YOLO. Jika berhasil terdeteksi data citra ditunjukkan dengan kotak pembatas atau *bounding box*. Jumlah *bounding box* yang tampil (berhasil dideteksi) ini yang digunakan untuk dapat menghitung banyaknya pohon kelapa sawit, seperti pada Gambar 4.36.



Gambar 4.36: Hasil Mendeteksi dan Menghitung Pohon Kelapa Sawit

4.7.4 Mendapatkan objek yang terdeteksi dengan format YOLO

Pada setiap kotak pembatas atau *bounding box* yang tampil menandakan bahwa itu merupakan objek pohon kelapa sawit. Setiap *bounding box* ini berisi kelas oil palm, koordinat x, koordinat y, width, dan height. Tetapi, objek yang terdeteksi yang memiliki titik tengah (koordinat x dan y) saat ini bukanlah lokasi atau piksel koordinat yang sebenarnya. Titik-titik ini masih

menggunakan format YOLO. Format YOLO hanya merepresentasikan posisi relative, maka dibutuhkan konversi terlebih dahulu ke format COCO. Format COCO merepresentasikan lokasi piksel asli dari *bounding box* pohon kelapa sawit.

Untuk mendapatkan gambaran bagaimana setiap objek pohon kelapa sawit tidak hanya terdeteksi dan dihitung, tetapi dapat memiliki posisi atau titik koordinat latitude dan longitude yang dapat digunakan untuk pemantauan lebih lanjut, maka dijelaskan dalam simulasi sesuai diagram alir pada Gambar 3.14.

Setiap objek yang berhasil dideteksi sebagai pohon kelapa sawit, maka ditandai dengan *bounding box* didalam polygon 100 x 100m, dan memiliki ID, lokasi koordinat latitude, longitude, serta nilai confidence, tampak seperti Gambar 4.37.



Gambar 4.37: Hasil deteksi dan penghitungan, serta menampilkan titik-titik koordinat latitude-longitude lokasi setiap objek pohon kelapa sawit

Gambar 4.37 merupakan salah satu objek yang terdeteksi sebagai pohon

kelapa sawit yang masuk dalam area deteksi polygon pada Gambar 4.36. yang telah diketahui ID, koordinat latitude, koordinat longitude, serta nilai confidence. Titik tengah dari setiap bounding box ini merepresentasikan koordinat latitude dan longitude.

Dalam mendapatkan gambaran bagaimana setiap objek memiliki titik koordinat tersebut, maka diacu pada satu objek pohon kelapa sawit pada Gambar 4.36. Gambar 4.37 merupakan nilai dari titik-titik awal hingga bisa ditampilkan titik koordinat latitude dan longitude objek pohon kelapa sawit pada Gambar 4.38. Perhitungan ini berlaku sama pada objek pohon kelapa sawit lainnya yang berhasil dideteksi pada sistem.

```
center lat long {'lat': -6.47484745212241, 'lng': 107.03319672664993}
z=0 {'x': 284.11249545006217, 'y': 132.61416744978712}
z=0 latlong {'lat': -6.474734588801301, 'lng': 107.0333732706901}
z=0 {'x': 284.11262099249075, 'y': 132.6140866762166}
yolo x=0.6543537378311157 y=0.3123517632484436 width=0.7570236921310425 height=0.42296940088272095 confidence=0.9860687255859375 label='oilpalm' x_center=None y_center=None
coco x=418.78639221191406 y=199.9951284790039 width=65.70877075195312 height=70.7952880859375 confidence=0.9860687255859375 label='oilpalm' x_center=451.640777587986 y_center=235.30277252197266
lat -6.474734588801301
lng 107.0333732706901
```

Gambar 4.38: Nilai Awal dari Titik-Titik pada Objek Pohon Kelapa Sawit

Berikut ini nilai yang dimiliki pada Tabel 4.27 agar tampak lebih jelas. Pada Gambar 4.37. sudah diketahui nilai-nilai dari format COCO.

Tabel 4.27: Nilai Titik Awal dari Titik-Titik pada Objek Pohon Kelapa Sawit

Komponen	Nilai
Center lat long	Latitude: -6.47484745212241; Longitude: 107.03319672664993
ID	5588
x yolo	0,6543537378311157
y yolo	0,312351763248443
Width yolo	0,7570236921310425
Height yolo	0,42296940088272095
Confidence	0,9860687255859375

Nilai pada format YOLO tercantum pada Tabel 4.27, yaitu diketahui x yolo, y yolo, width yolo, dan height yolo. Nilai-nilai ini digunakan untuk mengubah format YOLO menjadi format COCO.

4.7.5 Mengubah objek ke format COCO

Adanya perbedaan pada format YOLO dan COCO mengenai letak posisi dari objek, maka dibutuhkan konversi atau mengubah format YOLO ke format COCO. Karena format YOLO hanya merepresentasikan posisi relative dari titik tengah objek, maka diubah menjadi format COCO yang dimana merupakan titik piksel asli dari objek yang terdeteksi. Dengan menggunakan persamaan 20 hingga 25 pada subbab 3.8. nilai dari format YOLO ke COCO seperti pada Tabel 4.28.

Tabel 4.28: Nilai Komponen Format COCO

Komponen	Nilai
x coco	418,78639221191406
y coco	199,9051284790039
Width coco	65,70877075195312
Height coco	70,7952880859375
x_center	451,6407775878906
y_center	235,30277252197266
Confidence	0,9860687255859375

4.7.6 Mengubah Titik Piksel ke Piksel Koordinat

Setelah nilai format COCO didapatkan yang berupa nilai dari piksel setiap *bounding box* objek yang terdeteksi pohon kelapa sawit, maka berikutnya mengubah titik piksel menjadi titik koordinat dengan menggunakan *Mercator projection*.

Tahap ini digunakan untuk mendapatkan koordinat *easting* dan *northing*, dengan menggunakan persamaan 26 subabb 3.8. Koordinat *easting* dan *northing* mengacu pada koordinat geografis kartesius suatu titik. Koordinat *easting* mengacu pada koordinat x yang diproyeksikan atau jarak yang diukur ke arah timur (khatulistiwa), sedangkan koordinat *northing* disebut sebagai

koordinat y yang diproyeksikan atau jarak yang dikur ke arah utara. Koordinat ini memproyeksikan seperti map yang datar dengan ukuran map 256 x 256. Berdasarkan persamaan 25 subab 3.8, maka dapat dihitung titik piksel ke piksel koordinat sebagai berikut.

