

**RANCANG BANGUN MONITORING HIDROPONIK DENGAN SISTEM
IMAGE PROCESSING BERBASIS IoT**

PROYEK AKHIR

Laporan akhir ini dibuat dan diajukan untuk memenuhi salah satu syarat kelulusan
Sarjana Terapan Politeknik Manufaktur Negeri Bangka Belitung



Disusun Oleh:

Roman Sabda Agung	<i>NIM</i> :	1052255
Sabihisma Rabika Wahada	<i>NIM</i> :	1052256

**POLITEKNIK MANUFAKTUR NEGERI
BANGKA BELITUNG**

TAHUN 2025

LEMBAR PENGESAHAN

RANCANG BANGUN MONITORING HIDROPONIK DENGAN SISTEM IMAGE PROCESSING BERBASIS IOT

Oleh:

Roman Sabda Agung NIM 1052255

Sabihisma Rabika Wahada NIM 1052256

Laporan akhir ini telah disetujui dan disahkan sebagai salah satu syarat kelulusan
Program Sarjana Terapan Politeknik Manufaktur Negeri Bangka Belitung

Menyetujui,

Pembimbing 1



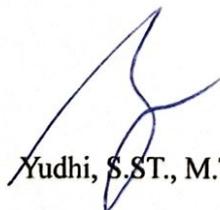
Aan Febriansyah, S.ST., M.T.

Pembimbing 2



Badriyah, S.P., M.Si

Penguji 1



Yudhi, S.ST., M.T.

Penguji 2



Nur Khasanah, S.P., M.Si.

PERNYATAAN BUKAN PLAGIAT

Yang bertanda tangan dibawah ini :

Nama Mahasiswa 1 : Roman Sabda Agung NIM : 1052255

Nama Mahasiswa 2 : Sabihisma Rabika Wahada NIM : 1052256

Dengan Judul : Rancang Bangun Monitoring Hidroponik Dengan Sistem Image
Processing Berbasis *IoT*

Menyatakan bahwa laporan akhir ini adalah hasil kerja kami sendiri dan bukan merupakan plagiat. Pernyataan ini kami buat dengan sebenarnya dan bila ternyata dikemudian hari melanggar pernyataan ini, kami bersedia menerima sanksi yang berlaku.

Sungailiat, 15 Juli 2025

Nama Mahasiswa

Tanda Tangan

1. Roman Sabda Agung



.....

2. Sabihisma Rabika Wahada



.....

ABSTRAK

Penelitian ini merancang dan membangun sistem monitoring hidroponik berbasis Internet of Things (IoT) yang terintegrasi dengan teknologi image processing untuk meningkatkan efisiensi pemantauan pertumbuhan tanaman dan memudahkan petani dalam memantau kondisi tanaman dari jarak jauh, termasuk lebar tanaman dan jumlah daun tanaman hidroponik. Sistem menggunakan kamera yang dipasang secara strategis untuk menangkap gambar tanaman secara periodik, kemudian diproses menggunakan Raspberry Pi 4 Model B dengan algoritma YOLOv8 untuk mendeteksi daun dan keseluruhan tanaman pakcoy. Dataset dikumpulkan selama satu bulan melalui pengambilan gambar, anotasi, augmentasi, dan pelatihan model menggunakan Google Colab. Hasil analisis visual seperti tinggi tanaman dan jumlah daun ditampilkan secara real-time melalui aplikasi smartphone berbasis Firebase. Sistem ini memudahkan petani dalam memantau kondisi tanaman dari jarak jauh dan memungkinkan deteksi dini terhadap gangguan pertumbuhan, kekurangan nutrisi, atau serangan hama. Hasil dari penerapan sistem ini, diharapkan produktivitas dan kualitas hasil panen hidroponik dapat meningkat secara signifikan.

Kata Kunci: hidroponik, YOLOv8, image processing, Raspberry Pi, IoT

ABSTRACT

This research designs and builds an Internet of Things (IoT)-based hydroponic monitoring system integrated with image processing technology to enhance the efficiency of plant growth monitoring and facilitate farmers in remotely observing the condition of their plants, including plant width and leaf count. The system employs strategically placed cameras to periodically capture images of the plants, which are then processed using a Raspberry Pi 4 Model B with the YOLOv8 algorithm to detect the leaves and stems of pakcoy plants. A dataset was collected over one month through image capture, annotation, augmentation, and model training using Google Colab. Visual analysis results, such as plant height and leaf count, are displayed in real-time through a Firebase-based smartphone application. This system enables farmers to monitor plant conditions from a distance and allows for early detection of growth disturbances, nutrient deficiencies, or pest attacks. With the implementation of this system, it is expected that the productivity and quality of hydroponic harvests can significantly improve.

Keywords: hydroponics, YOLOv8, image processing, Raspberry Pi, IoT

KATA PENGANTAR

Puji syukur kita panjatkan atas kehadiran Allah SWT yang telah memberikan rahmat dan hidayah-Nya sehingga penulis dapat menyelesaikan Laporan Proyek Akhir yang berjudul “**Rancang Bangun Monitoring Hidroponik dengan Sistem *Image Processing* Berbasis Iot**” tepat pada waktunya. Shalawat serta salam kepada Rasulullah Muhammad S.A.W yang telah membawa manusia ke jalan yang damai, terang benderang dan penuh ilmu pengetahuan.

Penyusunan laporan akhir ini selain merupakan salah satu persyaratan kelulusan pendidikan Diploma IV di Politeknik Manufaktur Negeri Bangka Belitung juga dimaksudkan untuk menambah ilmu pengetahuan pada bidang *image processing*.

Pada kesempatan ini, penulis ingin mengucapkan rasa terima kasih kepada berbagai pihak yang telah memberikan bantuan dan dukungan baik materil maupun nonmateril yang diberikan kepada penulis dalam menyelesaikan laporan akhir ini. Oleh karena itu, izinkan penulis mengucapkan rasa terima kasih kepada:

1. Allah SWT. Yang telah memberikan rahmat dan hidayah-Nya hingga penulis dapat menyelesaikan laporan proyek akhir ini.
2. Kedua Orang Tua dan keluarga yang selalu mendukung serta mendoakan penulis sehingga dapat menyelesaikan laporan akhir ini.
3. Bapak I Made Andik Setiawan, M.Eng., Ph.D., selaku Direktur di Politeknik Manufaktur Negeri Bangka Belitung.
4. Bapak Aan Febriansyah, S.ST., M.T. selaku Koordinator program Studi Teknik Elektronika dan dosen pembimbing 1 yang telah banyak meluangkan waktu, tenaga dan pikiran, serta memberikan saran dalam pengerjaan Proyek Akhir maupun makalah.
5. Ibu Badriyah, S.P., M.Si. selaku dosen pembimbing 2 yang telah memberikan saran dan solusi pada pengerjaan Proyek Akhir maupun makalah.

6. Seluruh Staff Komisi Proyek Akhir yang sudah membantu kegiatan Proyek Akhir.

7. Kepada rekan-rekan mahasiswa yang telah membantu dalam pengerjaan laporan akhir ini.

8. Semua pihak yang telah memberikan bantuan yang tidak dapat penulis sebutkan satu persatu sehingga mengantarkan penulis untuk menyelesaikan laporan akhir ini.

Dalam penyusunan Laporan ini tentunya masih banyak terdapat kekurangan, kesalahan dan kekhilafan karena keterbatasan kemampuan penulis, untuk itu sebelumnya penulis mohon maaf yang sebesar-besarnya. Penulis juga mengharapkan kritik dan saran dari semua pihak demi perbaikan yang bersifat membangun atas laporan ini.

Akhir kata dengan segala kerendahan hati penulis mengucapkan rasa terima kasih dan semoga laporan ini dapat bermanfaat bagi penulis maupun kita bersama.

