

**WEB PENGHITUNG CACAH TBS
MENGUNAKAN ALGORITMA YOLOv8**

SKRIPSI

Diajukan sebagai salah satu syarat untuk kelulusan Sarjana Program Studi

Teknik Pertanian pada Minat Teknik Informatika pertanian di Instiper Yogyakarta



Disusun Oleh :

FANZORIYANUS TELAUMBANUA
20/22075/TP/STIP

Dosen Pembimbing :

1. Dr. Ir. Hermantoro, MS
2. Teddy Suparyanto, S.Pd, M.TI

**JURUSAN TEKNIK PERTANIAN
FAKULTAS TEKNOLOGI PERTANIAN
INSTITUT PERTANIAN STIPER
YOGYAKARTA**

2024

**WEB PENGHITUNG CACAH TBS MENGGUNAKAN
ALGORITMA YOLOv8**

Disusun Oleh :

FANZORIYANUS TELAUMBANUA

20/22075/TP/STIP

Telah dipertahankan di depan Dewan Penguji

Pada tanggal 10 September 2024

Skripsi ini telah diterima sebagai salah satu persyaratan yang diperlukan guna memperoleh gelar Sarjana Teknologi Pertanian

Fakultas Teknologi Pertanian Institut Pertanian STIPER Yogyakarta

Yogyakarta, 10 September 2024

Menyetujui,

Dosen Pembimbing I



(Dr. Ir. Hermantoro, MS)

Dosen pembimbing II



(Teddy Suparyanto, S.Pd, M.TI)

Mengetahui

Dekan Fakultas Teknologi Pertanian



(Dekan, S.P., M.P)

WEB PENGHITUNG CACAH TBS

MENGGUNAKAN ALGORITMA YOLOv8

Fanzoriyanus Telaumbanua¹, Hermantoro², Teddy Suparyanto³

Jurusan Teknik Pertanian, Fakultas Teknologi Pertanian, Institut Pertanian

STIPER Yogyakarta

Jl. Nangka II Maguwoharjo, Depok, Sleman, Daerah Istimewa Yogyakarta 55282

Email: fanzotel@gmail.com

ABSTRAK

Kelapa sawit (*Elaeis guineensis*) telah menjadi salah satu tanaman dan komoditas yang sangat penting di dunia. Sebagai sumber utama nabati, Tandan Buah Segar (TBS) adalah buah kelapa sawit yang umumnya dipanen ketika buah mulai membrondol dari tandan. *You Only Look Once* (YOLO) adalah algoritma yang berbasis CNN dan menggunakan pendekatan jaringan saraf tunggal (*Single Neural Network*) untuk mendeteksi objek dalam gambar. Penelitian ini bertujuan menghitung buah sawit di TPH dengan melalui foto atau video menggunakan model pelatihan YOLOv8 berbasis web dengan pengujian SDLC atau System development Life Cycle. Pengujian tingkat kebenaran menggunakan confusion matrix yang menunjukkan seberapa yakin sistem bahwa objek yang dideteksi adalah benar-benar sebuah sawit atau brondolan sistem YOLOv8 berbasis web ini dikembangkan dengan tujuan mempermudah penghitungan buah sawit (TBS) di TPH dimana sebelumnya sistem yang digunakan masih manual. Hasil penelitian menunjukkan bahwa sistem telah berhasil mengidentifikasi brondolan dan buah sawit dengan tingkat kebenaran yang cukup

Kata kunci : Kelapa sawit, TBS, SDLC (*system development life cycle*) sistem YOLOv8, *Confusion matrix*.

KATA PENGANTAR

Pujian dan syukur disampaikan kepada Tuhan Yang Maha Esa, karena dengan berkat dan rahmat-Nya, penulis berhasil menyelesaikan skripsi yang berjudul “**Web Penghitung Cacah Tbs Menggunakan Algoritma Yolov8**”

Penulisan skripsi ini merupakan upaya terbaik dari penulis untuk memenuhi persyaratan dalam meraih gelar Sarjana Teknologi Pertanian di Jurusan Teknik Pertanian, Fakultas Teknologi Pertanian, Institut Pertanian STIPER Yogyakarta. Dengan rendah hati dan ketulusan, penulis ingin menyampaikan rasa terima kasih kepada semua pihak yang telah memberikan bantuan dalam menyelesaikan skripsi ini, termasuk:

1. Kepada Tuhan , Yang Maha Pengasih dan Maha Penyayang, kami mengucapkan rasa syukur yang tak terhingga atas nikmat kesehatan dan kesempatan yang telah diberikan-Nya kepada Peneliti, sehingga dapat menyelesaikan skripsi ini dengan baik. Semoga segala usaha dan hasil yang telah diperoleh dapat menjadi bentuk pengabdian yang bermanfaat bagi diri sendiri, masyarakat, dan bangsa.
2. Kepada Kedua orang tua, Bapak Aluinasokhi Telaumbanua dan Ibu Yutisa Ndruru. Peneliti menyadari bahwa tiada kata yang mampu sepenuhnya menggambarkan rasa syukur ini. Namun, dengan penuh rasa terimakasih yang sebesar – besarnya kepada kalian. Terimakasih atas segalanya, terimakasih atas doa, dukungan dan cinta yang tiada hentinya yang diberikan kepada peneliti. Terimakasih telah

menjadi orang tua yang supportif. Terimakasih telah berjuang mengorbankan banyak waktu, tenaga dan upaya untuk mendukung peneliti meraih impian. Peneliti berharap dengan terselesaikannya skripsi ini, dapat menjadi bentuk penghormatan dan apresiasi atas segala perjuangan dan kasih sayang yang kalian berikan. Untuk Bapak dan Ibu semoga diberikan Kesehatan, kebahagiaan, keberkahan dan umur panjang. Semoga segala doa yang telah kalian panjatkan untuk peneliti menjadi jembatan menuju kesuksesan di dunia.

3. Kepada Saudara penulis, Yusni Telaumbanua, Karimawati Telaumbanua, Iman Wati Telaumbanua, Diriyanus Telaumbanua, Feliyanus Telaumbanua, Yuniar Telaumbanua yang selalu memberikan semangat, motivasi, dan doa.
4. Bapak Dr. Ir. Harsawardana, M.Eng, selaku Rektor Institut Pertanian STIPER Yogyakarta.
5. Ibu Dr. Ngatirah, S.P., M.P, selaku Dekan Fakultas Teknologi Pertanian STIPER Yogyakarta yang menaungi di Fakultas.
6. Bapak Arief Ika Uktoro, STP., M.Sc. selaku Ketua Jurusan Teknik Pertanian.
7. Dr. Ir. Hermantoro, MS, selaku dosen pembimbing ke satu penulis dalam penyelesaian skripsi ini.
8. Bapak Teddy Suparyanto, S.Pd., M.TI, selaku dosen pembimbing kedua penulis dalam penyelesaian skripsi ini.
9. Bapak dan Ibu Dosen Fakultas Teknologi Pertanian Institut Pertanian STIPER Yogyakarta yang memberikan ilmu, wawasan, pengetahuan, dan pengalaman kepada penulis selama menjalani kuliah.

10. Bapak dan Ibu Staff Administrasi Fakultas Teknologi Pertanian Institut Pertanian STIPER Yogyakarta yang telah memberikan dukungan administratif dan bantuan yang sangat berarti selama penulis menjalani proses akademis, terima kasih atas dedikasi dan kerja kerasnya.
11. Teman-teman seangkatan mahasiswa Jurusan Teknik Pertanian, khususnya STIP-A TA 2020, yang telah memberikan dorongan, masukan, serta semangat agar penulis akhirnya dapat menyelesaikan skripsi ini. Demikianlah skripsi ini penulis susun, dengan segala keterbatasan sebagai manusia biasa, tentu tidak lepas dari kekurangan dan kesalahan. Oleh karena itu, penulis sangat mengharapkan saran dan kritik yang konstruktif guna kesempurnaan karya ini. Akhir kata, penulis mengucapkan terima kasih atas segala dukungan dan bantuan yang telah diberikan, semoga skripsi ini dapat bermanfaat bagi mereka yang membutuhkan dan dimanfaatkan sebagaimana mestinya.

Yogyakarta

Fanzoriyanus Telaumbanua

DAFTAR ISI

	Hal
PROPOSAL PENELITIAN.....	i
HALAMAN PENGESAHAN	ii
DAFTAR ISI.....	iii
DAFTAR GAMBAR.....	iv
BAB I.....	1
A. Latar Belakang	1
B. Rumusan Masalah.....	3
C. Tujuan Penelitian	3
D. Manfaat penelitian	4
BAB II.....	6
A. Kelapa sawit (<i>Elaeis guineensis</i>)	6
B. Menghitung buah sawit.....	7
C. TPH (Tempat pengumpulan buah).....	7
D. YOLO	8
E. SDLC	8
F. Confusion Matrix	9
BAB III.....	10
A. Lokasi dan Waktu Penelitian.....	10
B. Alat dan Bahan Penelitian.....	10
C. Metode Penelitian	11
BAB VI.....	16
A. Requirement Analysis.....	16
B. Desain	20
C. Implementasi.....	25
D. Confusion Matrix.....	32

BAB V.....	43
A. Kesimpulan.....	43
B. Saran.....	43
DAFTAR PUSTAKA.....	44
LAMPIRAN	

DAFTAR GAMBAR

Gambar 3. 1 Model <i>Waterfal</i>	11
Gambar 3. 2 <i>Flowchart</i> Alur Penelitian	12
Gambar 4.1 Use Case Diagram.....	19
Gambar 4.2 Tampilan awal desain web.....	23
Gambar 4. 3 Hasil deteksi sawit menggunakan YOLOv8.....	24
Gambar 4. 4 mengaktifkan lingkungan virtual dengan perintah “Conda activate yolov8_custom”	25
Gambar 4.5 Untuk membuka aplikasi labelimg.....	25
Gambar 4.6 proses untuk memberikan lebelan	26
Gambar 4.7 Hasil dari pelebelan	27
Gambar 4.8 Persiapan dataset	28
Gambar 4.9 Konfigurasi dataset.....	28
Gambar 4.10 Halaman utama dalam web	32
Gambar 4.11 Dashboard adalah tempat file yang pernah digunakan.....	33
Gambar 4.12 Hasil deteksi menggunakan YOLOv8.....	34
Gambar 4.13 Gambar yang menunjukkan tingkat kebenaran suatu objek.....	35
Gambar 4.14 Confusion matrix normalized.....	36
Gambar 4.15 . F1-Confidence curve	37
Gambar 4.16 Recall- Confidence Curve	38
Gambar 4.17 Visualisasi data.....	39
Gambar 4.18 plot matriks.....	41

BAB I

PENDAHULUAN

A. Latar Belakang

Kelapa sawit (*Elaeis guineensis*) telah menjadi salah satu tanaman dan komoditas yang sangat penting di dunia. Sebagai sumber utama minyak nabati, kelapa sawit dikenal sebagai tanaman yang paling produktif dalam menghasilkan minyak. Pada usia produktifnya (lebih dari 6 tahun), satu tanaman kelapa sawit dapat memproduksi sekitar 200 kg tandan buah segar setiap tahun, yang setara dengan 40 kg minyak sawit kasar (CPO). Di lahan seluas 1 hektar, kelapa sawit mampu menghasilkan rata-rata 5.000 kg minyak per tahun, jauh lebih tinggi dibandingkan dengan tanaman penghasil minyak nabati lainnya seperti kedelai dan jagung, yang masing-masing hanya menghasilkan 375 kg dan 145 kg. Kelapa sawit termasuk dalam keluarga yang sama dengan kelapa dan kelompok pinang-pinangan (palem) lainnya. Seperti anggota keluarga *Arecaceae* lainnya, kelapa sawit memiliki ciri khas berupa daun majemuk berwarna hijau yang berbentuk menyirip dan menempel pada pelepah. Setiap pelepah, yang panjangnya berkisar antara 7–9 meter, membawa 250–400 helai daun. Pada tahap awal pertumbuhannya, kelapa sawit dapat menumbuhkan 20–30 pelepah setiap tahun, tetapi jumlah ini akan berkurang seiring bertambahnya usia tanaman, dengan rata-rata sekitar 1,5 pelepah per bulan. Akibatnya, kelapa sawit menghasilkan biomassa yang sangat besar, mencapai lebih dari 6 ton per hektar per tahun (Nugroho, 2019).

Artificial Intelligence (AI), atau yang dalam bahasa Indonesia disebut Kecerdasan Buatan, adalah cabang dari ilmu komputer yang bertujuan untuk menciptakan sistem dan mesin yang mampu menjalankan tugas-tugas yang biasanya memerlukan kecerdasan manusia. AI melibatkan penggunaan algoritma dan model matematika yang memungkinkan komputer dan sistem lain untuk belajar dari data, mengenali pola, serta membuat keputusan yang cerdas. Beberapa konsep penting dalam AI mencakup *machine learning* (pembelajaran mesin), *neural networks* (jaringan saraf tiruan), *natural language processing* (pemrosesan bahasa alami), dan lainnya. Pengembangan AI telah membawa dampak signifikan di berbagai bidang seperti pengenalan suara,

pengenalan wajah, kendaraan otonom, pengobatan, dan banyak lagi (Eriana & Zein, 2023). Web, atau World Wide Web, merupakan sistem global yang memungkinkan pertukaran informasi melalui internet. Web terdiri dari kumpulan halaman atau situs yang diakses menggunakan browser, dengan protokol HTTP (Hypertext Transfer Protocol). Setiap halaman web dapat menyajikan teks, gambar, video, serta tautan ke halaman lainnya. Diperkenalkan pertama kali pada tahun 1991 oleh Tim Berners-Lee, yang menciptakan HTML (Hypertext Markup Language) untuk merancang halaman web, web terus berkembang dari versi statis (Web 1.0) ke versi dinamis dan interaktif (Web 2.0), hingga era terkini yang mendukung aplikasi berbasis kecerdasan buatan dan blockchain (Web 3.0). Kini, web menjadi platform utama untuk komunikasi, bisnis, hiburan, dan pendidikan secara global.

You Only Look Once (YOLO) adalah algoritma yang berbasis CNN dan menggunakan pendekatan jaringan saraf tunggal (single neural network) untuk mendeteksi objek dalam gambar. Jaringan ini mampu memprediksi setiap bounding box dengan menggunakan fitur dari seluruh gambar. YOLO dapat secara langsung memprediksi bounding box dan probabilitas objek dalam satu evaluasi. Arsitektur YOLO sebagian besar terinspirasi dari backbone GoogleLeNet, yang terdiri dari 24 lapisan konvolusi untuk ekstraksi fitur, diikuti oleh 2 Fully Connected Layer (FCN) untuk memprediksi koordinat bounding box dan klasifikasi objek. Algoritma deteksi objek YOLO bekerja dengan memulai dari gambar masukan yang dibagi menjadi grid, di mana setiap sel bertugas mendeteksi objek. YOLO secara efisien memprediksi beberapa kotak pembatas serta tingkat kepercayaan untuk masing-masing sel, yang menunjukkan lokasi dan keberadaan objek dengan akurat. Selain itu, YOLO menghitung probabilitas setiap kelas objek dalam sel-sel tersebut. Algoritma ini menghasilkan kotak pembatas yang tepat dan tidak saling tumpang tindih di sekitar objek yang terdeteksi, seperti anjing atau sepeda. Sebagai langkah akhir, YOLO menyaring hasil dengan menggunakan ambang batas kepercayaan dan teknik non-maximal suppression untuk memastikan hasil yang lebih murni. Semua proses ini dilakukan secara real-time, menunjukkan kehebatan dan kecepatan algoritma YOLO dalam mendeteksi objek. (Muhlashin & Stefanie, 2023)

Sebuah situs web adalah kumpulan halaman web yang biasanya merupakan bagian dari suatu nama domain atau subdomain di World Wide Web (WWW) di Internet. Web page adalah dokumen yang ditulis dalam format HTML (*Hyper Text Markup Language*) dan hampir selalu diakses melalui HTTP, yaitu protokol yang mengirimkan informasi dari server website ke pengguna melalui web browser. Halaman-halaman web ini bisa bersifat statis atau dinamis, membentuk suatu struktur yang saling terkait dan terhubung melalui jaringan halaman. Website statis memiliki konten yang tetap dan jarang berubah, dengan informasi yang disampaikan secara satu arah dari pemilik website. Sebaliknya, website dinamis memiliki konten yang sering berubah dan bersifat interaktif dua arah, melibatkan pemilik dan pengguna website. Contoh website statis adalah profil perusahaan, sedangkan contoh website dinamis adalah platform seperti Friendster atau Multipliy. Dalam pengembangannya, website statis hanya bisa diperbarui oleh pemiliknya, sementara website dinamis bisa diperbarui oleh baik pengguna maupun pemilik (Noviantoro et al., 2022).