Diketahui:

$$\text{latitude} = -6.47484745212241; // (\varphi)$$

$$\text{longitude} = 107.03319672664993 // (\lambda)$$

$$\text{mapWidth} = 256;$$

$$\text{mapHeight} = 256;$$

Mendapatkan nilai koordinat x:

$$x = (\text{longitude}+180)*(256/360)$$

$$x = (107.03319672664993 + 180)*(256/360)$$

$$x = 204.11249545006217$$

Mengkonversi dari derajat ke radians:

$$\text{latRad} = \text{latitude} * \pi / 180;$$

$$\text{latRad} = -6.47484745212241 * 3.141592653589793 / 180$$

$$\text{latRad} = -0.1130074066$$

Mendapatkan nilai koordinat y: //MercN = Mercator North

$$\text{mercN} = \ln(\tan((\pi/4) + (\text{latRad}/2)));$$

$$\text{mercN} = \ln(\tan((3.141592653589793/4) + (-0.1130074066/2)))$$

$$\text{mercN} = \ln(0.72889446009)$$

$$\text{mercN} = \ln(0.89292855444)$$

$$\text{mercN} = -0.11324870753S$$

Maka, nilai y:

$$y = (\text{mapHeight}/2) - (\text{mapWidth} * \text{mercN} / (2 * \pi));$$

$$y = (256/2) - (256 * (-0.11324870753)) / (2 * 3.141592653589793))$$

$$y = 132.61416744978712$$

Maka, nilai piksel koordinat (x, y) diketahui senilai (204.11249545006217,

132.61416744978712). Nilai ini merupakan titik piksel koordinat dari 1 objek pohon kelapa sawit yang digunakan pada Gambar 4.37.

4.7.7 Mengubah Piksel Koordinat ke Latitude-Longitude

Tahap selanjutnya mengubah piksel koordinat ke latitude-longitude, yang dimana bahwa setiap objek pohon kelapa sawit yang terdeteksi memiliki lokasi masing-masing. Titik latitude-longitude merupakan titik koordinat geografis yang digunakan untuk mengetahui letak atau posisi dari suatu pohon, hasil ini dapat digunakan untuk pemantauan setiap pohon kelapa sawit yang ada di suatu area atau lahan tertentu.

Dengan menggunakan persamaan 26, hasil latitude dan longitude sebagai berikut.

Diketahui: math.pi ($\phi = 3.141592653589793$)

Mendapatkan Longitude:

$$\text{longitude} = (\text{point.x} - \text{origin.x}) / \text{self.pixelsPerLonDegree}_\text{—}$$

$$\text{longitude} = (\text{point.x} - (\text{mapWidth}/2)) / (\text{mapWidth}/360)$$

$$\text{longitude} = (204.11262099249-128)/(256/360)$$

$$\text{longitude} = 107.03337332706901$$

Mendapatkan Latitude:

$$\text{latRadians} = (\text{point.y} - \text{origin.y}) / -\text{self.pixelsPerLonRadian}_\text{—}$$

$$\text{latRadians} = (132.61416744978712-128)/-(256/(2^* \phi))$$

$$\text{latRadians} = -0.1132467250392861$$

$$\text{latitude} = \text{proj.radiansToDegrees}(2^*\text{math.atan}(\text{math.exp}(\text{latRadians})) - \text{math.pi} / 2)$$

$$\text{latitude} = \text{proj.radiansToDegrees}(2^*\text{math.atan}($$

$$2.718281^{-0.1132467250392861}$$

)) - math.pi/2)

latitude = proj.radiansToDegrees(2 * math.atan(0.892930324666564)) - math.pi/2)

latitude = proj.radiansToDegrees(2 * 0.7288954450137763 - math.pi/2)

latitude = proj.radiansToDegrees(2 * 0.7288954450137763 - (3.141592653589793/2))

latitude = radiansToDegrees(-0.11300543676)

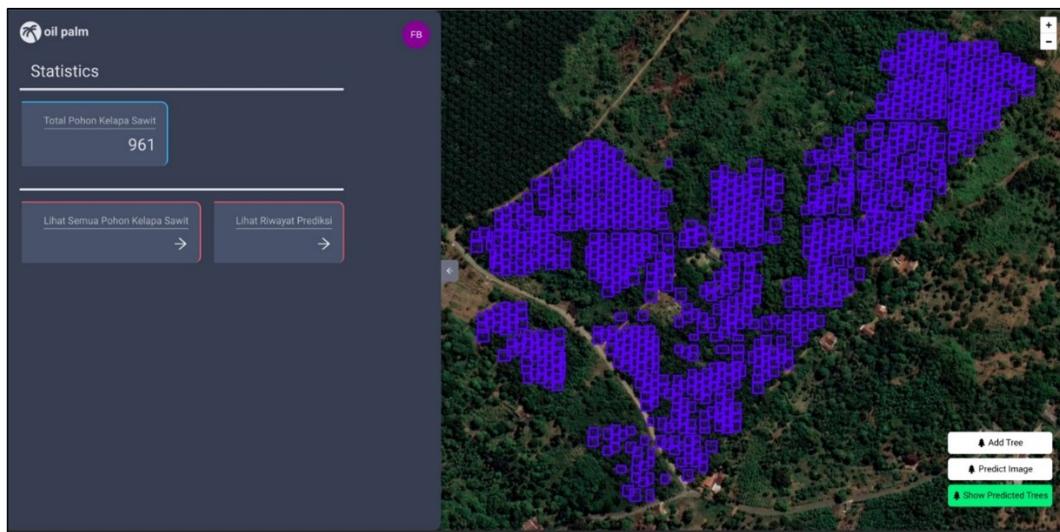
latitude = -0.11300543676/(math.pi /180)

latitude = -0.11300543676/(3.141592653589793/180)