Sungailiat, 15 Juli 2025

Hormat kami,

Penulis

DAFTAR ISI

LEMBAR PENGESAHAN	ii
PERNYATAAN BUKAN PLAGIAT	iii
ABSTRAK	iv
ABSTRACT	v
KATA PENGANTAR.....	vi
DAFTAR ISI	viii
DAFTAR GAMBAR	x
DAFTAR TABEL.....	xii
DAFTAR LAMPIRAN	xiii
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	3
1.3 Tujuan Penelitian.....	3
BAB II DASAR TEORI.....	4
2.1 Hidroponik.....	4
2.2 Image Processing.....	5
2.3 Artificial Intellegent (AI).....	6
2.3.1 Machine Learning.....	7
2.3.2 Deep Learning.....	9
2.4 Convolutional Neural Network (CNN)	11
2.5 YOLO (You Only Look Once)	12
2.5.1 YOLO (You Only Look Once) v8	14
2.6 Visual Studio Code	14
2.7 Python.....	15
2.8 OpenCV	16
2.9 Rapsberry Pi	18
2.9.1 <i>Raspberry Pi</i> 4 Model B	19
2.10 Roboflow	20
2.11 Google Colab.....	21
2.12 Internet of Things (IoT).....	21

BAB III METODE PELAKSANAAN	23
3.1 Perancangan sistem	24
3.2 Pengujian Komponen	24
3.3 Pembuatan Alat.....	25
3.4 Pengujian Sistem	25
3.4.1 Akses <i>Raspberry Pi</i> Secara Headless.....	25
3.4.2 VNC (Virtual Network Computing)	26
3.4.3 File sudah masuk ke <i>Raspberry pi</i>	27
3.4.4 Pengujian di <i>Raspberry Pi</i>	27
3.5 Pembuatan Laporan Akhir.....	28
BAB IV PEMBAHASAN.....	29
4.1 Sistem Kerja Alat.....	29
4.2 Penerapan Sistem	30
4.2.1 Pengambilan Dataset.....	30
4.2.2 Images/Data <i>Pre-Processing</i>	33
4.2.3 Pembuatan Label Dataset.....	34
4.2.4 Image Augmentation.....	35
4.2.5 Training Dataset.....	37
4.2.5.1 Hasil Training Dataset	39
4.3 Pengujian Sistem	45
4.3.1 Hasil Pengujian Sistem	47
BAB V KESIMPULAN DAN SARAN.....	74
5.1 Kesimpulan.....	74
5.2 Saran	74
DAFTAR PUSTAKA	76
Lampiran 1 Daftar Riwayat Hidup.....	80
Lampiran 2 Program Keseluruhan	83

DAFTAR GAMBAR

Gambar 2. 1 Kebun Hidroponik.....	4
Gambar 2. 2 Skema AI.....	6
Gambar 2. 3 Ilustrasi visual terkait perbedaan deep learning dan tradisional machine learning dalam fitur ekstraksi dan pembelajaran	10
Gambar 2. 4 Proses Dasar Deteksi Gambar CNN Dalam Satu Tahap	11
Gambar 2. 5 Penggunaan YOLO	13
Gambar 2. 6 Logo Visual studio Code	15
Gambar 2. 7 Logo Python	16
Gambar 2. 8 Logo openCV	17
Gambar 2. 9 Rapsberry Pi 4 Model B	18
Gambar 2. 10 Logo Roboflow	20
Gambar 2. 11 Logo Google Colab	21
Gambar 3. 1 Flowchart Alur Penelitian.....	23
Gambar 3. 2 Cara Access Raspberry Secara Headless.....	25
Gambar 3. 3 Tampilan Vnc Membuat Laptop Jadi Monitor Raspberry.....	26
Gambar 3. 4 file proyek yang diinput ke raspberry pi	27
Gambar 3. 5 Pengujian di Raspberry	27
Gambar 4. 1 Flowchart.....	29
Gambar 4. 2 Contoh Gambar Dataset dari Sudut Atas.....	31
Gambar 4. 3 Contoh Gambar Dataset dari Sudut Samping	31
Gambar 4. 4 Pre-Processing dataset.....	33
Gambar 4. 5 Hasil Anotasi Tanaman Pakcoy	34
Gambar 4. 6 Hasil Anotasi Tanaman Kailan	35
Gambar 4. 7 Proses Augmentasi Dataset	36
Gambar 4. 8 Proses Training Dataset di Google Colab	38
Gambar 4. 9 Model Segmentasi YOLOv8	39
Gambar 4. 10 Gambar Hasil Training Dataset 100 epochs.....	39
Gambar 4. 11 Confusion Matrix	40
Gambar 4. 12 Grafik Loss Training dan Validation	42

Gambar 4. 13 Grafik Metrics Precision dan Recall	42
Gambar 4. 14 Program Prediksi Model Training Dataset	44
Gambar 4. 15 Hasil Prediksi Hidroponik Kailan	44
Gambar 4. 16 Hasil Prediksi Pakcoy.....	45
Gambar 4. 17 Integrasi dengan Firebase.....	46
Gambar 4. 18 Hasil Pengujian Model.....	46



DAFTAR TABEL

Tabel 4. 1 Pembagian Dataset	45
Tabel 4. 2 Hasil Pengujian Tanaman Pakcoy Hari Ke-1	56
Tabel 4. 3 Pengambilan Data 1 Tanaman Pakcoy Secara Manual	58
Tabel 4. 4 Hasil Pengujian Pakcoy Hari Ke-2.....	50
Tabel 4. 5 Pengambilan Data 2 Tanaman Pakcoy Secara Manual	60
Tabel 4. 6 Hasil Pengujian Pakcoy Hari Ke-3.....	61
Tabel 4. 7 Pengambilan Data 3 Tanaman Pakcoy Secara Manual	63
Tabel 4. 8 Hasil Pengujian Pakcoy Hari Ke-4.....	63
Tabel 4. 9 Pengambilan Data 4 Tanaman Pakcoy Secara Manual	56
Tabel 4. 10 Hasil Pengujian Pakcoy Hari Ke-5.....	66
Tabel 4. 11 Pengambilan Data 5 Tanaman Pakcoy Secara Manual	67
Tabel 4. 12 Hasil Pengujian Kailan Hari Ke-1	59
Tabel 4. 13 Pengambilan Data 1 Tanaman Kailan Secara Manual	70
Tabel 4. 14 Hasil Pengujian Kailan Hari Ke-2.....	71
Tabel 4. 15 Pengambilan Data 2 Tanaman Kailan Secara Manual	73
Tabel 4. 16 Hasil Pengujian Kailan Hari Ke-3.....	74
Tabel 4. 17 Pengambilan Data 3 Tanaman Kailan Secara Manual	75
Tabel 4. 18 Hasil Pengujian Kailan Hari Ke-4.....	76
Tabel 4. 19 Pengambilan Data 4 Tanaman Kailan Secara Manual	78
Tabel 4. 20 Hasil Pengujian Kailan Hari Ke-5.....	70
Tabel 4. 21 Pengambilan Data 5 Tanaman Kailan Secara Manual	70

DAFTAR LAMPIRAN

Lampiran 1 Daftar Riwayat Hidup.....	89
Lampiran 2 Program Keseluruhan	92



BAB I

PENDAHULUAN

1.1 Latar Belakang

Hidroponik adalah metode bercocok tanam yang semakin diminati di Indonesia dalam beberapa tahun terakhir. Teknik ini memungkinkan tanaman tumbuh tanpa menggunakan tanah, dengan cara menyalurkan nutrisi langsung ke akar melalui larutan air [1]. Metode hidroponik semakin banyak dikenal masyarakat karena kurangnya lahan untuk menanam secara tradisional serta dapat meningkatkan hasil produksi tanaman dengan kualitas yang lebih terjaga karena bebas dari resiko serangan hama dan penyakit yang berasal dari tanah, tanaman tumbuh lebih sehat, bibit berkembang dengan baik dan produktivitasnya tinggi. Pengelolaan dan budidaya tanaman hidroponik lebih praktis karena lingkungannya lebih bersih serta tanaman terlindung dari hujan. Hasil panen dari hidroponik ini juga memiliki kualitas yang unggul, daya tahan lebih lama serta harga jual yang tinggi [2]. Petani hidroponik sering mengintegrasikan sistem ini dengan rumah kaca (*greenhouse*) untuk menciptakan lingkungan pertumbuhan yang lebih sehat dan terkendali. Penggunaan rumah kaca membantu mengurangi ketergantungan pada bahan kimia berbahaya yang dapat merusak tanaman sekaligus menekan penyebaran hama [3].

Sebagian orang tertarik dengan sistem hidroponik adalah fleksibilitasnya, karena dengan berbagai teknik yang dapat dilakukan oleh petani sesuai kebutuhannya. Teknik yang trend di Indonesia yaitu *Deep Flow Technique* (DFT), *Nutrient Film Technique* (NFT), *ebb and flow* serta hidroponik drip. Bersamaan dengan pesatnya perkembangan teknologi, sistem hidroponik semakin maju melalui riset yang berkelanjutan untuk mengoptimalkan pertumbuhan tanaman dan mempercepat hasil panen.

Uniknya, sejumlah petani tidak hanya menjadikan hidroponik sebagai pekerjaan, tapi dijadikan sebagai hobi yang berkelanjutan. Akan tetapi, para petani tersebut masih mengalami kesulitan dalam mendeteksi persoalan yang terjadi pada

tanaman tersebut seperti adanya serangan hama, kelainan pada tumbuhan, kurangnya nutrisi serta penyakit yang berpotensi menghambat pertumbuhan tanaman hidroponik. Oleh karena itu, solusi baru yang kreatif dalam memonitoring tanaman hidroponik menjadi sangat penting untuk mendukung pertanian yang modern.

Beriringan dengan era modern yang terjadi sekarang, *image processing* termasuk bagian yang bisa digunakan untuk menganalisis dan memproses gambar dengan efektif. Sistem monitoring hidroponi sangat berperan penting dalam mendeteksi kondisi tanaman misalnya lebar tanaman dan jumlah daun pada tanaman hidroponik. Menggunakan analisis visual yang tepat, petani hidroponik dapat mengambil langkah pencegahan sebelum permasalahan menjadi lebih serius.

