

**SISTEM PENDETEKSI DAN PENGHITUNG POLEN HIDUP DAN MATI  
PADA TANAMAN KELAPA SAWIT MENGGUNAKAN ALGORITMA  
YOLO V5 BERBASIS *ARTIFICIAL INTELLIGENCE***

**(Skripsi)**

**Oleh :**

**AHLUL HALLIWAL AQDI  
NPM. 1915031036**



**PROGRAM STUDI TEKNIK ELEKTRO  
JURUSAN TEKNIK ELEKTRO  
FAKULTAS TEKNIK  
UNIVERSITAS LAMPUNG  
BANDAR LAMPUNG  
2023**

## **ABSTRAK**

### **SISTEM PENDETEKSI DAN PENGHITUNG POLEN HIDUP DAN MATI PADA TANAMAN KELAPA SAWIT MENGGUNAKAN ALGORITMA YOLO V5 BERBASIS *ARTIFICIAL INTELLIGENCE***

Oleh

**AHLUL HALLIWAL AQDI**

Teknologi saat ini telah mengalami perkembangan yang sangat pesat dan memiliki peranan penting dalam perubahan pola kehidupan manusia. Salah satu perkembangan teknologi itu adalah *Artificial Intelligence* (AI). Dalam industri pertanian, juga perlu adanya implementasi AI dalam mendukung pertanian berkelanjutan. Salah satunya yaitu implementasi AI pada industri kelapa sawit di Indonesia. Pada industri kelapa sawit, kebutuhan akan bahan tanam kelapa sawit yang berkualitas tinggi sangat diperlukan untuk menunjang perkembangan perkebunan kelapa sawit di Indonesia. Salah satu upaya yang dilakukan dalam menghasilkan bahan tanam kelapa sawit berkualitas tinggi yaitu dengan melakukan persilangan atau penyerbukan buatan. Keberhasilan dari proses penyerbukan buatan ini dipengaruhi oleh polen dari bunga jantan tanaman kelapa sawit, sehingga perlu diketahui viabilitas dari polen tersebut. Proses yang dilakukan dalam mengetahui viabilitas polen ini menggunakan algoritma YOLO V5. Dalam hal ini, terdapat beberapa model yang dilatih untuk mendapatkan nilai presisi yang paling tinggi diantaranya, YOLO V5S dengan nilai presisi 0,9121, YOLO V5M dengan nilai presisi 0,9183, YOLO V5L dengan nilai presisi 0,9204, serta YOLO V5X dengan nilai presisi 0,9196. Dari hasil pelatihan model tersebut, YOLO V5L memiliki nilai presisi tertinggi sehingga digunakan untuk mendeteksi dan menghitung viabilitas polen. Hasil dari pengujian ini didapatkan nilai presisi 99,59%, *recall* 97,88%, dan akurasi 97,80%.

Kata kunci: *Artificial Intelligence*, polen, algoritma YOLO V5.

## **ABSTRACT**

### **A SYSTEM FOR DETECTING AND COUNTING LIVE AND DEAD POLLEN IN OIL PALM PLANTS USING THE YOLO V5 ALGORITHM BASED ON ARTIFICIAL INTELLIGENCE**

**By**

**AHLUL HALLIWAL AQDI**

Technology today has experienced very rapid development and has an important role in changing the pattern of human life. One of the technological developments is Artificial Intelligence (AI). In the agricultural industry, there is also a need for the implementation of AI in supporting sustainable agriculture. One of them is the implementation of AI in the palm oil industry in Indonesia. In the palm oil industry, the need for high-quality palm planting material is needed to support the development of oil palm plantations in Indonesia. One of the efforts made in producing high-quality oil palm planting material is by crossing or artificial pollination. The success of this artificial pollination process is influenced by pollen from male flowers of oil palm plants, so it is necessary to know the viability of the pollen. The process carried out to know the viability of this pollen uses the YOLO V5 algorithm. In this case, several models are trained to get the highest precision value including, YOLO V5S with a precision value of 0.9121, YOLO V5M with a precision value of 0.9183, YOLO V5L with a precision value of 0.9204, and YOLO V5X with a precision value of 0.9196. From the model training results, YOLO V5L has the highest precision value so it is used to detect and calculate pollen viability. The results of this test obtained 99.59% precision, 97.88% recall, and 97.80% accuracy.

**Keyword:** Artificial Intelligence, pollen, YOLO V5 algorithm.

**SISTEM PENDETEKSI DAN PENGHITUNG POLEN HIDUP DAN  
MATI PADA TANAMAN KELAPA SAWIT MENGGUNAKAN  
ALGORITMA YOLO V5 BERBASIS *ARTIFICIAL INTELLIGENCE***

**Oleh**

**Ahlul Halliwal Aqdi**

**Skripsi**

**Sebagai Salah Satu Syarat Untuk Mendapatkan Gelar  
SARJANA TEKNIK**

**Pada**

**Jurusan Teknik Elektro  
Fakultas Teknik Universitas Lampung**



**PROGRAM STUDI TEKNIK ELEKTRO  
JURUSAN TEKNIK ELEKTRO  
FAKULTAS TEKNIK  
UNIVERSITAS LAMPUNG  
BANDAR LAMPUNG  
2023**

Judul Skripsi : **SISTEM PENDETEKSI DAN  
PENGHITUNG POLEN HIDUP DAN  
MATI PADA TANAMAN KELAPA  
SAWIT MENGGUNAKAN  
ALGORITMA YOLO V5 BERBASIS  
ARTIFICIAL INTELLIGENCE**

Nama Mahasiswa : **Ahul Halliwal Aqdi**

Nomor Pokok Mahasiswa : 1915031036

Program Studi : Teknik Elektro

Fakultas : Teknik

**MENYETUJUI**

1. Komisi Pembimbing

  
**Dr. Eng. Ir. Helmy Fitriawan, S.T., M.Sc.** NIP 197509282001121002

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

2. Mengetahui

Ketua Jurusan Teknik Elektro

Ketua Program Studi Teknik Elektro

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

  
**Dr. Eng. Nining Purwasih, S.T., M.T.**  
NIP 197404222000122001

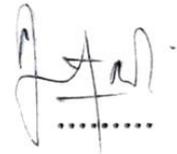
**MENGESAHKAN**

1. Tim Penguji

Ketua : **Dr. Eng. Ir. Helmy Fitriawan, S.T., M.Sc.** .....



Sekretaris : **Sumadi, S.T., M.T.**



Penguji  
Bukan Pembimbing : **Dr. Eng. F.X. Arinto Setyawan, S.T., M.T.** .....



2. Dekan Fakultas Teknik Universitas Lampung



**Dr. Eng. Ir. Helmy Fitriawan, S.T., M.Sc.** †  
NIP 19750928200121002

Tanggal Lulus Ujian Skripsi : **03 Mei 2023**

## SURAT PERNYATAAN

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

Apabila pernyataan saya tidak benar, maka saya bersedia dikenai sanksi akademik sesuai dengan hukum yang berlaku.

Bandar Lampung, 03 Mei 2023



Ahlul Halliwal Aqdi  
NPM 1915031036

## RIWAYAT HIDUP



Penulis lahir di Pagelaran pada tanggal 09 Februari 2001, sebagai anak kedua dari tiga bersaudara, dari Bapak Lokesworo dan Ibu Sri Erna Ningsih. Riwayat Pendidikan penulis dimulai dari SD Negeri 2 Pagelaran, SMP Negeri 1 Pringsewu, dan melanjutkan Pendidikan ke SMK Telkom Lampung dengan jurusan teknik jaringan akses. Penulis terdaftar sebagai mahasiswa teknik elektro Universitas Lampung pada tahun 2019 melalui jalur SBMPTN. Selama berkuliah, penulis aktif mengikuti kegiatan organisasi diantaranya, anggota Himpunan Mahasiswa Teknik Elektro (HIMATRO) Universitas Lampung, staff komisi 1 DPM U KBM Universitas Lampung, Koordinator acara panitia khusus Pemira Fakultas Teknik, Kepala dinas pemuda olahraga dan kreativitas mahasiswa Badam Eksekutif Mahasiswa Fakultas Teknik (BEM FT) Universitas Lampung. Pada semester 5, penulis mengambil konsentrasi elektronika dan kendali (elkaken) dan menjadi asisten laboratorium teknik elektronika. Penulis juga pernah mengikuti studi independen bersertifikat di PT. Artificial Intelligence Center Indonesia dan magang bersertifikat sebagai Artificial Intelligence Developer di PT. RPN Pusat Penelitian Kelapa Sawit di Medan, Sumatera Utara. Prestasi yang pernah dimenangkan yaitu mendapat juara 2 penulisan esai nasional dan pernah mendapat pendanaan dalam Program Mahasiswa Wirausaha (PMW) Universitas Lampung.

## PERSEMBAHAN

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

Puji syukur kehadiran Allah SWT atas segala nikmat, rahmat serta hidayah-  
Nya tak lupa pula shalawat serta salam kepada Nabi Muhammad SAW.

**Kupersembahkan karyaku ini dengan penuh rasa hormat, cinta, dan  
kasih sayang**

Kepada :

**Ayahanda dan Ibunda Tercinta**

**Bapak Lokesworo dan Ibu Sri Erna Ningsih**

**Kakak dan Adik Tersayang**

**Nurul Fahma Hidayah dan Aqila Nurlatifah**

Terimakasih pula saya ucapkan kepada Lembaga yang telah mendidik,  
mendewasakan dan tempat berpores dalam berfikir dan bertindak.  
Jurusan Teknik Elektro, Fakultas Teknik, Universitas Lampung

## MOTTO

لَا حَوْلَ وَلَا قُوَّةَ إِلَّا بِاللَّهِ الْعَلِيِّ الْعَظِيمِ

“Tiada daya dan upaya kecuali dengan kekuatan Allah yang maha tinggi lagi maha agung.”

“Sesungguhnya beserta kesulitan itu ada kemudahan. Maka apabila engkau telah selesai (dari sesuatu urusan), tetaplah bekerja keras (untuk urusan yang lain), dan hanya kepada Tuhanmulah engkau berharap.”

**(Al-Insyirah : 6-8)**

“Usaha dan doa tergantung pada cita-cita. Manusia tiada memperoleh selain apa yang telah diusahakannya. -  
Jalaluddin Rumi.”

**(Jalaluddin Rumi)**

## SANWACANA

Alhamdulillah puji syukur kehadirat Allah SWT atas segala karunia, hidayah, serta inayahnya sehingga penulis dapat menyelesaikan skripsi ini yang berjudul “Sistem Pendeteksi Dan Penghitung Polen Hidup Dan Mati Pada Tanaman Kelapa Sawit Menggunakan Algoritma YOLO V5 Berbasis *Artificial Intelligence*.” Yang merupakan salah satu syarat untuk memperoleh gelar Sarjana Teknik pada Jurusan Teknik Elektro Fakultas Teknik Universitas Lampung.

Selama perjalanan mengerjakan Skripsi ini, penulis mendapatkan dukungan moril dan bantuan pemikiran dari berbagai pihak. Oleh karena itu dalam kesempatan kali ini, penulis ingin mengucapkan terima kasih kepada:

1. Orang tua dan keluarga yang selalu mendukung dan memberi dorongan penulis untuk menyelesaikan dunia perkuliahan ini.
2. Ibu Herlinawati, S.T., M.T. sebagai Ketua Jurusan Teknik Elektro Universitas Lampung.
3. Ibu Dr. Eng. Nining Purwasih, S.T., M.T. sebagai Ketua Program Studi Teknik Elektro Universitas Lampung.
4. Bapak Dr. Eng. Ir. Helmy Fitriawan, S.T., M.Sc. sebagai Dosen Pembimbing Utama, terima kasih atas kesediaan waktunya untuk membimbing dan memberikan ilmu.
5. Bapak Sumadi, S.T., M.T. sebagai Dosen Pembimbing Akademik dan sebagai Dosen Pembimbing Pendamping, terima kasih atas kesediaan waktunya untuk membimbing dan memberikan ilmu.
6. Bapak Dr. Eng. F.X. Arinto Setyawan, S.T., M.T. selaku Dosen Penguji, terimakasih atas masukan, bimbingan, dan arahnya.
7. Bapak Nanang Supena, S.P., M.P sebagai Dosen Pendamping dari pihak Pusat Penelitian Kelapa Sawit, terima kasih atas ilmu yang telah diberikan dan segala bantuan yang telah diberikan kepada penulis.
8. Seluruh Dosen dan karyawan Jurusan Teknik Elektro Universitas Lampung.

9. Bapak Muzakky Al-Maududy selaku mentor magang, terima kasih atas bimbingan dan ilmu yang telah diberikan.
10. AIWA (*Artificial Intelligence, Web, Android*) Developer, terima kasih atas kenangan dan ilmu yang telah diberikan.
11. Bismillah Komisarlis (Agung dan Zakky), terima kasih atas segala bantuan yang diberikan selama kuliah.
12. Asisten Laboratorium Teknik Elektronika, terima kasih atas segala dukungan, cerita, dan ilmu yang diberikam.
13. Eternity selaku teman angkatan seperjuangan.
14. Seluruh teman-teman kuliah dan organisasi yang tidak bisa penulis sebutkan satu persatu, terima kasih atas segala cerita, ilmu, dan pengalaman yang telah diberikan.

Semoga Allah SWT membalas perbuatan kebaikan kalian atas terselesaikannya skripsi ini. Penulis menyadari bahwa banyak hal kurang dalam penulisan skripsi ini, maka dari itu Penulis terbuka untuk menerima masukan kritik dan saran yang dapat membangun Penulis kedepannya. Semoga penulisan skripsi ini dapat bermanfaat bagi pembaca.