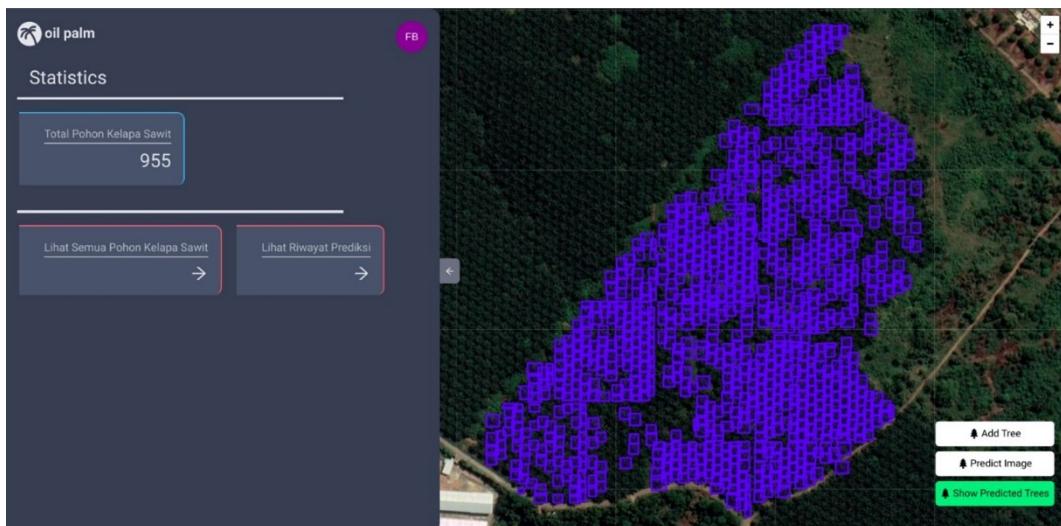
latitude = -6.474734588801301

Maka, titik koordinat latitude dan longitude diketahui senilai (-6.474734588801301, 107.03337332706901). Nilai ini merupakan titik koordinat geografis dari 1 objek pohon kelapa sawit yang digunakan pada Gambar 4.37, dan terbukti bahwa penerapan pada sistem berhasil untuk mendapatkan lokasi setiap pohon kelapa sawit.

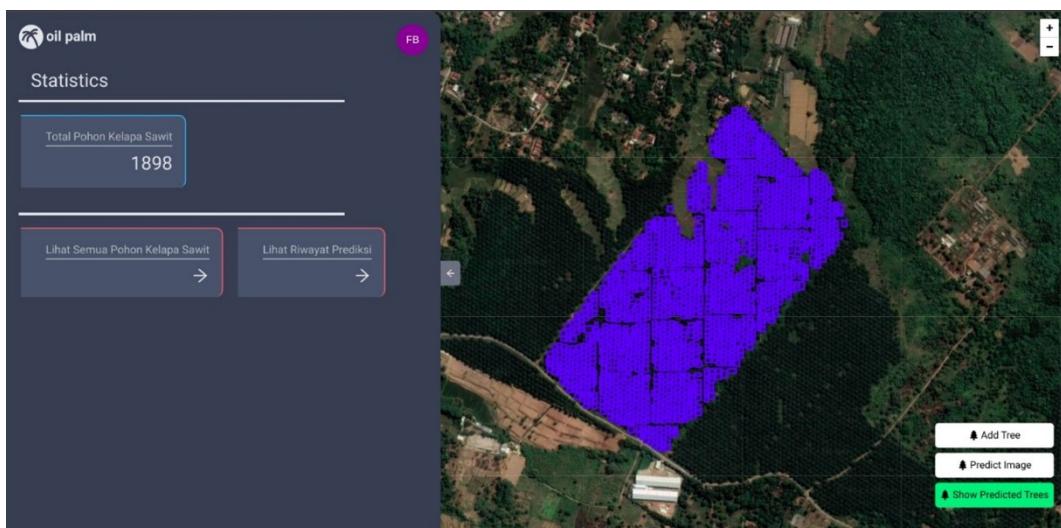
Selanjutnya dilakukan ujicoba dengan 4 blok pada lahan area di Kebun Pendidikan Kelapa Sawit IPB-Cargil, Jonggol, seperti pada Gambar 4.39 blok 1, Gambar 4.40 blok 2, Gambar 4.41 blok 3, dan Gambar 4.42 blok 4.



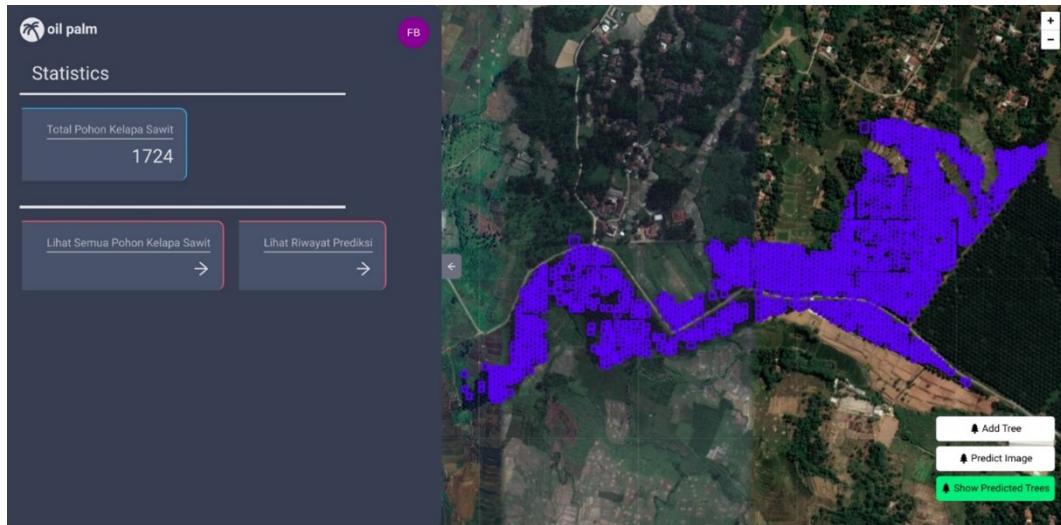
Gambar 4.39: Hasil Uji Blok 1



Gambar 4.40: Hasil Uji Blok 2



Gambar 4.41: Hasil Uji Blok 3



Gambar 4.42: Hasil Uji Blok 4

Tabel 4.29 merupakan akumulasi hasil uji penerapan deteksi dan menghitung pohon kelapa sawit dengan menggunakan sistem yang diterapkan. Berdasarkan hasil studi lapang dan wawancara bahwa pohon kelapa sawit awal menurut perhitungan tanam sebanyak 5670 pohon. Hasil akumulasi dan persentase keberhasilan menggunakan sistem dari citra yang berhasil dideteksi seperti pada Tabel 4.29.