Seiring dengan pesatnya perkembangan teknologi, banyak masyarakat yang tidak memiliki cukup waktu untuk melakukan pengecekan kondisi tanaman hidroponik secara berkala karena kesibukan di luar rumah. Oleh karena itu, integrasi Internet of Things (IoT) dalam sistem hidroponik menjadi solusi inovatif yang memungkinkan pemantauan kondisi tanaman secara *real-time*. Menggunakan teknologi ini, petani lebih mudah dan akurat dalam mengontrol pertumbuhan tanaman tanpa harus selalu berada di lokasi.

Sistem yang dikembangkan dalam penelitian ini memanfaatkan kamera yang dipasang pada tiang menghadap ke bawah, yang kemudian dihubungkan dengan *Raspberry Pi* sebagai minicomputer untuk menjalankan program *image processing*. *Raspberry Pi* dipilih karena memiliki spesifikasi yang mumpuni untuk memproses gambar dengan cepat dan efisien [4]. Hasil monitoring akan ditampilkan pada *website* firebase, yang mempermudah petani mengakses informasi terhadap kondisi tanaman secara *real-time*. Tujuan dari penelitian ini adalah untuk memberikan kemudahan kepada petani hidroponik dalam memantau kondisi tanaman yang memanfaatkan teknologi *Internet of Things*. Harapannya sistem yang dirancang bisa meningkatkan efisiensi, mempercepat mendeteksi masalah dan menghasilkan tanaman hidroponik yang berkualitas dan berkelanjutan.

1.2 Rumusan Masalah

1. Bagaimana merancang dan membangun sistem *image processing* untuk memantau lebar tanaman dan jumlah daun secara otomatis pada tanaman hidroponik?
2. Apa saja komponen dan sensor kamera yang diperlukan untuk mendukung kualitas *monitoring* tanaman hidroponik?
3. Bagaimana integrasi data hasil pengolahan citra ke dalam basis data yang dapat diakses secara *real-time* melalui *website* firebase?

1.3 Tujuan Penelitian

1. Merancang dan membangun sistem *image processing* yang mampu memantau lebar tanaman dan jumlah daun secara otomatis.
2. Mengidentifikasi dan mengimplementasikan komponen serta sensor kamera yang sesuai untuk sistem monitoring tanaman hidroponik.
3. Mengembangkan sistem pemantauan berbasis Internet of Things (IoT) yang memungkinkan akses data hasil monitoring secara *real-time* melalui *website* firebase.

BAB II

DASAR TEORI

2.1 Hidroponik

Hidroponik adalah metode menanam tanaman tanpa menggunakan tanah sebagai media tanam, melainkan air yang menjadi media tanam. Hidroponik sangat cocok untuk mengatasi keterbatasan lahan serta memiliki ketahanan terhadap perubahan iklim, seperti kekeringan, banjir, dan musim tanam yang pendek. Konsep dari hidroponik sendiri yaitu membuat tanaman terapung pada air yang diberi nutrisi sebagai media tanam. Pengembangan teknologi dalam budidaya dengan sistem hidroponik menjadi salah satu alternatif yang dapat diterapkan untuk meningkatkan produktivitas cocok tanam dan sebagai upaya konservasi lahan di Indonesia [5][6].



Gambar 2. 1 Kebun Hidroponik

Terdapat beberapa jenis sistem hidroponik yang umum digunakan, masing-masing memiliki karakteristik dan mekanisme kerja yang berbeda. Sistem sumbu (*Wick System*) merupakan salah satu sistem hidroponik paling sederhana dari semua sistem hidroponik karena tidak memiliki bagian yang bergerak sehingga tidak menggunakan pompa dan listrik [6]. Sistem NFT (*Nutrient Film Technique*) merupakan sistem pemberian nutrisi tanaman dengan mengalirkan larutan nutrisi setinggi sekitar 3 mm pada perakaran tanaman (Hendra & Andoko, 2014). Sistem rakit apung (*Water Culture System*) adalah sistem hidroponik yang

menggunakan styrofoam sebagai penopang untuk membuat tanaman yang ditanam terapung di atas larutan nutrisi [7]. Selain itu, terdapat juga sistem Aeroponik yang menyemprotkan larutan nutrisi langsung ke akar tanaman yang digantung di udara, sangat efisien namun memerlukan peralatan yang canggih dan mahal.

Keuntungan dari konsep tanaman hidroponik adalah dapat meningkatkan kualitas dan hasil pertanian dengan menggunakan sumber daya yang optimal, metode ini tidak berlaku untuk pertanian konvensional yang membutuhkan banyak air dan nutrisi, pertanian hidroponik sering dikombinasikan dengan penggunaan rumah kaca, penggunaan rumah kaca dapat mengurangi penyebaran hama pada tanaman [8].

2.2 Image Processing

Image processing atau disebut pengolahan citra merupakan salah satu proses pengambilan sinyal input dalam bentuk citra dan output yang dihasilkan juga berupa citra dengan memberikan algoritma serta teknik tertentu untuk meningkatkan kualitas citra agar mudah diinterpretasi oleh manusia maupun sebuah mesin untuk melakukan analisa terhadap citra. Penggunaan sistem *image processing* ini dapat dimanfaatkan dalam berbagai bagian, seperti medis, otomotif, keamanan, manufaktur, pertanian dan lain-lain. Digunakan pada bidang pertanian, metode *image processing* bisa melakukan klasifikasi untuk membantu para petani agar memberikan perawatan khusus secara efektif dan efisien pada tanaman yang tidak normal [9].

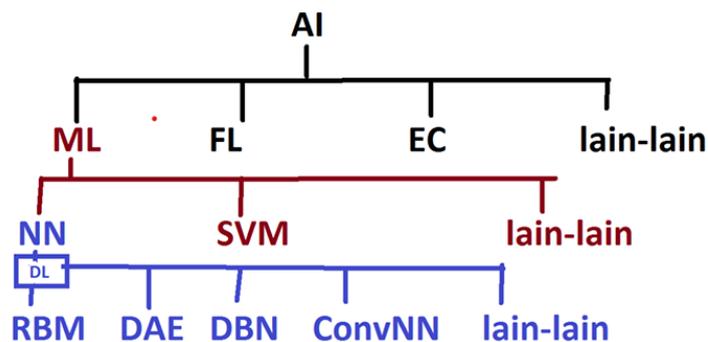
Metode *image processing* memiliki hubungan dengan *deep learning* terutama pada pengelolaan dan analisis gambar yang memiliki beberapa teknik pada analisis gambar seperti edit size, filter, maupun deteksi tepi. *Deep learning* membuat proses ekstraksi yang dilakukan secara manual menjadikan *image processing* bisa diotomatisasi serta ditingkatkan akurasi. Hal ini memungkinkan sistem mempelajari pola-pola yang kompleks dalam citra, seperti pengenalan objek, segmentasi, dan klasifikasi menjadi lebih baik dibandingkan metode tradisional. Sinergi antara *image processing* dan *deep learning* kedepannya membuat manusia

mampu menyediakan alat yang lebih canggih untuk memahami serta menganalisis data visual lebih akurat.

Image processing memiliki teknik analisis yang sering digunakan, seperti Convolutional Neural Network (CNN). CNN didesain untuk mengolah data dua dimensi. Teknik ini termasuk jenis *Deep Neural Network* karena kedalaman jaringan yang tinggi dan banyak diaplikasikan pada data citra [10]. Selain itu, CNN secara khusus dibuat untuk membantu komputer agar dapat melihat beberapa bagian maupun keseluruhan citra sekaligus pada satu waktu.

2.3 Artificial Intellegent (AI)

Kecerdasan Buatan atau *Artificial Intellegent* (AI) adalah teknik yang digunakan untuk meniru kecerdasan yang dimiliki oleh makhluk hidup maupun benda mati untuk menyelesaikan sebuah persoalan [11]. *Artificial Intellegent* salah satu teknologi yang sudah dimanfaatkan dalam kehidupan sehari-hari, seperti berkomunikasi, menerjemahkan bahasa, *mobile banking*, dan bernavigasi. Proses yang terjadi dalam *Artificial Intelligence* mencakup *learning*, *reasoning*, dan *self-correction*. Proses ini mirip dengan manusia yang melakukan analisis sebelum memberikan keputusan [12]. *Artificial Intelligence* (AI) dapat dibagi menjadi beberapa sub-bidang, termasuk pembelajaran mesin (*machine learning*), jaringan saraf tiruan (*neural networks*), logika fuzzy (*fuzzy logic*), dan algoritma genetika (*genetic algorithms*). Dalam konteks keamanan siber, *machine learning* adalah salah satu teknik AI yang paling banyak digunakan [13].



Gambar 2. 2 Skema AI

Artificial Intelligent (AI) saling terkait erat dengan *image processing*, karena peran AI digunakan untuk mengenali pola gambar, seperti wajah, objek, maupun teks. Selain itu, AI dapat mengklasifikasi gambar ke kategori tertentu berdasarkan fitur yang didapatkan dari data penelitian. Salah satu algoritma AI yang dipakai yaitu *Convolutional Neural Network* (CNN). Keunggulan algoritma Convolutional Neural Network (CNN) dalam menjalankan tugas pengenalan gambar karena algoritma CNN memiliki struktur *neuron deep layers* dan proses pelatihan data berulang [14].