Bandar Lampung, 03 Mei 2023  
Penulis,



**Ahlul Halliwal Aqdi**

## DAFTAR ISI

	<b>Halaman</b>
<b>ABSTRAK</b> .....	<b>ii</b>
<b>ABSTRACT</b> .....	<b>iv</b>
<b>LEMBAR PENGESAHAN</b> .....	<b>vii</b>
<b>SURAT PERNYATAAN</b> .....	<b>viii</b>
<b>RIWAYAT HIDUP</b> .....	<b>ix</b>
<b>PERSEMBAHAN</b> .....	<b>x</b>
<b>MOTTO</b> .....	<b>xi</b>
<b>SANWACANA</b> .....	<b>xii</b>
<b>DAFTAR ISI</b> .....	<b>xiv</b>
<b>DAFTAR GAMBAR</b> .....	<b>xvi</b>
<b>DAFTAR TABEL</b> .....	<b>xviii</b>
<b>BAB I</b> .....	<b>1</b>
<b>PENDAHULUAN</b> .....	<b>1</b>
1.1. Latar Belakang .....	1
1.2. Tujuan Penelitian .....	4
1.3. Manfaat Penelitian .....	4
1.4. Rumusan Masalah .....	4
1.5. Batasan Masalah.....	4
1.6. Hipotesis.....	5
1.7. Sistematika Penulisan.....	5
<b>BAB II</b> .....	<b>7</b>
<b>TINJAUAN PUSTAKA</b> .....	<b>7</b>
2.1. Penelitian Terdahulu .....	7
2.2. Polen.....	8
2.3. Citra.....	9
2.4. Pengolahan Citra Digital .....	10
2.5. <i>Computer Vision</i> .....	11
2.6. <i>Artificial Intelligence</i> .....	12
2.6.1. <i>Machine Learning</i> .....	12
2.6.2. <i>Deep Learning</i> .....	14
2.7. <i>Convolutional Neural Network</i> .....	15
2.7.1. Operasi Konvolusi .....	16
2.7.2. <i>Arsitektur Convolutional Neural Network (CNN)</i> .....	16

2.8.	<i>You Only Look Once (YOLO)</i> .....	21
2.9.	<i>Object Detection</i> .....	30
2.10.	<i>Python</i> .....	30
2.11.	Pelabelan Data.....	31
2.12.	<i>Streamlit</i> .....	31
2.13.	<i>Visual Studio Code</i> .....	32
<b>BAB III</b>	.....	<b>33</b>
<b>METODE PENELITIAN</b>	.....	<b>33</b>
3.1.	Waktu dan Tempat Penelitian .....	33
3.2.	Alat dan Bahan.....	33
3.3.	Prosedur Penelitian.....	34
3.4.	Perancangan Program Pengolahan Citra.....	37
3.5.	Perancangan Sistem Aplikasi Web .....	38
<b>BAB IV</b>	.....	<b>40</b>
<b>HASIL DAN PEMBAHASAN</b>	.....	<b>40</b>
4.1.	Dataset.....	40
4.2.	Pelabelan Dataset .....	41
4.3.	Pelatihan Model .....	45
4.3.1.	Membuat <i>Virtual Environment</i> .....	46
4.3.2.	Pembagian Dataset.....	46
4.3.3.	Proses Latih Model .....	47
4.4.	Konfigurasi <i>Streamlit</i> .....	60
4.5.	Hasil Pengujian .....	61
<b>BAB V</b>	.....	<b>73</b>
<b>KESIMPULAN DAN SARAN</b>	.....	<b>73</b>
5.1.	KESIMPULAN .....	73
5.2.	SARAN .....	74
<b>DAFTAR PUSTAKA</b>	.....	<b>75</b>

## DAFTAR GAMBAR

### Halaman

Gambar 2.1	Polen Bunga Kelapa Sawit .....	9
Gambar 2.2	Pengolahan Citra Digital.....	10
Gambar 2.3	Koordinat Dalam Citra Digital [12].....	11
Gambar 2.4	Pemetaan Objek .....	13
Gambar 2.5	Bagian-bagian Artificial Intelligence.....	14
Gambar 2.6	Contoh Jaringan Convolutional Neural Network .....	15
Gambar 2.7	Arsitektur CNN.....	17
Gambar 2.8	<i>Convolution Layer</i> .....	17
Gambar 2.9	<i>Feature Map</i> .....	18
Gambar 2.10	<i>Pooling Layer (max pool)</i> .....	19
Gambar 2.11	<i>Fully Connected Layer</i> .....	20
Gambar 2.12	Arsitektur R-CNN.....	21
Gambar 2.13	Arsitektur Fast R-CNN .....	21
Gambar 2.14	Arsitektur Faster R-CNN .....	22
Gambar 2.15	<i>Intersection Over Union</i> .....	24
Gambar 2.16	Contoh Hasil IOU .....	24
Gambar 2.17	Penentuan nilai <i>confusion matrix</i> .....	25
Gambar 2.18	Konsep Desain YOLO .....	26
Gambar 2.19	Model YOLO V5 .....	29
Gambar 2.20	Proses Deteksi YOLO.....	29
Gambar 2.21	Pelabelan Dataset.....	31
Gambar 3.1	Diagram alir prosedur penelitian .....	35
Gambar 3.2	Diagram Alir Perancangan Program Pengolahan Citra .....	37
Gambar 3.3	Diagram Blok Perancangan Sistem .....	39
Gambar 4.1	Citra Polen Kelapa Sawit.....	41
Gambar 4.2	Pelabelan Citra Polen.....	41
Gambar 4.3	Langkah Pelabelan Data .....	43
Gambar 4.4	Keluaran Hasil Pelabelan Data .....	44
Gambar 4.5	Kotak Pembatas Pelabelan Citra.....	44
Gambar 4.6	Membuat <i>Virtual Environmnet</i> .....	46
Gambar 4.7	<i>Git Clone</i> YOLO V5.....	47
Gambar 4.8	<i>Install Requirements</i> .....	48
Gambar 4.9	Parameter Pelatihan Model.....	49
Gambar 4.10	Dataset.yaml .....	50
Gambar 4.11	Proses Pelatihan Model.....	51
Gambar 4.12	Struktur YOLO V5 .....	52
Gambar 4.13	Potongan sel pada grid.....	54
Gambar 4.14	IOU .....	55
Gambar 4.15	<i>Non-Max Suppression</i> .....	56
Gambar 4.16	Grafik Presisi YOLO V5s.....	57
Gambar 4.17	Grafik Presisi YOLO V5m .....	57

Gambar 4.18 Grafik Presisi YOLO V5l .....	58
Gambar 4.19 Grafik Presisi YOLO V5x .....	59
Gambar 4.20 Grafik Perbandingan Model.....	59
Gambar 4.21 Tampilan Aplikasi Web Streamlit .....	61
Gambar 4.22 Data Hasil Prediksi Polen Pertama .....	62
Gambar 4.23 Data Hasil Prediksi Polen Kedua.....	64
Gambar 4.24 Data Hasil Prediksi Polen Ketiga.....	66
Gambar 4.25 Data Hasil Prediksi Polen Keempat.....	68
Gambar 4.26 Data Hasil Prediksi Polen Kelima .....	70

## DAFTAR TABEL

	<b>Halaman</b>
Tabel 2.1 Kecepatan Model Deteksi [22] .....	22
Tabel 4.1 Pembagian Dataset Citra Polen .....	40
Tabel 4.2 Bentuk vektor dari y label pada objek di setiap sel .....	54
Tabel 4.3 Bentuk vektor y label polen.....	54
Tabel 4.4 Perbandingan dari hasil latih model YOLO .....	60
Tabel 4.5 Pengujian Citra Polen Pertama .....	62
Tabel 4.6 Perbandingan Nilai Aktual dan Hasil Prediksi Pengujian Pertama Kondisi Normal .....	63
Tabel 4.7 Pengujian Citra Polen Kedua.....	64
Tabel 4.8 Perbandingan Nilai Aktual dan Hasil Prediksi Pengujian Kedua Kondisi Normal .....	65
Tabel 4.9 Pengujian Citra Polen Ketiga .....	66
Tabel 4.10 Perbandingan Nilai Aktual dan Hasil Prediksi Pengujian Ketiga Kondisi Normal.....	67
Tabel 4.11 Pengujian Citra Polen Keempat.....	68
Tabel 4.12 Perbandingan Nilai Aktual dan Hasil Prediksi Pengujian Keempat Kondisi Normal .....	69
Tabel 4.13 Pengujian Citra Polen Kelima .....	70
Tabel 4.14 Perbandingan Nilai Aktual dan Hasil Prediksi Pengujian Kelima Kondisi Citra Normal.....	71
Tabel 4.15 <i>Precision, Recall</i> , Akurasi dan Waktu Hasil Prediksi Pengujian Kondisi Normal.....	72

# **BAB I**

## **PENDAHULUAN**

### **1.1. Latar Belakang**

Perkembangan teknologi saat ini telah mengalami perkembangan yang sangat pesat dan memiliki peranan penting dalam perubahan pola kehidupan manusia. Salah satu perkembangan teknologi itu adalah *artificial intelligence*. *Artificial intelligence* (AI) atau dalam bahasa Indonesia berarti kecerdasan buatan, merupakan kemampuan komputer yang mengadaptasi dari perilaku kecerdasan manusia seperti cara manusia dalam mengamati, memahami, memprediksi, mengambil keputusan dan menyelesaikan masalah sehari-hari. Kehadiran AI dalam kehidupan manusia tentu sangat berguna diantaranya dapat meningkatkan produktivitas, dimana pekerjaan manusia dapat dilakukan dengan waktu yang lebih cepat dan efisien. Tujuan utama dari kecerdasan buatan adalah untuk mengembangkan sistem yang lebih canggih dan lebih kompleks yang akan mengungguli manusia dalam hal apa pun. Hal ini termasuk kinerja tugas-tugas yang lebih rumit seperti bermain catur dan memecahkan persamaan. Oleh karena itu, tujuan masa depan AI adalah menyempurnakan semua aktivitas manusia dan memberikan solusi yang lebih baik untuk masalah daripada yang dapat dilakukan manusia [1]. Saat ini, otomatisasi pekerjaan dengan AI mulai banyak dilakukan dalam berbagai aspek pekerjaan dengan harapan dapat menurunkan kesalahan pengguna dalam bekerja.

Dalam industri pertanian, juga perlu adanya implementasi AI dalam mendukung pertanian berkelanjutan. Salah satunya yaitu implementasi AI pada industri kelapa sawit di Indonesia. Pada industri kelapa sawit, kebutuhan

akan bahan tanam kelapa sawit yang berkualitas tinggi sangat diperlukan untuk menunjang perkembangan perkebunan kelapa sawit di Indonesia. Kelapa sawit (*Elaeis guineensis Jacq.*) merupakan salah satu jenis tanaman perkebunan yang penting dan penghasil minyak tertinggi. Luas areal produksi kelapa sawit di Indonesia terus meningkat berdasarkan data dari Departemen Pertanian 2011 produksi CPO (*Crude Palm Oil*) di Indonesia sebesar 19.844.901 ton dengan luas areal sebesar 8.430.206 ha [2]. Minyak sawit diproduksi di seluruh dunia karena banyak dibutuhkan untuk konsumsi. Produksi minyak sawit digunakan untuk berbagai macam makanan, kosmetik, produk kebersihan, dan sumber *biofuel* atau *biodiesel* [3]. Produksi buah kelapa sawit ini dipengaruhi oleh proses penyerbukan yang menggunakan polen atau serbuk sari sebagai alat perkembangbiakkan jantan. Penyerbukan ini biasanya dilakukan dengan cara proses polinasi kepala butik oleh polen melalui penyerbukan sendiri (dibantu angin), melalui serangga penyerbuk (*Elaidobius Kamerunicus*) dan melalui bantuan manusia [4]. Permasalahan penyerbukan alami biasanya disebabkan karena kurangnya jumlah bunga jantan dan kurang efektifnya serangga penyerbuk (*Elaidobius Kamerunicus*) pada saat musim penghujan. Salah satu upaya yang dilakukan dalam menghasilkan bahan tanam kelapa sawit berkualitas tinggi yaitu dengan melakukan persilangan atau penyerbukan buatan. Dalam proses persilangan ini dilakukan dengan cara mengambil polen dari bunga jantan kemudian dilakukan penyerbukan dengan bunga betina. Polen atau serbuk sari ini merupakan organ jantan pada tumbuhan yang memiliki peran penting dalam proses reproduksi tanaman, sehingga perlu diketahui viabilitasnya karena berpengaruh terhadap keberhasilan proses persilangan.

Proses yang biasa dilakukan dalam mengetahui viabilitas polen dengan cara mengamati sampel polen yang sebelumnya sudah disiapkan kedalam wadah, kemudian sampel polen tersebut diambil dan di letakkan kedalam media dengan ditetesi larutan borax dan sukrosa dengan komposisi 0,125 g untuk borax, 0,25 g untuk sukrosa, dan 100 ml aquades kemudian diinkubasi pada suhu 40°C selama kurang lebih 2-3 jam [5]. Setelah selesai diinkubasi, sampel

polen tersebut diambil dan diamati menggunakan mikroskop cahaya dan ditangkap menggunakan kamera mikroskop tersebut sehingga keluarannya berupa gambar dengan format JPG (*Joint Photographic Group*). Mikroskop tersebut sudah terhubung dengan komputer pengguna sehingga tangkapan gambar dari kamera ini langsung tersimpan kedalam komputer pengguna. Setelah gambar polen di dapatkan kemudian pengguna melakukan pengamatan terhadap polen hidup dan mati serta melakukan perhitungan persentase polen hidup terhadap jumlah polen total secara manual dengan viabilitas yang baik umumnya di atas 50% [5], tentu proses ini membutuhkan banyak waktu.

Solusi yang dilakukan untuk mempermudah dalam proses perhitungan polen tersebut agar diketahui viabilitasnya, yaitu dengan cara melakukan implementasi *artificial intelligence* kedalam sistem pendeteksi dan perhitungan polen hidup dan mati. Sistem ini akan mengklasifikasikan polen hidup dan mati menggunakan data latih yang memiliki bobot dengan akurasi yang baik sehingga dapat melakukan proses deteksi dan perhitungan dengan cepat dan efisien. Hasil pengujian menggunakan sistem ini digunakan sebagai acuan pengguna dalam menentukan viabilitas polen untuk selanjutnya siap dilakukan penyerbukan. Sistem ini dibuat dengan menggunakan konsep pengguna *friendly* sehingga dapat memudahkan pengguna dalam mengakses dan menginput data pada saat proses pendeteksian dan perhitungan polen. Adapun algoritma yang digunakan dalam melakukan implementasi ini yaitu *You Only Look Once* (YOLO) menggunakan model YOLO v5l (*large*). Berdasarkan permasalahan dan ide tersebut penulis melakukan penelitian dengan judul “Sistem Pendeteksi Dan Penghitung Polen Hidup Dan Mati Pada Tanaman Kelapa Sawit Berbasis *Artificial Intelligence* Menggunakan Algoritma YOLO V5” dengan studi kasus di PT. Riset Perkebunan Nusantara (RPN) Pusat Penelitian Kelapa Sawit. Harapan dari penelitian ini yaitu dapat mempermudah pekerjaan pengguna pada divisi terkait di PT. Riset Perkebunan Nusantara (RPN) Pusat Penelitian Kelapa Sawit dalam melakukan proses perhitungan viabilitas polen secara cepat dan efisien.

## 1.2. Tujuan Penelitian

Adapun tujuan penelitian ini adalah sebagai berikut :

1. Mengembangkan program pendeteksi dan penghitung polen pada tanaman kelapa sawit menggunakan algoritma YOLO V5 berbasis *artificial intelligence* dengan 2 kelas yaitu hidup dan mati.
2. Membuat aplikasi berbasis web yang dapat mengklasifikasikan dan menghitung polen secara cepat dan efisien berbasis *artificial intelligence*.