Tabel 4.29: Hasil Presentase Prediksi Pohon Kelapa Sawit pada 4 Blok

Blok	Pohon Yang Terdeteksi
1	961
2	955
3	1898
4	1724
Jumlah	5538
Persentase % (Terdeteksi/Perhitungan Tanam (5670))	97.67%

Gambar 4.39 hingga Gambar 4.42 merupakan bukti penerapan yang telah dilakukan dengan menggunakan *convolutional neural network* dan perhitungan lokasi dari setiap pohon sawit untuk dapat dimonitoring dengan

menggunakan *Mercator projection*. *Bounding box* berwarna merupakan pohon kelapa sawit yang berhasil terdeteksi secara otomatis. Pada penelitian ini juga dihasilkan kebaruan atau novelty bahwa dihasilkan suatu identifier berdasarkan koordinat latitude-longitude untuk setiap satu pohon kelapa sawit yang terdeteksi pada sistem, sehingga bisa menjadi identifier yang unik untuk dapat bisa dimonitoring setiap pohon kelapa sawit tersebut.

BAB V

PENUTUP

5.1 Kesimpulan

Berdasarkan permasalahan dan hasil penelitian yang telah dilakukan, maka dapat ditarik kesimpulan secara umum bahwa purwarupa sistem berbasis web menggunakan model OBIA dan *Deep Learnig* (DL) yang telah dihasilkan dari penelitian ini mampu mendeteksi dan menghitung tegakkan kelapa sawit dalam satu wilayah tertentu secara cepat, akurat, dan *up-to-date*. Secara khusus penelitian ini dapat disimpulkan sebagai berikut:

1. Algoritma *template matching* dan *Balanced Iterative Reducing and Clustering Using Hierarchies* (BIRCH) dengan pendekatan OBIA berhasil digunakan untuk dapat membuat dataset otomatis dengan diberikan anotasi atau pelabelan 'oil palm' pada 205 citra dataset primer dengan waktu yang dibutuhkan untuk anotasi dataset otomatis selama 30 menit 24 detik. Berdasarkan ujicoba dengan 10 data citra dihasilkan 89,90% berhasil dideteksi sebagai dataset 'oil palm'.
2. Berdasarkan hasil evaluasi performance yang digunakan dari YOLOv5, YOLOv6, dan YOLOv7 dengan menggunakan Google Colab Pro dan DGX-A-100 Universitas Gunadarma bahwa model dengan algoritma YOLOv7 pada pelatihan lebih baik dibandingkan YOLOv5 dan YOLOv6 dengan hasil *Precision*, *Recall*, *F1-Score*, dan *best mAP @0.5*. pada pelatihan secara berurutan adalah 0,970, 0,976, 0,973, dan 0,993, sedangkan pada hasil pengujian mengalami peningkatan yaitu 0.984, 0.989, 0.986, dan 0.997. Hal ini menunjukkan bahwa model dapat menemukan dan mempelajari pola dari dataset dengan baik, sehingga dapat mengenali

pola pada data pengujian, dan pendugaan ketelitian tegakkan pohon kelapa sawit dengan akurasi best mAP 0,997 atau 97,70% lebih baik dibandingkan dengan penelitian sebelumnya. Berdasarkan segi waktu komputasi yang dibutuhkan untuk pelatihan dan pengujian, DGX-A-100 Universitas Gunadarma lebih baik dibanding Google Colab Pro, tercatat waktu pelatihan pada YOLOv7 sebesar 2948 detik, lebih baik dibandingkan dengan Google Colab Pro yaitu 4847 detik, sedangkan pada proses pengujian lebih besar pada Google Colab Pro sebesar 53 detik, berbeda 1 detik dengan DGX-A-100 Universitas Gunadarma.

3. Hasil purwarupa sistem berhasil dibuat dan diterapkan dengan model CNN dari algoritma YOLOv7 untuk mendeteksi dan menghitung pohon kelapa sawit, serta menghasilkan identifier berdasarkan titik koordinat latitude dan longitude untuk setiap pohon kelapa sawit berdasarkan citra satelit yang terintegrasi dengan Google Maps API, sehingga bisa menjadi identifier yang unik untuk memonitoring pohon kelapa sawit. Berdasarkan hasil uji coba pada 4 blok area Kebun Pendidikan dan Penelitian Kelapa Sawit IPB-Cargil, maka hasil presentase berhasil dideteksi objek kelapa sawit sebesar 97.67%.

5.2 Saran

Berdasarkan hasil yang dicapai dari hasil penelitian bahwa peneliti mengakui bahwa masih dalam lingkup keterbatasan dalam mengumpulkan dataset primer yang dapat digunakan sebagai pembelajaran oleh mesin, sehingga dapat menambah area dan cakupan dataset primer untuk dapat melatih model lebih baik. Data yang terwakili perkebunan kelapa sawit masih sedikit, dan dapat digunakan dengan penelitian dengan perkebunan kelapa sawit dengan tanah gambut. Penelitian ini dapat dikembangkan dan diterapkan pada

model CNN lainnya, seperti YOLOv8 atau *Region Based Convolutional Neural Network* (R-CNN) untuk mendeteksi objek dan mendukung manajemen perkebunan, seperti pemantauan perkebunan secara otomatis, mendeteksi dan menghitung pohon kelapa sawit, nutrisi pohon, pemetaan, dan pemupukan. Dapat meningkatkan performa model untuk dapat membuat waktu komputasi pada pelatihan dan pengujian lebih cepat. Pengembangan purwarupa sistem sudah berhasil mendeteksi, menghitung, dan mendapatkan letak atau titik koordinat dari setiap objek kelapa sawit yang berhasil dideteksi, sehingga selanjutnya, dapat menjadi acuan dan membantu para *stakeholder* untuk pemantauan dan pengambilan keputusan.