Revolusi signifikan telah dibawa oleh *Artificial Intelligent* (AI) dalam bidang *image processing*, baik itu dalam kehidupan sehari-hari maupun skala yang besar. Kemudahan AI yang didapat dalam kehidupan sehari-hari, yaitu pengenalan wajah, foto otomatis, dan aplikasi *augmented reality*. Sementara itu, penggunaan AI dalam skala besar digunakan untuk menganalisis citra medis, pengawasan keamanan, serta otomatisasi industri. Intinya, mengimplementasikan AI pada pengerjaan manusia adalah memperoleh hasil kinerja optimal dengan waktu proses yang cepat dan hasil yang maksimal [12].

2.3.1 Machine Learning

Machine Learning (ML) adalah cabang dari kecerdasan buatan yang mempelajari algoritma yang mampu belajar secara mandiri, langsung dari data input. Dalam dekade terakhir, teknik *machine learning* telah mengalami kemajuan pesat, seperti yang ditunjukkan oleh algoritma *Deep Learning* (DL) yang diimplementasikan dalam mobil otonom atau permainan strategi elektronik. Pada *machine learning* dilengkapi sejumlah aturan program yang dijalankan oleh algoritma, oleh karena itu pada teknik *machine learning* dapat dikategorikan sebagai instruksi yang dijalankan dan dipelajari secara otomatis untuk menghasilkan output yang optimal, hal ini dilakukan secara otomatis tanpa ada campur tangan manusia sedikitpun. Semua dilakukan secara otomatis untuk mengubah data menjadi beberapa pola dan diinputkan jauh ke dalam sistem untuk mendeteksi masalah otomatis [15].

Potensi *machine learning* dalam manajemen operasi membuat peneliti mempertimbangkan aplikasi *machine learning* dalam berbagai di bidang industri, terutama untuk pengolahan pola dan gambar, pemrosesan bahasa alami, optimasi operasi, penambahan data, dan penemuan pengetahuan. Kemampuan belajar representasi secara mandiri model *machine learning* dan terutama pada *deep learning* mampu mengekstrak pengetahuan dari data mentah, yang membebaskan kita dari langkah ekstraksi dan rekayasa fitur yang mahal. Salah satu hal yang terpenting bagi *machine learning* yaitu data, bisa dikatakan bahwa kerangka kerja dari *machine learning* terdapat beberapa fase yaitu, pengumpulan data, pembangunan model, penyetelan model, pengujian, serta penerapan model. Data ini bisa berasal dari berbagai sumber, termasuk jejaring sosial, log, blog, dan berbagai sensor seperti sensor suhu, sensor arus, sensor kelembaban, dan lainnya. Sifat data ini bisa sangat bervariasi. Misalnya, data dari jejaring sosial dapat berupa log, data suara, data gambar, data video (rangkaian gambar), dan data teks. Data dari sensor biasanya berbentuk vektor, dan tipe datanya bisa berupa float, integer, atau string. Data juga dapat disimpan dalam bentuk daftar (list) atau format lainnya; cara penyimpanan data ini ditentukan oleh konvensi yang diikuti oleh repositori penyimpanan data [16].

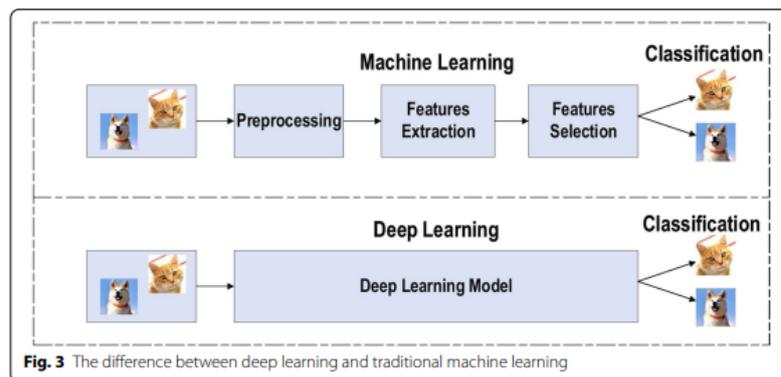
Tantangan pada saat memakai algoritma *machine learning* meliputi ketersediaan data, masalah data yang hilang, dan batasan yang berlaku oleh perangkat keras. Masalah tersebut masih bisa diatasi dengan tindakan pencegahan dan membuat langkah-langkah yang lebih terstruktur sebelum membangun model. *Exploratory Data Analysis* (EDA) adalah alat penting bagi insinyur *machine learning* untuk mendapatkan wawasan dari data mereka. EDA membantu mereka membuat keputusan yang terinformasi tentang data dan membangun model yang secara akurat mencerminkan data tersebut. Data mentah mungkin tidak langsung menunjukkan pola, korelasi, atau anomali, tetapi dengan EDA, ilmuwan data dapat menggunakan berbagai visualisasi untuk mengungkap wawasan tersembunyi dan lebih memahami struktur data. Ini termasuk mempelajari distribusi data, mengidentifikasi outlier dan nilai yang hilang, serta mengeksplorasi hubungan antar fitur [16]. *Exploratory Data Analysis* dapat digunakan untuk mengevaluasi kinerja model *machine learning*, dengan mengvisualisasikan hasilnya dari data dapat

digunakan untuk mengidentifikasi area di mana model tidak mampu untuk melakukan kerjanya dengan baik dan melakukan perbaikan untuk meningkatkan akurasi serta potensi kelemahan dalam data.

2.3.2 Deep Learning

Deep Learning adalah cabang ilmu dari *Machine Learning* yang algoritmanya terinspirasi dari struktur otak manusia. Struktur tersebut dinamakan *Artificial Neural Networks* atau disingkat ANN atau disebut juga Jaringan Saraf Tiruan atau disingkat JTT. *Deep Learning* juga memiliki algoritma tersendiri antara-lain: Convolutional Neural Network (CNN), *Long Short Term Memory Network* (LSTM), (RNN), *Self Organizing Maps* (SOM) (Setiawan, 2021). *Deep learning* telah memberi perubahan signifikan kepada banyak bidang, karena kemampuannya mengekstrak fitur secara otomatis dari data mentah tanpa memerlukan rekayasa pada fitur manual yang rumit. Metode *deep learning* memanfaatkan proses komputasi data yang besar yang dapat berlangsung dengan cepat dengan menggunakan *Central Processing Unit* (CPU), *Random Access Memory* (RAM), dan *Graphics Processing Unit* (GPU) [17].

Dalam teknik *machine learning* tradisional, tugas klasifikasi umumnya melibatkan proses berurutan yang mencakup pra-pemrosesan, ekstraksi fitur, seleksi fitur yang cermat, pembelajaran, dan klasifikasi. Keefektifan metode *machine learning* sangat bergantung pada seleksi fitur yang akurat, karena pemilihan fitur yang bias dapat mengakibatkan klasifikasi kelas yang salah. Sebaliknya, model *deep learning* memungkinkan pembelajaran dan klasifikasi dilakukan secara bersamaan dan menghilangkan kebutuhan akan langkah-langkah yang rumit. Kemampuan ini membuat *deep learning* sangat menguntungkan untuk mengotomatisasi pembelajaran fitur dalam berbagai tugas secara visual. menggambarkan perbedaan antara *deep learning* dan *machine learning* tradisional dalam hal ekstraksi fitur dan pembelajaran. *Deep learning* melibatkan proses pembelajaran representasi hierarkis dari data dengan memanfaatkan arsitektur yang memiliki banyak lapisan tersembunyi. Seiring dengan perkembangan fasilitas komputasi berkinerja tinggi, teknik *deep learning* yang memanfaatkan jaringan saraf dalam *deep neural networks* semakin banyak diminati.