## 1.3. Manfaat Penelitian

Adapun manfaat dari penelitian ini adalah sebagai berikut :

1. Memperoleh web aplikasi yang dapat mendeteksi dan menghitung polen hidup dan mati secara cepat dan efisien.
2. Mempermudah pengguna dalam mendeteksi dan menghitung polen hidup dan mati untuk mengetahui viabilitas polen.

## 1.4. Rumusan Masalah

Adapun rumusan masalah dari penelitian ini adalah sebagai berikut :

1. Bagaimana membuat program yang tepat untuk mendeteksi dan menghitung polen hidup dan mati menggunakan algoritma YOLO V5?
2. Bagaimana pengguna dapat mengoperasikan sistem ini dengan mudah dan *user friendly*?

## 1.5. Batasan Masalah

Adapun batasan masalah pada penelitian ini adalah sebagai berikut :

1. Citra yang diolah adalah citra digital yang diambil menggunakan kamera mikroskop dalam format *Joint Photographic Group* (JPG).

2. Citra diambil pada latar belakang terang dan tidak kabur.
3. Kriteria untuk menentukan polen hidup dan mati berdasarkan warna polen.
4. Keluaran dari penelitian ini adalah program *python* dan aplikasi web untuk mendeteksi dan menghitung polen hidup dan mati.

## **1.6. Hipotesis**

Pendeteksian dan perhitungan polen tanaman kelapa sawit menggunakan algoritma YOLO V5 dengan model YOLO V5l (*large*) dapat diaplikasikan untuk membantu pengguna dalam mengklasifikasikan dan menghitung polen hidup dan mati secara cepat dan efisien.

## **1.7. Sistematika Penulisan**

Sistematika penulisan yang digunakan dalam penulisan tugas akhir ini dibagi menjadi lima bab, yaitu :

### **BAB I PENDAHULUAN**

Pendahuluan terdapat latar belakang, tujuan penelitian, manfaat penelitian, rumusan masalah, batasan masalah, hipotesis, dan sistematika penulisan.

### **BAB II TINJAUAN PUSTAKA**

Tinjauan pustaka berisi teori-teori yang mendukung pendeteksian dan perhitungan polen tanaman kelapa sawit menggunakan algoritma YOLO V5 berbasis *artificial intelligence*.

### **BAB III METODE PENELITIAN**

Metode penelitian berisi waktu dan tempat penelitian, alat dan bahan yang digunakan, garis besar metode yang diusulkan, serta diagram alir penelitian dan metode yang diusulkan.

#### **BAB IV HASIL DAN PEMBAHASAN**

Hasil dan pembahasan menjelaskan hasil penelitian, analisis dan pembahasan hasil penelitian, dan perhitungan kinerja metode yang diusulkan.

#### **BAB V PENUTUP**

Penutup berisi tentang kesimpulan yang didapatkan dari hasil penelitian serta saran untuk pengembangan penelitian ini lebih lanjut.

## **BAB II** **TINJAUAN PUSTAKA**

### **2.1. Penelitian Terdahulu**

Berkaitan dengan penelitian yang dilakukan, referensi terhadap penelitian yang telah dilakukan sebelumnya sangat penting untuk menghindari plagiarisme atau duplikasi penelitian sebelumnya, hal ini juga dimaksudkan sebagai bahan penunjang penelitian agar tema penelitian ini terus berkembang. Berikut merupakan beberapa penelitian terdahulu yang telah dilakukan sebelumnya berdasarkan metode yang digunakan serta hasil yang diperoleh.

Pada penelitian yang dilakukan oleh Mariam L. Francies, *et al.*, dengan judul “*A Robust Multiclass 3D Object Recognition Based on Modern YOLO Deep Learning Algorithms*” membandingkan performa dari model algoritma YOLO yaitu YOLO V3, YOLO V4, dan YOLO V5 menggunakan *object recognition* (Pascal VOC 2007) sebagai masukkan jaringan yang dilakukan maka didapatkan hasil akurasi tertinggi yaitu YOLO V3 dengan nilai 77% dengan waktu 8 jam 4 menit, YOLO V4 dengan rata-rata akurasi 54,9% dalam waktu pelatihan model 6 jam 52 menit dan hasil modifikasi YOLO V5 dengan akurasi 55% serta memiliki waktu pelatihan model yang lebih sedikit yaitu 3 jam 10 menit [6].

Pada penelitian yang dilakukan oleh Elzbieta, *et al.*, dengan judul “*Detection and Recognition of Pollen Grains in Multilabel Microscopic Images*” melakukan komparasi (perbandingan) model YOLO V5s, YOLO V5l, RetinaNet dan Faster R-CNN untuk mendeteksi polen dengan tiga kelas yaitu

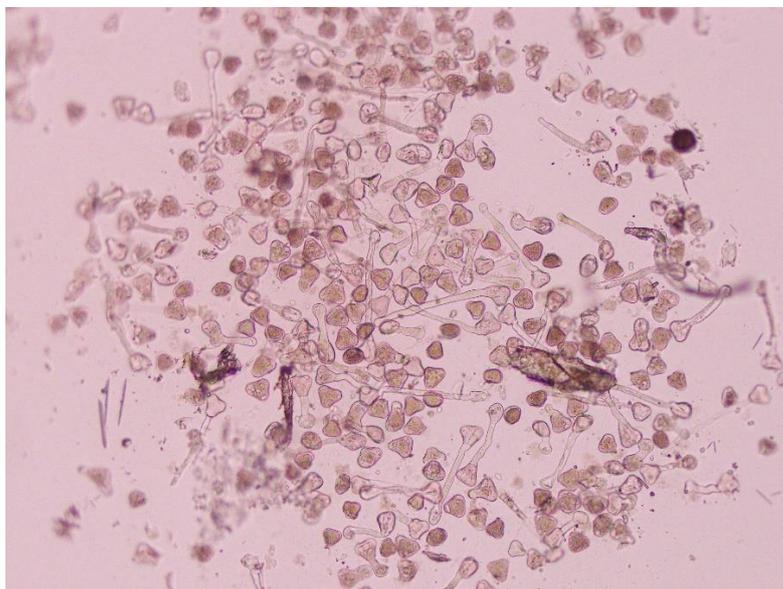
*alnus*, *Betula* dan *Corylus* serta membagi data latih sebanyak 265 gambar, data validasi 114 gambar dan data tes sebanyak 49 gambar. Berdasarkan parameter tersebut berhasil mendapatkan rata-rata nilai akurasinya sebesar 90,7 % untuk YOLO V5s, 91,5% untuk YOLO V5l, 79,5% untuk RetinaNet dan 51,0% untuk Faster R-CNN [7].

Pada penelitian yang dilakukan oleh Abdullah, et al., dengan judul “*Computer Vision Based Deep Learning Approach for the Detection and Classification of Algae Species Using Microscopic Images*” melakukan penelitian terhadap spesies *Algae* dengan empat kelas yaitu *Cosmarium*, *Scenedesmus*, *Closterium*, and *Spirogyra*. Penelitian ini bertujuan untuk mendeteksi dan mengklasifikasikan spesies *Algae* dengan melakukan perbandingan algoritma yaitu YOLO V3, YOLO V4 dan YOLO V5 serta mendapatkan rata-rata nilai akurasi (mAP) sebesar 75,3% untuk YOLO V3, 83,0% untuk YOLO V4, dan 90,1% untuk YOLO V5 [8].

## **2.2. Polen**

Serbuk sari atau polen merupakan alat perkembangbiakan jantan yang ada pada bunga yang memiliki bentuk yang sangat kecil seperti butiran-butiran halus dan berjumlah sangat banyak yang dihasilkan oleh tumbuhan berbunga atau *spermatophyta*. Serbuk sari diproduksi di dalam kepala sari atau antera, saat sudah masak kepala sari akan mengeluarkan serbuk sari yang dapat terbawa oleh angin, air, serangga, hewan ataupun manusia sampai dengan jarak yang jauh dari tempat asalnya [9]. Serbuk sari memiliki mantel keras yang melindungi sel-sel sperma selama proses gerakan mereka antara benang sari ke putik tanaman berbunga. Ketika serbuk sari di putik kompatibel pada tanaman berbunga, maka berkecambah dan menghasilkan tabung polen yang mentransfer sperma ke ovula dari ovarium reseptif. Transfer serbuk sari terhadap struktur reproduksi wanita (putik dalam *angiosperma*) disebut penyerbukan [10]. Proses penyerbukan ini bisa dilakukan oleh angin (*anemophilous*), serangga penyerbuk dan bantuan manusia. Berikut Gambar

2.1 merupakan polen atau serbuk sari pada tanaman kelapa sawit yang diambil menggunakan kamera mikroskop.



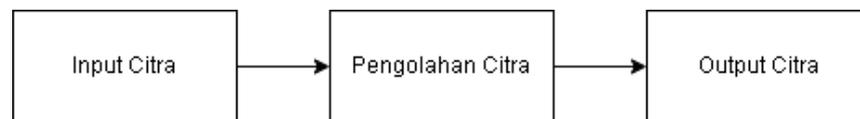
Gambar 2.1 Polen Bunga Kelapa Sawit

### 2.3. Citra

Citra atau gambar merupakan sebuah representasi, kemiripan, ataupun imitasi dari suatu obyek atau benda. Terdapat dua kategori dari citra yaitu citra tampak dan citra tidak tampak. Adapun contoh dari citra tampak yang sering dijumpai dalam kehidupan sehari-hari yaitu foto, gambar, dan lukisan, sedangkan citra tidak tampak yaitu data gambar dalam file (citra digital), dan citra yang direpresentasikan menjadi fungsi matematis. Berdasarkan contoh citra tersebut, hanya citra digital yang dapat diolah menggunakan komputer. Apabila jenis citra yang lain hendak diolah dengan komputer, maka harus diubah terlebih dahulu menjadi citra digital, sebagai contoh foto discan dengan *scanner*, persebaran panas tubuh foto ditangkap dengan kamera infra merah dan diubah menjadi informasi numeris, informasi densitas dan komposisi bagian dalam tubuh manusia ditangkap dengan bantuan pesawat sinar-x dan sistem deteksi radiasi menjadi informasi digital. Kegiatan untuk mengubah informasi citra fisik non digital menjadi digital disebut sebagai pencitraan (*imaging*) [11].

## 2.4. Pengolahan Citra Digital

Citra digital adalah gambar dua dimensi yang dihasilkan dari analog dua dimensi yang kontinu menjadi gambar melalui proses sampling. Gambar analog dibagi menjadi N baris dan M kolom sehingga menjadi gambar diskrit. Citra yang dapat diolah dengan komputer yaitu citra digital [12]. Yang disimpan dalam komputer hanyalah angka-angka yang menunjukkan besar intensitas pada masing-masing piksel. Karena berbentuk data numerik, maka citra digital dapat diolah dengan komputer. Pengolahan citra digital merupakan ilmu yang mempelajari tentang hal-hal yang berkaitan dengan perbaikan kualitas terhadap suatu gambar (meningkatkan kontras, perubahan warna, restorasi citra), transformasi gambar (translasi, rotasi transformasi, skala, geometrik), melakukan pemilihan citra ciri (*feature images*) yang optimal untuk tujuan analisis, melakukan penyimpanan data yang sebelumnya dilakukan reduksi dan kompresi, transmisi data, dan waktu proses data [12]. Berikut merupakan diagram sederhana dari sebuah proses pengolahan citra dapat dilihat pada Gambar 2.2 di bawah ini.

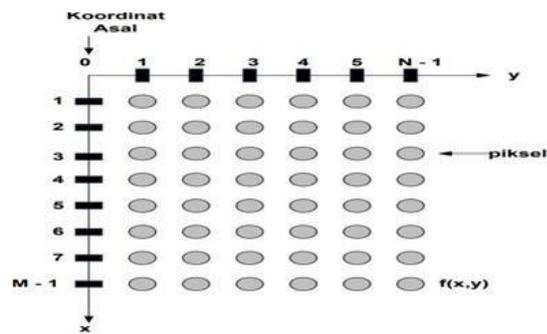


Gambar 2.2 Pengolahan Citra Digital

Secara umum pengolahan citra digital dapat diartikan sebagai pemrosesan gambar dua dimensi dengan menggunakan komputer. Citra digital merupakan sebuah *array* (larik) yang berisikan nilai-nilai real maupun kompleks yang dapat direpresentasikan dengan deretan bit tertentu.

Sebuah citra digital dapat diwakili oleh sebuah matriks yang terdiri dari M kolom dan N baris seperti pada Gambar 2.3 berikut, dimana perpotongan antara kolom dan baris tersebut disebut dengan piksel (*picture element*) yang merupakan elemen terkecil dari sebuah citra. Piksel mempunyai dua parameter, yaitu koordinat dan intensitas atau warna. Nilai yang terdapat

pada koordinat  $(x,y)$  adalah  $f(x,y)$  yang merupakan besar intensitas atau warna dari piksel di titik itu.



Gambar 2.3 Koordinat Dalam Citra Digital [12]

$$f = \begin{bmatrix} f(1,1) & f(1,2) & \dots & f(1,N) \\ f(2,1) & f(2,2) & \dots & f(2,N) \\ \dots & \dots & \dots & \dots \\ f(M,1) & f(M,2) & \dots & f(M,N) \end{bmatrix} \dots \dots \dots (1)$$

Berdasarkan gambaran diatas, maka secara matematis citra digital dapat dituliskan sebagai intensitas  $f(x,y)$  dimana nilai  $x$  (baris) dan  $y$  (kolom) adalah koordinat posisi dan  $f(x,y)$  adalah nilai fungsi pada setiap titik  $(x,y)$  yang menyatakan besar intensitas yang diterima oleh citra atau tingkat keabuan atau warna dari piksel di titik tersebut. Besarnya tergantung pada intensitas yang dipantulkan oleh objek. Ini berarti  $f(x,y)$  sebanding dengan energi yang dipancarkan oleh sumber cahaya, sehingga besar intensitas  $f(x,y)$  [12].

## 2.5. Computer Vision

*Computer vision* atau visi komputer dapat didefinisikan dengan pengertian pengolahan citra yang dikaitkan dengan akuisisi citra, pemrosesan, klasifikasi, penganan, dan pencakupan keseluruhan, pengambilan keputusan yang diikuti pengidentifikasian citra. Inti dari teknologi computer vision adalah untuk menduplikasi kemampuan penglihatan manusia ke dalam benda elektronik sehingga benda elektronik dapat memahami dan mengerti arti dari gambar yang dimasukkan [13].