B. Rumusan Masalah

1. Bagaimana cara membuat web untuk menghitung jumlah TBS di TPH menggunakan YOLOv8?
2. Bagaimana proses implementasi YOLOv8 berbasis web untuk menghitung jumlah TBS di TPH?
3. Bagaimana cara menghitung jumlah TBS menggunakan modul pelatihan YOLOv8?

C. Tujuan Penelitian

Tujuan penelitian ini adalah sebagai berikut:

1. Menghitung buah sawit di TPH.
2. Implementasi deteksi jumlah TBS di TPH melalui foto atau video menggunakan modul pelatihan YOLOv8 berbasis web.
3. Menguji kinerja modul pelatihan YOLOv8 dalam menghitung jumlah TBS di TPH berbasis web.

D. Manfaat Penelitian

1. Meningkatkan Efisiensi Operasional: Dengan menggunakan teknologi deteksi objek YOLOv8, web ini secara otomatis menghitung jumlah tandan buah segar (TBS) kelapa sawit, mengurangi ketergantungan pada metode manual. Hal ini mempercepat proses penghitungan dan meminimalkan kesalahan manusia, sehingga meningkatkan efisiensi operasional di perkebunan kelapa sawit.
2. Mengurangi Biaya Tenaga Kerja: Sistem otomatis ini mengurangi kebutuhan akan tenaga kerja tambahan untuk menghitung jumlah TBS. Hal ini tidak hanya menghemat biaya operasional, tetapi juga memungkinkan pekerja fokus pada tugas-tugas lain yang lebih penting dalam proses produksi kelapa sawit.
3. Peningkatan Akurasi dan Transparansi: Algoritma YOLOv8 memberikan hasil yang akurat dalam mendeteksi dan menghitung TBS. Dengan adanya sistem ini, perusahaan dapat memastikan penghitungan yang tepat, sehingga meningkatkan transparansi dalam pelaporan hasil panen dan memperkuat kepercayaan antara pekerja, manajer, dan pihak-pihak terkait.
4. Kecepatan dan Efisiensi YOLOv8 dirancang untuk mendeteksi objek secara

real-time dengan kecepatan tinggi, berkat arsitektur yang efisien dan penggunaan pendekatan anchor-free. Ini memungkinkan model untuk melakukan deteksi objek dengan latensi rendah, cocok untuk aplikasi yang membutuhkan deteksi cepat seperti kendaraan otonom atau pengawasan video.

5. Akurasi Deteksi Multi-Skala YOLOv8 memiliki kemampuan untuk mendeteksi objek dengan presisi tinggi pada berbagai skala, baik objek kecil maupun besar, berkat penggunaan feature pyramid network (FPN) dan path aggregation network (PAN) pada bagian neck nya. Ini menjadikannya ideal untuk aplikasi yang memerlukan deteksi objek yang bervariasi dalam ukuran dan jarak, seperti pengenalan wajah atau pelacakan objek dalam kerumunan.

BAB II

TINJAUAN PUSTAKA

A. Kelapa sawit (*Elaeis guineensis*)

Kelapa sawit adalah tanaman dari keluarga Arecaceae yang awalnya dibudidayakan di Amerika Selatan. Tanaman ini telah menjadi salah satu komoditas utama bagi petani dalam produksi minyak kelapa sawit. Dalam industri perkebunan, kelapa sawit dianggap sebagai komoditas bernilai tinggi yang dapat menggantikan kelapa dalam pembuatan minyak. Perkebunan kelapa sawit umumnya terletak di kawasan hutan yang jauh dari keramaian, karena membutuhkan lahan yang luas. Melihat potensi keuntungan ekonomi dari produk turunan kelapa sawit, banyak petani memilih untuk menanam komoditas ini, yang sekarang tersebar di berbagai pulau di Indonesia, seperti Kalimantan, Sumatera, dan Sulawesi. Dalam penyebarannya, terdapat dua jenis kelapa sawit yang biasanya dibudidayakan oleh petani, yaitu *Elaeis guineensis* Jacq. dan *Elaeis oleifera*. Perbedaan utama antara kedua jenis ini adalah bahwa *Elaeis guineensis* memiliki produktivitas yang tinggi sehingga banyak dibudidayakan, sedangkan *Elaeis oleifera* memiliki tanaman dengan ketinggian yang lebih rendah. Di Indonesia, sumber genetika tanaman kelapa sawit awalnya ditanam di Kebun Raya Bogor. Namun, keterbatasan varietas mengakibatkan produksi dari tanaman ini kurang maksimal. Setiap varietas kelapa sawit yang dikembangkan di dunia perkebunan memiliki kelebihan dan kekurangan masing-masing (Rahmawati, 2023).

B. Menghitung buah sawit

Umumnya, alat transportasi untuk mengangkut tandan buah segar di perkebunan kelapa sawit menggunakan kendaraan roda empat atau lebih, seperti Landiri, Jhondere, dan Dump Truck. TBS di TPH akan dilakukan perhitungan secara manual baru dilakukan pengangkutan ke atas truk. Dalam perkebunan, terdapat berbagai jenis transportasi dengan sistem pengangkutan yang berbeda, yang terbagi menjadi dua metode pengantaran: langsung dan tidak langsung. Pengantaran langsung (direct) adalah proses pengangkutan TBS dari Tempat Pengumpulan Hasil (TPH) langsung ke Pabrik Kelapa Sawit (PKS). Sementara itu, pengantaran tidak langsung (indirect) melibatkan proses pengangkutan TBS dari TPH ke stampingan (loading ramp afdeling) terlebih dahulu, sebelum dilanjutkan ke PKS menggunakan unit transportasi yang berbeda milik Perusahaan (Ngapiyatun et al., 2021).

C. TPH (Tempat pengumpulan buah)

Dalam proses pemanenan buah kelapa sawit, terdapat beberapa kendala terkait pengangkutan buah dari lahan menuju tempat pengumpulan hasil. Pemilihan mesin angkut yang sesuai dipengaruhi oleh berbagai faktor, terutama jenis mesin angkut, keterbatasan waktu, dan tenaga yang selama ini menjadi tantangan yang cukup kompleks. Oleh karena itu, perlu dilakukan kajian mengenai penggunaan mesin angkut tandan buah segar kelapa sawit (*Elaeis guineensis* Jacq) secara mekanis, yang efektif untuk mengumpulkan Tandan Buah Segar (TBS) kelapa sawit. Keberhasilan panen dan produksi kelapa sawit dipengaruhi oleh berbagai faktor, seperti mutu bahan tanam, keterampilan pekerja panen, ketersediaan alat panen, kelancaran transportasi, efisiensi organisasi pemanenan, kondisi lahan, serta insentif yang diberikan. Proses panen meliputi pemotongan tandan buah yang sudah matang, pengumpulan brondolan, pemangkasan pelepah, dan pengangkutan hasil ke tempat pengumpulan (TPH) serta ke pabrik. Faktor-faktor penting yang mendukung keberhasilan panen meliputi persiapan awal, penentuan kriteria kematangan buah, frekuensi pemanenan, metode panen, ketersediaan fasilitas, pengawasan yang baik, dan pengangkutan tandan. Semua faktor ini berpengaruh besar terhadap jumlah dan kualitas minyak sawit yang dihasilkan. Oleh karena itu, diperlukan manajemen pemanenan yang terencana dan efektif untuk mencapai

hasil optimal di perkebunan kelapa sawit. (Reztano et al., 2023)

D. YOLO

YOLOv8 adalah versi dari algoritma deteksi objek terbaru dari You Only Look Once (YOLO), yang pertama kali diperkenalkan oleh Joseph Redmon dan rekan-rekannya pada tahun 2016, telah mengalami berbagai iterasi sejak saat itu. YOLO telah menjadi pilihan utama untuk deteksi objek, dengan YOLOv5 menjadi salah satu versi yang paling dikenal. Namun, Ultralytics YOLOv8 menawarkan kemampuan paling canggih dan mengungguli versi sebelumnya dalam hal akurasi dan kecepatan deteksi. YOLOv8 juga mendukung kontribusi model, tugas, dan aplikasi dari pengguna, yang semakin meningkatkan fleksibilitas platformnya. Fitur ini memungkinkan pengguna untuk membagikan model dan tugas mereka, sehingga dapat mempercepat pengembangan proyek deteksi objek. YOLO adalah sebuah pendekatan dalam computer vision dan pemrosesan gambar yang digunakan untuk deteksi objek secara real-time dalam gambar. Pengenalan wajah menggunakan metode YOLOv8. Ultralytics adalah pengembang model YOLOv8 terbaru, yang dapat diterapkan untuk tugas deteksi objek, klasifikasi gambar, dan segmentasi objek. Dibandingkan dengan YOLOv5, YOLOv8 menawarkan banyak peningkatan dan perubahan dalam arsitektur dan pengalaman pengembangan (Putra et al., 2024).

E. SDLC

SDLC, atau Sistem Development Life Cycle, adalah siklus yang digunakan dalam pembuatan atau pengembangan sistem informasi untuk menyelesaikan masalah secara efektif. Dalam pengertian lain, SDLC adalah rangkaian tahapan kerja yang bertujuan untuk menghasilkan sistem berkualitas tinggi sesuai dengan keinginan pelanggan atau tujuan pembuatan sistem tersebut. SDLC menyediakan kerangka kerja yang mencakup langkah-langkah yang diperlukan untuk mengembangkan perangkat lunak. Sistem ini mencakup rencana lengkap untuk pengembangan, pemeliharaan, dan penggantian perangkat lunak tertentu. Teknik manajemen proyek yang terstruktur, seperti SDLC, membantu manajemen mengendalikan proyek dengan memecah tugas yang kompleks menjadi bagian-bagian yang lebih mudah dikelola. Dengan membagi proyek ke

dalam fase-fase kontrol logis, manajer dapat meninjau kemajuan setiap fase untuk memastikan keberhasilan sebelum melanjutkan ke fase berikutnya dan mengalokasikan sumber daya. Jumlah fase dalam siklus hidup proyek ditentukan oleh karakteristik proyek dan metodologi manajemen proyek yang digunakan. Sebuah proses lima langkah, misalnya, mungkin terdiri dari fase yang secara umum didefinisikan sebagai persiapan, pengadaan, pengujian, implementasi, dan pemeliharaan. (Rizal et al., 2022).

F. Confusion matrix

Klasifikasi teks adalah salah satu teknik utama dalam penambangan data teks, dengan tujuan untuk secara otomatis mengklasifikasikan dokumen ke dalam kategori yang sesuai berdasarkan kontennya. Berbagai algoritma telah dikembangkan untuk menangani tantangan ini. Seiring dengan perkembangan metode algoritma yang ada, beberapa proses baru juga telah dikembangkan untuk mengotomatisasi pengelompokan dokumen. Salah satu konsep penting dalam pengelompokan dokumen adalah aturan asosiasi. Aturan asosiasi berguna untuk menemukan hubungan atau korelasi menarik di antara item-item data dalam jumlah besar. Hubungan yang ditemukan dari sejumlah besar transaksi dapat membantu dalam pengambilan keputusan. Di sisi lain, matriks kebingungan (confusion matrix) memanfaatkan estimasi posterior maksimum untuk mempelajari pengklasifikasi. Matriks ini mengasumsikan bahwa setiap kemunculan kata dalam sebuah dokumen bersifat kondisional independen terhadap kata-kata lainnya. lainnya.

BAB III

METODE PENELITIAN

A. Lokasi dan Waktu Penelitian

Penelitian ini dilakukan di PT.xxxxx di area Jambi. Penelitian ini dimulai pada bulan Mei 2024 sampai Juli 2024.

B. Alat dan Bahan Penelitian

1. Perangkat Keras :

Laptop Merek HP. Model HP Laptop 14s-dq2xxx, BIOS: F.33, Processor 11th Gen Intel (R) i3-1115g4 @ 3.00GHz (4CPUs), ~2.9GHz. memory 12288MB RAM. Page File 9642MB used, 4720MB available. DirectX Version: directX 12Perangkat

2. Software :

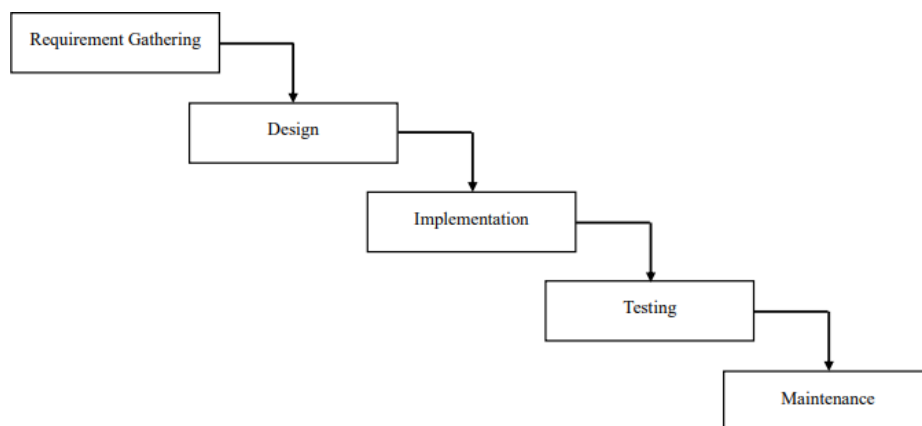
- a. YOLOv8
- b. Python 3.9.19
- c. Anaconda 24.1.2
- d. Lebelimg 1.8.6
- e. Ultralytics 8.2.77
- f. Flask Version: 3.0.3
- g. pytorch version 2.4.0+cu124
- h. fisual studi code 1.92.2

3. Data :

- a. Foto buah kelapa sawit.
- b. Data atribut kelapa sawit perhitungan jumlah pada buah kelapa sawit
- c. Data statistic, berisikan beberapa foto buah kelapa sawit