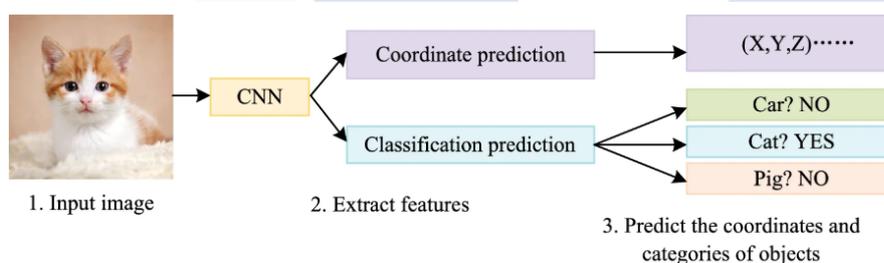


Gambar 2. 3 Ilustrasi visual terkait perbedaan *deep learning* dan tradisional *machine learning* dalam fitur ekstraksi dan pembelajaran

Algoritma *deep learning* memproses data melalui beberapa lapisan, di mana setiap lapisan secara bertahap mengekstrak fitur dan mengirimkan informasi ke lapisan selanjutnya. Lapisan awal mengekstrak karakteristik tingkat rendah, yang kemudian digabungkan oleh lapisan berikutnya untuk membentuk representasi yang salah satu arsitektur *deep learning* yang paling terkenal yaitu *Conventional Neural Network* (CNN), karena sudah dirancang khusus untuk memperoleh data berbentuk grid, seperti gambar. CNN menggunakan operasi konvolusi untuk mengekstrak fitur lokal data, sehingga efektif dalam tugas-tugas seperti klasifikasi gambar dan deteksi objek. Arsitektur yang lain yang bisa digunakan, seperti *Recurrent Neural Network* (RNN) dan *Long Short-Term* (LSTM). Dalam teknik ini, proses pembelajaran didasarkan pada *dataset* yang semi-terlabeli. Terkadang, *generative adversarial networks* (GANs) dan DRL (*Deep Reinforcement Learning*) digunakan dengan cara yang serupa seperti teknik ini. Selain itu, RNNs (*Recurrent Neural Networks*), yang mencakup GRUs (*Gated Recurrent Units*) dan LSTMs (*Long Short-Term Memory*), juga digunakan untuk pembelajaran semi-terawasi. Salah satu keuntungan dari teknik ini adalah meminimalkan jumlah data berlabel yang diperlukan [19].

2.4 Convolutional Neural Network (CNN)

Bidang *computer vision* telah mencapai kemajuan luar biasa di beberapa area, termasuk klasifikasi gambar, segmentasi semantik, deteksi objek, dan rekonstruksi gambar dengan resolusi super. Perkembangan pesat *deep convolutional neural network* (CNN) pada fitur pembelajaran ekspresi yang mandiri, serta ekstraksi fitur dari data input asli dapat dilakukan dengan melatih model CNN yang sesuai dengan aplikasi praktis. Teknologi *deep learning* membuat struktur CNN menjadi kompleks dan beragam, membuat CNN bertahap menggantikan metode *machine learning tradisional*. Convolutional Neural Network (CNN) adalah salah satu arsitektur *deep learning* yang terinspirasi oleh mekanisme kerja otak manusia, khususnya dalam memproses informasi visual. CNN dirancang untuk memproses data yang memiliki struktur *grid-like*, seperti gambar, dengan memanfaatkan operasi konvolusi untuk mengekstrak fitur secara hierarkis. Arsitektur CNN terdiri dari beberapa lapisan, termasuk lapisan konvolusi, *pooling*, dan *fully connected*, yang bekerja bersama untuk mempelajari representasi fitur dari data input. 2024. Perkembangan CNN telah membawa revolusi besar dalam bidang *computer vision*, terutama sejak kemunculan AlexNet pada tahun 2012, yang memenangkan kompetisi *ImageNet Large Scale Visual Recognition Challenge (ILSVRC)*.



Gambar 2. 4 Proses Dasar Deteksi Gambar CNN Dalam Satu Tahap

CNN telah banyak digunakan dalam berbagai aplikasi *computer vision*, seperti klasifikasi gambar, deteksi objek, dan prediksi video. Dalam klasifikasi gambar, CNN mampu membedakan objek berdasarkan fitur yang diekstrak dari gambar, seperti pada *dataset ImageNet*. Selain itu, CNN juga digunakan dalam deteksi objek, di mana model seperti R-CNN, Fast R-CNN, dan *YOLO (You Only Look Once)* telah menunjukkan performa yang luar biasa dalam mendeteksi dan

mengklasifikasikan objek dalam gambar secara *real-time*. CNN juga diterapkan dalam prediksi video, di mana model seperti ConvLSTM dan PredNet digunakan untuk memprediksi frame video masa depan berdasarkan frame sebelumnya. Kemampuan CNN dalam memproses data spasial dan temporal membuatnya sangat cocok untuk aplikasi-aplikasi tersebut[20]. Selain itu, CNN juga digunakan dalam bidang medis untuk analisis citra medis, seperti deteksi tumor dan klasifikasi penyakit.

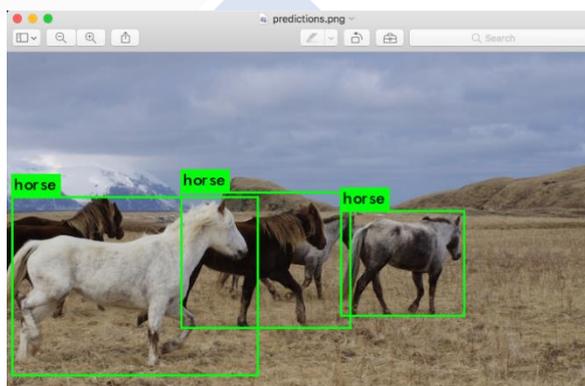
Arsitektur CNN terdiri dari beberapa komponen utama, yaitu lapisan konvolusi, lapisan *pooling*, fungsi aktivasi, dan lapisan *fully connected*. Lapisan konvolusi bertugas mengekstrak fitur dari gambar dengan menggunakan filter (kernel) yang digeser secara *sliding window*. Setiap filter akan menghasilkan *feature map* yang merepresentasikan fitur tertentu dari gambar. Lapisan *pooling* digunakan untuk mengurangi dimensi *feature map* dengan melakukan *down-sampling*, sehingga mengurangi kompleksitas komputasi dan mencegah *overfitting*. Fungsi aktivasi, seperti *ReLU (Rectified Linear Unit)*, digunakan untuk memperkenalkan *non-linearitas* ke dalam model, sehingga CNN dapat mempelajari pola yang lebih kompleks. Lapisan *fully connected* berperan dalam mengintegrasikan fitur yang telah diekstrak untuk melakukan klasifikasi atau regresi. Selain itu, teknik seperti *batch normalization* dan dropout digunakan untuk meningkatkan stabilitas dan generalisasi model[20]. Dengan kombinasi komponen-komponen ini, CNN mampu mempelajari representasi fitur yang semakin abstrak dan hierarkis, sehingga sangat efektif dalam menyelesaikan berbagai tugas *computer vision*.

2.5 YOLO (You Only Look Once)

YOLO (You Only Look Once) adalah salah satu algoritma deteksi objek berbasis *deep learning* yang terkenal karena kemampuannya dalam mendeteksi objek secara *real-time* dengan akurasi yang tinggi. Berbeda dengan metode deteksi objek tradisional seperti HARR, SIFT, dan HOG yang memerlukan proses ekstraksi fitur manual, *YOLO* menggunakan pendekatan end-to-end untuk mendeteksi objek dalam satu tahap. *YOLO* membagi gambar menjadi grid dan memprediksi *bounding*

box serta probabilitas kelas untuk setiap grid tersebut. Keunggulan utama *YOLO* terletak pada kecepatan dan akurasi, sehingga banyak digunakan dalam aplikasi seperti pengawasan keamanan, kendaraan otonom, dan analisis video [21]. Sejak diperkenalkan pertama kali pada tahun 2016, *YOLO* telah mengalami beberapa peningkatan versi, seperti *YOLOv2*, *YOLOv3*, *YOLOv4*, *YOLOv5*, dan versi seterusnya yang semakin meningkatkan performa dan efisiensinya.

YOLO dapat digunakan dalam berbagai aplikasi yang memerlukan deteksi objek *real-time*, seperti sistem pengawasan CCTV, kendaraan otonom, dan analisis lalu lintas. Untuk menggunakan *YOLO*, pengguna perlu menyiapkan model yang telah dilatih sebelumnya atau melatih model baru menggunakan *dataset* yang sesuai. *YOLO* mendukung berbagai sumber input, seperti kamera jaringan (RTSP), *file* video, dan kamera *USB*. Pengguna dapat mengatur parameter seperti ukuran input gambar dan frame rate sesuai dengan kebutuhan aplikasi. Selain itu, *YOLO* juga dapat dijalankan pada berbagai *platform hardware*, mulai dari perangkat *embedded* seperti *Jetson Nano* hingga *GPU high-end* seperti *GTX 1060*. Namun, performa *YOLO* sangat bergantung pada spesifikasi hardware yang digunakan, terutama dalam hal kecepatan pemrosesan dan akurasi deteksi [21].



Gambar 2. 5 Penggunaan *YOLO*

YOLO bekerja dengan membagi gambar input menjadi grid dan memprediksi bounding box serta probabilitas kelas untuk setiap grid tersebut. Setiap *bounding box* diprediksi dengan koordinat (*x*, *y*, *width*, *height*) dan skor kepercayaan yang menunjukkan seberapa yakin model bahwa objek tersebut ada dalam *bounding box*. *YOLO* menggunakan arsitektur convolutional neural network (CNN) untuk mengekstrak fitur dari gambar dan memprediksi *bounding box* serta

kelas objek. Keunggulan *YOLO* dibandingkan metode deteksi objek lainnya adalah kemampuannya untuk memproses seluruh gambar dalam satu tahap, sehingga lebih cepat dan efisien. Namun, *YOLO* memiliki keterbatasan dalam mendeteksi objek kecil atau objek yang tumpang tindih. Untuk mengatasi masalah ini, beberapa versi *YOLO* seperti *YOLOv4* dan *YOLOv5* telah diperkenalkan dengan peningkatan pada akurasi dan kecepatan deteksi [21]. Selain itu, penelitian terbaru juga mengusulkan teknik seperti *Adaptive Frame Control* (AFC) untuk meningkatkan kemampuan *real-time YOLO* pada perangkat dengan sumber daya terbatas.