## **2.6. Artificial Intelligence**

Kecerdasan buatan (AI) adalah kecerdasan yang diperlihatkan oleh mesin. Dalam ilmu komputer, bidang penelitian AI mendefinisikan sebagai studi tentang "agen cerdas" artinya setiap perangkat yang dapat memahami lingkungannya dan mengambil tindakan guna memaksimalkan peluang untuk tujuan tertentu. AI ini secara sederhana dapat meniru tingkah laku manusia dalam kegiatan sehari-hari misalnya seperti belajar menirukan sesuatu dan melakukan pemecahan masalah. Kemampuan AI saat ini sudah pada tahap dapat memahami bahasa manusia, bersaing menyelesaikan suatu game dengan tingkat kesulitan yang tinggi, adanya pengembangan self-driving dan juga penerapan untuk mengolah data yang kompleks [14].

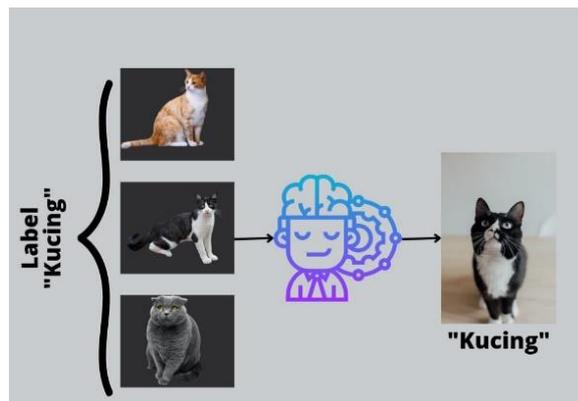
### **2.6.1. Machine Learning**

*Machine learning* merupakan cabang ilmu bagian dari kecerdasan buatan (*artificial intelligence*), dengan pemrograman untuk memungkinkan komputer menjadi cerdas berperilaku seperti manusia dan dapat meningkatkan pemahamannya melalui pengalaman secara otomatis [15]. Fokus dari *machine learning* yaitu terletak pada pengembangan sistem yang mampu belajar sendiri untuk mengambil keputusan tanpa harus berulang kali diprogram oleh manusia. Hal ini menjadikan mesin tidak hanya mampu berperilaku mengambil keputusan, namun juga dapat beradaptasi dengan perubahan yang terjadi. *Machine learning* bekerja apabila tersedia data sebagai input untuk dilakukan analisis terhadap kumpulan data besar (*big data*) sehingga menemukan pola tertentu. Data merupakan bahan input yang akan digunakan untuk melakukan pembelajaran (*training*) sehingga mesin dapat menghasilkan analisis yang benar. Didalam *machine learning* dikenal data latih dan data uji, data latih untuk melatih algoritma dalam *machine learning* dan data uji untuk mengetahui performa dari algoritma dalam *machine learning* yang telah dilatih yaitu ketika menemukan data baru yang

belum pernah diberikan dalam data latih [16]. *Machine learning* menggunakan teknik untuk menangani data besar (*big data*) dengan cara yang cerdas untuk memberikan hasil yang tepat. Berdasarkan teknik pembelajarannya, tipe-tipe *machine learning* dapat dibedakan menjadi *supervised learning*, *unsupervised learning* dan *reinforcement learning* berikut merupakan penjelasannya.

### 1. *Supervised learning*

*Supervised learning* merupakan salah satu teknik *machine learning* yang menggunakan *dataset* yang sudah berlabel (*labeled data*) untuk melakukan pembelajaran pada mesin, sehingga mesin mampu mengidentifikasi label *input* dengan menggunakan fitur yang dimiliki untuk selanjutnya melakukan prediksi maupun klasifikasi. Pemetaan objek data berlabel dapat dilihat pada Gambar 2.4 berikut.



Gambar 2.4 Pemetaan Objek

Berdasarkan Gambar 2.4 diatas dapat diketahui bahwa diberikan *input* dengan label kucing, kemudian data berlabel tersebut dilatih dan diproses oleh mesin maka ketika mesin diberikan data baru yang identik dengan data *input* akan menghasilkan *output* yang sama sesuai dengan label yang diberikan. Algoritma yang termasuk kedalam teknik *supervised learning* diantaranya *Decision Tree*, *K-Nearest Neighbor* (KNN), *Naive Bayes*, Regresi, dan *Super Vector Machine* [17].

### 2. *Unsupervised learning*

*Unsupervised learning* adalah teknik dengan menarik kesimpulan berdasarkan dataset yang merupakan input data *labeled response* [18].

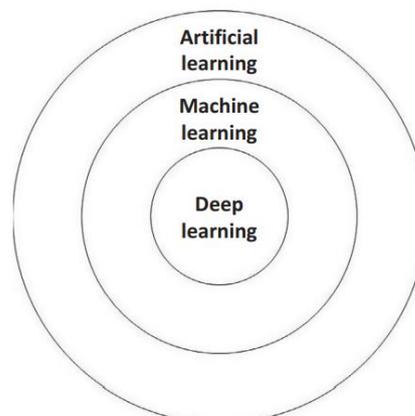
Teknik ini akan bekerja menemukan pola yang sesuai untuk mendapatkan suatu informasi dan tidak membutuhkan data berlabel diawal. Algoritma dalam *unsupervised learning* diantaranya DBSCAN, *Fuzzy C-Means*, *K-Means*, dan *Self Organizing Map*.

### 3. *Reinforcement Learning*

*Reinforcement learning* merupakan teknik *machine learning* yang melakukan pembelajaran berinteraksi dengan lingkungan sekitar (*environment*) untuk mendapatkan informasi. *Reinforcement learning* pembelajaran ini akan mendapatkan *reward* dan harus belajar mengetahui cara apa untuk mendapat *reward* tersebut.

#### 2.6.2. *Deep Learning*

*Deep Learning* adalah cabang ilmu *machine learning* berbasis Jaringan Syaraf Tiruan (JST) atau bisa dikatakan sebagai perkembangan dari JST. Dalam *deep learning*, sebuah komputer belajar mengklasifikasi secara langsung dari gambar atau suara. *Convolutional Neural Network* (CNN/ConvNet) adalah salah satu algoritma *deep learning* yang merupakan pengembangan dari *Multilayer Perceptron* (MPL) yang dirancang untuk mengolah data dalam bentuk dua dimensi, misalnya gambar atau suara. CNN dapat belajar langsung dari citra sehingga mengurangi beban dari pemrograman [19]. *Deep learning* merupakan bagian dari *artificial intelligence* yang secara urutan dapat dilihat pada Gambar 2.5 berikut.

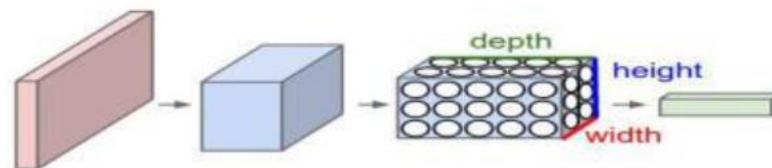


Gambar 2.5 Bagian-bagian Artificial Intelligence

*Deep learning* adalah bagian dari *machine learning*, dan menggabungkan komputasi model dan algoritma yang meniru arsitektur jaringan syaraf biologis di otak (jaringan syaraf tiruan (ANN)). Setiap kali otak menerima informasi baru, ia mencoba membandingkannya dengan informasi yang sudah diketahui untuk mencoba memahaminya. Otak menguraikan informasi melalui pelabelan dan penetapan item ke berbagai kategori, dan pada *deep learning* menggunakan konsep yang sama [20].

## 2.7. *Convolutional Neural Network*

*Convolutional Neural Network* (CNN) merupakan jaringan syaraf yang dikhususkan untuk memproses data yang memiliki *grid*. Sebagai contoh adalah berupa citra dua dimensi. Nama konvolusi sendiri merupakan operasi aljabar linear yang mengkalikan matriks dari *filter* pada citra yang akan di proses. Proses ini disebut lapisan konvolusi dan merupakan salah satu jenis dari banyak lapisan bisa dimiliki dalam satu jaringan. Lapisan konvolusi merupakan lapisan utama yang paling penting untuk digunakan. Jenis lapisan lain yang biasa digunakan adalah *Pooling Layer*, yakni lapisan yang digunakan untuk mengambil nilai maksimal atau nilai rata-rata dari bagian-bagian piksel pada citra. Adapun lapisan dari jaringan CNN dapat dilihat pada Gambar 2.6 berikut.



Gambar 2.6 Contoh Jaringan *Convolutional Neural Network*

Pada Gambar 2.6 menunjukkan setiap lapisan *input* yang dimasukkan memiliki volume yang berbeda dan mewakili dengan kedalaman, tinggi dan lebar. Setiap besaran yang didapatkan tergantung dari hasil filtrasi dari lapisan sebelumnya dan juga banyak *filter* yang digunakan. Model jaringan seperti ini sudah terbukti sangat ampuh dalam menangani permasalahan klasifikasi citra.

### 2.7.1. Operasi Konvolusi

Operasi konvolusi adalah operasi pada dua fungsi argumen bernilai nyata. Operasi ini menerapkan fungsi *output* sebagai *Feature Map* dari input citra. *Input* dan *output* ini dapat dilihat sebagai dua argumen bernilai riil. Secara formal operasi konvolusi dapat ditulis dengan rumus berikut.

$$s(t) = (x \times w) (t) \dots\dots\dots(2)$$

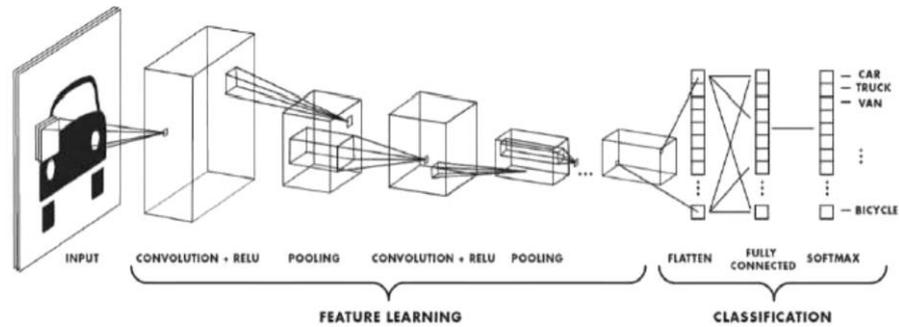
Fungsi  $s(t)$  memberikan *output* tunggal berupa *Feature Map*, argumen pertama adalah input yang merupakan  $x$  dan argumen kedua  $w$  sebagai kernel atau filter. *Kernel size* ini merupakan ukuran seberapa lokal filter input yang akan diatur pada proses konvolusi. Jika kita melihat input sebagai citra dua dimensi, maka kita bisa mengasumsikan  $t$  sebagai piksel dan menggantinya dengan  $i$  dan  $j$ . Operasi untuk konvolusi ke input dengan lebih dari satu dimensi dapat ditulis sebagai berikut.

$$S(i, j) = (K \times I) (i, j) = \sum \sum l(i - m, j - n) K (m, n) \dots\dots\dots(3)$$

Persamaan diatas adalah perhitungan dasar dalam operasi konvolusi dimana  $i$  dan  $j$  adalah piksel dari citra. Perhitungannya bersifat komutatif dan muncul saat  $K$  sebagai kernel,  $I$  sebagai *input* dan kernel yang dapat dibalik relatif terhadap *input*. Sebagai alternatif, operasi konvolusi dapat dilihat sebagai perkalian matriks antara citra masukan dan kernel dimana keluarannya dapat dihitung dengan *dot product*.

### 2.7.2. Arsitektur Convolutional Neural Network (CNN)

Arsitektur dari CNN juga terbagi menjadi 2 bagian besar yaitu *Feature Extraction Layer* dan *Fully-Connected Layer (MLP)* seperti pada Gambar 2.7 berikut.

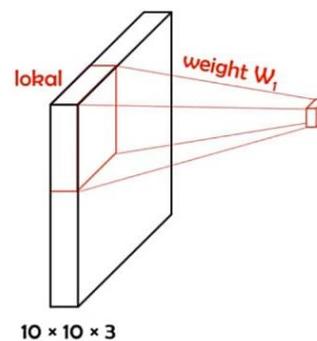


Gambar 2.7 Arsitektur CNN

Secara teknis, CNN terdiri dari tiga lapisan utama yaitu *convolution layer* yang ditunjukkan pada Gambar 2.8, *pooling layer*, dan *fully connected layer*.

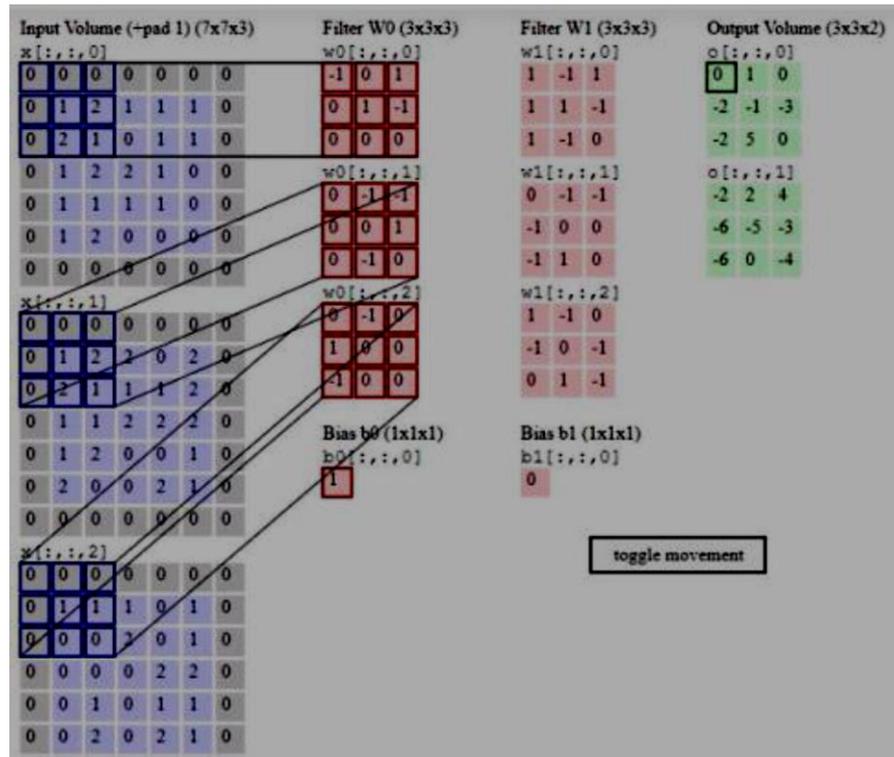
### 1. Convolution Layer

#### Convolution layer



Gambar 2.8 Convolution Layer

Gambar 2.8 diatas merupakan gambar dari proses *convolution layer* dimana terdapat input dengan  $10 \times 10 \times 3$  tetapi bobot diambil secara lokal. Parameter akan *dishare* dengan input yang lain, artinya input akan diambil dan bergeser sejauh nilai *stride* yang akan ditentukan . Panjang 10 piksels, tinggi 10 piksels dan tebal atau jumlah sesuai dengan *channel* dari *image* tersebut. Ketiga filter ini akan digeser keseluruh bagian dari gambar. Setiap pergeseran akan dilakukan operasi “dot” antara *input* dan nilai dari filter sehingga menghasilkan sebuah *output* atau biasa disebut sebagai *activation map* atau *feature map*. Proses dari *feature map* seperti pada Gambar 2.9 berikut.