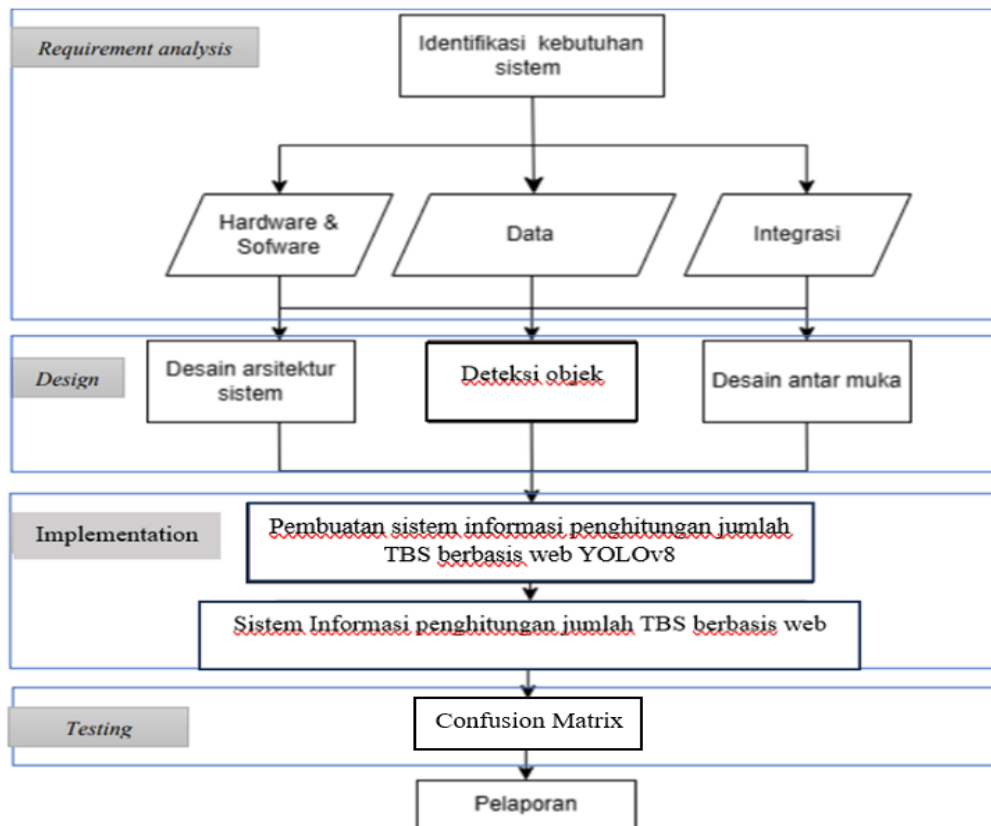
C. Metode Penelitian

Penggabungan sistem dilakukan dengan memanfaatkan komponen-komponen yang dapat digunakan kembali. Model Waterfall, yang pertama kali diperkenalkan oleh Winston Royce pada tahun 1970, adalah model klasik dengan alur sistem yang sederhana dan linear. Setiap tahap menghasilkan output yang menjadi input untuk tahap berikutnya. Model ini, yang diadaptasi dari proses rekayasa lain, menawarkan pendekatan yang lebih realistis dalam pengembangan perangkat lunak. Model Waterfall melibatkan tim SQA (*Software Quality Assurance*) dan terdiri dari 5 tahapan, di mana setiap tahap selalu diiringi dengan verifikasi atau pengujian.



Gambar 3. 1 Model *Waterfal*

Setelah tahap perencanaan, langkah berikutnya adalah menganalisis berbagai aspek, baik dari sisi fungsional maupun non-fungsional, serta mempertimbangkan kebutuhan pengguna aplikasi di masa depan.



Gambar 3. 2 Alur System Development Life Cycle (SDLC)

Gambar 3. 2 Karena keterbatasan sumber daya, penelitian ini hanya sampai pada tahap pengujian (testing) dalam siklus pengembangan perangkat lunak. Keputusan ini diambil untuk memastikan efisiensi dan efektivitas penggunaan sumber daya yang ada, serta untuk menjaga relevansi dan validitas hasil penelitian.

1. Requirement analysis

Pada Requirement analysis akan dilakukan pengambilan dokumentasi TBS (Tandan buah segar) di TPH (Tempat pengumpulan buah). Kemudian hasil foto TBS itu akan dilakukan perhitungan menggunakan algoritma YOLOv8. Pada bagian ini dilakukan identifikasi kebutuhan sistem adapun yang diamati ada 3 bagian sistem :

a. Hardware & Software

- 1) Hardware (*Client-Server Hosting*) Menggunakan model arsitektur client-server untuk hosting sistem. Hardware server, dapat menjalankan aplikasi web seperti platform e-commerce, sistem manajemen konten, serta dapat menggunakan hosting untuk menyimpan data secara aman dan dapat diakses dari mana saja.
- 2) Software adalah serangkaian instruksi atau program yang digunakan untuk mengoperasikan komputer dan perangkat lainnya. Software ini dapat berupa sistem operasi yang mengelola sumber daya perangkat keras, aplikasi yang memungkinkan pengguna melakukan tugas tertentu, atau alat-alat lain yang mendukung pengembangan dan pemeliharaan sistem komputer. Contohnya termasuk aplikasi seperti pengolah kata, program desain grafis, dan perangkat lunak pengelola basis data.

b. Data

Dara yang digunakan adalah 100 foto hasil dokumentasi TBS di TPH dengan hasil foto yang baik. Data dokumentasi Tandan Buah Segar (TBS) di Tempat Penumpukan Hasil (TPH) dalam bentuk foto merupakan cara modern untuk mengumpulkan dan memantau data panen sawit. Foto-foto ini diambil di lokasi TPH dan berfungsi sebagai sumber data visual yang mendokumentasikan jumlah dan kondisi tandan buah segar yang dipanen. Metode ini memanfaatkan teknologi pengolahan gambar dan kecerdasan buatan untuk mendeteksi, menghitung, dan mengklasifikasikan TBS secara otomatis

c. Integrasi

Data dokumentasi Tandan Buah Segar (TBS) di Tempat Penumpukan Hasil (TPH) dalam bentuk foto adalah metode yang efisien dan akurat untuk mengumpulkan data panen sawit. Foto-foto ini diambil secara langsung di lokasi TPH dan dimanfaatkan sebagai sumber data visual, yang memungkinkan deteksi, penghitungan, serta klasifikasi TBS secara otomatis menggunakan teknologi pengolahan gambar seperti YOLO. Dengan mengandalkan gambar sebagai sumber data, proses

dokumentasi menjadi lebih cepat dan minim kesalahan dibandingkan dengan penghitungan manual.

2. Design

Setelah tahap analisis kebutuhan, langkah berikutnya adalah menyusun desain sistem, seperti desain antarmuka pengguna (UI) dan desain database. Pada tahap ini, penulis mengonversi hasil identifikasi masalah ke dalam bentuk rancangan sistem, yang mencakup struktur data, arsitektur sistem, serta prosedur pengkodean yang akan diimplementasikan. Bahasa pemrograman yang digunakan yaitu Python. Desain web adalah proses merancang tampilan, tata letak, dan elemen visual dari sebuah situs web untuk menciptakan pengalaman pengguna yang baik dan menarik. Desain ini meliputi pemilihan warna, tipografi, gambar, ikon, serta penataan konten agar situs web mudah digunakan dan responsif di berbagai perangkat, seperti desktop dan ponsel. Beberapa desain dalam web

a. Desain Arsitektur Sistem

Desain Arsitektur Sistem adalah proses perancangan yang menitikberatkan pada struktur menyeluruh dari suatu perangkat lunak. Hal ini mencakup pengaturan komponen utama, modul, dan hubungan antara bagian-bagian sistem, serta bagaimana sistem tersebut berinteraksi dengan perangkat keras atau sistem lainnya. Tujuan dari arsitektur ini adalah untuk menjamin bahwa sistem dapat berjalan secara skalabel, aman, dan efisien. Misalnya, dalam sistem berbasis cloud, arsitektur perlu merancang cara layanan backend berkomunikasi dan diakses oleh frontend melalui API. Desain ini juga mempertimbangkan pemilihan pola arsitektur seperti client-server, microservices*, atau monolithic yang sesuai dengan kebutuhan sistem.

b. Desain Objek

Desain Objek Menitikberatkan pada cara objek-objek dalam perangkat lunak, yang berfungsi sebagai entitas dengan atribut dan perilaku, dirancang dan diatur. Pada tahap ini, setiap objek diciptakan untuk mewakili elemen-elemen tertentu dalam domain bisnis atau sistem yang lebih besar. Desain objek menerapkan konsep pemrograman berorientasi objek (OOP) seperti pewarisan, enkapsulasi, dan

polimorfisme, sehingga objek yang dihasilkan bersifat modular dan bisa digunakan kembali. Sementara itu,

3. Desain Antarmuka

Desain Antarmuka menitikberatkan pada interaksi pengguna dengan sistem, termasuk elemen-elemen UI (User Interface) seperti tombol, formulir, atau menu. Desain antarmuka yang baik memperhatikan aspek UX (User Experience), memastikan tampilan mudah digunakan, dapat diakses, dan menarik, sehingga pengguna dapat berinteraksi dengan sistem secara optimal dan efisien.

BAB IV

HASIL DAN PEMBAHASAN

A. Requirement Analysis

1. Requirement analysis

Pada Requirement analysis akan dilakukan pengambilan dokumentasi TBS (Tandan buah segar) di TPH (Tempat pengumpulan buah). Kemudian hasil foto TBS itu akan dilakukan perhitungan menggunakan algoritma Yolov8. Pada bagian ini dilakukan identifikasi kebutuhan sistem adapun yang diamati ada 3 bagian sistem :

a). Hardware & Software

1. Hardware (*Client-Server Hosting*) Menggunakan model arsitektur client-server untuk hosting sistem. Hardware server, dapat menjalankan aplikasi web seperti platform e-commerce, sistem manajemen konten, serta dapat menggunakan hosting untuk menyimpan data secara aman dan dapat diakses dari mana saja.
2. Software adalah serangkaian instruksi atau program yang digunakan untuk mengoperasikan komputer dan perangkat lainnya. Software ini dapat berupa sistem operasi yang mengelola sumber daya perangkat keras, aplikasi yang memungkinkan pengguna melakukan tugas tertentu, atau alat-alat lain yang mendukung pengembangan dan pemeliharaan sistem komputer. Contohnya termasuk aplikasi seperti pengolah kata, program desain grafis, dan perangkat lunak pengelola basis data.
3. Integrasi adalah proses menggabungkan kemampuan deteksi objek berbasis algoritma YOLOv8 ke dalam aplikasi atau platform berbasis web. Integrasi ini memungkinkan deteksi objek dilakukan secara real-time atau melalui pengunggahan gambar langsung di antarmuka web, yang kemudian diproses oleh model YOLOv8 untuk mendeteksi dan menghitung objek secara otomatis. Dalam implementasinya, YOLOv8, yang dilatih untuk mendeteksi objek tertentu, dihubungkan dengan backend web melalui API atau layanan cloud. Gambar atau video yang diunggah pengguna melalui antarmuka web kemudian dikirim ke

server, di mana YOLOv8 melakukan analisis dan deteksi objek. Hasilnya, seperti jumlah objek yang terdeteksi atau lokasi objek, dikirim kembali ke frontend web dan ditampilkan kepada pengguna dalam bentuk yang mudah dipahami, seperti kotak pembatas di gambar atau laporan deteksi. Proses integrasi ini memanfaatkan berbagai teknologi web, seperti Python untuk menghubungkan YOLOv8, Flask atau Django untuk backend, dan HTML, CSS, serta JavaScript untuk antarmuka web. Hasil akhirnya adalah sebuah aplikasi web interaktif yang mampu memanfaatkan kekuatan YOLOv8 untuk mendeteksi objek secara efisien.

b). Data

Dara yang digunakan adalah 100 foto hasil dokumentasi TBS di TPH dengan hasil foto yang baik. Data dokumentasi Tandan Buah Segar (TBS) di Tempat Penumpukan Hasil (TPH) dalam bentuk foto merupakan cara modern untuk mengumpulkan dan memantau data panen sawit. Foto-foto ini diambil di lokasi TPH dan berfungsi sebagai sumber data visual yang mendokumentasikan jumlah dan kondisi tandan buah segar yang dipanen. Metode ini memanfaatkan teknologi pengolahan gambar dan kecerdasan buatan untuk mendeteksi, menghitung, dan mengklasifikasikan TBS secara otomatis

Foto yang diambil di TPH memberikan gambaran visual yang akurat tentang jumlah TBS yang ada pada waktu tertentu. Setiap gambar dapat dianalisis menggunakan algoritma pengolahan citra, seperti YOLO (*You Only Look Once*) atau model deteksi objek lainnya, untuk mengidentifikasi dan menghitung TBS. Selain itu, kualitas dan kematangan buah juga bisa dianalisis dari gambar tersebut. Keuntungan menggunakan data foto adalah kecepatan dan efisiensinya, mengurangi ketergantungan pada penghitungan manual yang memerlukan waktu lebih lama dan berisiko kesalahan.

Proses pengumpulan data dimulai dengan pengambilan foto dari TBS yang sudah dipanen dan ditumpuk di TPH. Pengambilan gambar bisa dilakukan secara manual dengan kamera atau menggunakan drone untuk cakupan area yang lebih luas. Gambar-gambar ini kemudian diunggah ke sistem yang dilengkapi dengan perangkat lunak analisis

gambar. Setelah diunggah, algoritma akan mendeteksi jumlah TBS, menganalisis ukuran, dan memvalidasi hasil panen. Foto-foto ini juga berfungsi sebagai bukti visual yang bisa dilacak kembali untuk audit atau laporan hasil panen.

c). Integrasi

Data dokumentasi Tandan Buah Segar (TBS) di Tempat Penumpukan Hasil (TPH) dalam bentuk foto adalah metode yang efisien dan akurat untuk mengumpulkan data panen sawit. Foto-foto ini diambil secara langsung di lokasi TPH dan dimanfaatkan sebagai sumber data visual, yang memungkinkan deteksi, penghitungan, serta klasifikasi TBS secara otomatis menggunakan teknologi pengolahan gambar seperti YOLO. Dengan mengandalkan gambar sebagai sumber data, proses dokumentasi menjadi lebih cepat dan minim kesalahan dibandingkan dengan penghitungan manual.

Proses pengumpulan data melalui foto melibatkan pengambilan gambar TBS di TPH menggunakan kamera atau drone. Gambar tersebut kemudian diunggah ke sistem analisis yang dilengkapi dengan algoritma pendeteksi objek, yang secara otomatis menghitung jumlah TBS dan mengidentifikasi kualitas tandan. Pendekatan ini memberikan akurasi yang lebih tinggi dalam pemantauan hasil panen, serta menyediakan bukti visual yang dapat digunakan untuk audit atau verifikasi data.

Meskipun penggunaan dokumentasi foto ini sangat efektif, beberapa tantangan tetap ada, seperti perubahan kondisi pencahayaan dan sudut pengambilan gambar yang tidak ideal. Namun, dengan pengaturan yang tepat dan penggunaan perangkat keras yang sesuai, tantangan ini dapat diatasi. Penggunaan data foto dalam pengambilan keputusan juga memberikan keuntungan signifikan, termasuk kemampuan untuk mengukur hasil panen dengan lebih baik, mengoptimalkan manajemen sumber daya, dan menyimpan data sebagai arsip digital untuk keperluan pelacakan jangka panjang.

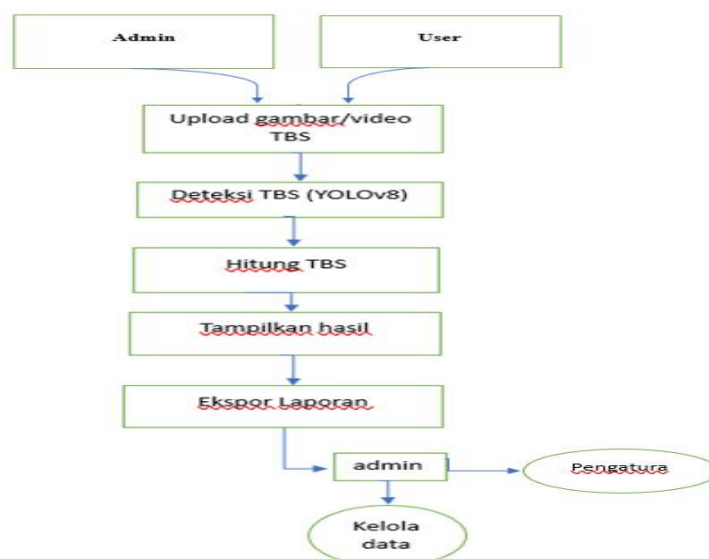
Analisis kebutuhan web ini, dimulai dengan mengambil dokumentasi TBS (Tandan buah segar) di TPH (Tempat penumpukan hasil) dalam bentuk foto untuk mengidentifikasi kebutuhan. Hasilnya menunjukkan efisiensi akses dan pengelolaan

web penghitung cacah kelapa sawit, memungkinkan pemantauan kelapa sawit yang lebih efektif. Teridentifikasi dua (2) informasi utama di TPH, yaitu buah sawit dan brondolan.

