

BAB II

LANDASAN TEORI

2.1 Kelapa Sawit

Tanaman kelapa sawit merupakan salah satu jenis tanaman perkebunan yang menduduki posisi penting dalam sektor pertanian dan sektor perkebunan. Kelapa sawit merupakan komoditi andalan Indonesia yang perkembangannya demikian pesat. Lahan yang optimal untuk kelapa sawit harus mengacu pada tiga faktor yaitu lingkungan, sifat fisik lahan dan sifat kimia tanah atau kesuburan tanah. Tanaman kelapa sawit di perkebunan komersial dapat tumbuh dengan baik pada kisaran suhu 24-28oC.

Proses produksi dikatakan baik apabila proses tersebut menghasilkan produk yang memenuhi standar yang telah ditetapkan Namun pada kenyataannya dalam proses produksi masih sering terjadi berbagai penyimpangan dan hambatan yang mengakibatkan produk dianggap cacat Mangoensoekardjo dan Semangun (2008) dikutip oleh Hudori (2015) mengatakan bahwa parameter kualitas yang diperhitungkan dalam standar perdagangan *crude palm oil (CPO)* adalah kadar free fatty acid (FFA),Kadar FFA menentukan kualitas nya,semakin tinggi kadar FFA buah sawit menunjukan bahwa CPO yang di hasilkan mengalami degradasi,sehingga kualitas nya menurun. [1]. Salah satu alat pengendalian yang dapat digunakan dengan pengendalian kualitas secara statistik yang dilakukan menggunakan alat bantu Dalam menjamin kualitas CPO agar berada pada kondisi baik dan stabil maka perlu dilakukan pengendalian kualitas melalui proses statistik

dikenal dengan seven tools. Seven tools merupakan metode grafik paling sederhana untuk menyelesaikan masalah termasuk menyelesaikan masalah yang berhubungan dengan CPO agar bisa menjaga dan memperbaiki kualitas produk sesuai dengan yang diharapkan [2].

2.2 Deteksi Ojek

Deteksi objek adalah cabang dari visi komputer yang bertujuan untuk mengidentifikasi dan menentukan lokasi objek dalam sebuah gambar atau video. Teknik ini menggabungkan klasifikasi objek dan lokalisasi objek dalam satu kerangka kerja. Model *deep learning* modern, seperti *Yolo (You Only Look Once)*, sangat populer dalam tugas deteksi objek karena efisiensinya dalam pengolahan real-time. Deteksi objek (Object Detection) adalah teknik visi komputer untuk menemukan contoh objek dalam gambar atau video. Konsep deteksi objek ini memegang peranan penting karena dapat di manfaatkan dalam berbagai hal, misalnya mendeteksi pelanggar lalu lintas, melatih mobil otonom, serta termasuk dalam deteksi kematangan buah sawit. Tujuan deteksi objek adalah untuk mereplikasi kecerdasan yang dimiliki manusia dalam melihat benda menggunakan komputer. Cara kerja deteksi objek adalah deteksi objek menempatkan keberadaan objek dalam gambar dan menggambar kotak pembatas di sekitar objek itu. memperkirakan kotak pembatas dan memberikan skor kepercayaan. Sel ini memprediksi probabilitas kelas untuk menetapkan kelas dari setiap objek. Deteksi akhir ini akan terdiri dari kotak pembatas unik yang sesuai dengan objeknya. [3]

2.3 *Deep Learning*

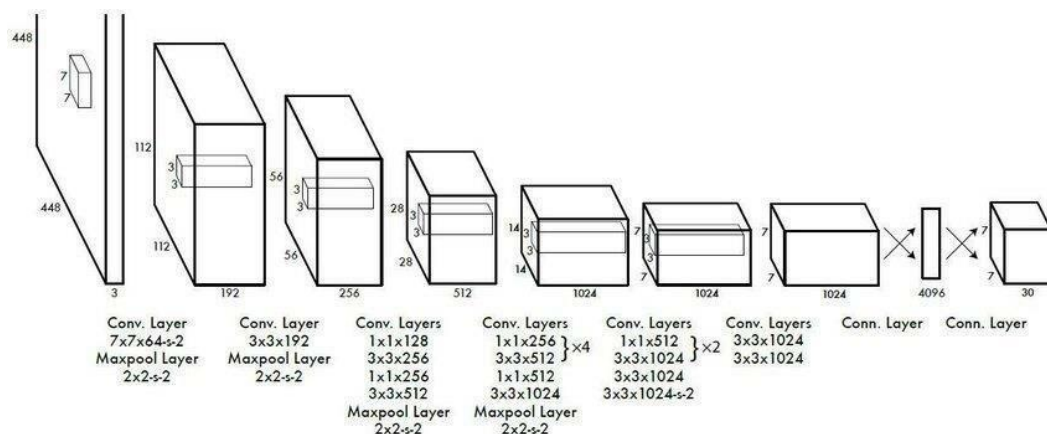
Menurut Batubara (Pemelajaran dalam (bahasa Inggris: Deep Learning) atau sering dikenal dengan istilah pemelajaran struktural mendalam (bahasa Inggris: deep structured learning) atau pemelajaran hierarki (bahasa Inggris: hierarchical learning) adalah salah satu cabang dari ilmu pemelajaran mesin (bahasa Inggris: machine learning) yang terdiri algoritma pemodelan abstraksi tingkat tinggi pada data menggunakan sekumpulan fungsi transformasi non-linear yang ditata secara berlapis - lapis dan mendalam [4]. Pada Deep Learning lapisan representasi tersebut bernama *Neural Networks*. *Neural Networks* memiliki struktur yang bertumpuk yang berarti suatu lapisan akan berada diatas lapisan lain. *Neural Networks* mengambil konsep dari bidang neurobiology dan terinspirasi dari kemampuan dalam memahami sesuatu seperti yang dilakukan oleh otak manusia. Meskipun konsep utama *Deep Learning* dikembangkan atas inspirasi tersebut, tetapi model *Deep Learning* bukanlah model dari otak manusia. Hal tersebut dikarenakan belum ada bukti yang menunjukkan bahwa otak manusia bekerja layaknya model deep learning seperti sekarang ini^[4].