Gambar 2.9 Feature Map

Pada Gambar 2.9 dijelaskan bagaimana pergeseran *stride* tiap piksel nya. Pada *feature map* diatas menggunakan 2 *stride* sehingga akan melompat 2 kotak tiap pergeserannya. Semakin kecil nilai *stride* maka akan semakin bagus hasil nilai yang didapatkan. Tiap pergeseran akan dijumlahkan dengan filter sehingga akan menghasilkan *output* [21]. Untuk menghitung dimensi dari *feature map*, bisa menggunakan rumus dibawah ini :

$$V = \frac{W - F + 2P}{s} + 1 \dots\dots\dots(4)$$

V = Volume

W = Panjang atau Tinggi input

F = Panjang atau Tinggi filter

P = Zero Padding

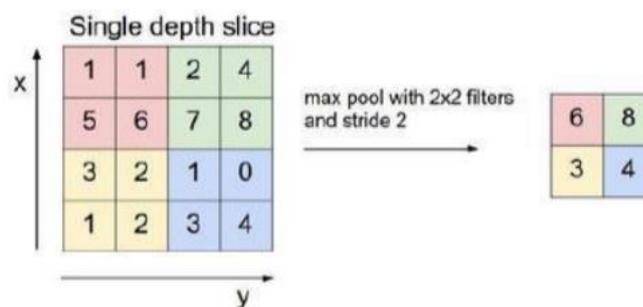
S = Stride

K = Kernel

$$\text{Output Volume} = V \times V \times K \dots\dots\dots(5)$$

## 2. Pooling Layer

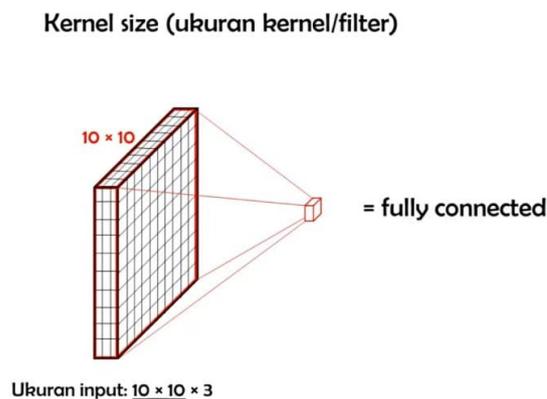
*Pooling layer* dilakukan untuk mengurangi varian, mengurangi kerumitan perhitungan dan ekstrak fitur tingkat rendah dari lingkungan sekitar. *Pooling layer* terdapat dua jenis yaitu *Max pooling* dan *Average pooling*. *Max pooling* mengekstrak fitur yang paling penting seperti tepi sedangkan *average pooling* lebih halus dibandingkan dengan *max pooling*. Meskipun keduanya digunakan untuk alasan yang sama, *max pooling* lebih baik untuk mengekstrak fitur. *Average pooling* terkadang tidak dapat mengekstrak fitur yang baik karena dibutuhkan semua untuk menghitung dan menghasilkan nilai rata-rata yang mungkin atau tidak penting untuk deteksi objek. *Average pooling* membawa semua ke dalam hitungan dan mengalirkannya ke lapisan berikutnya yang berarti semua nilai benar-benar digunakan untuk pemetaan fitur dan pembuatan keluaran yang merupakan perhitungan yang sangat umum. Jika tidak diperlukan semua masukan dari lapisan konvolusi maka akan mendapatkan akurasi yang buruk untuk *average pooling*. Pada klasifikasi objek *pooling layer* yang banyak digunakan adalah *average pooling*. Pemilihan *pooling layer* tergantung pada jenis *dataset* yang digunakan. Berikut Gambar 2.10 merupakan menunjukkan *pooling layer*.



Gambar 2.10 *Pooling Layer (max pool)*

## 3. Fully Connected Layer

Proses pada *Fully Connected Layer* berfungsi untuk melakukan proses klasifikasi yaitu dengan menggunakan *softmax* yang sesuai dengan proses yang digambarkan pada Gambar 2.11.



Gambar 2.11 *Fully Connected Layer*

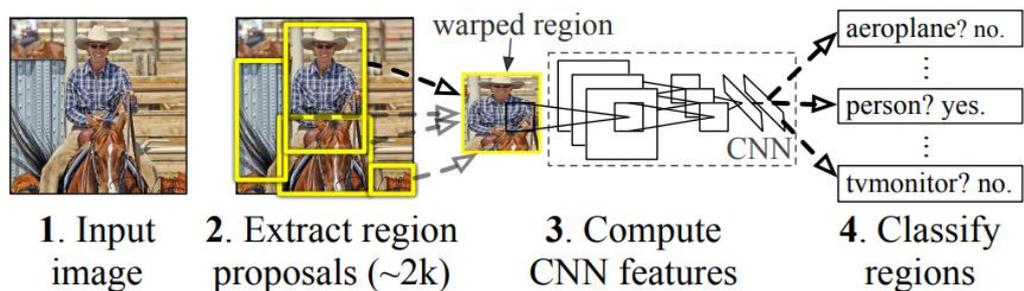
Dari Gambar 2.11 diatas, dimana *layer* 1 akan dilakukan *feed forwarding* menuju *layer* 2 dengan menggunakan fungsi aktivasi *ReLU*. Pada *layer* 2 akan dilakukan klasifikasi dengan menggunakan fungsi aktivasi *softmax*. Lapisan *Fully-Connected* biasanya digunakan pada metode Multi lapisan *Perceptron* dan bertujuan untuk mengolah data sehingga bisa diklasifikasikan. Perbedaan antara lapisan *Fully-Connected* dan lapisan konvolusi biasa adalah neuron di lapisan konvolusi terhubung hanya ke daerah tertentu pada input. Sementara lapisan *Fully-Connected* memiliki neuron yang secara keseluruhan terhubung. Namun, kedua lapisan tersebut masih mengoperasikan *dot product*, sehingga fungsinya tidak begitu berbeda.

#### 4. Aktivasi ReLU

Fungsi aktivasi ReLU atau *Rectified Linear Unit* merupakan lapisan aktivasi pada model CNN yang mengaplikasikan fungsi  $f(x)=\max(0,x)$  dimana fungsi ini melakukan *thresholding* dengan nilai nol terhadap nilai piksel pada input citra. Aktivasi ini membuat seluruh nilai piksel yang bernilai kurang dari nol pada suatu citra akan dijadikan 0 [19].

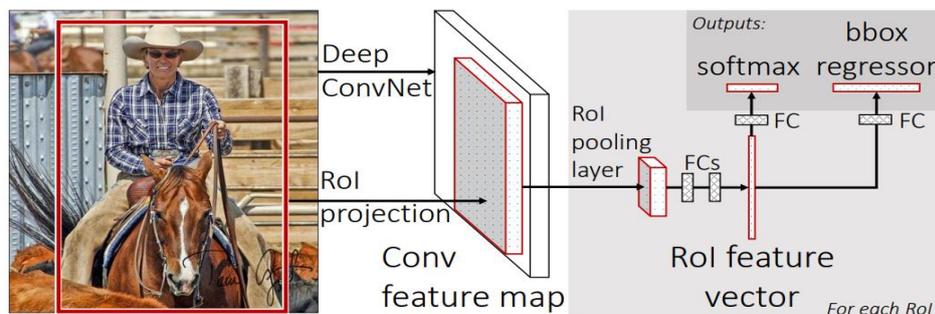
## 2.8. You Only Look Once (YOLO)

YOLO (*You Only Look Once*) merupakan algoritma dengan pendekatan baru dalam object detection / computer vision menggunakan jaringan syaraf tiruan (CNN). YOLO V1 diperkenalkan pertama kali melalui jurnal yang berjudul “*You Only Look Once: Unified, Real-Time Object Detection*” oleh Joseph Redmon, SantoshDivvala, Ross Girshick, dan Ali Farhadi pada tahun 2015 [22]. Ide pertama dari kreator YOLO adalah dengan mendesain sebuah algoritma yang memanfaatkan jaringan tunggal *Convolutional Neural Network* yang berguna memprediksi beberapa *bounding box* dan probabilitas kelas untuk *bounding box* tersebut, dan YOLO berlatih pada citra penuh secara langsung mengoptimalkan kinerja pendeteksian, yang dimana dengan cara ini bisa mengalahkan kecepatan algoritma pendahulunya seperti R-CNN dapat dilihat pada Gambar 2.12 , Fast R-CNN pada Gambar 2.13, dan Faster R-CNN dapat dilihat pada Gambar 2.14 berikut.



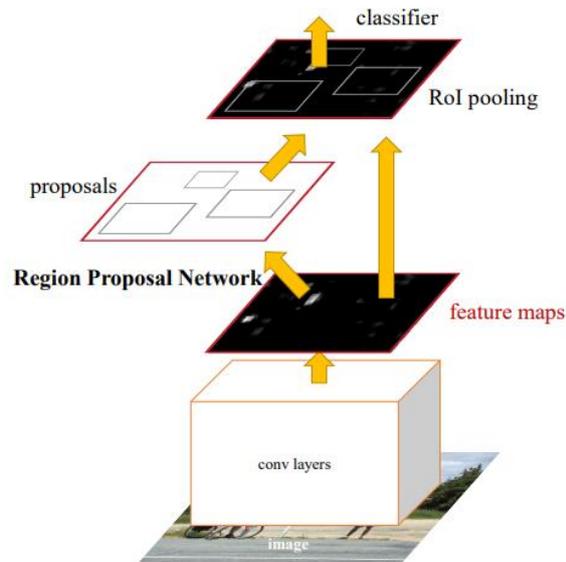
Gambar 2.12 Arsitektur R-CNN

Pada R-CNN masukkan gambar diolah dan diproses menggunakan *selective search* untuk mengekstrak sejumlah 2000 region proposals. Masing-masing proposal dimasukkan kedalam CNN untuk diklasifikasi [23].



Gambar 2.13 Arsitektur Fast R-CNN

Pada Fast R-CNN mengambil seluruh gambar masukkan dan diproses kedalam CNN kemudian menggunakan proposal *bounding box* untuk menghasilkan *feature map* menggunakan *selective search*. Selanjutnya, untuk setiap proposal objek, *Region of Interest (ROI) pooling layer* mengekstrak fitur dari *feature map*, yang kemudian setiap proposal diproses dan diklasifikasi seperti Gambar 2.14 [24].



Gambar 2.14 Arsitektur *Faster R-CNN*

Pada arsitektur *Faster R-CNN*, seperti *Fast R-CNN* yaitu mengambil keseluruhan gambar masukan dan memprosesnya kedalam CNN, tetapi pada proses pengambilan proposalnya menggunakan *Region Proposal Network*, hal ini lebih cepat dibandingkan menggunakan *Selective Search* yaitu 2s/gambar menggunakan CPU dan selanjutnya *feature map* setiap proposal diklasifikasi [25].

Perbandingan kecepatan model deteksi dapat dilihat pada Tabel 2.1 berikut.

Tabel 2.1 Kecepatan Model Deteksi [22]

Model	Kecepatan Deteksi	
R-CNN	0.05 FPS	20 s/gambar
FAST R-CNN	0.5 FPS	2 s/gambar
FASTER R-CNN	7 FPS	140 ms/gambar
YOLO	45 FPS	22 ms/gambar

Kerangka kerja YOLO dibagi menjadi 3 bagian utama, yaitu :

1. *Backbone* : *Convolutional Neural Network* yang menggabungkan dan membentuk fitur gambar pada berbagai jenis detail gambar.
2. *Neck* : Serangkaian lapisan jaringan yang mencampur dan menggabungkan fitur gambar dan meneruskannya ke lapisan prediksi.
3. *Head* : Menggunakan fitur dari *Neck* untuk menghasilkan kotak pembatas, dan memprediksi kategori.

Algoritma YOLO akan membagi citra dalam grid berukuran  $S \times S$ , apabila dalam *region S* terdapat objek maka region tersebut akan melakukan deteksi. Setiap region akan memprediksi *bounding box* dan peta kelas pada masing masing grid. Sehingga setiap *region* akan memiliki *bounding box* dan nilai *confidence*, secara umum nilai *confidence* memiliki persamaan sebagai berikut :

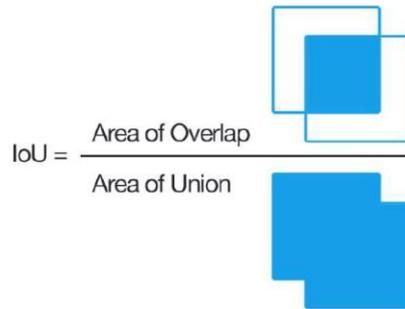
$$Confidence = Pr(Object) \times IoU \dots\dots\dots (6)$$

$Pr()$  merupakan nilai objektivitas, *Object* apabila tidak terdapat objek maka akan bernilai nol dan bernilai satu jika terdapat sebuah objek.  $IoU$  merupakan nilai rasio *Intersection Over Union* antara kotak *ground truth* dan kotak prediksi. *Truth* merupakan area kepastian objek sedangkan *pred* merupakan area kotak prediksi. *Setiap Bounding Box* berisi 5 nilai prediksi yaitu  $x, y, w, h, conf$ . Nilai  $x, y$  mewakili pusat kotak terhadap batas batas dari *region*. Nilai  $w, h$  mewakili nilai lebar dan tinggi dari keseluruhan citra. Sedangkan *conf* mewakili nilai *confidence* antara kotak *ground truth* dan kotak prediksi [22].

Selain itu *grid cell* juga memprediksi probabilitas kelas  $Pr(Class\ i\ | \ Object)$ . Probabilitas diberikan pada *region* yang terdapat sebuah objek, satu probabilitas hanya akan diberikan per satu *region grid*, terlepas dari jumlah kotak prediksi [22]. Pada saat pengujian, YOLO mengkalkulasi probabilitas kelas kondisional dengan prediksi kotak *confidence* dengan persamaan:

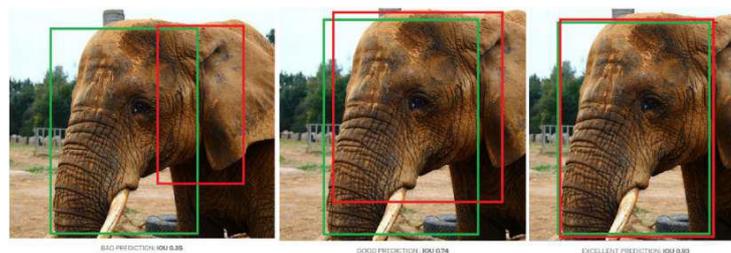
$$\Pr (Class i | Object) \times \Pr (Object) \times IOU = \Pr (Class i) \times IoU \dots\dots\dots(7)$$

Adapun parameter yang digunakan untuk mengukur model yaitu *intersection over union* yang ditunjukkan pada Gambar 2.15 berikut.