2.5.1 YOLO (You Only Look Once) v8

YOLOv8 yang merupakan versi kedelapan dari model "*You Only Look Once*" berfungsi sebagai kerangka kerja utama dalam penelitian ini yang ditujukan untuk deteksi objek dalam gambar. Model ini dibangun di atas arsitektur Convolutional Neural Network (CNN), yang dirancang khusus untuk menganalisis citra dengan struktur grid. CNN adalah tipe jaringan saraf tiruan yang sangat efisien dalam memproses dan memahami data visual. Selain itu, penelitian ini juga memanfaatkan *Pytesseract* sebagai metode untuk mengonversi citra menjadi teks, sehingga memungkinkan ekstraksi informasi dari gambar yang dianalisis.

2.6 Visual Studio Code

Microsoft Visual Studio Code (VS Code) adalah sebuah editor kode sumber yang dikembangkan oleh Microsoft. VS Code dirancang untuk menyediakan lingkungan pengembangan yang ringan namun powerful, yang dapat digunakan untuk berbagai bahasa pemrograman. Editor ini mendukung fitur-fitur seperti penyorotan sintaksis, penyelesaian kode otomatis, integrasi dengan Git, dan debugging. VS Code juga memiliki ekosistem ekstensi yang luas, memungkinkan pengguna untuk menyesuaikan dan memperluas fungsionalitasnya sesuai dengan kebutuhan pengembangan mereka. Dengan antarmuka yang intuitif dan dukungan lintas platform (Windows, macOS, dan Linux), VS Code menjadi pilihan populer di kalangan pengembang perangkat lunak [22].



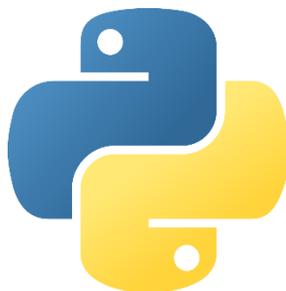
Gambar 2. 6 Logo Visual studio Code

Microsoft Visual Studio Code (VS Code) adalah editor kode sumber yang bersifat *open-source* dan dapat diunduh secara gratis. VS Code dirancang untuk mendukung berbagai bahasa pemrograman seperti JavaScript, Python, C++, dan banyak lagi melalui penggunaan ekstensi. Salah satu fitur unggulan VS Code adalah *IntelliSense*, yang menyediakan penyelesaian kode cerdas berdasarkan variabel, fungsi, dan modul yang tersedia. Selain itu, VS Code juga dilengkapi dengan fitur debugging yang memungkinkan pengembang untuk mengeksekusi kode langkah demi langkah, memeriksa variabel, dan mengevaluasi ekspresi secara *real-time*. Editor ini juga mendukung integrasi dengan sistem kontrol versi seperti Git, memungkinkan pengembang untuk melakukan commit, pull, dan push langsung dari dalam editor. Dengan dukungan komunitas yang besar dan pembaruan rutin dari Microsoft, VS Code terus berkembang menjadi salah satu alat pengembangan perangkat lunak yang paling banyak digunakan di dunia [22].

2.7 Python

Python adalah bahasa pemrograman tingkat tinggi yang bersifat interpretatif, mudah dipelajari, dan memiliki sintaksis yang sederhana dan intuitif. Dikembangkan oleh Guido van Rossum pada awal tahun 1990-an, Python dirancang untuk meningkatkan keterbacaan kode dan memungkinkan pengembang untuk mengekspresikan konsep dalam lebih sedikit baris kode dibandingkan dengan bahasa pemrograman lain seperti C++ atau Java. Python bersifat *open-source*, yang berarti siapa pun dapat menggunakan, memodifikasi, dan mendistribusikan ulang kode sumbernya tanpa biaya. Bahasa ini mendukung berbagai paradigma pemrograman, termasuk pemrograman berorientasi objek, fungsional, dan prosedural, sehingga cocok untuk berbagai jenis proyek pengembangan perangkat lunak. Python juga memiliki komunitas yang besar dan aktif, yang berkontribusi

pada pengembangan pustaka dan kerangka kerja yang luas, menjadikannya salah satu bahasa pemrograman paling populer di dunia [23].



Gambar 2. 7 Logo Python

Python memiliki ekosistem yang kaya dengan berbagai pustaka dan kerangka kerja yang mendukung berbagai aplikasi, mulai dari pengembangan web, analisis data, hingga kecerdasan buatan. Beberapa pustaka populer seperti NumPy, Pandas, dan Matplotlib memungkinkan pengguna untuk melakukan komputasi numerik, manipulasi data, dan visualisasi data dengan efisien. Selain itu, Python juga mendukung integrasi dengan platform big data seperti *Apache Spark* dan *Hadoop* melalui pustaka seperti PySpark, yang memungkinkan pengolahan data skala besar. Python juga dikenal karena kemampuannya dalam pengembangan model *machine learning* dan *deep learning*, dengan dukungan dari pustaka seperti *Scikit-learn*, *TensorFlow*, dan *PyTorch*. Meskipun Python memiliki beberapa keterbatasan dalam hal kinerja dan skalabilitas untuk *dataset* yang sangat besar, integrasinya dengan alat-alat seperti Dask dan Ray membantu mengatasi tantangan tersebut. Dengan demikian, Python menjadi pilihan utama bagi para ilmuwan data, analis, dan pengembang yang membutuhkan fleksibilitas dan kemudahan dalam mengelola dan menganalisis data [23].

2.8 OpenCV

OpenCV (*OpenSource Computer Vision Library*) adalah sebuah library *open-source* yang dirancang untuk memfasilitasi pengembangan aplikasi dalam bidang *computer vision* dan *image processing*. Pertama kali dikembangkan oleh Intel pada tahun 1999, OpenCV kini dikelola oleh *OpenCV Foundation* dan didukung oleh komunitas global yang besar. Library ini menyediakan lebih dari

2500 algoritma yang dioptimalkan untuk berbagai tugas pengolahan gambar, termasuk deteksi wajah, pelacakan objek, segmentasi gambar, dan banyak lagi. OpenCV mendukung berbagai bahasa pemrograman, termasuk C++, Python, Java, dan MATLAB, sehingga memudahkan pengembang untuk mengintegrasikannya ke dalam berbagai platform dan aplikasi. Dengan dukungan untuk sistem operasi seperti Windows, Linux, macOS, Android, dan iOS, OpenCV menjadi alat yang sangat fleksibel dan populer di kalangan peneliti dan pengembang [24].



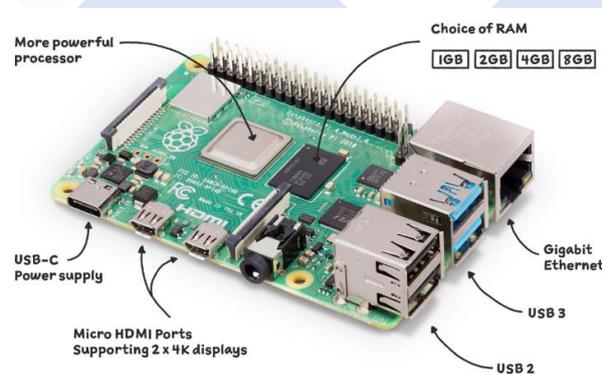
Gambar 2. 8 Logo openCV

OpenCV menyediakan berbagai fungsi dan modul yang memungkinkan pengembang untuk melakukan tugas-tugas kompleks dalam pengolahan gambar dan visi komputer dengan mudah. Salah satu fitur utama OpenCV adalah kemampuannya untuk melakukan deteksi wajah menggunakan algoritma *Haar Cascade* yang dikembangkan oleh Viola dan Jones. Algoritma ini menggunakan fitur berbasis *Haar-like features* dan teknik *AdaBoost* untuk melatih *classifier* yang dapat mendeteksi wajah dengan akurasi tinggi. Selain itu, OpenCV juga mendukung algoritma deteksi wajah berbasis *Local Binary Patterns (LBP)* yang lebih ringan dan cocok untuk aplikasi *real-time*. OpenCV juga menyediakan dukungan untuk integrasi dengan perangkat keras seperti kamera dan sensor, serta kemampuan untuk memproses video secara *real-time*. Dengan dukungan untuk *machine learning* melalui integrasi dengan library seperti *TensorFlow* dan *PyTorch*, OpenCV menjadi alat yang sangat kuat untuk pengembangan aplikasi visi komputer yang canggih. Keberadaan dokumentasi yang lengkap dan komunitas yang aktif juga membuat OpenCV menjadi pilihan utama bagi pengembang yang

ingin mengimplementasikan solusi visi komputer dalam berbagai bidang, mulai dari keamanan, robotika, hingga analisis medis [24].