Dengan struktur ini, hubungan yang terorganisir antar tabel memungkinkan integrasi data dan analisis komprehensif terkait buah sawit sawit di TPH. pada lampiran data dokumentasi foto di TPH yang menjadi dasar dari analisis kebutuhan ini.

2. Use Case Diagram

Penelitian berjudul "Penghitung Cacah TBS dengan Menggunakan Algoritma YOLOv8" dapat dijelaskan melalui beberapa interaksi sistem dalam bentuk use case diagram. Pengguna memiliki kemampuan untuk mengunggah gambar atau video tandan buah segar (TBS). Setelah itu, sistem akan menjalankan fungsi deteksi TBS dengan menggunakan algoritma YOLOv8, yang secara otomatis mendeteksi dan menghitung jumlah TBS yang ada dalam gambar atau video tersebut. Setelah proses deteksi selesai, sistem akan menampilkan hasil perhitungan TBS kepada pengguna. Pengguna kemudian dapat mengekspor hasil deteksi tersebut dalam bentuk laporan, baik dalam format PDF atau Excel. Selain itu, admin memiliki hak istimewa untuk mengelola data dalam sistem, seperti menghapus data lama atau memperbarui informasi pengguna dan sistem. Sistem juga akan menyimpan log aktivitas untuk m



Gambar 4.1 Use Case Diagram

B. Desain

Setelah tahap analisis kebutuhan, langkah berikutnya adalah menyusun desain sistem, seperti desain antarmuka pengguna (UI) dan desain database. Pada tahap ini, penulis mengonversi hasil identifikasi masalah ke dalam bentuk rancangan sistem, yang mencakup struktur data, arsitektur sistem, serta prosedur pengkodean yang akan diimplementasikan. Bahasa pemrograman yang digunakan yaitu Python.

1. Model arsitektur

Desain sistem ini memanfaatkan model arsitektur client-server. Pada sisi server, web ini diimplementasikan menggunakan fraks sebagai back-end, yang menyediakan layanan web dan berkomunikasi dengan basis data. YOLOv8 (*You Only Look Once version 8*) adalah model deep learning yang dirancang untuk deteksi objek secara real-time dengan akurasi tinggi dan latensi rendah. Arsitektur YOLOv8 lebih ringan dan efisien dibandingkan versi sebelumnya, dengan berbagai peningkatan pada komponen inti. YOLOv8 terdiri dari backbone, neck, dan head. Backbone bertanggung jawab untuk mengekstraksi fitur penting dari gambar menggunakan convolutional layers, sedangkan neck memproses fitur-fitur ini melalui layer seperti PANet (*Path Aggregation Network*) atau FPN (*Feature Pyramid Network*), memungkinkan deteksi multi-skala. Pada head, prediksi dilakukan pada berbagai skala, di mana model mendeteksi objek dan menentukan kotak pembatas serta klasifikasinya. Dengan desain ini, YOLOv8 mampu melakukan deteksi cepat dengan presisi tinggi.

Salah satu keunggulan utama YOLOv8 adalah kemampuannya untuk melakukan deteksi objek dalam satu tahap, yang menghemat waktu komputasi dibandingkan model lain yang memerlukan beberapa tahap deteksi. Model ini juga dilengkapi dengan peningkatan pada penyesuaian anchor boxes, yang memungkinkan model lebih fleksibel dalam mendeteksi objek dari berbagai ukuran dan posisi. Algoritma optimasi seperti Mosaic augmentation dan data augmentation lainnya juga digunakan untuk meningkatkan performa pada gambar yang lebih bervariasi. YOLOv8 mendukung deteksi objek dalam kondisi pencahayaan yang sulit atau dengan sudut yang kompleks, menjadikannya ideal untuk aplikasi industri, termasuk penghitungan TBS dalam perkebunan atau sistem pengawasan otomatis.

2. Deteksi objek

Dalam melakukan deteksi objek, kita harus memastikan semua library yang dibutuhkan terinstal. Anda dapat menggunakan yolov8 melalui pip, labelimg, ultralytics. Kemudian dapat memuat model YOLOv8 yang sudah dilatih (pretrained) atau melatih model yang kita inginkan menggunakan anaconda.

Bagian menu deteksi yang di gunakan :

a. *Backbone* YOLOv8

Bagian pertama dari desain arsitektur YOLOv8 adalah *backbone*, yang berfungsi untuk mengekstraksi fitur-fitur penting dari gambar input. Backbone pada YOLOv8 menggunakan serangkaian convolutional layers untuk mendeteksi pola, tepi, tekstur, dan fitur visual lainnya dari gambar. Proses ini dikenal sebagai feature extraction, di mana informasi dari gambar diolah secara bertahap melalui layer yang semakin dalam, menghasilkan representasi fitur yang lebih abstrak. YOLOv8 menggunakan teknik modern seperti CSP (*Cross Stage Partial networks*) dan residual connections yang meningkatkan efisiensi pemrosesan data dan mengurangi redundansi informasi, sehingga mempercepat proses deteksi objek sambil menjaga akurasi.

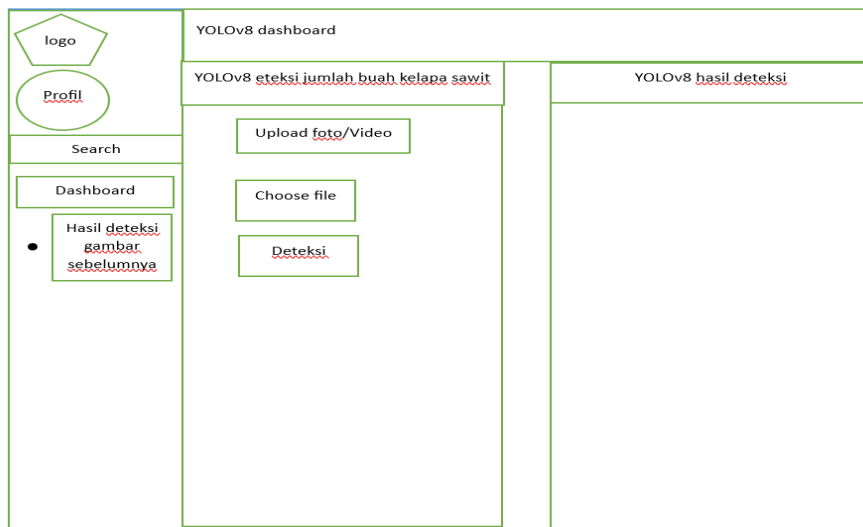
b. Neck YOLOv8

Setelah fitur diekstraksi oleh backbone, bagian neck mengambil peran penting dalam memproses fitur tersebut untuk deteksi multi-skala. Neck pada YOLOv8 menggunakan arsitektur seperti PANet (*Path Aggregation Network*) atau *Feature Pyramid Network* (FPN) yang dirancang untuk menggabungkan fitur dari berbagai level kedalaman (*shallow dan deep layers*). Dengan deteksi multi-skala ini, YOLOv8 mampu mendeteksi objek dengan ukuran yang bervariasi, baik objek kecil maupun besar, dalam satu gambar. Bagian neck ini memainkan peran vital karena memungkinkan model untuk mempertahankan informasi resolusi tinggi pada fitur penting yang membantu meningkatkan kemampuan deteksi objek, terutama dalam skenario kompleks di mana objek mungkin tumpang tindih atau tersembunyi sebagian.

c. Head YOLOv8

Bagian terakhir dari desain YOLOv8 adalah head, yang bertanggung jawab untuk melakukan prediksi akhir. Pada tahap ini, YOLOv8 menghasilkan kotak pembatas (*bounding boxes*) yang mengelilingi objek yang terdeteksi, serta memberikan label klasifikasi untuk setiap objek tersebut. Head juga menggunakan teknik anchor boxes, yang dirancang untuk mencocokkan berbagai ukuran dan bentuk objek dalam gambar. Prediksi dilakukan pada beberapa skala untuk memastikan bahwa objek dari berbagai ukuran dapat terdeteksi dengan akurat. Pada output ini, YOLOv8 tidak hanya memberikan koordinat kotak pembatas, tetapi juga nilai probabilitas untuk setiap kelas objek yang memungkinkan model menghasilkan hasil deteksi yang lebih presisi. Sistem ini mendukung deteksi real-time dengan latensi rendah, yang menjadikannya ideal untuk berbagai aplikasi praktis, termasuk pengawasan, penghitungan hasil panen, atau aplikasi industri lainnya. Head YOLOv8 adalah bagian akhir dari arsitektur model YOLOv8 (*You Only Look Once version 8*), yang bertanggung jawab untuk menghasilkan prediksi akhir dari deteksi objek. Bagian ini terdiri dari beberapa lapisan konvolusi dan output yang berfungsi untuk memprediksi koordinat bounding box, kelas objek, dan confidence score dari setiap objek yang terdeteksi. YOLOv8 menggunakan pendekatan *anchor-free* pada bagian head-nya, berbeda dengan versi sebelumnya yang menggunakan *anchor-based*. Hal ini membuat deteksi lebih cepat dan sederhana karena model tidak perlu mencari *anchor* yang sesuai. Selain itu, head YOLOv8 dirancang untuk memproses keluaran dari fitur yang diekstraksi oleh *backbone* dan *neck*, sehingga dapat mendeteksi objek pada berbagai skala dengan presisi yang lebih tinggi.

3. Tampilan awal

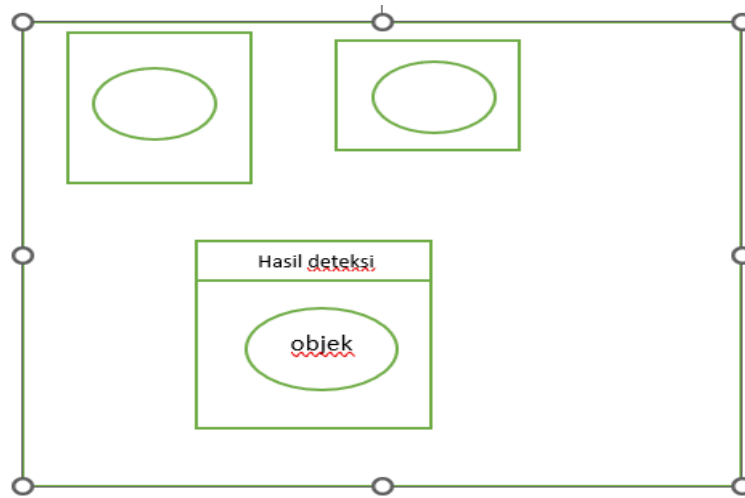


Gambar 4.2 Tampilan awal desain web

- a. Bagian Logo (Pojok kiri atas):
 - 1) Ada tempat untuk logo.
 - 2) Di bawah logo, terdapat ikon profil dengan label "Profil."
- Sidebar (Panel kiri):
- b. Terdapat kolom pencarian (Search).
 - 1) Ada tombol "Dashboard" untuk navigasi.
 - 2) Di bawah tombol dashboard, terdapat pilihan lain yang dapat diklik untuk melihat hasil deteksi gambar sebelumnya, dengan label "Hasil deteksi gambar sebelumnya."
- c. Bagian Utama (Tengah):
 - a) Di bagian header, tertulis "YOLOv8 dashboard."
 - b) Di bawahnya terdapat dua bagian:
 - a). Bagian kiri: "YOLOv8 deteksi jumlah buah kelapa sawit."
 - b). Bagian kanan: "YOLOv8 hasil deteksi."
 - c). Bagian kiri menampilkan tombol untuk mengunggah foto atau video, memilih file, dan ada tombol "Deteksi" untuk memulai proses deteksi.
- d. Bagian Kanan:

Area ini digunakan untuk menampilkan hasil deteksi, dengan label "YOLOv8 hasil deteksi."

1. Hasil dekteksi



Gambar 4.3 Hasil deteksi sawit menggunakan YOLOv8

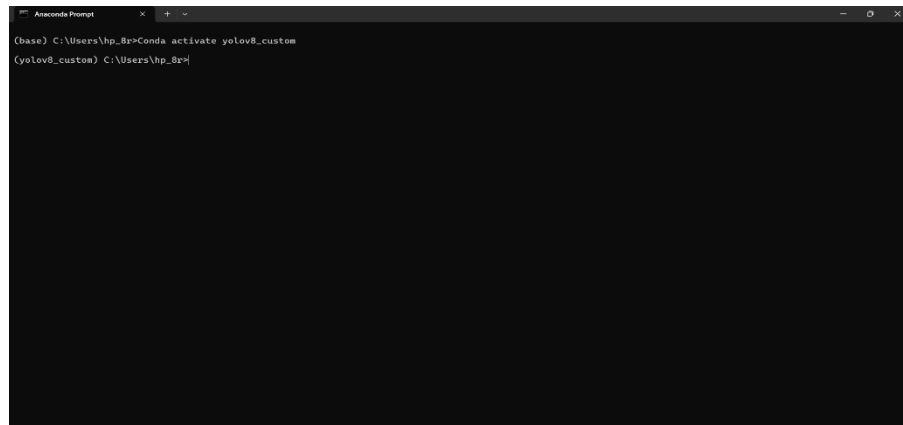
Gambar 4.3 Ini merupakan tempat di mana hasil deteksi dari objek akan ditampilkan setelah proses deteksi dilakukan. Hasil deteksi buah sawit menggunakan YOLOv8 menunjukkan tingkat akurasi yang tinggi dalam identifikasi Tandan Buah Segar (TBS) dan brondolan sawit di Tempat Penampungan Hasil (TPH). Sistem berbasis web ini mampu mendeteksi buah sawit dengan tingkat kepercayaan lebih dari 0,85, yang berarti sistem memiliki keandalan dalam membedakan antara buah sawit dan objek lain di sekitarnya. Confusion matrix digunakan untuk mengukur tingkat kepercayaan deteksi, menunjukkan bahwa pengembangan sistem menggunakan YOLOv8 efektif dalam meningkatkan efisiensi proses perhitungan buah sawit yang sebelumnya dilakukan secara manual.