Gambar 2.15 *Intersection Over Union*

Pada Gambar 2.15 diatas menunjukkan IoU atau *Intersection Over Union* yaitu metrik yang mengukur akurasi model dalam mendeteksi objek pada kumpulan data latih. IoU membandingkan *bounding box ground truth* dan *bounding box prediksi* yang terletak pada suatu objek dalam citra gambar. Nilai IoU yang dihasilkan akan berfungsi sebagai nilai *confidence* untuk ke tahap proses *Non Max Supression*. Contoh IoU dapat dilihat pada Gambar 2.16 dan dapat didefinisikan dengan persamaan berikut:



Gambar 2.16 Contoh Hasil IOU

$$IoU = \frac{A \cap B}{A \cup B} \dots\dots\dots(8)$$

$$Precision = \frac{TP}{TP + FP} \dots\dots\dots(9)$$

$$Recall = \frac{TP}{TP + FN} \dots\dots\dots(10)$$

$$Akurasi = \frac{TP + TN}{TP + FP + TN + FN} \dots\dots\dots(11)$$

Dimana :

A = *Bounding Box Ground Truth*

B = *Bounding Box Prediksi*

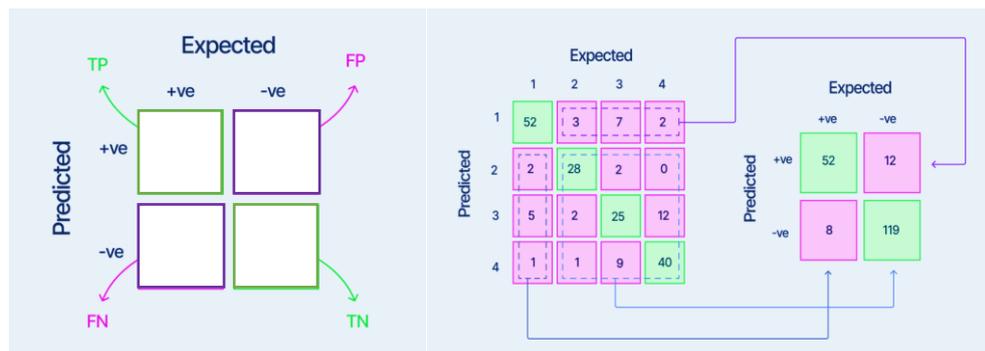
TP = *True Positive*

TN = *True Negative*

FP = *False Positive*

FN = *False Negative*

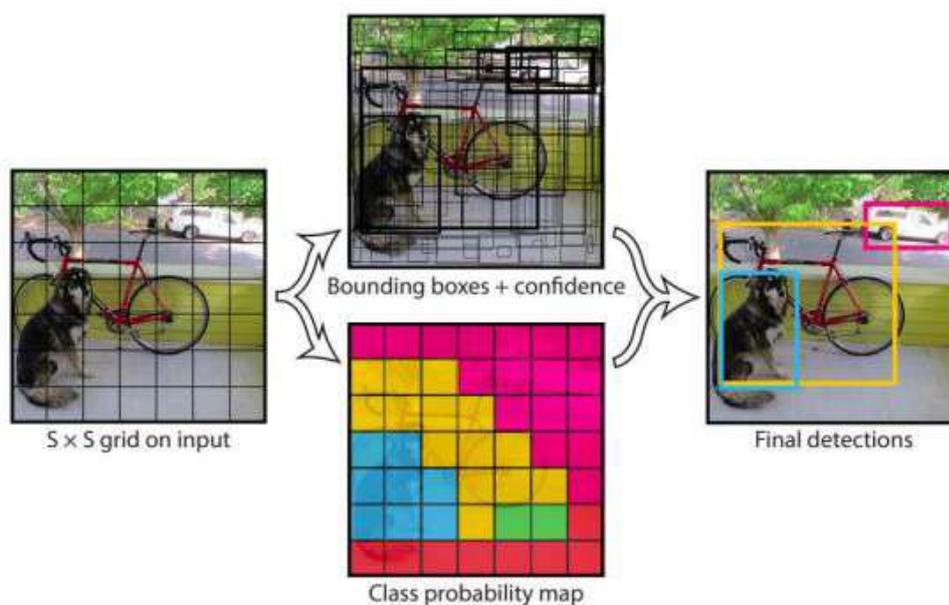
*True Positive* merupakan objek yang mengacu pada kelas positif dan diklasifikasikan benar, *True Negative* merupakan objek pada kelas negatif dan diklasifikasikan benar, *False Positive* merupakan objek kelas negatif dan diklasifikasikan salah sebagai kelas positif, dan *False Negative* merupakan objek dalam kelas positif yang diklasifikasikan salah sebagai kelas negatif. Adapun penentuan dari nilai-nilai tersebut yang memiliki *multiple classes* dapat dilihat pada Gambar 2.17 berikut [26].



Gambar 2.17 Penentuan nilai *confusion matrix*

IoU memiliki rentang nilai antara 0 dan 1. Nilai IoU yang mendekati 0 dianggap sebagai prediksi yang buruk, sedangkan nilai IoU yang mendekati 1 dianggap sebagai prediksi yang baik. Umumnya nilai *threshold* secara *default* akan di set 0.5, akan tetapi dalam penerapannya nilai *threshold* bisa di set berapapun sesuai kebutuhan penelitian. Biasanya nilai  $IoU > 0.5$  akan

dianggap sebagai prediksi yang baik. Dalam gambar diatas, gambar pertama dianggap sebagai *bad prediction* karena nilai  $IoU < 0.5$ , gambar kedua dianggap sebagai *good prediction* karena nilai  $IoU > 0.5$ , sedangkan gambar ketiga dianggap sebagai *excellent prediction* karena nilai  $IoU$  mendekati 1 atau  $IoU > 0.9$ . *Non Max Supression* berfungsi untuk menghilangkan *bounding box* yang memiliki nilai *overlap* / tumpang tindih kecil terhadap objek dalam suatu gambar. *Non Max Supression* akan memanfaatkan dua nilai yaitu  $IoU$  dan *Threshold*. *Threshold* merupakan nilai yang ditentukan oleh peneliti sebagai ambang batas. Apabila Nilai  $IoU$  dari suatu *bounding box* lebih kecil dari nilai *Threshold* maka *bounding box* tersebut akan di hilangkan. Berikut Gambar 2.18 merupakan konsep desain dari YOLO.



Gambar 2.18 Konsep Desain YOLO

*Loss function / cost function* merupakan fungsi yang digunakan untuk mengukur seberapa baik performa yang dihasilkan model dalam mendeteksi sebuah objek. Fungsi loss yang baik adalah fungsi yang menghasilkan error yang paling rendah. Jika model memiliki kelas lebih dari dua atau multiclass maka perlu adanya cara untuk mengukur perbedaan nilai antara probabilitas hasil prediksi dan probabilitas nilai aktual. Fungsi loss yang digunakan dalam algoritma YOLO adalah *Mean Square Error* (MSE). *Mean Square Error* (MSE) merupakan fungsi loss yang paling sering umum untuk digunakan dan

sering disebut juga sebagai L2 Loss, dimana cara kerjanya yaitu dengan menghitung rata-rata perbedaan kuadrat nilai aktual dan nilai prediksi. Hasil dari MSE akan selalu bernilai positif terlepas dari nilai prediksi dan nilai aktualnya. Untuk mencapai model dengan performa terbaik, peneliti harus mencoba mengurangi nilai L2 Loss sekecil mungkin atau mencapai nilai sempurna yaitu 0. Berikut merupakan rumus persamaan untuk menghitung nilai MSE [22] :

$$MSE = \sum_{i=0}^{S^2} 1_i^{obj} \sum_{c \in classes} (P_i(c) - (\hat{P})_i(c))^2 \dots\dots\dots(12)$$

Dimana :

MSE = Mean Square Error

S = Jumlah Grid

$1_i^{obj}$  = Objek, bernilai 1 jika terdapat objek, dan bernilai 0 jika tidak terdapat objek

$P_i(c)$  = Nilai Prediksi

$(\hat{P})_i(c)$  = Nilai Aktual

Adapun rumus dari *loss function* adalah sebagai berikut[22].

$$lf = a + b + c + d + e \dots\dots\dots(13)$$

$$\begin{aligned} & \lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^B 1_{ij}^{obj} [(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2] \\ & + \lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^B 1_{ij}^{obj} \left[ (\sqrt{w_i} - \sqrt{\hat{w}_i})^2 + (\sqrt{h_i} - \sqrt{\hat{h}_i})^2 \right] \\ & + \sum_{i=0}^{S^2} \sum_{j=0}^B 1_{ij}^{obj} (c_i - \hat{c}_i)^2 \\ & + \lambda_{noobj} \sum_{i=0}^{S^2} \sum_{j=0}^B 1_{ij}^{obj} (c_i - \hat{c}_i)^2 \\ & + \sum_{i=0}^{S^2} 1_{ij}^{obj} \sum_{c \in classes} (p_i(c) - \hat{p}_i(c))^2 \end{aligned}$$

Dimana :

lf = loss Function

a = Menghitung *loss* untuk x dan y kotak prediksi

b = Menghitung *loss* untuk *widht* dan *height* kotak prediksi

c = Menghitung *loss* untuk *confidence* kotak prediksi

d = Menghitung *loss* sel yang tidak memiliki objek

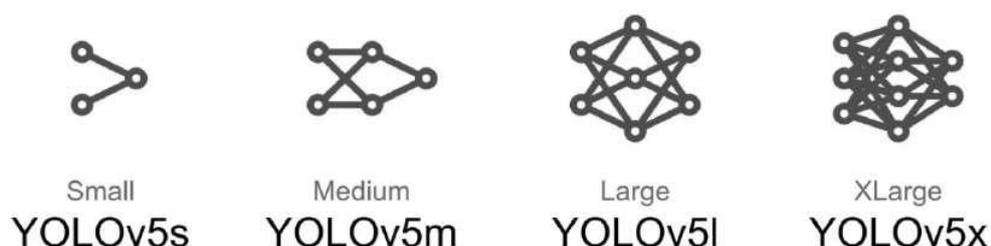
e = Menghitung *loss* YOLO

Tidak lama setelah YOLO V4 rilis pada April 2020, Glenn Jocher dan tim *Ultralytics* LLC melakukan publikasi versi terbaru dari Seri YOLO, versi terbaru tersebut diberi nama YOLO V5. Glenn Jocher merupakan peneliti, pengembang dan CEO *Ultralytics* LLC. Secara umum arsitektur YOLO V5 tidak jauh berbeda dengan YOLO versi sebelumnya. YOLO V5 dikembangkan menggunakan bahasa pemrograman *python*, bukan seperti YOLO versi sebelumnya yang menggunakan bahasa pemrograman C. Hal ini yang membuat instalasi dan integrasi pada perangkat IoT lebih mudah digunakan. Selain itu YOLO V5 dikembangkan menggunakan kerangka *library PyTorch* dimana memiliki komunitas yang lebih besar dari komunitas *Darknet*, yang menyebabkan *PyTorch* akan mendapatkan dukungan kontribusi dan potensi pertumbuhan yang lebih baik di masa depan. Meskipun dikembangkan dalam dua bahasa pemrograman dan kerangka kerja yang berbeda, kinerja antara YOLO V4 dan YOLO V5 tidak jauh berbeda, akan tetapi setelah beberapa saat YOLO V5 terbukti lebih berperforma baik akurasi maupun kecepatan daripada YOLO V4 dalam beberapa kasus, dan juga sebagian komunitas computer vision lebih percaya akan performa YOLO V5.

YOLO memiliki tiga kerangka kerja utama yaitu *backbone*, *neck* dan *head*. Bagian *backbone* YOLO V5 menggunakan CSPDarknet53 yang di adopsi dari CSPDarknet YOLO V4, bagian ini memecahkan pengulangan informasi gradien di *backbone* dan mengintegrasikan perubahan gradien ke dalam peta fitur sehingga dapat meningkatkan akurasi, mengurangi kecepatan inferensi, dan mengurangi ukuran bobot model dengan mengurangi jumlah parameter. Bagian *neck* YOLO V5 menggunakan *Path Aggregation Network* (PANet) untuk meningkatkan arus informasi. PANet ini pengembangan dari *Feature Pyramid Network* (FPN) di YOLO V3 yang mencakup lapisan bottom up dan top down, dengan ini dapat meningkatkan propagasi level rendah fitur dalam model. PANet meningkatkan lokalisasi di lapisan bawah, yang meningkatkan akurasi lokalisasi objek. Bagian *head* YOLO V5 sama dengan YOLO V3 dan

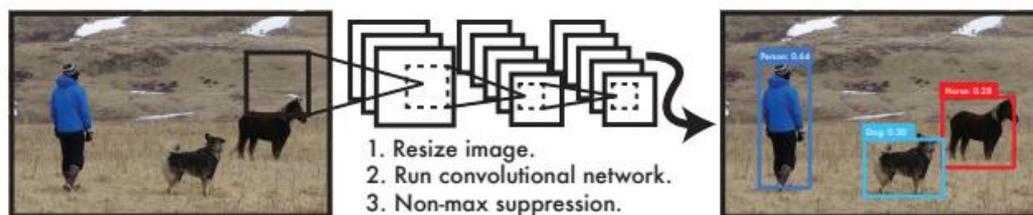
YOLO V4 yang menghasilkan tiga keluaran peta fitur yang berbeda untuk mencapai skala multi *prediction*, sehingga dapat membantu dalam meningkatkan prediksi objek kecil hingga besar secara efisien.

YOLO adalah metode deteksi objek *one-stage*. Terdapat beberapa versi *pretrained model* YOLO v5 diantaranya adalah YOLO v5s, YOLO v5m, YOLO v5l, dan YOLO v5x. Berikut Gambar 2.19 merupakan perbandingan *pre-trained model* YOLO V5.



Gambar 2.19 Model YOLO V5

Pada Gambar 2.19 diatas merupakan gambaran model dari tipe YOLO. Parameter merupakan variabel konfigurasi internal model yang nilainya diberikan dari data pretrained model yang digunakan. Model ini digunakan untuk menentukan keterampilan model dalam membuat prediksi. Semakin tinggi tipe YOLO maka akan memiliki jaringan *neural network* yang lebih banyak sehingga parameter yang digunakan semakin besar. Adapun proses deteksi dari YOLO dapat dilihat pada Gambar 2.20 berikut.