2.4 *Algoritma Yolo (you only look once)*

You Only Look Once (YOLO) adalah salah satu pendekatan untuk melakukan pendeteksian objek *secara real-time* berbasis *Convolutional Neural Network*. *Yolo* menggunakan pendekatan jaringan syaraf tunggal (*Single neural network*) untuk melakukan pendeteksian objek pada sebuah frame. Jaringan ini

menggunakan fitur dari semua gambar untuk memprediksi setiap bounding box yang dapat melakukan prediksi pada kotak-kotak pembatas dan probabilitas secara langsung dalam satu evaluasi [5].

Jaringan deteksi *Yolo* memiliki 24 lapisan konvolusi (*convolutional layer*) yang diikuti oleh 2 lapisan yang terhubung penuh (*fully connected layer*). Beberapa lapisan konvolusi menggunakan lapisan reduksi 1x1 sebagai alternative dalam mengurangi kedalaman feature maps yang diikuti oleh 3x3 lapisan konvolusional (*convolutional layer*) seperti pada Gambar 2.1

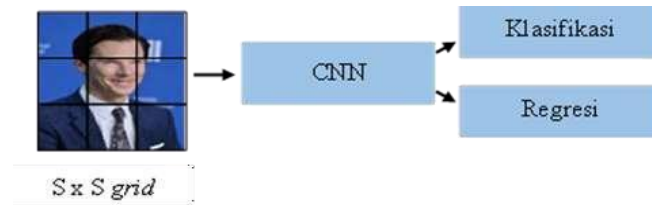


Gambar 2.1 Arsitektur Yolo

Sumber: (Azhar., 2021)

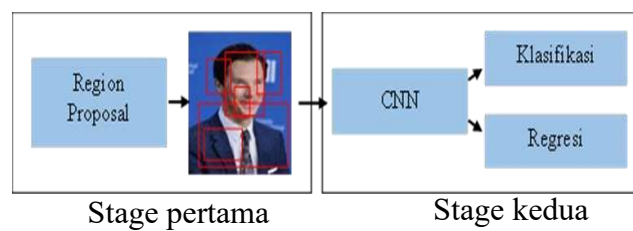
Terdapat dua pendekatan dalam melakukan deteksi wajah yaitu multi stage detector dan *one stage detector*. *Multi stage detector* terdiri dari dua tahapan yaitu proses region proposal dan proses klasifikasi. Sedangkan, *One stage detector* menggabungkan proses region proposal dan proses klasifikasi dalam satu network. Salah satu metode yang

menerapkan konsep *one stage detector* adalah *You Only Look Once* (YOLO).



Gambar 2.2. Multi Stage Detector

sumber: (Yusqi Alfian Thoriq., 2023).

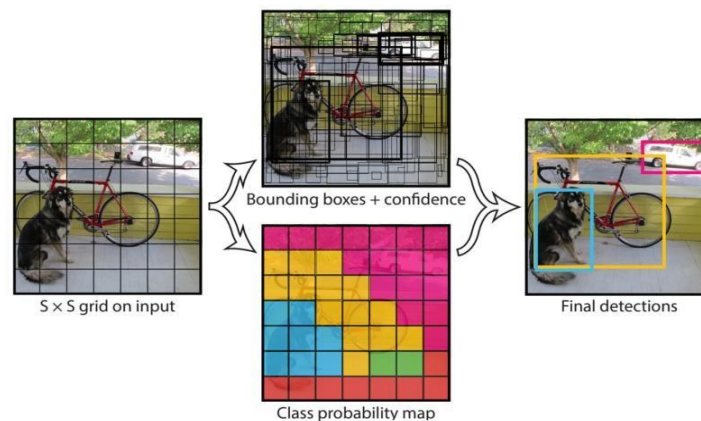


Gambar 2.3 One Stage Detector

Sumber: (Yusqi Alfian Thoriq., 2023)

Pada *Yolo* akan dibentuk grid dengan ukuran $S \times S$, dimana setiap grid pada citra akan memprediksi bounding box B dan Nilai probabilitas kelas C . Terdapat lima prediksi dalam bounding box B yaitu confidence score (p), x , y , w , dan h . Nilai confidence score menyatakan ada atau tidaknya objek pada grid tertentu. Nilai x dan y menyatakan titik pusat dari suatu objek, w dan h merupakan lebar dan tinggi bounding box dari objek [6].

saat dilakukan dengan pengujian data uji. *Yolo* mengaplikasikan arsitektur yang hampir sama dengan *Convolutional neural networks*. [7].