DAFTAR PUSTAKA

- Abid Haleem, Mohd Javaid, M. A. Q. R. S. (2022), ‘Understanding the role of digital technologies in education: A review’, **3**, 275–285.
- Akoglu, H. (2018), ‘User's guide to correlation coefficients’, *Turkish Journal of Emergency Medicine* **18**(3), 91–93.
- Altunian, G. (2021), ‘What Is Signal-to-Noise Ratio and Why Does It Matter?’, URL: <https://www.lifewire.com/signal-to-noise-ratio-3134701>. (Diakses tanggal 07 Juli 2022).
- Annur, C. M. (2023), ‘Demografi Pengguna Televisi di Indonesia Berdasarkan Usia (2022)’, URL: <https://databoks.katadata.co.id/datapublish/2022/12/09/survei-nielsen-indonesia-majoritas-pengguna-television-di-ri-berusia-50-tahun-ke-atas>. (diakses tanggal 20 Januari 2023).
- Apple-Inc. (2012), ‘Compressor 4 - User Manual, MPEG-2 Reference Information’, URL: <https://help.apple.com/compressor/mac/4.0/en/compressor/usermanual/#chapter=18%26section=5>. (Diakses tanggal 25 April 2021).
- Baek, S. & Kim, B. J. (2008), Digital Multimedia Broadcasting (DMB) in Korea, in ‘Multimedia Technologies’, IGI Global, pp. 1377–1390.
- Centauri, A. (2013), ‘Why Decibels are used to measure Signal to Noise Ratio?’, URL: <https://electronics.stackexchange.com/q/82823>. (Diakses tanggal 21 Juli 2022).

- Cheon, M. & Lee, J.-S. (2016), ‘On ambiguity of objective image quality assessment’, *Electronics Letters* **52**(1), 34–35.
- Cheon, M., Vigier, T., Krasula, L., Lee, J., Callet, P. L. & Lee, J.-S. (2021), ‘Ambiguity of objective image quality metrics: A new methodology for performance evaluation’, *Signal Processing: Image Communication* **93**, 116150.
- Chochliouros, I. (2009), ‘Digital Video Broadcasting (DVB) Evolution, Encyclopedia of Multimedia Technology and Networking, Second Edition’, *IGI Global Science and Technology*.
- Dai, L., Wang, Z. & Yang, Z. (2012), ‘Next-generation digital television terrestrial broadcasting systems: Key technologies and research trends’, *IEEE Communications Magazine* **50**(6), 150–158.
- Dihin, R. A., Hamza, N. R. & Toman, Z. H. (2020), ‘Full-Reference Facial Image Quality Assessment and Identification by Two Proposed Measures’, *Journal of Southwest Jiaotong University* **55**(2), 1.
- Ding, K., Ma, K., Wang, S. & Simoncelli, E. P. (2021), ‘Comparison of Full-Reference Image Quality Models for Optimization of Image Processing Systems’, *International Journal of Computer Vision* **129**(4), 1258–1281.
- Dost, S., Saud, F., Shabbir, M., Khan, M. G., Shahid, M. & Lovstrom, B. (2022), ‘Reduced reference image and video quality assessments: review of methods’, *EURASIP Journal on Image and Video Processing* **2022**(1).
- DTVstatus (2017), ‘DTV Status Compares The Worldwide Digital Broadcasting Systems ATSC, DTMB, DVB-T/DVB-T2, and ISDB-T’, URL: <http://en.dtvstatus.net/>. (Diakses tanggal 27 Juni 2021).
- DVBProject (2013), Digital Video Broadcasting (DVB) Bluebook A116-2, in ‘Digital Video Broadcasting’, Digital Video Broadcasting.

- Ebrahimi, T. & Pereira, F. (2002), *The MPEG-4 Book*, illustrated edition edn, Prentice Hall PTR.
- El-Saba, M. (2018), *Telecommunication Systems and Data Networks*, third edn.
- Emmet (2022), ‘Stream TV from your Raspberry Pi with Tvheadend’, URL: <https://pimylifeup.com/raspberry-pi-tvheadend/>. (Diakses tanggal April 2022).
- Ferzli, R. & Karam, L. (2009), ‘A no-reference objective image sharpness metric based on the notion of just noticeable blur (JNB)’, *IEEE Transactions on Image Processing* **18**(4), 717–728.
- Fischer, W. (2010), *Digital Video and Audio Broadcasting Technology*, third edn, Springer Berlin Heidelberg.
- Foundation, T. (2015), ‘Tvheadend 4.0 User Guide’, URL: https://docs.tvheadend.org/about_project/. (Diakses tanggal 5 Oktober 2021).
- Frost, P. (2014), ‘How is signal strength calculated?’, Electrical Engineering Stack Exchange. URL: <https://electronics.stackexchange.com/q/98691> (version: 2014-02-04). (Diakses tanggal 30 Oktober 2022).
- Gultom, A. D. (2018), ‘Digitalisasi Penyiaran Televisi di Indonesia’, *Buletin Pos dan Telekomunikasi* **16**(2), 91.
- Havinal, R. (2020), ‘The role and potential of information technology in agricultural development’.
- Hendrickson, J. (2022), ‘How to Check Your Wi-Fi Signal Strength’, URL: <https://www.howtogeek.com/426642/how-to-check-your-wi-fi-signal-strength/>. (Diakses tanggal 30 Oktober 2022).