2.9 Rapsberry Pi

Raspberry Pi adalah sebuah komputer papan tunggal (*single-board computer*) yang dikembangkan oleh *Raspberry Pi Foundation*. Komputer ini dirancang untuk tujuan edukasi, namun karena kemampuannya yang serbaguna dan harganya yang terjangkau, *Raspberry Pi* telah menjadi populer di berbagai bidang, termasuk penelitian, pengembangan prototipe, dan aplikasi Internet of Things (*IoT*). *Raspberry Pi* memiliki ukuran yang kecil, hemat daya, dan dilengkapi dengan port *General Purpose Input/Output* (GPIO) yang memungkinkannya untuk terhubung dengan berbagai sensor dan perangkat eksternal. Dengan prosesor yang cukup kuat dan dukungan untuk berbagai sistem operasi berbasis Linux, *Raspberry Pi* mampu menjalankan aplikasi yang kompleks, mulai dari pemrosesan data hingga pengendalian perangkat otomatis. Keunggulan utama *Raspberry Pi* adalah kemampuannya untuk digunakan dalam berbagai proyek, mulai dari yang sederhana hingga yang kompleks, menjadikannya alat yang sangat berharga bagi peneliti, pengembang, dan penggemar teknologi [25].



Gambar 2. 9 Rapsberry Pi 4 Model B

Raspberry Pi dapat digunakan dalam berbagai aplikasi, mulai dari proyek sederhana seperti mengendalikan lampu rumah hingga proyek kompleks seperti sistem otomasi industri. Untuk memulai penggunaan *Raspberry Pi*, pengguna perlu menginstal sistem operasi yang kompatibel, seperti Raspbian, pada kartu microSD.

Setelah sistem operasi terinstal, *Raspberry Pi* dapat dihubungkan ke monitor, keyboard, dan mouse untuk digunakan seperti komputer biasa. Selain itu, *Raspberry Pi* juga dapat diakses secara remote melalui SSH atau VNC, memungkinkan pengguna untuk mengontrolnya dari jarak jauh. Port GPIO pada *Raspberry Pi* memungkinkan pengguna untuk menghubungkan berbagai sensor, motor, dan perangkat lainnya, sehingga dapat digunakan untuk proyek *IoT*, robotika, atau sistem monitoring. Dengan dukungan komunitas yang besar, pengguna dapat dengan mudah menemukan tutorial, proyek contoh, dan dukungan teknis untuk memaksimalkan penggunaan *Raspberry Pi* [25].

Raspberry Pi juga banyak digunakan dalam aplikasi pengolahan gambar (*image processing*) karena kemampuannya untuk menjalankan algoritma pengolahan gambar secara *real-time*. Dengan dukungan kamera *Raspberry Pi* dan pustaka OpenCV, pengguna dapat mengembangkan sistem pengenalan wajah, deteksi objek, dan analisis gambar lainnya. Contoh aplikasi *image processing* yang menggunakan *Raspberry Pi* termasuk sistem pengawasan berbasis kamera, deteksi gerakan, dan pengenalan pola. *Raspberry Pi* juga dapat digunakan dalam proyek-proyek yang memerlukan analisis gambar secara *real-time*, seperti deteksi penyakit pada tanaman, pemantauan lalu lintas, dan sistem keamanan berbasis pengenalan wajah. Meskipun memiliki keterbatasan dalam hal daya pemrosesan dibandingkan dengan komputer desktop, *Raspberry Pi* tetap menjadi pilihan yang populer untuk aplikasi *image processing* karena biayanya yang rendah dan kemampuannya untuk diintegrasikan dengan perangkat keras lainnya [25].

2.9.1 Raspberry Pi 4 Model B

Raspberry Pi adalah modul komputer mini yang berfungsi mirip dengan komputer konvensional, dilengkapi dengan input dan output seperti papan mikrokontroler. Di antara produk *Raspberry Pi*, model 4 B menonjol dengan keunggulan dalam kecepatan prosesor, multimedia, kinerja, memori, dan konektivitas dibandingkan generasi sebelumnya. Selain efisien dalam konsumsi daya, *Raspberry Pi 4 B* juga memiliki pin GPIO yang dapat diprogram,

memungkinkan pengumpulan dan pengendalian modul sensor yang memerlukan port serial [26].

Secara umum, *Raspberry Pi* Model B memiliki kapasitas RAM sebesar 512 MB, yang lebih besar dibandingkan dengan Model A yang hanya memiliki 256 MB. Model B juga dilengkapi dengan port Ethernet untuk koneksi LAN. Desain *Raspberry Pi* didasarkan pada SoC (System on a Chip) Broadcom BCM2835, yang mengintegrasikan prosesor ARM1176JZF-S dengan kecepatan 700 MHz, GPU VideoCore IV, dan RAM 256 MB [26].

Raspberry Pi Model B menawarkan kinerja desktop yang setara dengan sistem PC x86. Fitur-fitur produk ini meliputi prosesor quad-core 64-bit berkinerja tinggi, dukungan untuk layar ganda dengan resolusi hingga 4K melalui dua port micro-HDMI, dekoder video perangkat keras hingga 4Kp60, serta kapasitas RAM hingga 4GB. Selain itu, perangkat ini dilengkapi dengan LAN nirkabel dual band 2,4/5,0 GHz, Bluetooth 5.0, GB Ethernet, USB 3.0, dan kemampuan PoE melalui add-on PoE HAT yang terpisah. LAN nirkabel dual band dan Bluetooth memiliki sertifikasi kepatuhan modular, memungkinkan papan ini dirancang menjadi produk akhir dengan pengujian dan kepatuhan yang signifikan berkurang, sehingga mengurangi biaya dan waktu ke pasar [26].

2.10 Roboflow

Roboflow adalah situs web yang sangat membantu bagi para pengembang dalam mengolah *dataset* untuk proyek penglihatan komputer. Ketika membuat proyek *computer vision*, para pengembang sering menggunakan roboflow karena situs web ini menawarkan berbagai fitur yang memudahkan dan membantu pengembang dalam membuat proyek tersebut [27].



Gambar 2. 10 Logo Roboflow

Langkah pertama yang harus dilakukan saat menggunakan roboflow adalah mengunggah *dataset* yang kemudian akan diorganisir sehingga proses anotasi

menjadi lebih mudah. Data ini kemudian digunakan sebagai input untuk proses pelatihan model. Setelah model selesai dilatih, pengembang dapat langsung melakukan proses deployment atau menggunakan model yang telah dilatih dalam sistem penglihatan komputer untuk melihat hasil deteksi dari model yang telah dilatih. Oleh karena itu, pengembang dapat menentukan apakah model yang telah dilatih sudah selesai atau belum [27].

2.11 Google Colab



Gambar 2. 11 Logo Google Colab

Google Colab merupakan sebuah lingkungan pengembangan terintegrasi (IDE) yang dirancang khusus untuk pemrograman Python. Keunggulan dari platform ini adalah pemrosesan yang dilakukan tidak bergantung pada perangkat keras lokal, melainkan dikerjakan oleh server Google yang dilengkapi dengan perangkat keras berkinerja tinggi. Hal ini memungkinkan pengguna untuk menjalankan kode yang kompleks tanpa harus khawatir mengenai keterbatasan spesifikasi komputer pribadi. Selain itu, dari segi perangkat lunak, Google Colab telah menyediakan hampir seluruh pustaka (library) yang diperlukan dalam pengembangan proyek, mulai dari analisis data hingga pembelajaran mesin. Dengan demikian, pengguna dapat lebih fokus pada pengembangan dan eksperimen tanpa perlu repot menginstal berbagai pustaka secara manual [28].