Gambar 2.4. Proses Deteksi Yolo

Sumber: (Salamah., 2022)

2.5 Computer Vision

Computer vision adalah bidang ilmu komputer yang bertujuan untuk memberikan kemampuan pada komputer atau mesin untuk memahami, menganalisis, dan menafsirkan informasi visual dari gambar dan video. Tujuan utama dari computer vision adalah untuk mengembangkan sistem yang dapat melihat dan memahami dunia sekitarnya dengan cara yang mirip dengan cara manusia melakukannya rekonstruksi 3D, analisis citra medis dan pengenalan tulisan tangan [7]. Teknologi ini mengintegrasikan metode pemrosesan citra digital, pembelajaran mesin, dan algoritma deep learning untuk mengidentifikasi pola, objek, atau informasi spesifik dalam data visual. computer vision memungkinkan

komputer untuk melakukan tugas-tugas seperti deteksi objek, segmentasi gambar, klasifikasi, dan pelacakan pergerakan objek.



Sumber:[Aprianti, Winda,Maliha, Umi,2016]

2.6 Open CV

OpenCV merupakan salah satu library yang memiliki fungsi-fungsi pemrograman pada computer vision secara real-time. OpenCV bersifat open-source yang dapat digunakan untuk membantu hal-hal yang bersifat akademis dan komersil. Pada OpenCV terdapat antarmuka untuk C, C++, Python, dan Java yang dapat dijalankan pada Windows, Mac, Linux, dan Android OpenCV memiliki lebih dari 2500 algoritma yang telah dioptimalkan (Budiarjo, 2020). OpenCV memiliki banyak sekali fitur yang dapat dimanfaatkan, beberapa diantaranya yaitu dapat membaca data gambar atau video dari file serta sebaliknya, menjalankan berbagai

standar algoritma computer vision seperti line detection, edges detection, dan sebagainya [9].

2.6 Roboflow

Roboflow adalah framework pengembang computer vision untuk pengumpulan data yang lebih baik ke prapemrosesan, dan teknik pelatihan model. Roboflow memiliki kumpulan data publik yang tersedia bagi pengguna dan juga memiliki akses bagi pengguna untuk mengunggah data khusus mereka sendiri. Cara kerja roboflow adalah mengkonversi dataset yang sudah di beri label ke dalam *YOLOv8* [10]. Roboflow menawarkan solusi menyeluruh mulai dari pengelolaan dataset hingga deployment model secara real-time di berbagai perangkat. Platform ini sering digunakan dalam penelitian yang melibatkan algoritma deep learning seperti YOLO (You Only Look Once) karena menyediakan berbagai fitur penting yang memudahkan proses pengolahan data dan pelatihan model. Roboflow memungkinkan pengguna untuk mengimpor, mengatur, dan mengelola dataset dalam berbagai format. Dataset dapat dikonversi otomatis agar kompatibel dengan framework deep learning seperti TensorFlow, PyTorch, dan YOLO. Roboflow sangat bermanfaat bagi peneliti maupun praktisi di bidang computer vision, terutama yang bekerja dengan proyek berbasis deep learning. Dengan fitur-fitur yang lengkap, Roboflow membantu mempercepat proses pengolahan data serta memastikan kualitas dataset yang lebih baik, yang pada akhirnya meningkatkan akurasi model yang dihasilkan.

2.7 Confusion Matrix

Confusion matrix adalah sebuah tabel yang menunjukkan kinerja dari sebuah model klasifikasi yang memiliki data jawaban benar (supervise). Dari tabel yang didapatkan, untuk model klasifikasi yang dimiliki dapat dihitung akurasi, presisi, F-Score dan masih banyak lagi variabel yang dapat dihitung berdasarkan kondisi data yang diprediksi atau diklasifikasikan [11]. Confusion matrix dapat dilihat dengan beberapa ketentuan sebagai berikut:

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

Berdasarkan dari data Akurasi, presisi, recall dan f-score adalah Ketentuan yang digunakan dalam Confusion matrix. Ketentuan adalah kinerja model dalam seberapa tepat model dalam mengklasifikasi data secara keseluruhan.