- Hoelzer, S. (2005), ‘MPEG-2 overview and MATLAB codec project’, URL: https://users.cs.cf.ac.uk/Dave.Marshall/Multimedia/Lecture_Examples/Compression/mpeg2.pdf. (Diakses tanggal 25 April 2021).
- Hou, H.-A. (2010), ‘DVB-T2 technology overview - Industrial Technology Research Institute’, URL: https://www.ncc.gov.tw/chinese/files/10051/1661_2.pdf. (Diakses tanggal 2 Oktober 2021).
- Huynh-Thu, Q. & Gha, M. (2009), ‘No-reference temporal quality metric for video impaired by frame freezing artefacts’.
- ISO & IEC (2022), ‘13818-1:2022 Information technology - Generic coding of moving pictures and associated audio information - Part 1: Systems’, URL: <https://www.iso.org/obp/ui/#0023iso:std:83239:en>. (Diakses tanggal 25 Juli 2022).
- ITU-R BT500 (2019), *BT500-14: Methodology for the Subjective Assessment of the Quality of Television Pictures*, International Telecommunication Union - Radiocommunication, International Telecommunication Union - Radiocommunication.
- ITU-R P.910 (2022), ‘International telecommunication union - telecommunication standarization’.
- Jason Dedrick, Kenneth L. Kraemer, E. S. (2014), ‘Information technology and productivity in developed and developing countries’, pp. 97–122.
- Kennedy, G. & Davis, B. (2009), *Electronic Communication System, Fourth Edision*, 4 edn, Tata McGraw Hill Education Private Limited.
- Kessel, R. (2018), ‘Why isn’t signal-to-noise ratio expressed in a percentage instead of in dB?’.

- Kieser, R., Reynisson, P. & Mulligan, T. J. (2005), ‘Definition of signal-to-noise ratio and its critical role in split-beam measurements’, *ICES Journal of Marine Science* **62**(1), 123–130.
- Kominfo, H. (2022), ‘Update Perkembangan Analog Switch Off (ASO)’, URL: https://www.kominfo.go.id/content/detail/43905/siaran-pers-no-346hmkominfo082022-tentang-update-perkembangan-analog-switch-off-aso/0/siaran_pers. (Diakses tanggal 2 Oktober 2022).
- Kruger, L. G. & Guerrero, P. F. (2002), *Digital Television: An Overview*, Novinka Books.
- Kumar, S. & Prajapati, P. (2015), ‘A Review Paper on Image Quality Assessment Metrics’, *International Journal of Advance Research in Computer Science and Management* **2**, 129–132.
- Kusuma, T. M. (2005), A Perceptual-based Objective Quality Metric for Wireless Imaging, phdthesis, Curtin University of Technology.
- Lee, Y.-S. & Kook, J. (2019), ‘Integrated DVB-X2 Receiver Architecture with Common Acceleration Engine’, *Applied Sciences* **9**(19), 3983.
- Lundstrom, L.-I. (2006), *Understanding Digital Television*, Focal Press.
- Markoulidakis, I., Rallis, I., Georgoulas, I., Kopsiaftis, G., Doulamis, A. & Doulamis, N. (2021), ‘Multiclass confusion matrix reduction method and its application on net promoter score classification problem’, *Technologies* **9**(4), 81.
- Martínez-Rach, M. O., Piñol, P., López, O. M., Malumbres, M. P., Oliver, J. & Calafate, C. T. (2014), ‘On the Performance of Video Quality Assessment Metrics under Different Compression and Packet Loss Scenarios’, *The Scientific World Journal* **2014**, 1–18.

- Marziliano, P., Dufaux, F., Winkler, S. & Ebrahimi, T. (2002), A no-reference perceptual blur metric, *in* ‘Proceedings. International Conference on Image Processing’, IEEE.
- Matthews, V. O., Idiaké, S. U., Noma-Osaghae, E. & Nwukor, F. (2018), ‘Comparative Analysis of the Effect of Distance on Signal Strength using Selected Communication Media Comparative Analysis of the Effect of Distance on Signal Strength using Selected Communication Media’, *International Journal of Advances in Scientific Research and Engineering* **4**(8), 82–91.
- M.H. Pinson, S. W. (2004), ‘A New Standardized Method for Objectively Measuring Video Quality’, *IEEE Transactions on Broadcasting* **50**(3), 312–322.
- Mittal, A., Moorthy, A. K. & Bovik, A. C. (2012), ‘No-reference image quality assessment in the spatial domain’, *IEEE Transactions on Image Processing* **21**(12), 4695–4708.
- Mustafa, W., Yazid, H., Jaafar, M., Zainal, M., Abdul Nasir, A. S. & Mazlan, N. (2017), ‘A Review of Image Quality Assessment (IQA): SNR, GCF, AD, NAE, PSNR, ME’, *Journal of Advanced Research in Computing and Applications* **7**, 1–7.
- Nguyen, H. H. & Shwedyk, E. (2009), *A First Course in Digital Communications*, Cambridge University Press.
- Ong, E., Lin, W., Lu, Z., Yang, X., Yao, S., Pan, F., Jiang, L. & Moschetti, F. (2003), ‘A no-reference quality metric for measuring image blur’.
- Pauliks, R., Tretjaks, K., Belahs, K. & Pauliks, R. (2013), A survey on some measurement methods for subjective video quality assessment, *in* ‘2013 World Congress on Computer and Information Technology (WCC-IT)’, IEEE.

- Powers, D. M. W. (2020), ‘Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation’.
- Reimers, U. (1998), ‘Digital video broadcasting’, *IEEE Communications Magazine* **36**(6), 104–110.
- Reimers, U. (2006), ‘DVB The Family of International Standards for Digital Video Broadcasting’, *Proceedings of the IEEE*. **94**(1), 173–182.
- Robin, M. & Poulin, M. (2000), *Digital Televisin Fundamentals 2nd Edition*, McGraw-Hill Professional.
- Rodriguez, D. Z., Rosa, R., AbrahÃ£o, J., Bressan, G. & Costa Alfaia, E. (2014), ‘Video Quality Assessment in Video Streaming Services Considering User Preference for Video Content’, *IEEE Transactions on Consumer Electronics* **60**, 436–444.
- Salomon, D. & Motta, G. (2010), *Handbook of Data Compression*, fifth edn, Springer London.
- Seshadrinathan, K., Soundararajan, R., Bovik, A. C. & Cormack, L. K. (2010), ‘Study of Subjective and Objective Quality Assessment of Video’, *IEEE Transactions on Image Processing* **19**(6), 1427–1441.
- Shahreza, H. O., Amini, A. & Behroozi, H. (2019), ‘No-reference video quality assessment using recurrent neural networks’.
- Sheikh, H., Sabir, M. & Bovik, A. (2006), ‘A Statistical Evaluation of Recent Full Reference Image Quality Assessment Algorithms’, *IEEE Transactions on Image Processing* **15**(11), 3440–3451.
- Suryanto (2021), ‘Indonesia mulai masuki era TV Digital’, URL: https://www.kominfo.go.id/content/detail/3400/indonesia-mulai-masuki-era-tv-digital/0/sorotan_media. (Diakses tanggal 26 April 2021).