Gambar 2.20 Proses Deteksi YOLO

## 2.9. *Object Detection*

Deteksi objek adalah sebuah proses untuk menentukan di mana objek berada dalam gambar yang diberikan (lokalisasi objek) dan menentukan kategori mana saja yang dimiliki objek itu (klasifikasi objek). Pada umumnya, deteksi objek dapat dibagi menjadi tiga tahapan: pemilihan wilayah objek yang di deteksi, ekstraksi fitur, dan klasifikasi dari objek yang ditangkap. Dalam penerapannya, objek deteksi banyak digunakan dalam berbagai hal seperti klasifikasi objek pada gambar, deteksi rambu untuk autonomous car, dan lain sebagainya.

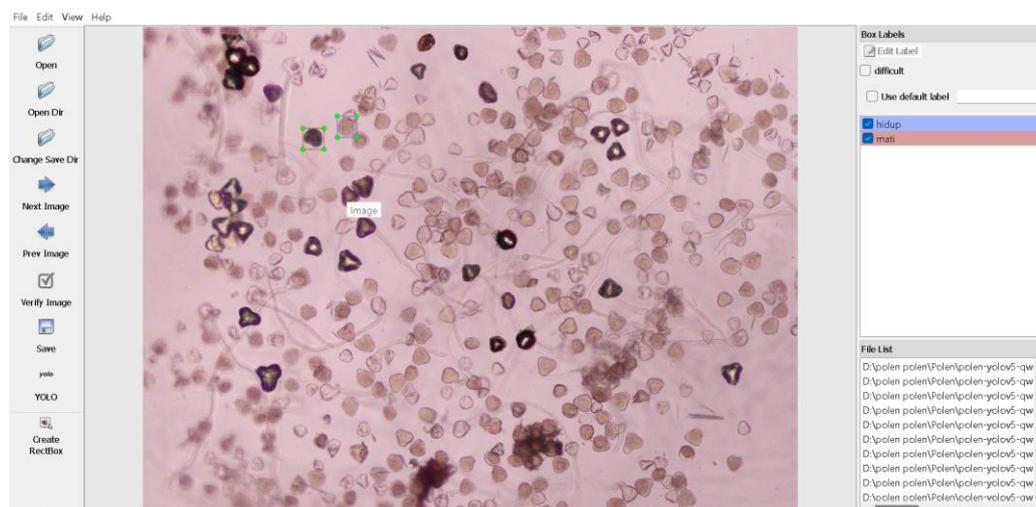
## 2.10. *Python*

Bahasa pemrograman *python* dipilih dalam penelitian, karena memiliki keunggulan seperti *readability*, efisien, multifungsi, interoperabilitas, dan memiliki dukungan komunitas yang memadai. *Readability*, dengan *python* memiliki *source code* yang sederhana, sehingga mudah ditulis, mudah diingat dan juga digunakan ulang. Hal ini memudahkan pengembangan aplikasi, mulai dari tahap pengkodean, pengujian, perbaikan jika ada kesalahan, *bug* atau *error* lainnya. Efisien, dengan menggunakan *python* memiliki *library* yang lengkap, *python code* akan lebih sederhana bila dibandingkan dengan kode yang ditulis dengan bahasa pemrograman lainnya seperti misalnya *java*, *C*, *C#* maupun *C++*. Multifungsi, dengan menggunakan *python* dapat membuat *website*, aplikasi jaringan, aplikasi bidang robotika, sampai dengan aplikasi kecerdasan buatan. *Python* memiliki banyak modul yang siap digunakan untuk pengembangan aplikasi sesuai kebutuhan. Interoperabilitas, dengan menggunakan *python* memiliki interoperabilitas yang tinggi karena mampu berinteraksi dengan bahasa-bahasa pemrograman lain. Program yang dikembangkan *python* dapat diperasionalisasi pada hampir semua sistem operasi baik Windows, Linux, Mac OS, Unix, dan juga sistem operasi pada perangkat lunak berbasis mobile seperti Android atau IOS. Dukungan Komunitas, dengan menggunakan *python* memiliki dukungan komunitas

yang sangat kuat, karena *python* bersifat *open source*. Dengan komunitas yang baik, mempermudah pengguna untuk saling berbagi, dan mengembangkan bahasa pemrograman *python* menjadi bahasa yang handal.

## 2.11. Pelabelan Data

*Image labelling* atau pelabelan gambar merupakan suatu proses memberikan keterangan berupa nama atau kata kunci pada suatu objek yang bertujuan untuk pelatihan model. Memberikan anotasi gambar ini dengan cara memberikan pembatas pada objek tertentu yang akan dikenalkan. Dalam hal ini menggunakan anotasi *rectangular boxes* yaitu dengan memberikan kotak pembatas pada objek yang akan dikenali oleh mesin dengan dua kelas yaitu polen hidup dan polen mati dengan format YOLO menggunakan aplikasi labelimg. Proses pelabelan data menggunakan labelimg dapat dilihat pada Gambar 2.20 berikut.



Gambar 2.21 Pelabelan Dataset

## 2.12. Streamlit

*Streamlit* merupakan *open-source framework* dengan bahasa pemrograman *Python* yang memudahkan pengembangan aplikasi web pada *machine learning* karena memiliki fitur untuk mempermudah pengembangan model

*machine learning* dan dapat dengan mudah untuk disesuaikan dengan keinginan. Streamlit menawarkan visualisasi melalui antarmuka yang interaktif dan dapat mengkomunikasikan informasi yang dapat dengan mudah dipahami oleh pengguna. Proses *deploy* pada streamlit cukup mudah melalui platform cloud-sharing Streamlit dan dapat diakses di berbagai *platform*.

### **2.13. Visual Studio Code**

*Visual Studio Code (VS Code)* merupakan sebuah teks editor ringan dan handal yang dibuat oleh Microsoft untuk sistem operasi *multiplatform* yang mana dapat digunakan untuk berbagai sistem operasi seperti *Linux, Mac,* dan *Windows*. *VS Code* mendukung bahasa pemrograman *Java Script, Typescript,* dan *Python* serta bahasa pemrograman lainnya seperti *Node.js, C++, C#, Java* dan lainnya. Banyak fitur yang disediakan oleh *VS Code* diantaranya *Intellisense, Git Integration, Debugging,* dan fitur ekstensi yang menambah kemampuan teks editor. Dalam hal ini, menggunakan *VS Code* dengan bahasa pemrograman *python* sebagai *code editor*.

## **BAB III METODE PENELITIAN**

### **3.1. Waktu dan Tempat Penelitian**

Penelitian ini dimulai pada November 2022 sampai dengan April 2023. Penelitian ini dilakukan di PT. RPN Pusat Penelitian Kelapa Sawit Unit Marihat, Kabupaten Simalungun, Sumatera Utara dan Laboratorium Terpadu Jurusan Teknik Elektro Universitas Lampung.

### **3.2. Alat dan Bahan**

Adapun alat dan bahan yang akan digunakan dalam penelitian ini adalah sebagai berikut.

1. Citra Polen Bunga Kelapa Sawit.  
Data citra polen bunga kelapa sawit digunakan sebagai data latih yang telah diberi label terlebih dahulu, data validasi digunakan pada saat proses pelatihan model, dan data uji digunakan untuk melakukan pengujian dari hasil pelatihan model.
2. Super Komputer DIKTI AI *Center* dengan *Graphics Card* Nvidia DGX-A100 2x40 GB dan CPU 16 *core* AMD EPYC 7742 dengan RAM 128 GB yang berjalan pada sistem operasi Ubuntu 20.04.  
Super komputer ini digunakan sebagai media untuk melakukan proses pelatihan model menggunakan YOLO V5.
3. Laptop Lenovo IdeaPad 3 14ALC6 dengan Prosesor AMD Ryzen 7 5700U Radeon *Graphics* 1.80 GHz RAM 8 GB.

Laptop digunakan sebagai media untuk menjalankan hasil pelatihan model yang telah dilakukan dan melakukan *deployment* ke aplikasi web *streamlit*.

4. *Visual Studio Code*.

*Visual Studio Code* digunakan sebagai media menulis dan mengedit dengan menggunakan bahasa pemrograman *python*.

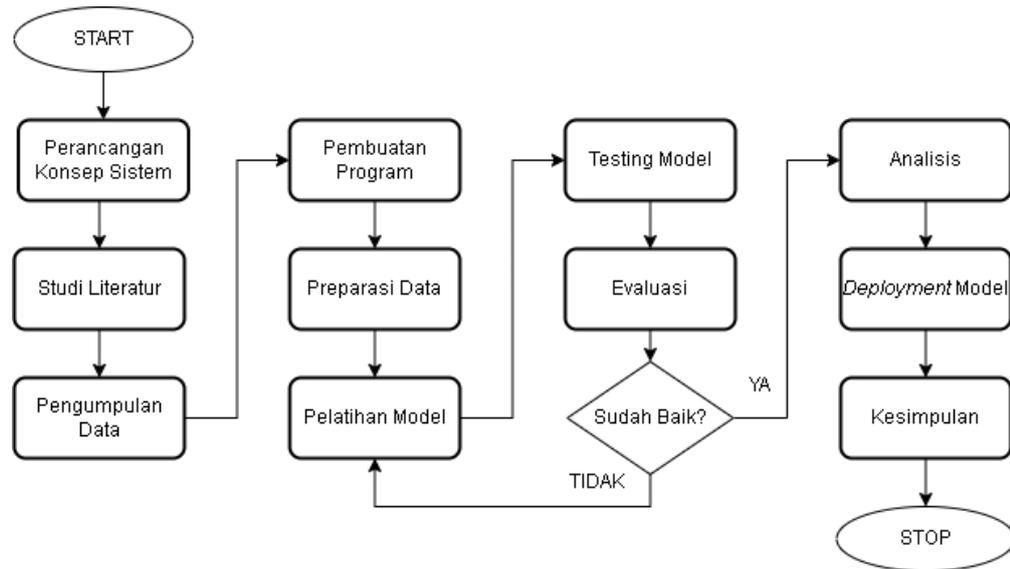
5. Perangkat lunak *python* 3.10.2.

*Python* digunakan sebagai bahasa pemrograman utama untuk membangun sistem deteksi dan hitung polen.

### **3.3. Prosedur Penelitian**

Prosedur penelitian ini dimulai dengan mencari topik atau ide perancangan sistem. Dengan adanya ide atau topik perancangan sistem, penelitian dilanjutkan untuk mencari dan mempelajari literatur yang berkaitan dengan topik perancangan sistem untuk menentukan spesifikasi sistem. Jika spesifikasi sistem tidak tersedia atau kurang efektif, maka pencarian literatur dilakukan kembali agar mendapatkan spesifikasi sistem yang efektif.

Penelitian dilanjutkan dengan merancang sistem dan dilanjutkan dengan pengujian sistem. Jika pengujian sistem tidak berhasil, maka perancangan sistem perlu dilakukan kembali agar sistem yang diharapkan dapat bekerja dengan baik saat diuji kembali. Setelah pengujian sistem berhasil, penelitian dilanjutkan dengan pengambilan data. Data yang sudah diambil akan dilakukan analisis untuk mendapatkan kesimpulan dari sistem yang telah berhasil dibuat. Berikut Gambar 3.1 merupakan diagram alir prosedur penelitian yang dilakukan.



Gambar 3.1 Diagram alir prosedur penelitian

Berdasarkan Gambar 3.1 Diagram alir prosedur penelitian diatas dapat dijelaskan tahapannya sebagai berikut.

1. Perancangan Konsep Sistem

Tahapan ini bertujuan untuk membuat sebuah kerangka pembuatan sistem yang akan digunakan sebagai acuan untuk melakukan penelitian ini.

2. Studi Literatur

Tahap selanjutnya yaitu menyempurnakan kerangka pembuatan sistem yang telah dirancang sebelumnya dengan mencari referensi penelitian terdahulu dan mencari referensi pembuatan sistem yang relevan untuk mendapatkan informasi yang dapat mendukung penelitian ini.

3. Pengumpulan Data

Tahap ini merupakan proses pengumpulan data dari pengguna untuk keperluan pelatihan model dan pengujian sistem deteksi dan hitung polen untuk mengetahui viabilitasnya.

4. Pembuatan Program

Tahap ini merupakan proses mencari referensi dan pengembangan program menggunakan *python* untuk membangun Sistem Pendeteksi Dan Penghitung Polen Hidup Dan Mati Pada Tanaman Kelapa Sawit Berbasis *Artificial Intelligence* Menggunakan Algoritma YOLO V5.

#### 5. Preparasi Data

Tahap ini merupakan proses memberikan label pada citra polen yang telah dikumpulkan dengan dua kelas yaitu hidup dan mati serta membaginya kedalam tiga bagian yaitu data latih, data validasi dan data uji.

#### 6. Pelatihan Model

Tahap ini merupakan proses pelatihan model (*training*) dengan menggunakan data yang sudah terlabel dari data latih dan validasi untuk mendapatkan bobot sesuai dengan tujuan dari penelitian ini.

#### 7. Pengujian Model

Tahap ini bertujuan untuk melakukan pengujian terhadap model yang telah dilatih untuk mengetahui tingkat akurasi.

#### 8. Evaluasi

Tahap ini bertujuan untuk mengevaluasi hasil dari pelatihan model yang sudah dilakukan pada tahap sebelumnya untuk mendapatkan hasil yang lebih akurat.

#### 9. Analisis

Tahap ini merupakan proses untuk memastikan hasil akurasi apakah sudah sesuai dengan tujuan dari penelitian ini yaitu dapat mendeteksi dan menghitung polen yang kemudian dilanjutkan ke tahap *deployment* model.

#### 10. *Deployment* Model

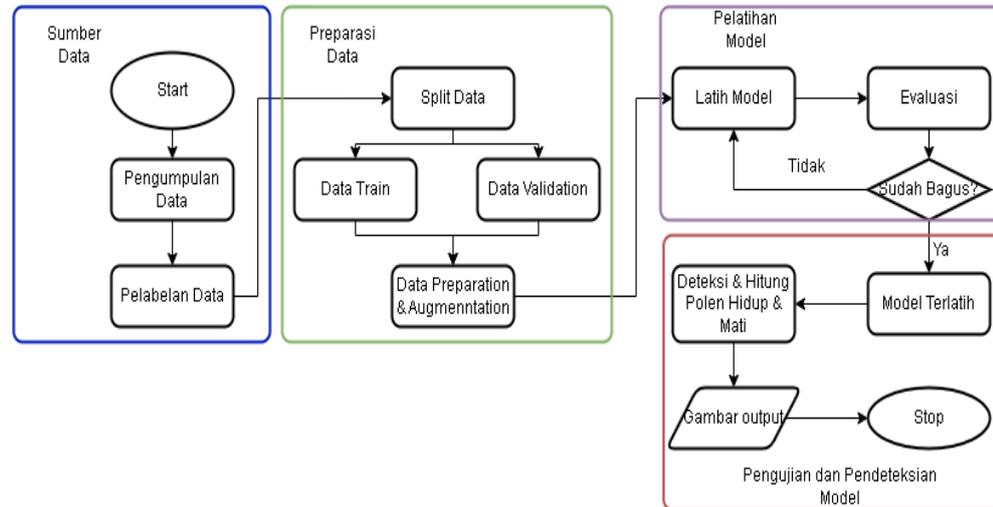
Tahap ini bertujuan untuk mentransfer bobot yang didapatkan dari hasil pelatihan model dengan aplikasi web *streamlit* agar memudahkan pengguna dalam menggunakan sistem ini.