Akurasi adalah seberapa tepat model dalam memprediksi data Positive dan negative dengan benar dengan keseluruhan jumlah data.

$$Accuracy = \frac{TP+TN}{TP+FP+TN+FN} \dots \dots \dots (1)$$

Presisi adalah seberapa tepat model dalam membandingkan data secara keseluruhan dengan jumlah Positive yang di klasifikasi secara benar.

$$Precision = \frac{TP}{TP+FP} \dots\dots\dots(2)$$

Recall adalah sistem melakukan perbandingan jumlah data benar Positive dengan jumlah data yang sebenarnya Positive secara keseluruhan.

$$Recall = \frac{TP}{TP+FN} \dots\dots\dots(3)$$

F-Score didapatkan dari penggabungan hasil presisi dan recall dengan menggunakan rata-rata harmonic presisi dan recall.

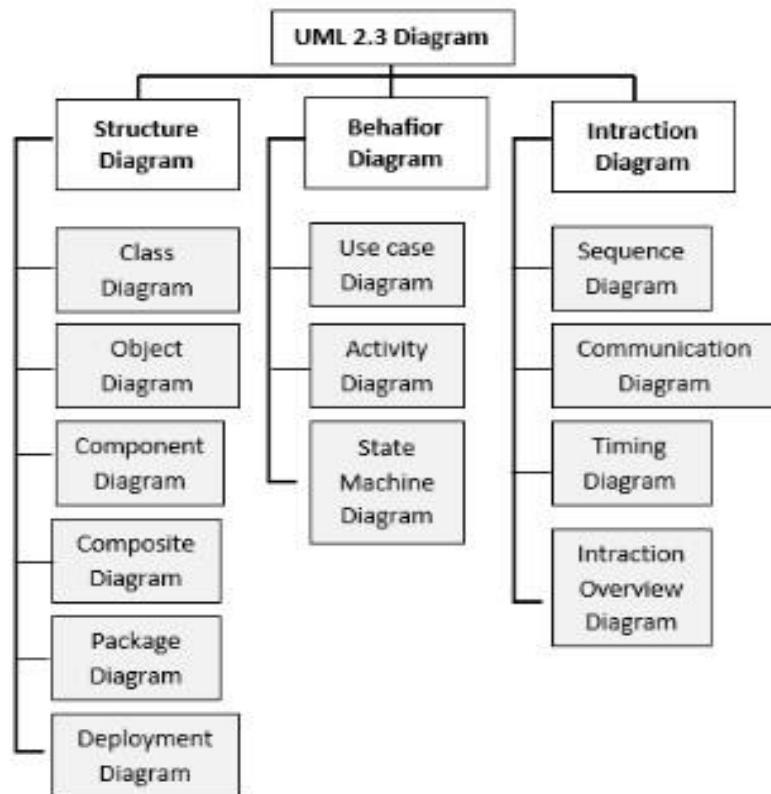
$$F - Score = \frac{precision \cdot recall}{precision + recall} \dots\dots\dots(4)$$

2.8 UML (*Unified Modeling Language*)

UML (Unified Modeling Language) adalah sebuah bahasa yang berdasarkan grafik/gambar untuk memvisualisasi, menspesifikasikan, membangun, dan pendokumentasian dari sebuah sistem pengembangan software berbasis OO (ObjectOriented)[12]. UML sendiri juga memberikan standar penulisan sebuah sistem blue print, yang meliputi konsep bisnis proses, penulisan kelas-kelas dalam. UML berfungsi sebagai alat komunikasi yang efektif antara tim pengembang dan pemangku kepentingan non-teknis. Diagram UML dapat

digunakan untuk menjelaskan konsep teknis dengan cara yang mudah dipahami. Dengan menggunakan UML, pengembang dapat merancang sistem dengan lebih terstruktur, yang pada akhirnya meningkatkan kualitas perangkat lunak dan memudahkan pemeliharaan di masa depan .

bahasa program yang spesifik, skema database, dan komponen-komponen yang diperlukan dalam sistem software.



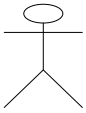
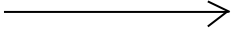
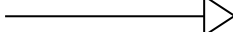
Gambar 2.6 Diagram UML

Sumber:(Voutama, A., & Novalia, E. (2022)k

2.8.1 Use Case Diagram

Use Case mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat (Rosa dan Shalahuddin, 2013).

Tabel 2.1 Use Case Diagram

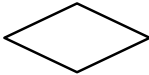
Simbol	Deskripsi
<i>Use Case</i> 	Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit dan aktor.
Aktor/Actor  nama aktor	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi.
Asosiasi/association 	Komunikasi antar aktor dan <i>Use Case</i> yang berpartisipasi.
Ekstensi/extend <<extend>> 	Relasi <i>Use Case</i> tambahan ke sebuah <i>Use Case</i> dimana <i>Use Case</i> yang ditambah dapat berdiri sendiri walau tanpa <i>Use Case</i> tambahan.
Generalisasi/generalization 	Hubungan generalisasi dan spesialisasi antara dua buah <i>Use Case</i> yang mana fungsi yang satu lebih umum dari yang lainnya.
Menggunakan include/Use Case <<include>>  uses 	Relasi <i>Use Case</i> tambahan ke sebuah <i>Use Case</i> dimana <i>Use Case</i> yang ditambahkan memerlukan <i>Use Case</i> ini untuk menjalankan fungsinya.