C. Implementasi

1. Melabeling data

a. Install Aplikasi LabelImg

- Unduh dan instal aplikasi LabelImg di komputer Anda. LabelImg tersedia untuk berbagai platform, termasuk Windows, macOS, dan Linux.
- Setelah diunduh, instal sesuai dengan sistem operasi yang Anda gunakan

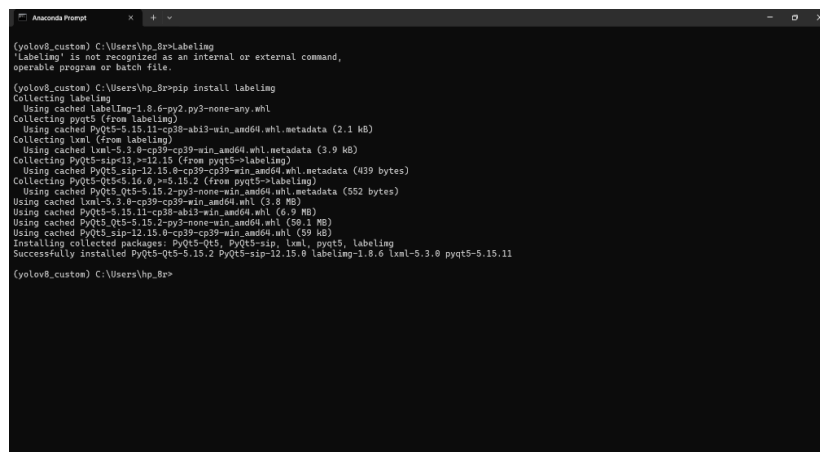


```
Anaconda Prompt
(base) C:\Users\hp_8r>conda activate yolov8_custom
(yolov8_custom) C:\Users\hp_8r>
```

Gambar 4.4 mengaktifkan lingkungan virtual dengan perintah “Conda activate yolov8_custom”

b. Buka Aplikasi dan Muat Gambar

- Buka aplikasi LabelImg.
- Klik tombol Open atau Open Dir untuk memuat gambar atau folder yang berisi gambar yang ingin Anda beri label.



```
Anaconda Prompt
(yolov8_custom) C:\Users\hp_8r>labelimg
'Labelimg' is not recognized as an internal or external command,
operable program or batch file.

(yolov8_custom) C:\Users\hp_8r>pip install labelimg
Collecting labelimg
  Using cached labelimg-1.0.6-py2.py3-none-any.whl
Collecting pyqt5 (from labelimg)
  Using cached PyQt5-5.15.11-cp39-abi3-win_amd64.whl.metadata (2.1 kB)
Collecting lxml (from labelimg)
  Using cached lxml-5.3.0-cp39-cp39-win_amd64.whl.metadata (3.9 kB)
Collecting PyQt5-sip==12.15 (from pyqt5->labelimg)
  Using cached PyQt5_sip-12.15.0-cp39-cp39-win_amd64.whl.metadata (439 bytes)
Collecting PyQt5-Qt5==5.15.2 (from pyqt5->labelimg)
  Using cached PyQt5-Qt5-5.15.2-py2.py3-none-win_amd64.whl.metadata (552 bytes)
Using cached lxml-5.3.0-cp39-cp39-win_amd64.whl (3.8 MB)
Using cached PyQt5-5.15.11-cp38-abi3-win_amd64.whl (6.9 MB)
Using cached PyQt5-Qt5-5.15.2-py2.py3-none-win_amd64.whl (59.1 MB)
Using cached PyQt5_sip-12.15.0-cp39-cp39-win_amd64.whl (59 kB)
Installing collected packages: PyQt5-Qt5, PyQt5-sip, lxml, pyqt5, Labelimg
Successfully installed PyQt5-Qt5-5.15.2 PyQt5-sip-12.15.0 labelimg-1.0.6 lxml-5.3.0 pyqt5-5.15.11

(yolov8_custom) C:\Users\hp_8r>
```

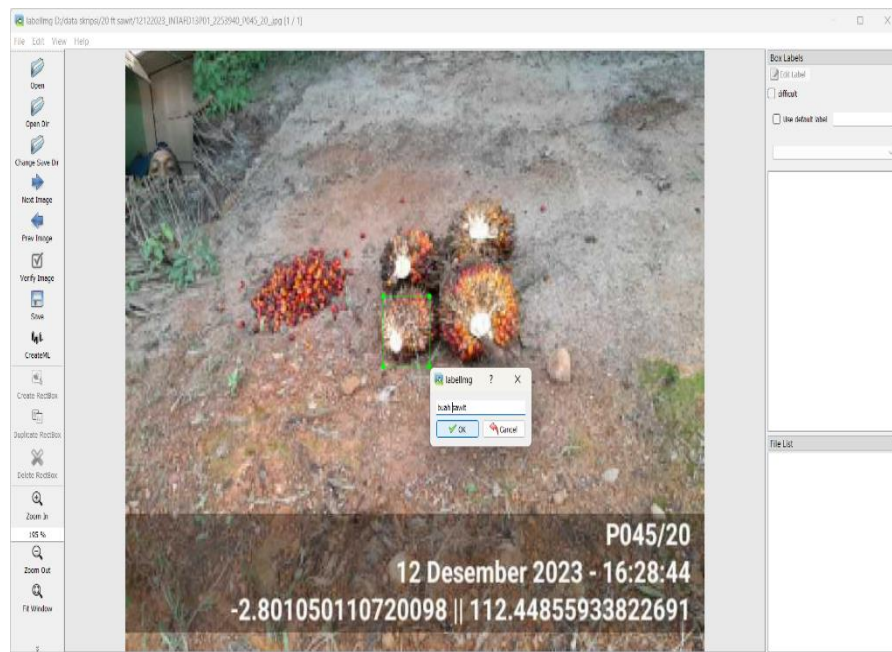
Gambar 4.5 Untuk membuka aplikasi labeling

c. Pilih Format Label (PascalVOC atau YOLO)

- Sebelum memulai pelabelan, pilih format output file yang ingin Anda gunakan. LabelImg mendukung dua format utama:
 1. PascalVOC (XML): Format default untuk dataset VOC.
 2. YOLO (TXT): Format yang digunakan untuk deteksi objek dengan model YOLO.

d. Mulai Pelabelan Objek

- Klik tombol Create RectBox atau tekan tombol W pada keyboard untuk memulai pelabelan.
- Pilih objek pada gambar dengan menarik kotak pembatas (bounding box) di sekitar objek yang ingin dilabeli.
- Setelah kotak dibuat, sebuah jendela pop-up akan muncul untuk meminta Anda memasukkan nama label atau kelas objek (misalnya, "TBS", "Mobil", "Orang", dll.).
- Klik OK setelah memberi label pada objek tersebut.



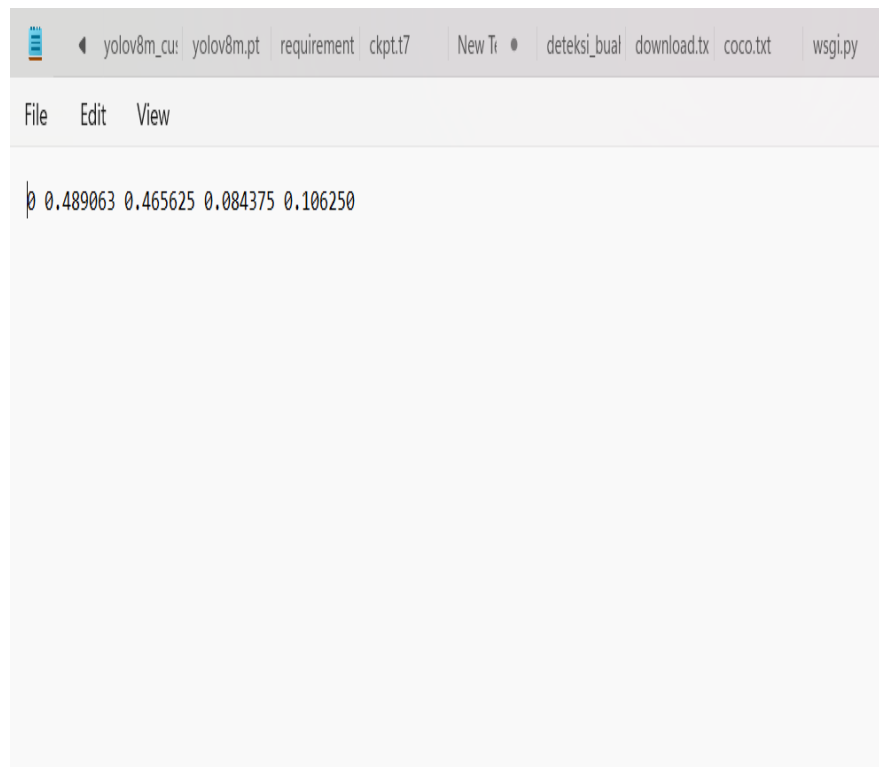
Gambar 4.6 proses untuk memberikan lebelan

e. Simpan Label

- Setelah selesai memberi label pada gambar, simpan file label tersebut dengan menekan tombol Save atau tekan Ctrl+S.
- File label akan disimpan dalam format yang dipilih sebelumnya (XML untuk PascalVOC atau TXT untuk YOLO) di folder yang sama dengan gambar.

f. Melanjutkan Pelabelan Gambar Lainnya

- Untuk melabeli gambar lainnya, gunakan tombol panah Next Image atau Prev Image untuk berpindah ke gambar berikutnya atau sebelumnya dalam folder.
- Ulangi proses pelabelan hingga semua gambar yang diinginkan selesai.

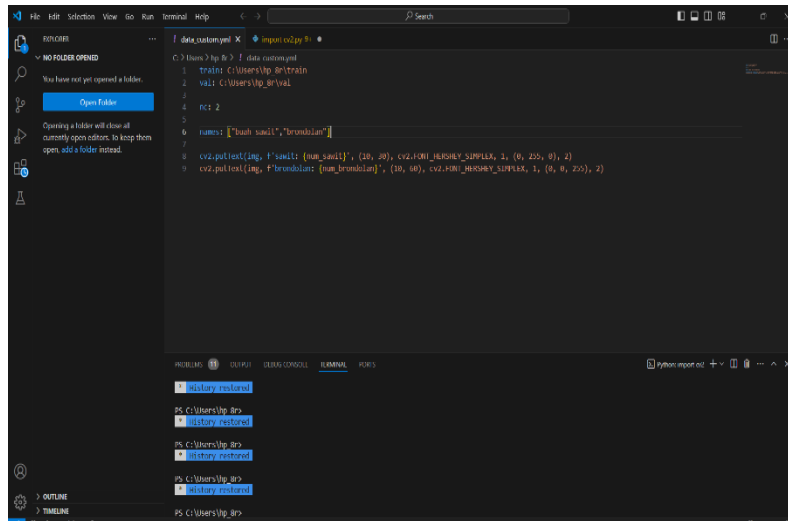


Gambar 4.7 Hasil dari pebelan

2. Pelatihan algoritma YOLOv8

a. Persiapkan Dataset:

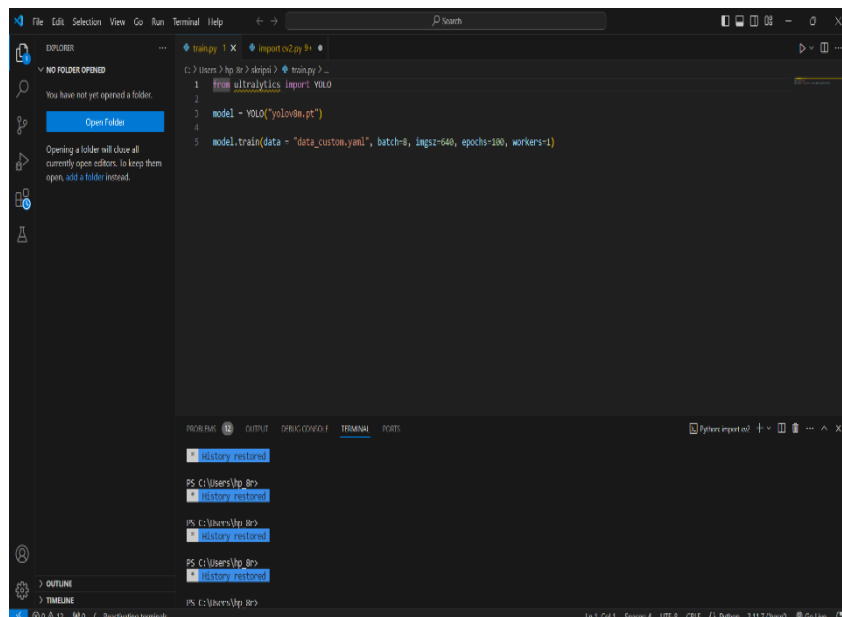
Siapkan dataset Anda dalam format yang kompatibel dengan YOLOv8. Biasanya, data pelatihan diorganisasi dalam folder dengan gambar dan file label yang sesuai.



Gambar 4.8 Persiapan dataset

b. Buat File Konfigurasi Dataset:

Buat file konfigurasi YAML yang mendefinisikan jalur ke data pelatihan dan validasi.



Gambar 4.9 Konfigurasi dataset

2. Mulai Pelatihan:

- Jalankan pelatihan model dengan perintah berikut. Gantilah data.yaml dan model.cfg sesuai dengan konfigurasi Anda: “yolo train data=data.yaml model=yolov8n.pt epochs=50 batch=16 imgsz=640”
- Dalam perintah ini:
 1. data=data.yaml merujuk ke file konfigurasi dataset.
 2. model=yolov8n.pt adalah model YOLOv8 yang digunakan sebagai titik awal (pretrained model).
 3. epochs=50 menentukan jumlah epoch pelatihan.
 4. batch=16 adalah ukuran batch.
 5. imgsz=640 menentukan ukuran gambar input.

3. Monitoring dan Evaluasi

- Monitor Proses Pelatihan:

Selama pelatihan, Anda dapat memantau proses melalui log yang ditampilkan di terminal. Ini biasanya mencakup metrik seperti loss dan akurasi.

- Evaluasi Model:

Setelah pelatihan selesai, evaluasi model Anda dengan menggunakan dataset validasi untuk memeriksa performa. Gunakan skrip atau perintah yang disediakan oleh YOLOv8 untuk mengevaluasi hasil model.

4. Simpan dan Gunakan Model

- Simpan Model:

Model yang telah dilatih akan disimpan di direktori output. Anda dapat menggunakan model ini untuk inferensi atau untuk melanjutkan pelatihan di masa mendatang.

- Gunakan Model untuk Inferensi:

Setelah pelatihan selesai, gunakan model untuk inferensi pada gambar baru atau video dengan perintah berikut: “yolo predict model=yolov8n.pt

```
source=/path/to/image.jpg”
```

Untuk menjalankan pelatihan “Yolo task=detect mode=train epochs=100 data=data_custom.yml model=yolov8m.pt imgsz=640” Perintah ini digunakan untuk melatih model YOLOv8 pada tugas deteksi objek. Berikut adalah rincian setiap bagiannya:

1. yolo

- yolo adalah perintah utama yang digunakan untuk menjalankan YOLOv8. YOLOv8 dilengkapi dengan CLI (Command Line Interface), sehingga perintah ini mengeksekusi berbagai tugas yang terkait dengan model YOLOv8, seperti pelatihan, prediksi, dan evaluasi.

2. task=detect

- task=detect menunjukkan bahwa tugas yang akan dilakukan adalah deteksi objek. YOLOv8 mendukung beberapa tugas lain seperti segmentasi atau klasifikasi, tetapi dalam konteks ini, tugas utamanya adalah mendeteksi objek dalam gambar.

3. mode=train

- mode=train menunjukkan bahwa mode yang dipilih adalah pelatihan. Ini berarti model akan dilatih menggunakan dataset yang disediakan. YOLOv8 mendukung beberapa mode seperti:
 - a) predict: Untuk melakukan inferensi atau prediksi.
 - b) val: Untuk melakukan evaluasi terhadap model.

4. epochs=100

- epochs=100 menentukan jumlah epoch untuk pelatihan, yaitu 100 epoch. Setiap epoch mewakili satu kali iterasi penuh dari pelatihan di seluruh dataset.
 - a) Semakin tinggi jumlah epoch, semakin lama waktu pelatihan, tetapi biasanya ini meningkatkan kemampuan model untuk belajar lebih baik dari data.
 - b) Anda dapat menyesuaikan jumlah epoch tergantung pada kebutuhan pelatihan dan seberapa baik model berkinerja selama pelatihan.

5. data=data_custom.yml

- 1) data=data_custom.yml merujuk pada file konfigurasi data (dalam format

YAML) yang mendefinisikan lokasi dataset pelatihan dan validasi.

a) File ini biasanya berisi informasi seperti jalur ke gambar dan label, jumlah kelas, dan nama kelas.