- Tom (2022), ‘What Is The Difference Between TV Signal Strength and Quality?’, URL: <https://www.smartaerials.co.uk/blog/what-is-the-difference-between-tv-signal-strength-quality>. (Diakses tanggal 30 Juli 2022).
- Varga, D. (2021), ‘No-Reference Image Quality Assessment with Global Statistical Features’, *Journal of Imaging* **7**(2), 29.
- Victoria, A. (2020), ‘Information Technology’.
- Wang, Z. (2003), ‘Objective Video Quality Assessment, The Handbook of Video Databases: Design and Applications’, *B.Furht and O. Marqure, ed., CRC Press.* pp. 1041–1078.
- Wang, Z., Bovic, A. & Hamid, S. (2002), ‘No-reference perceptual quality assessment of JPEG compressed images’, *IEEE International Conference on Image Processing* .
- Welvaert, M. & Rosseel, Y. (2013), ‘On the Definition of Signal-To-Noise Ratio and Contrast-To-Noise Ratio for fMRI Data’, *PLoS ONE* **8**(11), e77089.
- Xu, J., Tao, C., Mao, X., Lu, X., Bao, J., Drobe, B. & Chen, H. (2021), ‘Blur Detection Sensitivity Increases in Children Using Orthokeratology’, *Frontiers in Neuroscience* **15**.
- Ya’acob, N., Tajudin, N., Zamiri, M. M. . M., Yusof, A. L., Sarnin, S. S. & Idris, A. (2019), ‘Measurement of digital video broadcasting-second generation terrestrial (DVB-T2) signal in Kuala Lumpur, Malaysia’, *Indonesian Journal of Electrical Engineering and Computer Science* **13**(3), 1286.
- yen Ong, C., Song, J., Pan, C. & Li, Y. (2010), ‘Technology and standards of digital television terrestrial multimedia broadcasting, Topics in Wireless Communications’, *IEEE Communications Magazine* **48**(5), 119–127.

Yin, R. K. (n.d.), *Case Study Research and Applications : Design and Methods*,
SAGE Publications, Inc.

LAMPIRAN

1. Program Memperoleh Kualitas Sinyal dari TVheadend

Program ditulis dalam bahasa python sebagai berikut.

Program here ...

2. Program Pengukuran Metrik Objektif NR-IQA Blockiness

Program ditulis dalam bahasa python sebagai berikut.

Program here ...

3. Tabel hasil pengukuran obyektif dan subyektif pada dataset

Tabel 5.1: A sample long table.

nama	m_blok	m_blur	m_grad	std_grad	move_fr	m_temp	mos	class
000.mp4	87	52	4.84	4.76	95	75	3.8	3
001.mp4	88	55	0.29	0.25	100	75	4.64	5
002.mp4	73	1	-0.26	0.12	50	90	2.24	2
003.mp4	79	8	3.85	8.5	71	90	1.6	1
004.mp4	80	10	0.33	4.94	75	85	1.84	1
005.mp4	73	2	3.67	8.12	71	90	2.04	2
006.mp4	72	42	-0.51	0.59	49	90	1.48	1
007.mp4	48	13	3.9	9.97	71	35	1	1
008.mp4	72	10	-0.35	0.63	50	90	1.64	1
009.mp4	86	52	1.63	0.78	98	80	3.96	3
010.mp4	78	9	0	0.31	100	85	4.2	4
011.mp4	77	15	1.63	6.58	73	75	3.2	3
012.mp4	73	4	-0.47	1.67	50	90	1.56	1
013.mp4	78	23	0.31	0.25	100	75	4.2	4
014.mp4	82	15	0.84	0.68	99	75	3.84	3
015.mp4	66	30	11.8	17.7	63	65	1.16	1

016.mp4	77	16	-0.09	0.25	50	75	1.76	1
017.mp4	67	9	-0.23	0.29	50	90	1.96	2
018.mp4	69	8	-0.44	0.56	50	85	1.64	1
019.mp4	95	52	3.91	1.32	96	90	4.32	5
020.mp4	72	2	0.38	1.98	75	80	1.96	2
021.mp4	86	52	5.45	7.42	70	100	4.16	4
022.mp4	87	52	2.06	3.89	73	85	4.68	5
023.mp4	85	52	0.83	0.98	99	100	3.2	3
024.mp4	82	8	1.33	0.15	99	75	4.36	5
025.mp4	82	1	27.66	30.23	47	50	2	2
026.mp4	76	40	0.07	0.1	100	85	4.28	4
027.mp4	88	70	0.22	0.08	100	85	4.6	5
028.mp4	89	52	0.85	0.46	99	90	4.36	5
029.mp4	72	61	-3.1	3.11	47	80	1.84	1
030.mp4	73	46	0.06	0.16	100	90	3.72	3
031.mp4	84	85	2.53	3.23	97	85	4.32	5
032.mp4	73	69	-0.9	0.27	49	90	1.4	1
033.mp4	72	80	0.21	0.1	100	90	3.56	3
034.mp4	72	34	0.33	0.24	100	100	4.12	4
035.mp4	79	15	1.29	2.43	99	90	1.76	1
036.mp4	74	5	-0.75	0.61	49	80	1.2	1
037.mp4	85	52	1.25	0.19	99	85	4.08	3
038.mp4	73	21	4.74	6.49	70	90	1.36	1
039.mp4	73	12	0.68	1.01	99	85	2.24	2
040.mp4	86	50	0.86	0.18	99	65	4.56	5
041.mp4	85	52	0.14	0.12	100	90	4.68	5
042.mp4	81	35	3.68	9.04	71	90	2.36	2
043.mp4	87	52	12.89	1.4	87	70	3.88	3
044.mp4	83	18	0.42	0.37	100	75	3.88	3
045.mp4	75	2	1.67	1.68	98	85	4.16	4
046.mp4	88	62	0.61	0.29	99	85	4.28	4
047.mp4	83	35	0.96	1.2	99	85	2.16	2
048.mp4	83	19	2.03	1.8	98	80	3.92	3
049.mp4	71	4	0.14	0.44	100	90	2.48	2
050.mp4	80	97	0.06	0.36	100	75	4.76	5
051.mp4	75	15	4.21	10.84	71	85	2.04	2
052.mp4	90	52	0.81	0.7	99	80	4.32	5
053.mp4	75	41	14.21	21.08	61	70	1.64	1
054.mp4	87	52	1.15	1.15	99	75	4.2	4
055.mp4	82	14	0.73	0.44	99	85	4.52	5
056.mp4	89	52	2.63	1.55	97	90	4.52	5
057.mp4	83	12	0.26	0.68	100	80	4.68	5
058.mp4	71	4	-0.19	0.22	50	100	3.96	3