Sumber: (Aprianti , 2016)

2.8.2 Activity Diagram

Activity Diagram menggambarkan *workflow* (aliran kerja) dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Simbolsimbol *Activity Diagram* dapat dilihat pada Tabel 2.2.

Tabel 2.2 Activity Diagram

Simbol	Deskripsi
Status awal 	Status awal aktivitas pada sebuah diagram aktivitas memiliki sebuah status awal.
Aktivitas 	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
Percabangan/ <i>join</i> 	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu.
Penggabungan/ <i>join</i> 	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu.
Status akhir 	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir.

Sumber: (Aprianti , 2016)

2.8.3 *Sequence Diagram*

Sequence Diagram adalah salah satu jenis diagram dalam Unified Modeling Language (UML) yang menggambarkan bagaimana objek atau komponen dalam sistem saling berinteraksi satu sama lain melalui urutan pesan yang dikirimkan.

Sequence diagram menggambarkan interaksi antar objek di dalam dan di sekitar sistem (termasuk pengguna, *display*, dan sebagainya) berupa *message* yang digambarkan terhadap waktu. Dapat dilihat pada Tabel 2.3.

Tabel 2.3 Squence Diagram

NO	GAMBAR	NAMA	KETERANGAN
1		<i>LifeLine</i>	Objek <i>entity</i> , antarmuka yang saling berinteraksi.
2		<i>Message</i>	Spesifikasi dari komunikasi antar objek yang memuat informasiinformasi tentang aktifitas yang terjadi
3		<i>Message</i>	Spesifikasi dari komunikasi antar objek yang memuat informasiinformasi tentang aktifitas yang terjadi

Sumber : (Aprianti , 2016)

2.8.4 Class Diagram

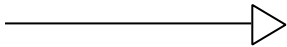
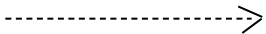
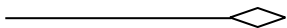
Class Diagram adalah diagram dalam UML (Unified Modeling Language) yang digunakan untuk memodelkan struktur sistem dalam bentuk kelas, atribut, metode, dan relasi antar kelas.

Class diagram menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan di buat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan metode atau operasi.

Simbol-simbol yang ada pada *Class diagram* ditunjukkan oleh Tabel 2.4

Tabel 2.4 Class Diagram

Simbol	Deskripsi			
Kelas <table><tr><td>Nama_kelas</td></tr><tr><td>+atribut</td></tr><tr><td>+operasi()</td></tr></table>	Nama_kelas	+atribut	+operasi()	Kelas pada struktur system
Nama_kelas				
+atribut				
+operasi()				
Antarmuka/interface ○ Nama_interface	Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek.			
Asosiasi/association _____	Relasi antarkelas dengan makna umum, asosiasi biasanya disertai dengan <i>multiplicity</i> .			
Asosiasi berarah/directed association —————>	Relasi antarkelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .			

Generalisasi 	Relasi antarkelas dengan makna generalisasi/spesialisasi (umum khusus).
Kebergantungan/<i>dependency</i> 	Relasi antarkelas dengan makna kebergantungan antarkelas.
Agregasi/<i>aggregation</i> 	Relasi antarkelas dengan makna semua-bagian.

Sumber: (Aprianti , 2016)

2.8.5 Objek Diagram

Rosa dan Shalahuddin (2013) mendefinisikan bahwa *object diagram* menggambarkan struktur sistem dari segi penamaan objek dan jalannya objek dalam sistem. *Object diagram* memastikan bahwa semua kelas yang sudah didefinisikan pada *Class diagram* harus dipakai objeknya, karena jika tidak, pendefinisian kelas itu tidak dapat dipertanggung jawabkan. Simbol-simbol *Object diagram* ditunjukkan oleh Tabel 2.5.

Tabel 2.5 Objek Diagram

Simbol	Deskripsi
Objek	Objek dari kelas yang berjalan saat sistem dijalankan.
Nama_objek: nam_kelas	
Atribut = nilai	

Link	Relasi antar objek.
-------------------	----------------------------

Sumber: (Aprianti . 2016)

2.9 *Python*

Python adalah bahasa pemrograman yang bersifat freeware atau bebas, yang berarti tidak ada batasan dalam penyalinan atau distribusinya. Python dilengkapi dengan source code, debugger, profiler, antarmuka, fungsi sistem, GUI (Graphics User Interface), dan baris datanya. Python menjadi bahasa resmi yang terintegrasi dalam Raspberry Pi. Nama Raspberry Pi sendiri terinspirasi dari kata "Python", dan dikatakan bahwa Python adalah bahasa yang secara alami terkait dengan Raspberry Pi. Python mendukung berbagai paradigma pemrograman, terutama namun tidak terbatas pada pemrograman berorientasi objek, pemrograman imperatif, dan pemrograman fungsional. Salah satu fitur yang dimiliki oleh Python adalah sebagai bahasa pemrograman dinamis dengan manajemen memori otomatis. Seperti halnya bahasa pemrograman dinamis lainnya, Python sering digunakan sebagai bahasa skrip, meskipun pada kenyataannya, penggunaannya lebih luas dan mencakup berbagai konteks pemanfaatan yang tidak terbatas pada bahasa skrip. Python dapat digunakan untuk bahasa pemrograman serbaguna yang mudah digunakan, kuat, dan memiliki banyak aplikasi di berbagai bidang. Dengan sintaks yang sederhana dan pustaka yang luas, Python menjadi pilihan utama dalam pengembangan software modern

berbagai tujuan dalam pengembangan perangkat lunak dan dapat dijalankan di berbagai platform sistem operasi.