#### 11. Kesimpulan

Tahap ini merupakan tahap akhir yaitu menarik kesimpulan berdasarkan hasil penelitian dan pengujian yang telah dilakukan.

### 3.4. Perancangan Program Pengolahan Citra

Perancangan program untuk mengolah citra dijelaskan melalui diagram alir pada Gambar 3.2 berikut.



Gambar 3.2 Diagram Alir Perancangan Program Pengolahan Citra

Berdasarkan Gambar 3.2 tahapan untuk mengolah citra adalah sebagai berikut:

#### 1. Pengumpulan data

Data yang diambil berupa citra gambar polen bunga jantan kelapa sawit dengan format JPG (*Joint Photographic Group*) yang ditangkap melalui kamera mikroskop sejumlah 415 data termasuk data yang sudah augmentasi dan akan di *resize* sebesar 640x640 pixels.

#### 2. Pelabelan Data

Setelah data terkumpul, maka dilakukan proses pelabelan data menggunakan anotasi *bounding box* yaitu proses pelabelan objek dengan *rectangular boxes* melalui aplikasi labelimg. Proses pelabelan data ini mengklasifikasikan objek kedalam dua kelas yaitu hidup dan mati dengan format anotasi YOLO.

#### 3. Preparasi Data

Proses ini bertujuan untuk membagi data kedalam dua bagian yaitu data latih, data validasi dan data uji. Dimana data latih merupakan data yang digunakan untuk melatih model dalam *deep learning*, data

validasi merupakan data yang digunakan untuk mengukur kemampuan model yang telah dilatih dalam memprediksi suatu objek, data latih dan data validasi ini digunakan pada saat proses pelatihan model sedangkan data uji merupakan data yang digunakan untuk menguji model yang telah dilatih apakah memiliki akurasi yang baik atau belum dalam mengenali objek.

4. Pelatihan model merupakan proses untuk mengajarkan dan mengenalkan mesin agar dapat mendeteksi objek dan menghitungnya sesuai dengan data latih dan data validasi yang sudah diberi label dengan kelas yang telah ditentukan, yaitu polen hidup dan mati. Pada saat proses pelatihan model ini menggunakan model YOLO V5l (Large) dengan *resize* ukuran gambar sebesar 640 x 640 pixels.

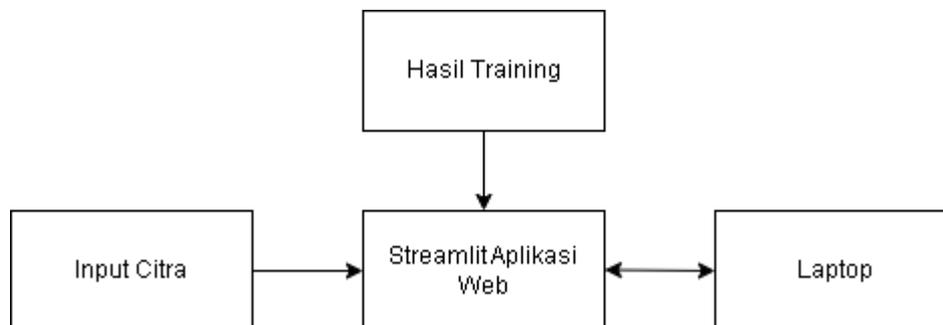
5. Pengujian Model

Hasil dari pelatihan model (*training* model) ini berupa bobot (*weight*) selanjutnya dilakukan pengujian dengan data tes (*data testing*). Proses deteksi ini didapatkan setelah melakukan pelatihan data menggunakan model YOLO V5l untuk dapat mengklasifikasikan polen hidup dan mati secara otomatis. Proses selanjutnya yaitu mesin dapat melakukan penghitungan polen berdasarkan deteksi *bounding box* yang dilakukan mesin setelah dilakukan pelatihan data, yang kemudian hasil penghitungan tersebut ditampilkan dalam citra polen untuk memudahkan pengguna mengetahui viabilitasnya, proses ini dilakukan dengan menggunakan aplikasi web *streamlit* dengan minimal *confidence threshold* 0,5 untuk menghindari terjadinya deteksi ganda pada objek. Apabila belum memiliki hasil yang akurat maka dilakukan evaluasi dan dilakukan pelatihan data kembali hingga mendapatkan hasil yang akurat.

### 3.5. Perancangan Sistem Aplikasi Web

Proses perancangan sistem aplikasi web ini melakukan *deployment* hasil pelatihan model dengan aplikasi web *streamlit*. Pengguna akan mengakses

aplikasi web tersebut dan memasukkan citra polen kemudian di proses oleh aplikasi web *streamlit* yang sudah dilakukaN konfigurasi di laptop pengguna dan *outputnya* akan tersimpan di laptop pengguna. Berikut Gambar 3.3 merupakan diagram blok perancangan sistem.



Gambar 3.3 Diagram Blok Perancangan Sistem

Berdasarkan Gambar 3.3 dapat diketahui bahwa proses awal yang dilakukan pada saat melakukan *deployment* yaitu mengambil bobot (*weight*) dari hasil training yang telah dilakukan dan menghubungkannya dengan program *streamlit* aplikasi web yang telah dikembangkan dan dikonfigurasi dengan laptop pengguna. Setelah bobot dari hasil pelatihan model terkonfigurasi dengan *streamlit* aplikasi web dan maka pengguna dapat memasukkan gambar polen kelapa sawit yang kemudian akan diproses oleh *streamlit* aplikasi web dan hasilnya akan muncul di laptop pengguna. Laptop pengguna yang telah *disetting* tersebut akan mengambil data dari *input* citra, memproses gambar polen tersebut menggunakan *streamlit* aplikasi web dan menampilkan hasil deteksi dan penghitungan polen.

## **BAB V**

### **KESIMPULAN DAN SARAN**

#### **5.1. KESIMPULAN**

Adapun kesimpulan dari penelitian ini adalah sebagai berikut.

1. Penelitian ini bertujuan untuk membangun sebuah sistem yang dapat memprediksi dan menghitung jumlah polen pada bunga jantan kelapa sawit yang bertujuan untuk mengetahui viabilitas dari polen tersebut. Pada penelitian ini telah dilakukan perbandingan pelatihan model menggunakan 4 *pre-trained* model YOLO V5 untuk mendeteksi polen dan didapatkan hasil presisi sebesar 91,21% untuk YOLO v5s (*small*) dengan waktu latih model 0,743 jam, 91,83% untuk YOLO v5m (*medium*) dengan waktu latih model 0,927 jam, 92,04% untuk YOLO v5l (*large*) dengan waktu latih model 1,197 jam dan 91,96% untuk YOLO v5x (*xlarge*) dengan waktu latih model 1,699 jam. Berdasarkan nilai-nilai tersebut, maka bobot yang dipilih untuk melakukan pengujian yaitu YOLO v5l (*large*) dengan nilai presisi 92,04%.
2. Pada hasil pengujian pertama hingga kelima citra polen kondisi normal menggunakan bobot YOLO v5l didapatkan nilai presisi total polen hidup sebesar 99,59%, nilai *recall* total polen hidup sebesar 97,88%, nilai akurasi total polen sebesar 97,80% dan waktu deteksi rata-rata didapatkan selama 2,517 detik. Kecerahan dan warna *background* memengaruhi hasil deteksi dan hitung polen. Setiap citra memiliki ciri yang berbeda sehingga perlu mengatur nilai *confidence threshold* pada sistem. Setiap citra polen dilakukan sebanyak 4 kali pengujian dengan melakukan rotasi dan hasilnya berpengaruh terhadap hasil deteksi dan perhitungan objek sehingga memiliki nilai viabilitas yang berbeda.

## 5.2. SARAN

Adapun saran dari penelitian ini yaitu memperbanyak dataset dengan kondisi *background* yang berbeda, memperbanyak proses augmentasi dan menambahkan objek polen mati untuk memperkaya fitur yang akan dikenalkan oleh mesin sehingga akan mendapatkan model yang lebih baik dalam mendeteksi objek pada citra. Pada saat proses pelatihan model, diperlukan *device* yang memiliki *Graphics Processing Unit* (GPU) agar mempercepat proses pelatihan.

## DAFTAR PUSTAKA

- [1] Indrasen, P., "How Artificial Intelligence in Impacting Real Life," *International Journal of Advance Research and Development*, vol. 2, no. 10, pp. 96-100, 2017.
- [2] Firdaus, M., & Lubis, I., "Analisis Produksi Kelapa Sawit (*Elaeis Guineensis* Jacq.) Di Kebun Buatan, Kabupaten Pelalawan, Riau," *Buletin Agrohorti*, vol. 6, no. 2, pp. 281–286, 2018.
- [3] Sobari, E., Hasibuan, A. A., Subandi, M., & Rusli, M. D., "Peningkatan Buah Kelapa Sawit ( *Elaeis Guinensis* Jacq. ) dengan Memanfaatkan Ukuran Pollen dan Waktu dalam Penyerbukan Buatan," *Zuriat*, vol. 20, no. 2, pp. 62–66, 2018.
- [4] Sobari, E., A.A. Hasibuan, M. Subandi, "Effect of pollen sizes on number of oil palm (*Elaeis guinaensis* Jacq.) in artificial pollination," *Kultivasi*, vol. 18, no. 1, pp. 805-810, 2019.
- [5] Agus, E., P., "Meningkatkan Fruit Set Kelapa Sawit dengan Teknik Hatch and Carry *Elaeidobius Kamerunicus*, Indonesian Oil Palm Research Institute," 2012.
- [6] Fancies, M., L., Ata, M., M., and Mohamed, M., A., " A Robust Multiclass 3D Object Recognition Based on Modern YOLO Deep Learning Algorithms," John Wiley & Sons, Ltd., 2021.
- [7] Kubera, E., et al., "Detection and Recognition of Pollen Grains in Multilabel Microscopic Images," *Sensors*, vol. 22, no. 7, pp. 1-12, 2022.
- [8] Abdullah, Sikandar A., Zihauallah, K., Ali, H., Ali, A., and Hee-Cheol, K., "Computer Vision Based Deep Learning Approach for the Detection and Classification of Algae Species Using Microscopic Images," *MDPI*, vol. 14, no. 14, pp. 2-14, 2022.
- [9] Fakhrizal, T., "Morofologi Serbuk Sari *Familia Poacea* di Kampus Universitas Syiah Kuala Banda Aceh," *Jurnal Biotik*. vol. 3, no. 2, pp. 116-127, 2015.
- [10] Sudarmono & Sahromi, "Pollen Atau Serbuk Sari : Aspek Morfologi, Sistematika Dan Aplikasinya Pada Tumbuhan Keluarga Mentol," *Jurnal Sains Natural*, vol. 2, no. 1, pp. 12-16, 2012.
- [11] Wikaria, G., Haryono, S., Jenny, O., "Penerapan Metode Konvolusi Dalam Pengolahan Citra Digital," *Jurnal Ilmiah Matematika, Statistika (MatStat)*, vol. 12, no. 2, pp. 103-113, 2012.

- [12]Munantri, N., Z., Sofyan, H., & Yanu, M., "Aplikasi Pengolahan Citra Digital Untuk Identifikasi Umur Pohon," *Telematika*, vol. 16, no. 2, pp. 97-104, 2019.
- [13]Prabowo, D. A., & Abdullah, D., "Deteksi dan Perhitungan Objek Berdasarkan Warna Menggunakan *Color Object Tracking*," *Pseudocode*, vol. 5, no. 2, pp. 85-91, 2018.
- [14]Pariwat, O., "*Artificial Intelligence, Machine Learning and Deep Learning*," IEEE, pp. 1-6, 2017.
- [15]Kusuma, P., D., "*Machine Learning Teori, Program, dan Studi Kasus*," Yogyakarta: Deepublish, 2020.
- [16]Fikriya, Z., A., Irawan, M., I., and Soetrisno, "Implementasi *Extreme Learning Machine* untuk Pengenalan Objek Citra Digital," *J. Sains dan Seni ITS*, vol. 6, no.1, pp. A18-A23, 2017.
- [17]Pamungkas, F., S., Prasetya, B., D., and Kharisudin, I., "Perbandingan Metode Klasifikasi *Supervised Learning* pada Data Bank *Customers* Menggunakan Python," *Prosiding Seminar Nasional Matematika*, vol. 3, pp. 689-694, 2020.
- [18]Nurhayati, Busman, & Iswara, R. P., "Pengembangan Algoritma *Unsupervised Learning Technique* Pada *Big Data Analysis* di Media Sosial sebagai media promosi *Online* Bagi Masyarakat," *J. Tek. Inform.*, vol. 12, no.1, pp. 79-96, 2019.
- [19]Ilahiyah, S., dan Nilogiri, A., " Implementasi *Deep Learning* Pada Identifikasi Jenis Tumbuhan Berdasarkan Citra Daun Menggunakan *Convolutional Neural Network*," *JUSTINDO*, vol. 3, no. 2, pp. 49-56, 2018.
- [20]Bernard, F., "*Guest Editorial: Discovery and Artificial Intelligence*," *American Journal of Roentgenology*," vol. 209, no. 6, pp. 1189-1190, 2017.
- [21]Li, F. F., Johnson, J., & Yeung, S., "*Convolutional neural networks (Lecture notes)*," *Computer Science*, Stanford University, 2019.
- [22]Joseph, R., Santosh D., and Ross G., "*You Only Look Once: Unified, Real-Time Object Detection*," IEEE, 2016.
- [23]Ross, G., Jeff D., Trevor, D., & Jitendra Malik, "*Rich feature hierarchies for accurate object detection and semantic segmentation*," *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 580-587, 2014.
- [24]Ross, G., "Fast R-CNN," *IEEE International Conference on Computer Vision (ICCV)*, pp. 1440-1448, 2015.

- [25] Shaoqing, R., Kaiming, H., Ross, G, Jian, S., “*Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks*,” *IEEE Trans Pattern Anal Mach Intell.* vol. 39, no. 6, pp. 1137-1149, 2016.
- [26] V7 Labs. 13 September 2022. *Confusion Matrix: How To Use It & Interpret Results [Examples]*. Diakses pada 15 Februari 2023, dari <https://www.v7labs.com/blog/confusion-matrix-guide>.