6. model=yolov8m.pt

model=yolov8m.pt menentukan model YOLOv8 versi "m" (medium) yang akan digunakan sebagai titik awal pelatihan. YOLOv8 menyediakan beberapa ukuran model dengan trade-off antara kecepatan dan akurasi :

- a) yolov8n.pt (nano): Model kecil, lebih cepat, tetapi dengan akurasi yang lebih rendah.
- b) yolov8s.pt (small): Model kecil-menengah, lebih cepat, tetapi dengan akurasi yang sedikit lebih baik.
- c) yolov8m.pt (medium): Model ukuran menengah, memberikan keseimbangan antara kecepatan dan akurasi.
- d) yolov8l.pt (large): Model besar, lebih lambat tetapi lebih akurat.
- e) yolov8x.pt (extra-large): Model terbesar dengan akurasi tertinggi tetapi dengan waktu inferensi yang lebih lambat.
- f) Penggunaan pretrained model ini memungkinkan model untuk memulai pelatihan dari titik yang sudah "terlatih" (transfer learning) sehingga lebih cepat mencapai konvergensi dan menghasilkan akurasi yang lebih baik.

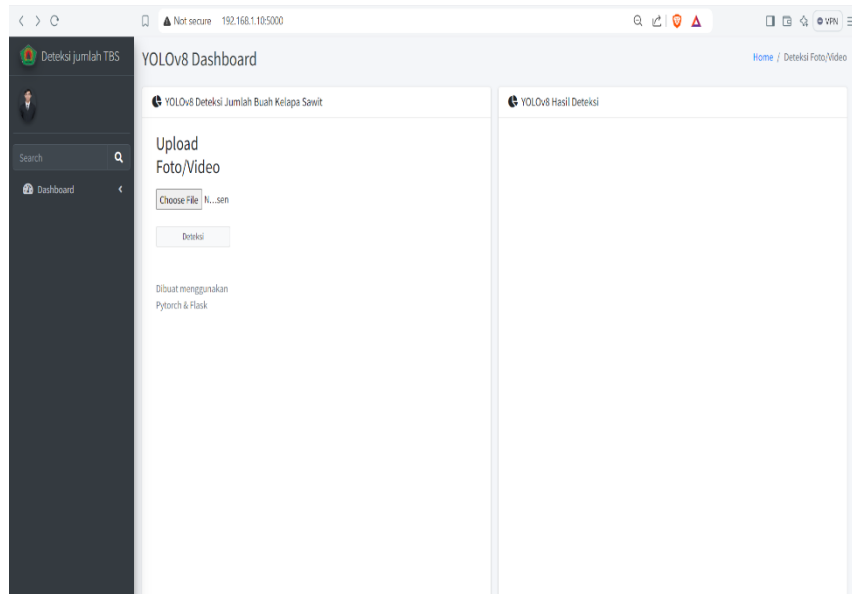
7. imgsz=640

- imgsz=640 menentukan ukuran input gambar yang digunakan dalam pelatihan. Gambar akan diskalakan ke resolusi 640x640 piksel.
 - 1) Ukuran gambar yang lebih besar memungkinkan deteksi lebih detail, tetapi membutuhkan lebih banyak memori dan waktu komputasi.
 - 2) Resolusi ini adalah ukuran standar untuk YOLOv8, dan Anda bisa menyesuaikannya tergantung pada kebutuhan dataset dan kapasitas perangkat keras (GPU).

D. Confusion matrix

1. Proses confusion matrix

a. Tampilan awal

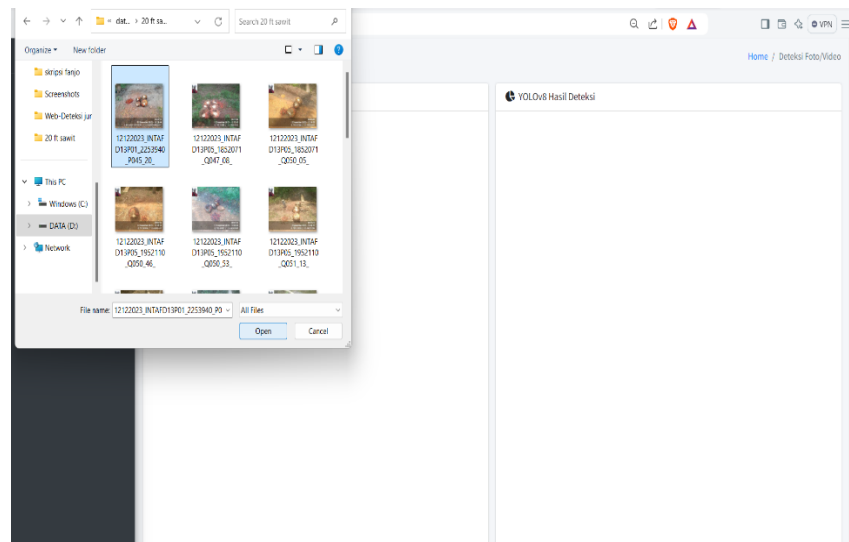


Gambar 4.10 Halaman utama dalam web

Gambar 4.10 Halaman utama web penghitung cacah TBS menggunakan algoritma YOLOv8, ini menampilkan sebuah antarmuka pengguna (user interface) dari sebuah aplikasi web yang digunakan untuk mendeteksi jumlah Tandan Buah Segar (TBS) pada tanaman kelapa sawit. Aplikasi ini dibangun menggunakan kerangka kerja Flask, yang merupakan teknologi populer untuk pengembangan aplikasi pembelajaran mesin.

Pada bagian kiri, terdapat menu navigasi yang memungkinkan pengguna untuk beralih antara halaman beranda (dashboard) dan halaman untuk mengunggah gambar atau video. Pengguna dapat memilih file gambar atau video yang ingin diproses untuk deteksi jumlah TBS. Setelah file diunggah, aplikasi akan memproses gambar atau video tersebut menggunakan model YOLOv8, yang merupakan salah satu model deteksi objek yang populer. Hasil deteksi kemudian akan ditampilkan pada bagian kanan layar, menunjukkan lokasi dan jumlah TBS yang terdeteksi pada gambar atau video tersebut.

b. Dashboard file



Gambar 4.11 Dashboard adalah tempat file yang pernah digunakan

Gambar ini menampilkan tampilan dashboard yang kemungkinan besar digunakan untuk mengelola dan mengorganisir data gambar. Dashboard ini menunjukkan adanya beberapa folder yang berisi kumpulan gambar, yang ditandai dengan nama-nama file. Folder-folder ini terorganisir dengan baik dalam struktur direktori yang jelas, memudahkan pengguna untuk menemukan gambar yang diinginkan. Fitur pencarian juga tersedia untuk membantu mempercepat proses pencarian.

Dashboard ini juga mengindikasikan adanya proses pengolahan gambar menggunakan model deteksi objek YOLOv8. Hal ini terlihat dari adanya folder "YOLOv8 Hasil Deteksi" yang berisi hasil dari proses deteksi objek pada kumpulan gambar tersebut. Fitur-fitur lain yang terdapat pada dashboard ini antara lain kemampuan untuk memvisualisasikan hasil deteksi. Secara keseluruhan, dashboard ini memberikan tampilan yang terpusat dan interaktif untuk mengelola dan menganalisis data gambar dalam konteks proyek deteksi objek.

C. Deteksi Hasil



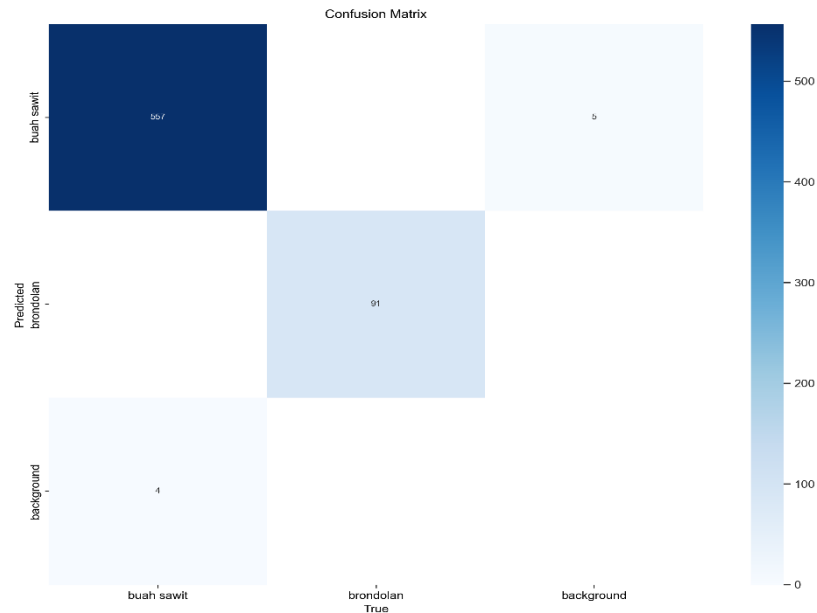
Gambar 4.12 Hasil deteksi menggunakan YOLOv8

Hasil deteksi gambar menunjukkan bahwa sistem telah berhasil mengidentifikasi beberapa objek, yaitu buah sawit dan brondolan. Objek-objek ini ditandai dengan kotak berwarna biru dan disertai dengan label serta tingkat kepercayaan (confidence score). Tingkat kepercayaan ini menunjukkan seberapa yakin sistem bahwa objek yang terdeteksi adalah benar-benar buah sawit atau brondolan. Pada gambar ini, tingkat kepercayaan untuk sebagian besar deteksi buah sawit cukup tinggi, yaitu di atas 0.85, yang mengindikasikan bahwa sistem sangat yakin dengan hasil deteksinya.

2. Hasil confusion matrix

Hasil testing dengan menggunakan Confusion matrix :

1. Confusion matrix



Gambar 4.13 Gambar yang menunjukkan tingkat kebenaran suatu objek

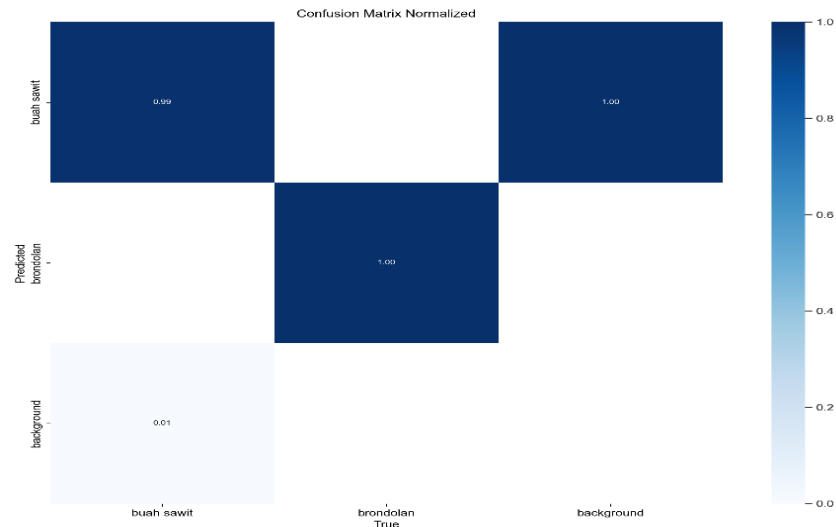
a) Sumbu-sumbu:

- 1) Sumbu-x (horizontal): Menunjukkan kelas aktual (true label) dari data. Dalam kasus ini, kelas aktualnya adalah "buah sawit", "brondolan", dan "background".
- 2) Sumbu-y (vertikal): Menunjukkan kelas yang diprediksi (predicted label) oleh model. Kelas-kelas yang diprediksi juga sama dengan kelas aktual.

b). Angka dalam Sel: Setiap angka dalam sel mewakili jumlah (atau proporsi) data yang seharusnya berada di kelas tertentu (sumbu-x) tetapi diprediksi oleh model berada di kelas lain (sumbu-y).

c). Warna: Warna yang digunakan pada *heat map* ini menunjukkan tingkat kepercayaan model terhadap prediksinya. Warna biru tua menunjukkan tingkat kepercayaan yang tinggi, sedangkan warna putih menunjukkan tingkat kepercayaan yang rendah.

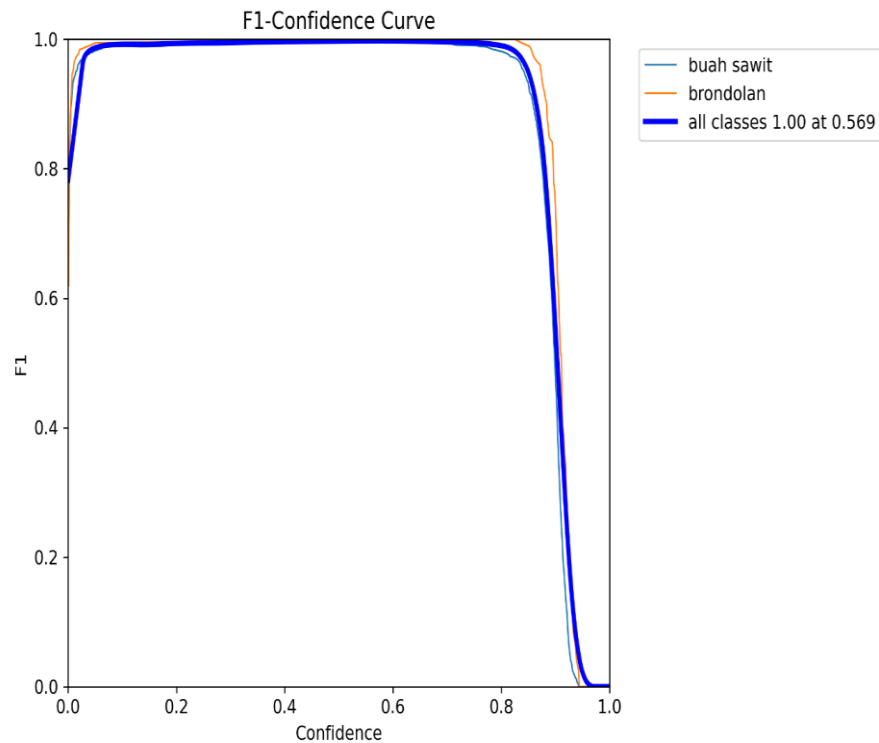
2. Confusion matrix normalized



Gambar 4.14 Confusion matrix normalized

Normalisasi matriks kebingungan adalah teknik yang digunakan untuk menskalakan nilai-nilai dalam matriks kebingungan agar lebih mudah diinterpretasikan dan dibandingkan, terutama saat menghadapi dataset yang tidak seimbang. Matriks kebingungan adalah tabel yang digunakan untuk mengevaluasi kinerja model klasifikasi dengan membandingkan klasifikasi yang diprediksi dan yang sebenarnya. Normalisasi matriks kebingungan dapat memberikan pemahaman yang lebih jelas tentang kinerja model dengan mengubah hitungan mentah menjadi proporsi atau persentase.

3. F1-Confidence curve



Gambar 4.15 . F1-Confidence curve

Gambar 4.15 F1-Confidence Curve ini menggambarkan hubungan antara tingkat kepercayaan (confidence) suatu model dalam mengklasifikasikan suatu objek (dalam hal ini, buah sawit dan brondolan) dengan nilai F1-score yang dihasilkan.