Gambar 2.7 Logo Python

Sumber: <https://www.python.org/>

2.10 *Flask*

Flask merupakan sebuah kerangka kerja web yang dikodekan dengan menggunakan bahasa pemrograman Python dan termasuk dalam kategori microframework. *Flask* berperan sebagai kerangka kerja untuk pengembangan aplikasi web dan mengatur tampilan dari suatu situs web. Dengan memanfaatkan *Flask* dan bahasa pemrograman Python, pengembang dapat membuat situs web yang terstruktur dan mengelola perilaku suatu situs web dengan lebih mudah.

Flask, sebagai web framework, ditulis dalam bahasa Python. Oleh karena itu, sebelum menggunakan Flask, seorang pengembang web perlu instal

Python di perangkatnya. Oleh karena itu, seorang pengembang web yang ingin menggunakan Flask untuk pengembangan web perlu memiliki pengetahuan dasar tentang bahasa pemrograman Python.

Meskipun disebut sebagai microframework, hal ini tidak berarti Flask memiliki keterbatasan dalam hal fungsionalitas. Istilah "microframework" pada Flask merujuk pada tujuan untuk membuat inti aplikasi sebersih mungkin, tetapi

Flask termasuk dalam kategori microframework karena tidak memerlukan alat atau pustaka khusus dalam penggunaannya. Sebagian besar fungsi dan komponen umum, seperti validasi formulir, database, dan lainnya, tidak terpasang secara default di *Flask*. komponen-komponen tersebut seolah-olah diimplementasikan oleh *Flask* itu sendiri. Logo *flask* dapat di lihat pada gambar 2.8.