059.mp4	78	10	1.16	0.17	99	90	3.72	3
060.mp4	88	52	3.63	4.03	96	100	3.88	3
061.mp4	74	10	-4.59	4.65	45	80	1.16	1
062.mp4	65	15	-4.4	3.07	46	80	1.28	1
063.mp4	66	6	0.01	0.03	100	75	4.56	5
064.mp4	74	50	-0.49	0.82	50	80	1.28	1
065.mp4	80	13	0.11	0.38	100	85	4.2	4
066.mp4	81	6	0.57	0.12	99	85	4.2	4
067.mp4	79	38	12.31	30.32	63	75	2.92	2
068.mp4	85	52	1.66	0.08	98	90	3.64	3
069.mp4	86	52	1.21	1.16	99	90	4.24	4
070.mp4	77	8	0.19	0.18	100	90	3.92	3
071.mp4	73	12	0.37	0.86	100	70	2.92	2
072.mp4	79	48	0.22	0.28	100	80	4.24	4
073.mp4	94	52	4.01	4.5	96	85	4.32	5
074.mp4	85	52	0.5	0.5	100	75	4.64	5
075.mp4	89	52	1.03	0.16	99	70	4.72	5
076.mp4	72	48	1.79	2.97	98	90	1.96	2
077.mp4	82	18	0.53	2.85	74	80	2.24	2
078.mp4	88	52	0.25	0.27	100	75	4.48	5
079.mp4	88	68	5.7	1.91	94	90	4.12	4
080.mp4	83	1	1.42	2.47	99	90	2.6	2
081.mp4	80	5	-0.88	2.69	49	90	2.44	2
082.mp4	87	52	4.95	6.8	70	85	4.2	4
083.mp4	76	60	0.38	0.26	100	80	3.56	3
084.mp4	88	52	6.71	9.62	68	90	4	3
085.mp4	87	52	3.86	1.54	96	80	4.12	4
086.mp4	79	15	0.31	0.17	100	75	4.12	4
087.mp4	79	29	0.07	1.13	100	85	2.64	2
088.mp4	78	21	0.1	0.52	100	80	4.2	4
089.mp4	83	12	1.75	1.51	98	90	4.36	5
090.mp4	88	61	8.22	5.61	92	90	3.96	3
091.mp4	70	14	-0.25	0.51	50	80	3.56	3
092.mp4	74	12	2.21	2.6	98	75	2.12	2
093.mp4	67	40	-0.28	0.59	50	80	2	2
094.mp4	68	50	-0.23	0.38	50	85	1.6	1
095.mp4	76	44	1.3	2.92	74	65	1.04	1
096.mp4	95	52	4	1.57	96	90	4.16	4
097.mp4	80	0	0.29	0.46	100	90	4.68	5
098.mp4	82	9	0.1	0.26	100	75	4.2	4
099.mp4	94	70	2.68	1.82	97	90	4.2	4
100.mp4	87	76	2.12	2.95	98	90	4.68	5
101.mp4	79	65	-0.47	0.58	50	85	2.2	2

102.mp4	92	52	9.33	16.3	66	100	4.12	4
103.mp4	78	21	1.86	7.08	73	85	2.32	2
104.mp4	86	52	1.36	2.3	99	75	4.12	4

Program here ...



RIWAYAT HIDUP

IDENTITAS DIRI

Nama : Guntur Eka Saputra
Tempat/Tanggal Lahir : Jakarta, 30 Januari 1992
Jenis Kelamin : Laki-laki
Pekerjaan : Dosen
NIP / NIDN : 171011 / 0314049301
Jabatan Fungsional : Lektor
Unit Kerja : Laboratorium Pengembangan Teknik Elektro
Institusi : Universitas Gunadarma
Telp : 085782382323
Email Staff : sandy_sr@staff.gunadarma.ac.id
Email Pribadi : sunariveru@gmail.com

PENDIDIKAN FORMAL

Tahun	Pendidikan	Institusi
1999	TK	TK Aisyiah 37 Jakarta
2005	SD	SD N Menteng 02 Jakarta
2008	SMP	SMP Negeri 216 Jakarta
2011	SMA	SMA Negeri 28 Jakarta
2015	S1 Teknik Elektro	Universitas Gunadarma
2016	S2 Teknik Elektro	Universitas Gunadarma
—	S3 Teknologi Informasi	Universitas Gunadarma

PENGALAMAN DILUAR PENDIDIKAN FORMAL

Tahun	Jabatan
2016	<i>Linneaus Palme Scholarship Swedia - Student Exchange</i>
2017 -	Dosen Tetap Universitas Gunadarma
2017 -	Reviewer Nasional Program Kreatifitas Mahasiswa - Kemdikbud
2017 -	Pilot Drone Licences Asosiasi Pilot Drone Indonesia - Kemenhub

PUBLIKASI ILMIAH

Tahun	Penulis-Judul	Keterangan
2020	Sandy Suryo Prayogo, Tubagus Maulana Kusuma - Analisis Sensitivitas Video MPEG-4 berdasarkan Struktur Frame Pada Transmisi DVB-T	Jurnal Ilmiah Informatika Komputer 25 (2), 86-97 , Universitas Gunadarma
2023	-	-
2023	-	-