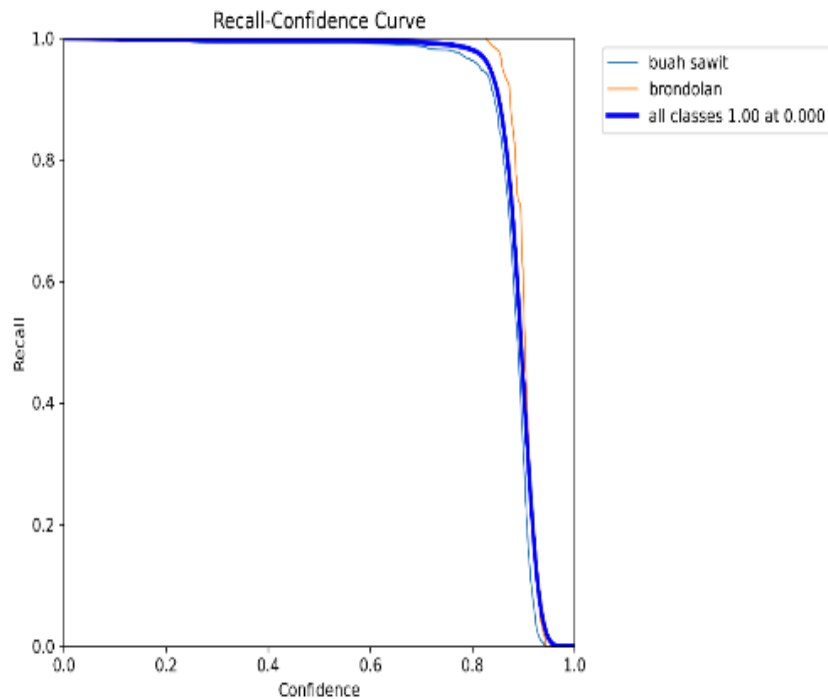
Elemen-elemen penting dalam gambar ini:

- a) Sumbu X (Confidence): Menunjukkan tingkat kepercayaan model dalam membuat prediksi. Nilai 0 berarti model sama sekali tidak yakin, sedangkan nilai 1 berarti model sangat yakin dengan prediksinya.
- b) Sumbu Y (F1-score): Merupakan metrik evaluasi yang mengukur akurasi dan recall model. Nilai F1-score berkisar antara 0 hingga 1, di mana nilai 1 menunjukkan kinerja sempurna.
- c) Kurva:
 - 1) Buah Sawit: Kurva ini menunjukkan hubungan antara tingkat kepercayaan model dalam mengklasifikasikan buah sawit dengan nilai F1-score yang dihasilkan.
 - 2) Brondolan: Kurva ini menunjukkan hubungan antara tingkat

kepercayaan model dalam mengklasifikasikan brondolan dengan nilai F1-score yang dihasilkan.

- 3) All Classes: Kurva rata-rata dari kedua kelas, menunjukkan kinerja model secara keseluruhan. Angka 1.00 menunjukkan F1-score tertinggi yang dicapai model pada tingkat kepercayaan tertentu (0.569).

4. Recall- Confidence Curve



Gambar 4.16 Recall- Confidence Curve

Gambar 4.16 menggambarkan hubungan antara tingkat kepercayaan (confidence) suatu model dalam mengklasifikasikan suatu objek (dalam hal ini, buah sawit dan brondolan) dengan nilai recall yang dihasilkan. Recall adalah metrik yang mengukur kemampuan model dalam mengidentifikasi semua contoh positif (dalam hal ini, semua buah sawit dan brondolan yang sebenarnya).

Elemen-elemen penting dalam gambar ini:

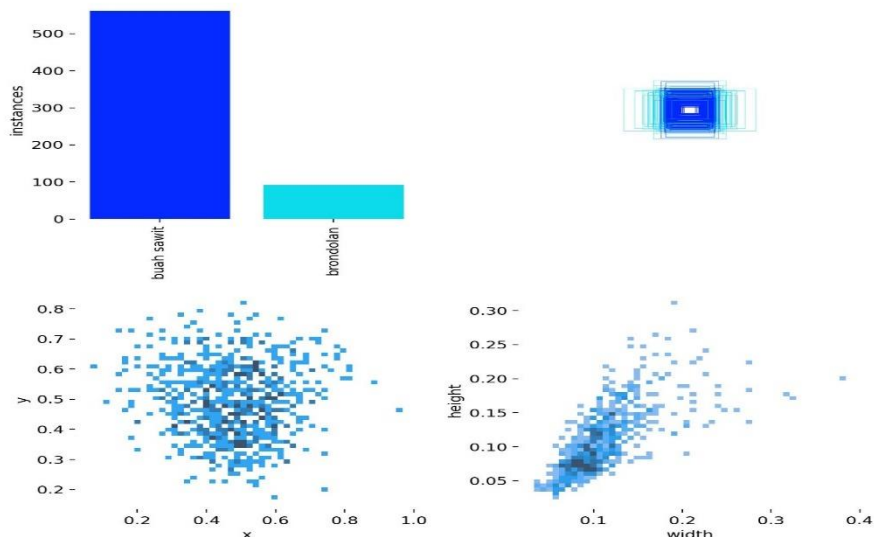
- a) Sumbu X (Confidence): Sama seperti sebelumnya, sumbu ini menunjukkan tingkat kepercayaan model dalam membuat prediksi. Nilai 0 berarti model sama sekali tidak yakin, sedangkan nilai 1 berarti model sangat yakin dengan prediksinya.
- b) Sumbu Y (Recall): Menunjukkan proporsi contoh positif yang

benar-benar diidentifikasi oleh model. Nilai recall yang tinggi berarti model mampu menemukan sebagian besar contoh positif yang ada.

c) Kurva:

- 1) Buah Sawit: Kurva ini menunjukkan hubungan antara tingkat kepercayaan model dalam mengklasifikasikan buah sawit dengan nilai recall yang dihasilkan.
- 2) Brondolan: Kurva ini menunjukkan hubungan antara tingkat kepercayaan model dalam mengklasifikasikan brondolan dengan nilai recall yang dihasilkan.
- 3) All Classes: Kurva rata-rata dari kedua kelas, menunjukkan kinerja model secara keseluruhan. Angka 1.00 menunjukkan recall tertinggi yang dicapai model pada tingkat kepercayaan tertentu (0.000).

5. Visualisasi data



Gambar 4.17 Visualisasi data

Gambar 4. 17 ini sepertinya merupakan visualisasi data yang berkaitan dengan karakteristik fisik dari dua jenis objek, yaitu "buah sawit" dan "brondolan". Visualisasi ini terbagi menjadi beberapa plot yang saling melengkapi untuk memberikan gambaran yang lebih komprehensif.

Penjelasan Setiap Plot

a). Bar Plot:

- Sumbu-X: Kategori objek (buah sawit dan brondolan)
- Sumbu-Y: Jumlah instance atau data titik untuk setiap kategori
- Interpretasi: Plot ini menunjukkan distribusi jumlah data untuk setiap kategori. Tampaknya jumlah data untuk "buah sawit" jauh lebih banyak dibandingkan dengan "brondolan".

b). Scatter Plot dengan Kotak Batas:

- Sumbu-X: Width (lebar)
- Sumbu-Y: Height (tinggi)
- Interpretasi: Plot ini menunjukkan distribusi data berdasarkan lebar dan tinggi objek. Kotak-kotak transparan mewakili batas-batas dari beberapa kelompok data. Ini mungkin menunjukkan adanya beberapa kelompok atau cluster dalam data.

c). Histogram 2D:

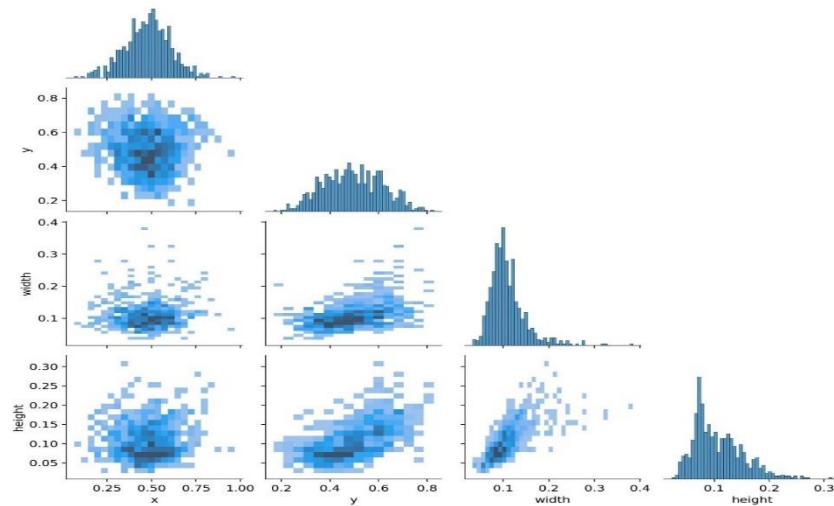
Plot Kiri:

- Sumbu-X: Nilai X (mungkin merupakan fitur lain dari objek)
- Sumbu-Y: Nilai Y (mungkin merupakan fitur lain dari objek)
- Interpretasi: Plot ini menunjukkan distribusi frekuensi gabungan dari dua fitur numerik. Tampaknya ada beberapa pola dalam distribusi data, misalnya konsentrasi data pada rentang nilai X tertentu.

Plot Kanan:

- Sumbu-X: Width (lebar)
- Sumbu-Y: Height (tinggi)
- Interpretasi: Plot ini serupa dengan plot sebelumnya, tetapi fokus pada distribusi lebar dan tinggi.

6. Plot matriks



Gambar 4.18 plot matriks

Gambar 4.18 di atas adalah sebuah plot matriks (matrix plot) yang menunjukkan distribusi dan hubungan antara tiga variabel: *width*, *height*, dan *y*.

Mari kita bahas bagian-bagian dari plot ini:

a) Diagram Histogram:

Setiap diagonal utama plot (kiri atas ke kanan bawah) menampilkan histogram untuk masing-masing variabel. Histogram ini menunjukkan distribusi frekuensi dari nilai-nilai variabel tersebut. Misalnya, histogram untuk variabel *width* menunjukkan berapa banyak data yang memiliki nilai *width* tertentu.

b) Diagram Scatter Plot:

Diagram di luar diagonal utama menunjukkan hubungan antara dua variabel. Misalnya, diagram di baris kedua kolom pertama menunjukkan hubungan antara variabel *y* dan *width*. Semakin banyak titik data yang berkumpul di suatu area, maka semakin kuat korelasi antara kedua variabel tersebut.

Dari plot ini, kita dapat mengamati beberapa hal:

a) Distribusi Variabel:

Variabel *width* dan *height* memiliki distribusi yang cenderung normal (bentuk lonceng), sementara variabel *y* memiliki distribusi yang sedikit miring ke kanan.

b) Hubungan Antar Variabel:

- 1) Terlihat adanya korelasi positif antara *width* dan *height*, artinya semakin besar nilai *width*, maka nilai *height* cenderung juga semakin besar.
- 2) Hubungan antara variabel *y* dengan *width* dan *height* juga terlihat cukup kuat, namun pola hubungannya lebih kompleks dan mungkin memerlukan analisis lebih lanjut.

Bab V

KESIMPILAN DAN SARAN

A. Kesimpulan

1. Instal model YOLOv8, latih dengan dataset TBS berkualitas, lalu integrasikan dengan framework web seperti Flask atau Django. Tambahkan library frontend untuk membuat antarmuka interaktif.
2. Implementasi YOLOv8 berbasis web: Model YOLOv8 dilatih, diintegrasikan ke aplikasi web, dan memproses video real-time dari kamera di TPH untuk mendeteksi TBS. Jumlah TBS yang terdeteksi ditampilkan secara real-time di antarmuka.
3. Langkah-langkah: Latih model, terapkan pada video, deteksi TBS di setiap frame, dan hitung jumlah TBS berdasarkan bounding box yang dihasilkan.

B. Saran

1. Integrasi Sensor Tambahan Menggabungkan aplikasi YOLOv8 dengan sensor berat atau RFID di TPH untuk meningkatkan akurasi penghitungan, sehingga tidak hanya berdasarkan deteksi visual, tetapi juga memperhitungkan jumlah dan berat buah sawit secara otomatis.
2. Optimasi Deteksi di Kondisi Minim Cahaya Menambahkan fitur optimasi pencahayaan atau algoritma pengolahan gambar untuk meningkatkan akurasi deteksi buah sawit pada kondisi cahaya rendah atau saat malam hari.
3. Menambahkan menu untuk menunjukkan jumlah suatu objek (Sawit)

DAFTAR PUSTAKA

- Eriana, E. S., & Zein, A. (2023). Artificial Intelligence. In *Eureka Media Aksara*.
- Muhlashin, M. N. I., & Stefanie, A. (2023). Klasifikasi Penyakit Mata Berdasarkan Citra Fundus Menggunakan YOLO V8. *JATI (Jurnal Mahasiswa Teknik Informatika)*, 7(2), 1363–1368. <https://doi.org/10.36040/jati.v7i2.6927>
- Ngapiyatun, S., Kurniadi, Z., Hidayat, N., Winarni, B., & Obeth, E. (2021). Hubungan Sistem Transportasi Dump Truck pada Pengangkutan Tandan Buah Segar ke Pabrik Kelapa Sawit. *Buletin Loupe*, 17(01), 35–40. <https://doi.org/10.51967/buletinloupe.v17i01.476>
- Noviantoro, A., Silviana, A. B., Fitriani, R. R., & Permatasari, H. P. (2022). Rancangan Dan Implementasi Aplikasi Sewa Lapangan Badminton Wilayah Depok Berbasis Web. *Jurnal Teknik Dan Science*, 1(2), 88–103. <https://doi.org/10.56127/jts.v1i2.108>
- Nugroho, A. (2019). Teknologi Agroindustri Kelapa Sawit. In *Lambung Mengkurat Universitas Press* (Issue August). https://www.researchgate.net/profile/Agung-Nugroho-13/publication/337315913_Buku_Teknologi_Agroindustri_Kelapa_Sawit/links/5dd1694792851c382f469b34/Buku-Teknologi-Agroindustri-Kelapa-Sawit.pdf
- Putra, W. P. N., Pradana, A. I., & Nurchim. (2024). Implementasi Sistem Penghitungan Volume Kendaraan Menggunakan. *Jurnal Fasilkom*, 14(2), 443–450.
- Rahmawati, A. (2023). Keragaman Genetik Varietas Kelapa Sawit (*Elaeis guineensis* Jacq.). *Jurnal Kridatama Sains Dan Teknologi*, 5(01), 35–40. <https://doi.org/10.53863/kst.v5i01.677>
- Reztano, S. L., Seno, Y. S. A., & Priyambada. (2023). Analisis Efektivitas Mekanisme Pengangkutan Buah dari Pohon Kelapa Sawit Menuju Tempat Pengumpulan Hasil (TPH) dengan Mesin Angkut Sepeda Motor Berkeranjang pada Berbagai Kondisi Medan. *Jurnal Mahasiswa Instiper*, 1(2), 1179–1184.
- Rizal, C., Supiyandi, Zen, M., & Eka, M. (2022). Perancangan Server Kantor Desa Tomuan Holbung Berbasis Client Server. *Bulletin of Information Technology (BIT)*, 3(1), 27–33. <https://doi.org/10.47065/bit.v3i1.255>
- Zakaria, R. N. (2024). Sistem presensi wajah mahasiswa menggunakan yolov8.