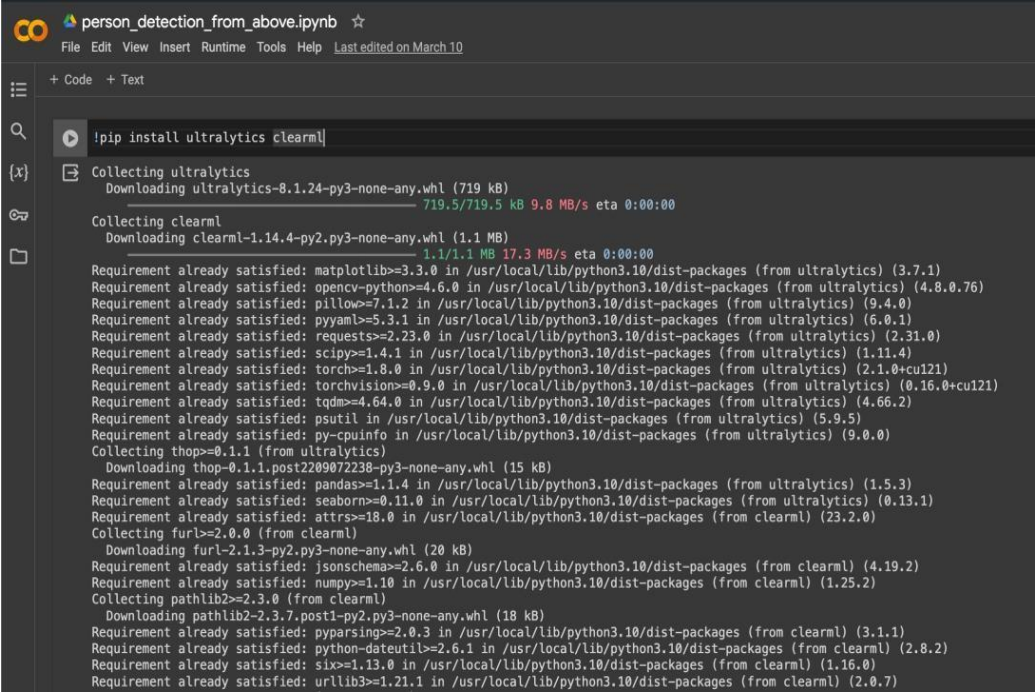


Gambar 2.8 Logo Flask

Sumber: *Fahri Muhamad Zulkarnaen(2020)*

2.11 Google Colaboratory

Google Colaboratory lebih umum disebut sebagai "*Google Colab*" atau hanya "*Colab*" adalah proyek penelitian untuk membuat prototipe model pembelajaran mesin pada opsi perangkat keras yang kuat seperti GPU dan TPU. Ini menyediakan lingkungan notebook Jupyter tanpa server untuk pengembangan interaktif. *Google Colab* gratis digunakan seperti produk G Suite lainnya (Bisong, 2019). *Google Colaboratory* berperan seperti komputer, sehingga bisa dilakukan penginstalan terhadap aplikasi-aplikasi yang dibutuhkan dalam proyek penelitian. merupakan contoh tampilan *google colab* yang sedang menginstall beberapa *library python*.



```
person_detection_from_above.ipynb ☆
File Edit View Insert Runtime Tools Help Last edited on March 10

+ Code + Text

!pip install ultralytics clearml

Collecting ultralytics
  Downloading ultralytics-8.1.24-py3-none-any.whl (719 kB)
    719.5/719.5 kB 9.8 MB/s eta 0:00:00
Collecting clearml
  Downloading clearml-1.14.4-py2.py3-none-any.whl (1.1 MB)
    1.1/1.1 MB 17.3 MB/s eta 0:00:00
Requirement already satisfied: matplotlib>=3.3.0 in /usr/local/lib/python3.10/dist-packages (from ultralytics) (3.7.1)
Requirement already satisfied: opencv-python>=4.6.0 in /usr/local/lib/python3.10/dist-packages (from ultralytics) (4.8.0.76)
Requirement already satisfied: pillow>=7.1.2 in /usr/local/lib/python3.10/dist-packages (from ultralytics) (9.4.0)
Requirement already satisfied: pyyaml>=5.3.1 in /usr/local/lib/python3.10/dist-packages (from ultralytics) (6.0.1)
Requirement already satisfied: requests>=2.23.0 in /usr/local/lib/python3.10/dist-packages (from ultralytics) (2.31.0)
Requirement already satisfied: scipy>=1.4.1 in /usr/local/lib/python3.10/dist-packages (from ultralytics) (1.11.4)
Requirement already satisfied: torch>=1.8.0 in /usr/local/lib/python3.10/dist-packages (from ultralytics) (2.1.0+cu121)
Requirement already satisfied: torchvision>=0.9.0 in /usr/local/lib/python3.10/dist-packages (from ultralytics) (0.16.0+cu121)
Requirement already satisfied: tqdm>=4.64.0 in /usr/local/lib/python3.10/dist-packages (from ultralytics) (4.66.2)
Requirement already satisfied: psutil in /usr/local/lib/python3.10/dist-packages (from ultralytics) (5.9.5)
Requirement already satisfied: py-cpuinfo in /usr/local/lib/python3.10/dist-packages (from ultralytics) (9.0.0)
Collecting thop>=0.1.1 (from ultralytics)
  Downloading thop-0.1.1.post2209072238-py3-none-any.whl (15 kB)
Requirement already satisfied: pandas>=1.1.4 in /usr/local/lib/python3.10/dist-packages (from ultralytics) (1.5.3)
Requirement already satisfied: seaborn>=0.11.0 in /usr/local/lib/python3.10/dist-packages (from ultralytics) (0.13.1)
Requirement already satisfied: attrs>=18.0 in /usr/local/lib/python3.10/dist-packages (from clearml) (23.2.0)
Collecting furl>=2.0.0 (from clearml)
  Downloading furl-2.1.3-py2.py3-none-any.whl (20 kB)
Requirement already satisfied: jsonschema>=2.6.0 in /usr/local/lib/python3.10/dist-packages (from clearml) (4.19.2)
Requirement already satisfied: numpy>=1.10 in /usr/local/lib/python3.10/dist-packages (from clearml) (1.25.2)
Collecting pathlib2>=2.3.0 (from clearml)
  Downloading pathlib2-2.3.7.post1-py2.py3-none-any.whl (18 kB)
Requirement already satisfied: pyparsing>=2.0.3 in /usr/local/lib/python3.10/dist-packages (from clearml) (3.1.1)
Requirement already satisfied: python-dateutil>=2.6.1 in /usr/local/lib/python3.10/dist-packages (from clearml) (2.8.2)
Requirement already satisfied: six>=1.13.0 in /usr/local/lib/python3.10/dist-packages (from clearml) (1.16.0)
Requirement already satisfied: urllib3>=1.21.1 in /usr/local/lib/python3.10/dist-packages (from clearml) (2.0.7)
```

Gambar 2.9 Google Colab

Sumber: *Google Colab*

2.10 Metode Penelitian

Metode penelitian yang digunakan dalam penelitian ini meliputi :

2.10.1 Metode Pengumpulan Data

Data memiliki peran yang sangat penting dalam proses penelitian karena menjadi dasar utama untuk mencapai tujuan penelitian. Data dapat berupa angka, teks, gambar, atau bentuk lain yang sesuai dengan kebutuhan studi yang dilakukan. Kualitas data sangat dipengaruhi oleh metode pengumpulan yang digunakan. Pengumpulan data yang efektif dapat meningkatkan validitas hasil penelitian dan mendukung pengambilan keputusan yang tepat. Dalam penelitian, data umumnya dibagi menjadi dua jenis, yaitu data primer dan data sekunder. Data primer adalah data yang dikumpulkan langsung dari sumber pertama melalui interaksi langsung, seperti wawancara, survei, atau observasi. Sementara itu, data sekunder adalah informasi yang berasal dari sumber yang telah ada sebelumnya, seperti dokumen resmi, laporan penelitian terdahulu, atau publikasi lainnya. Kedua jenis data ini.

2.10.2 Studi Pustaka

Studi pustaka adalah metode pengumpulan data atau informasi yang dilakukan melalui penelaahan berbagai literatur atau sumber tertulis yang relevan dengan topik penelitian. Studi pustaka bertujuan untuk mendapatkan landasan teori, konsep, atau kerangka berpikir yang mendukung penelitian. Sumber yang digunakan dalam studi pustaka dapat berupa buku, jurnal ilmiah, artikel, laporan

penelitian, dokumen resmi, atau publikasi lainnya yang dapat dipercaya dan sesuai dengan kebutuhan penelitian.

2.10.3 Observasi

Observasi adalah metode pengumpulan data dalam penelitian yang dilakukan dengan cara mengamati secara langsung objek, fenomena, atau aktivitas tertentu sesuai dengan tujuan penelitian. Metode ini sering digunakan dalam penelitian kualitatif maupun kuantitatif dan sangat berguna untuk memahami perilaku, interaksi sosial, atau situasi tertentu dengan lebih mendalam.

2.11 Kerangka Kerja Penelitian

Kerangka kerja penelitian yang akan dilakukan disajikan pada tabel dibawah ini yaitu sebagai berikut :

Tabel 2.6 Kerangka Kerja Penelitian

No	Kegiatan	Nov	Des	Jan	Feb	Mar
1	Penentuan Topik Penelitian	✓	✓			
2	Pengumpulan Data			✓		
3	Preprocessing dan Clening Data				✓	
4	Pelatihan Model dengan <i>Yolov8</i>				✓	✓
5	Pengujian Model					✓
6	Seminar Proposal					✓
7	Sidang Akhir					✓

Catatan :

- Durasi setiap kegiatan dapat disesuaikan dengan kebutuhan spesifikasi penelitian.
- Waktu pelaksanaan bergantung pada kalender akademik dan kesiapan

2.12 Penelitian Terdahulu

Tabel 2.7 Penelitian Terdahulu

No	Nama Peneliti	Judul Penelitian	Metode	Hasil Utama	Perbedaan penelitian
1	Genoveva & Syah (2024)	Model Machine Learning untuk Deteksi Tingkat Kematangan Tandan Buah Segar Kelapa Sawit Menggunakan Metode YOLOv8	Menggunakan dataset 6.592 citra TBS kelapa sawit dari platform Roboflow; model dilatih dengan metodologi CRISP-DM.	Precision 0,945; recall 0,947; mAP 0,98. Model mampu mendeteksi tingkat kematangan dengan baik.	Fokus pada deteksi kematangan kelapa sawit menggunakan YOLOv8 dan dataset yang besar dari Roboflow.
2	Aras (2024)	Deteksi Tingkat Kematangan Buah Tomat Menggunakan YOLOv5	Dataset gambar tomat pada berbagai tahap kematangan; model YOLOv5 dilatih untuk	Akurasi deteksi maksimum 73% dalam mengidentifikasi tingkat kematangan tomat.	Menggunakan YOLOv5 untuk mendeteksi kematangan buah tomat, berbeda dari fokus pada kelapa sawit.

			deteksi objek dan klasifikasi.		
3	Rifqi & Suharjito (2021)	Deteksi Kematangan Tandan Buah Segar (TBS) Kelapa Sawit Berdasarkan Komposisi Warna Menggunakan Deep Learning	Menggunakan model deep learning untuk menganalisis komposisi warna pada citra TBS kelapa sawit guna menentukan tingkat kematangan.	Model berhasil mengklasifikasi kematangan dengan akurasi yang memadai.	Fokus pada analisis komposisi warna, berbeda dari metode berbasis YOLO yang lebih umum digunakan belakangan ini.
4	Sari (20220	Deteksi Tingkat Kematangan Tandan Buah Segar Kelapa Sawit dengan Algoritme K-Means	Menggunakan algoritme K-Means untuk segmentasi warna pada citra TBS kelapa sawit guna menentukan tingkat kematangan.	Akurasi deteksi mencapai 85% dalam menentukan kematangan TBS.	Berbeda dari metode deep learning, menggunakan segmentasi berbasis K-Means untuk menentukan tingkat kematangan.
5	Muhammad Nur, Yolanda Eka Putri Dasneri, Ahmad Mas'ari (2019)	Pengendalian Kualitas Crude Palm Oil (CPO) di PT. Sebang Multi Sawit	Statistical Quality Control (SQC) dengan alat: histogram, peta kendali, dan diagram sebab akibat.	Ditemukan 14 data kadar FFA dan 8 data kadar air di luar batas kontrol. - Penyebab utama: kurangnya keterampilan pekerja,	Penelitian ini berfokus pada pengendalian kualitas minyak sawit, bukan deteksi kematangan buah.

				metode kerja tidak optimal,	
--	--	--	--	-----------------------------------	--