LAMPIRAN

Lampiran 1.1 Berikut adalah syntax file webapp.py yang digunakan untuk mengkonfigurasi tampilan halaman web yang digunakan untuk mendeteksi jumlah TBS di TPH menggunakan pelatihan algoritma YOLOv8 :

```
import argparse
import io
from PIL import Image
import datetime

import torch
import cv2
import numpy as np
import tensorflow as tf
from re import DEBUG, sub
from flask import Flask, render_template, request, redirect, send_file, url_for, Response
from werkzeug.utils import secure_filename, send_from_directory
import os
import subprocess
from subprocess import Popen
import re
import requests
import shutil
import time
import glob

from ultralytics import YOLO

app = Flask(__name__)

@app.route("/")
def hello_world():
    return render_template('index.html')

@app.route("/", methods=["GET", "POST"])
def predict_img():
    if request.method == "POST":
        if 'file' in request.files:
            f = request.files['file']
            basepath = os.path.dirname(__file__)
            filepath = os.path.join(basepath, 'uploads', f.filename)
            print("upload folder is ", filepath)
            f.save(filepath)
            global imgpath
            predict_img.imgpath = f.filename
            print("printing predict_img ::::: ", predict_img)
```

```

file_extension = f.filename.rsplit('.', 1)[1].lower()

if file_extension == 'jpg':
    img = cv2.imread(filepath)

    model = YOLO('yolov8m_custom.pt')
    detections = model(img, save=True)
    return display(f.filename)

elif file_extension == 'mp4':
    video_path = filepath
    cap = cv2.VideoCapture(video_path)

    frame_width = int(cap.get(cv2.CAP_PROP_FRAME_WIDTH))
    frame_height = int(cap.get(cv2.CAP_PROP_FRAME_HEIGHT))

    fourcc = cv2.VideoWriter_fourcc(*'mp4v')
    out = cv2.VideoWriter('output.mp4', fourcc, 30.0, (frame_width,
frame_height))

    model = YOLO('yolov8m_custom.pt')

    while cap.isOpened():
        ret, frame = cap.read()
        if not ret:
            break

        results = model(frame, save=True)
        print(results)
        cv2.waitKey(1)

        res_plotted = results[0].plot()
        cv2.imshow("result", res_plotted)

        out.write(res_plotted)

        if cv2.waitKey(1) == ord('q'):
            break

    return video_feed()

folder_path = 'runs/detect'
subfolders = [f for f in os.listdir(folder_path) if os.path.isdir(os.path.join(folder_path,
f))]
    latest_subfolder = max(subfolders, key=lambda x:
os.path.getctime(os.path.join(folder_path, x)))
    image_path = folder_path+'/'+latest_subfolder+'/'+f.filename
    return render_template('index.html', image_path=image_path)

```

```

@app.route('/<path:filename>')
def display(filename):
    folder_path = 'runs/detect'
    subfolders = [f for f in os.listdir(folder_path) if os.path.isdir(os.path.join(folder_path,
f))]
        latest_subfolder = max(subfolders, key=lambda x:
os.path.getctime(os.path.join(folder_path, x)))
    directory = folder_path+'/'+latest_subfolder
    print("printing directory: ",directory)
    files = os.listdir(directory)
    latest_file = files[0]

    print(latest_file)

    filename = os.path.join(folder_path, latest_subfolder, latest_file)

    file_extension = filename.rsplit('.', 1)[1].lower()

    environ = request.environ
    if file_extension == 'jpg':
        return send_from_directory(directory,latest_file,environ)

    else:
        return "Invalid file format"

def get_frame():
    folder_path = os.getcwd()
    mp4_files = 'output.mp4'
    video = cv2.VideoCapture(mp4_files)
    while True:
        success, image = video.read()
        if not success:
            break
        ret, jpeg = cv2.imencode('.jpg', image)

        yield (b'--frame\r\n'
            b'Content-Type: image/jpeg\r\n\r\n' + jpeg.tobytes() + b'\r\n\r\n')
        time.sleep(0.1)

@app.route("/video_feed")
def video_feed():
    print("function called")

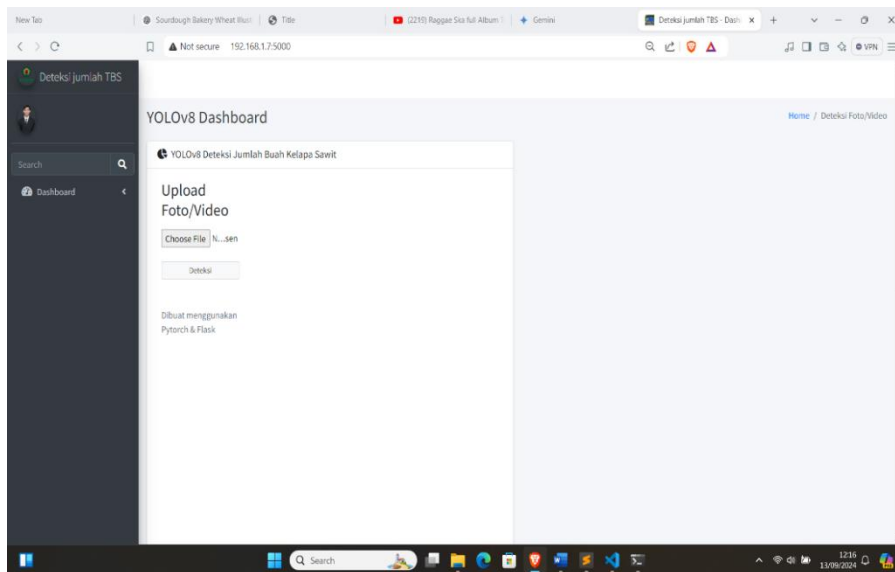
    return Response(get_frame(),

```



```
mimetype='multipart/x-mixed-replace; boundary=frame')
```

```
if __name__ == "__main__":  
    parser = argparse.ArgumentParser(description="Flask app exposing yolov8 models")  
    parser.add_argument("--port", default=5000, type=int, help="port number")  
    args = parser.parse_args()  
    model = YOLO('yolov8m_custom.pt')  
    app.run(host="0.0.0.0", port=args.port)
```



Gambar tampilan web Penghitung jumlah TBS

Lampiran 1.2 Syintax berikut digunakan untuk membuat judul halaman web

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <!-- <link rel="icon" href="/static/assets/favicon.ico" type="image/x-icon"> -->

  <title>
    Flask - {% block title %}{% endblock %} | Aarohi
  </title>

  {% block stylesheets %}{% endblock stylesheets %}

</head>
<body class="hold-transition {% block body_class %}{% endblock body_class %}">
<div class="wrapper">

  {% include 'navigation.html' %}

  {% include 'sidebar.html' %}

  {% block content %}{% endblock content %}

  {% include 'footer.html' %}

  <!-- Control Sidebar -->
  <aside class="control-sidebar control-sidebar-dark">
    <!-- Control sidebar content goes here -->
  </aside>
  <!-- /.control-sidebar -->

</div>
<!-- ./wrapper -->

<!-- Specific Page JS goes HERE -->
{% block javascripts %}{% endblock javascripts %}

</body>
</html>
```

1.3 syntax berikut digunakan untuk mengedit tampilan halaman web

```
{% extends "base.html" % }

{% block title % } Dashboard {% endblock % }

<!-- Element injected in the BODY element -->
{% block body_class % } sidebar-mini {% endblock body_class % }

<!-- Specific Page CSS goes HERE -->
{% block stylesheets % }

    <!-- Google Font: Source Sans Pro -->
    <link rel="stylesheet"
href="https://fonts.googleapis.com/css?family=Source+Sans+Pro:300,400,400i,700&displ
ay=fallback">
    <!-- Font Awesome -->
    <link rel="stylesheet" href="/static/assets/plugins/fontawesome-free/css/all.min.css">
    <!-- Ionicons -->
    <link rel="stylesheet"
href="https://code.ionicframework.com/ionicons/2.0.1/css/ionicons.min.css">
    <!-- Tempusdominus Bootstrap 4 -->
    <link rel="stylesheet" href="/static/assets/plugins/tempusdominus-bootstrap-
4/css/tempusdominus-bootstrap-4.min.css">
    <!-- iCheck -->
    <link rel="stylesheet" href="/static/assets/plugins/ichexk-bootstrap/ichexk-
bootstrap.min.css">
    <!-- JQVMap -->
    <link rel="stylesheet" href="/static/assets/plugins/jqvmap/jqvmap.min.css">
    <!-- Theme style -->
    <link rel="stylesheet" href="/static/assets/css/adminlte.min.css">
    <!-- overlayScrollbars -->
    <link rel="stylesheet"
href="/static/assets/plugins/overlayScrollbars/css/OverlayScrollbars.min.css">
    <!-- Daterange picker -->
    <link rel="stylesheet" href="/static/assets/plugins/daterangepicker/daterangepicker.css">
    <!-- summernote -->
    <link rel="stylesheet" href="/static/assets/plugins/summernote/summernote-
bs4.min.css">

{% endblock stylesheets % }

{% block content % }

<div class="content-wrapper">

    <!-- Content Header (Page header) -->
    <div class="content-header">
        <div class="container-fluid">
            <div class="row mb-2">
```

```

<div class="col-sm-6">
  <h1 class="m-0 text-dark">YOLOv8 Dashboard</h1>
</div><!-- /.col -->
<div class="col-sm-6">
  <ol class="breadcrumb float-sm-right">
    <li class="breadcrumb-item"><a href="index.html">Home</a></li>
    <li class="breadcrumb-item active">Deteksi Foto/Video</li>
  </ol>
</div><!-- /.col -->
</div><!-- /.row -->
</div><!-- /.container-fluid -->
</div>
<!-- /.content-header -->

<!-- Main content -->
<section class="content">
  <div class="container-fluid">
    <!-- Small boxes (Stat box) -->
    <div class="row">

      <!-- ./col -->
    </div>

  </div>
</div>
<!-- /.row -->
<!-- Main row -->
<div class="row">
  <!-- Left col -->
  <section class="col-lg-6 connectedSortable">
    <!-- Custom tabs (Charts with tabs)-->
    <div class="card">
      <div class="card-header">
        <h3 class="card-title">
          <i class="fas fa-chart-pie mr-1"></i>
          YOLOv8 Deteksi Jumlah Buah Kelapa Sawit
        </h3>

      </div><!-- /.card-header -->
      <div class="card-body">
        <div class="tab-content p-0">
          <!-- Morris chart - Sales -->
          <div class="chart tab-pane active" id="revenue-chart"
            style="position: relative; height: 700px;">

        </div>
      </div>
    </div>
  </div>
</div>
<!-- AI code-->

  <form class="form-signin col-lg-3" method=post enctype=multipart/form-data
name="form1">

  <h1 class="h3 mb-3 font-weight-normal">Upload Foto/Video</h1>

```

```

<input type="file" name="file" class="form-control-file" id="inputfile" >
<br/>
<button class="btn btn-block btn-default btn-sm " type="submit">Deteksi</button>
<p class="mt-5 mb-3 text-muted">Dibuat menggunakan Pytorch & Flask</p>
</form>

```

```

</div><!-- /.card-header -->
<div class="card-body">
  <div class="tab-content p-0">
    <!-- Morris chart - Sales -->
    <div class="chart tab-pane active" id="revenue-chart"
      style="position: relative; height: 700px;">

```

```

<!-- AI code-->

```

```

<!-- detected image display -->


```

```

<script>
document.getElementById("my-image").onload = function() {
  this.style.display = "block";
};
</script>

```

```

<!-- detected video display using opencv-->


```

```

<script>
document.getElementById("my-video-image").onload = function() {
  this.style.display = "block";
};
</script>

```

```

<!-- AI code ends here-->

```

```

</div>

```

```

</div>
</div><!-- /.card-body -->
</div>

```

```

        <!-- /.card -->

</section>

<!-- /.Left col -->
<!-- right col (We are only adding the ID to make the widgets sortable)-->

    <!-- right col -->
</div>
<!-- /.row (main row) -->
</div><!-- /.container-fluid -->
</section>
<!-- /.content -->

</div>

{% endblock content %}

<!-- Specific Page JS goes HERE -->
{% block javascripts %}

<!-- jQuery -->
<script src="/static/assets/plugins/jquery/jquery.min.js"></script>
<!-- jQuery UI 1.11.4 -->
<script src="/static/assets/plugins/jquery-ui/jquery-ui.min.js"></script>
<!-- Resolve conflict in jQuery UI tooltip with Bootstrap tooltip -->
<script>
    $.widget.bridge('uibutton', $.ui.button)
</script>
<!-- Bootstrap 4 -->
<script src="/static/assets/plugins/bootstrap/js/bootstrap.bundle.min.js"></script>
<!-- ChartJS -->
<script src="/static/assets/plugins/chart.js/Chart.min.js"></script>
<!-- Sparkline -->
<script src="/static/assets/plugins/sparklines/sparkline.js"></script>
<!-- JQVMap -->
<script src="/static/assets/plugins/jqvmap/jquery.vmap.min.js"></script>
<script src="/static/assets/plugins/jqvmap/maps/jquery.vmap.usa.js"></script>
<!-- jQuery Knob Chart -->
<script src="/static/assets/plugins/jquery-knob/jquery.knob.min.js"></script>
<!-- daterangepicker -->
<script src="/static/assets/plugins/moment/moment.min.js"></script>
<script src="/static/assets/plugins/daterangepicker/daterangepicker.js"></script>
<!-- Tempusdominus Bootstrap 4 -->
<script src="/static/assets/plugins/tempusdominus-bootstrap-4/js/tempusdominus-
bootstrap-4.min.js"></script>

```

```
<!-- Summernote -->
<script src="/static/assets/plugins/summernote/summernote-bs4.min.js"></script>
<!-- overlayScrollbars -->
<script
src="/static/assets/plugins/overlayScrollbars/js/jquery.overlayScrollbars.min.js"></script>
<!-- AdminLTE App -->
<script src="/static/assets/js/adminlte.js"></script>
<!-- AdminLTE dashboard demo (This is only for demo purposes) -->
<script src="/static/assets/js/pages/dashboard.js"></script>
<!-- AdminLTE for demo purposes -->
<script src="/static/assets/js/demo.js"></script>

{% endblock javascripts %}
```

1.4 syntax berikut digunakan untuk mengkonfigurasi halaman sidebar halaman web seperti logo, profil, dashboard dan lain-lain

```
<!-- Main Sidebar Container -->
<aside class="main-sidebar sidebar-dark-primary elevation-4">
  <!-- Brand Logo -->
  <a href="/" class="brand-link">
    
    <span class="brand-text font-weight-light">Deteksi jumlah TBS</span>
  </a>

  <!-- Sidebar -->
  <div class="sidebar">
    <!-- Sidebar user panel (optional) -->
    <div class="user-panel mt-3 pb-3 mb-3 d-flex">
      <div class="image">
        
      </div>

    </div>

    <!-- SidebarSearch Form -->
    <div class="form-inline">
      <div class="input-group" data-widget="sidebar-search">
        <input class="form-control form-control-sidebar" type="search"
        placeholder="Search" aria-label="Search">
        <div class="input-group-append">
          <button class="btn btn-sidebar">
            <i class="fas fa-search fa-fw"></i>
          </button>
        </div>
      </div>
    </div>

    <!-- Sidebar Menu -->
    <nav class="mt-2">
      <ul class="nav nav-pills nav-sidebar flex-column" data-widget="treeview"
      role="menu" data-accordion="false">
        <!-- Add icons to the links using the .nav-icon class
        with font-awesome or any other icon font library -->
        <li class="nav-item has-treeview {% if 'index' in segment % } menu-open {% endif
        % }">
          <a href="#" class="nav-link {% if 'index' in segment % } active {% endif % }">
            <i class="nav-icon fas fa-tachometer-alt"></i>
            <p>
              Dashboard
              <i class="right fas fa-angle-left"></i>
            </p>
          </a>
        </li>
      </ul>
    </nav>
  </div>
</aside>
```



```

</a>
<ul class="nav nav-treeview">
  <li class="nav-item">
    <a href="index.html" class="nav-link {% if 'index.html' in segment % } active
{% endif % }">
      <i class="far fa-circle nav-icon"></i>
      <p>Hasil Deteksi Gambar</p>
    </a>
  </li>

</ul>
</nav>
<!-- /.sidebar-menu -->
</div>
<!-- /.sidebar -->
</aside>

```

Hasil deteksi



Gambar hasil deteksi jumlah TBS menggunakan web