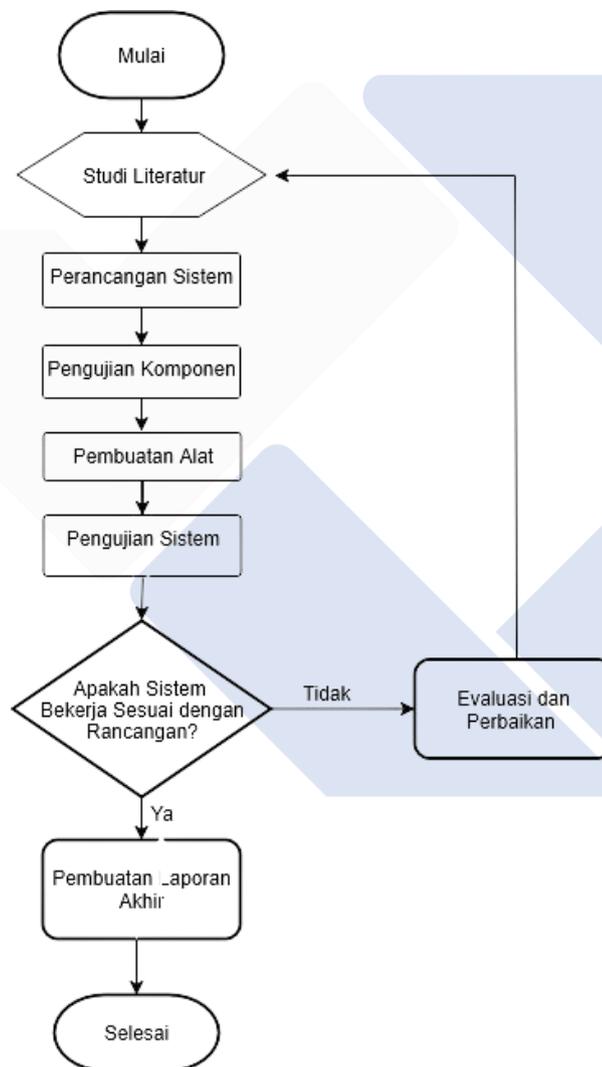
2.12 Internet of Things (IoT)

Internet of Things merupakan konsep yang menghubungkan berbagai objek fisik ke internet melalui jaringan sensor nirkabel, *radio-frequency identification* (RFID), dan teknologi komunikasi lainnya. *IoT* memungkinkan objek-objek tersebut untuk berkomunikasi dan bertukar data secara otomatis tanpa campur

tangan manusia secara langsung. Konsep ini telah berkembang pesat dalam beberapa tahun terakhir dan menjadi fondasi bagi berbagai aplikasi cerdas seperti smart home, smart grid, dan smart city. *IoT* bertujuan untuk menciptakan infrastruktur yang memungkinkan integrasi global dari berbagai perangkat, sehingga manusia dapat mengontrol dan memantau status objek tersebut secara *real-time* [29]. Internet of Things (IoT) tidak hanya terbatas pada konektivitas antar perangkat, tetapi juga mencakup pengolahan data yang dihasilkan oleh perangkat-perangkat tersebut. Dengan adanya *IoT*, data yang dikumpulkan dari berbagai sensor dapat dianalisis untuk memberikan informasi yang berguna dalam pengambilan keputusan. Teknologi seperti *wireless sensor networks* (WSN), *near-field communication* (NFC), dan *long-term evolution* (LTE) memainkan peran penting dalam menghubungkan perangkat-perangkat *IoT*. Selain itu, *IoT* juga menghadapi berbagai tantangan seperti keamanan data, interoperabilitas, dan skalabilitas. Meskipun demikian, potensi *IoT* dalam meningkatkan efisiensi dan kualitas hidup manusia sangat besar, terutama dalam bidang seperti kesehatan, transportasi, dan manajemen energi [29].

BAB III METODE PELAKSANAAN

Pelaksanaan proyek akhir yang berjudul "**Rancang Bangun Monitoring Hidroponik dengan Sistem *Image Processing* Berbasis *Iot***", dilakukan beberapa tahapan penelitian yang bertujuan untuk memudahkan penulis dalam proses pembuatan proyek akhir. Di bawah ini adalah tahapan-tahapan penelitian yang telah dirangkum dalam bentuk *flowchart* sebagai berikut:



Gambar 3. 1 Flowchart Alur Penelitian

3.1 Perancangan sistem

Ada beberapa hal yang perlu dilakukan agar sistem monitoring memiliki kinerja yang maksimal dalam melakukan monitoring pertumbuhan tanaman hidroponik. Berikut adalah tahapan yang perlu dilakukan dalam pembuatan alat ini :

1. Melakukan pengambilan *database* menggunakan kamera.
2. Melakukan perubahan ukuran pada *database* agar semua data memiliki dimensi 640x640 pixel.
3. Mengupload *database* yang telah dikumpulkan untuk proses anotasi menggunakan Roboflow.
4. Melakukan anotasi pada *database* dengan menggunakan *instance segmentation* pada objek yang akan dideteksi, seperti keseluruhan tanaman dan daun, sehingga hanya objek yang di segmentasi yang akan dilatih saat proses pelatihan.
5. Melakukan *preprocessing* dan *augmentasi database* untuk mengubah ukuran dan memperbanyak variasi model, sehingga mencerminkan berbagai bentuk yang mungkin ada di dunia nyata.
6. Merancang program monitoring hidroponik dengan menginput model yang telah dilatih ke dalam program dan melakukan integrasi dengan *IoT* seperti Firebase.

3.2 Pengujian Komponen

Tahap ini merupakan proses pengujian terhadap komponen-komponen yang akan digunakan dalam sistem, dengan tujuan memastikan bahwa seluruh komponen berfungsi dengan baik dan sesuai kebutuhan. Adapun pengujian komponen yang telah dilakukan meliputi:

1. Pengujian komponen *Raspberry Pi 4 Model B* dalam menjalankan program sederhana python dengan baik.
2. Pengujian *webcam* sebagai media dalam deteksi tanaman hidroponik.

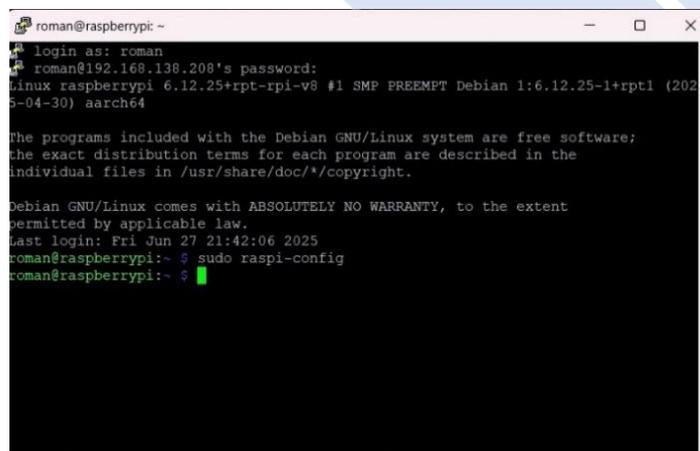
3.3 Pembuatan Alat

Proses pembuatan alat dilaksanakan di kostan, mengikuti rancangan sistem yang telah disusun sebelumnya. Langkah pertama adalah mengumpulkan *database* yang akan digunakan sebagai model pelatihan. Setelah *database* terkumpul, langkah selanjutnya adalah mengunggah data tersebut, diikuti dengan anotasi objek untuk menandai area yang relevan. Selanjutnya, dilakukan *preprocessing* dan *augmentasi* untuk meningkatkan variasi dalam *database*, sehingga model dapat belajar dari berbagai kondisi yang mungkin terjadi. Proses pelatihan model kemudian dilakukan, diakhiri dengan pengembangan program pemantauan hidroponik yang memanfaatkan *image processing* berbasis *IoT*.

3.4 Pengujian Sistem

Pada tahap ini akan dilakukan pengujian sistem yang telah dibuat sebelumnya sesuai dengan tahapan-tahapan yang telah dijelaskan. Alat akan diarahkan ke tanaman hidroponik, kemudian akan dilakukan pengujian apakah alat tersebut akan berfungsi dengan baik. Pengujian dilakukan dengan mengecek apakah program dapat melakukan capture gambar secara *real-time* pada objek yang akan di uji coba.

3.4.1 Akses *Raspberry Pi* Secara Headless

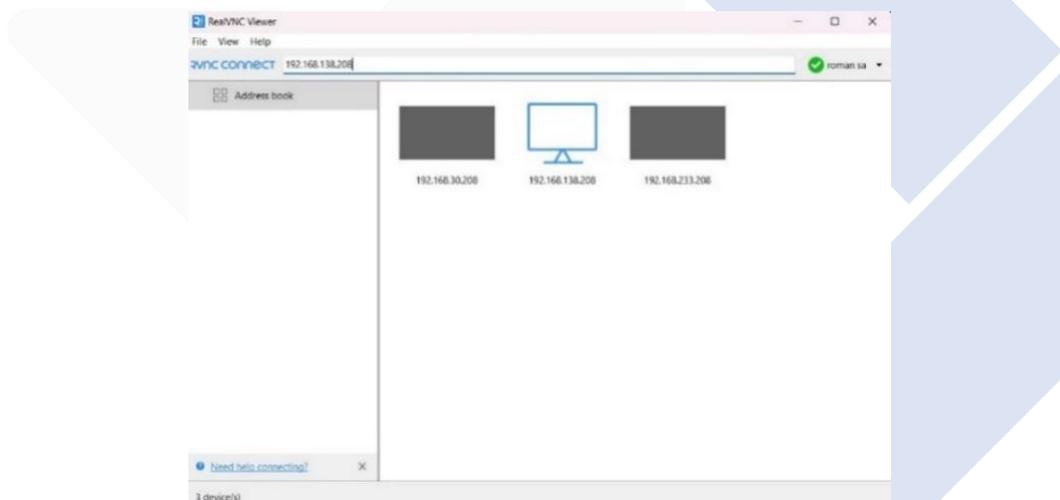


```
roman@raspberrypi: ~  
login as: roman  
roman@192.168.138.208's password:  
Linux raspberrypi 6.12.25+rpt-rpi-v8 #1 SMP PREEMPT Debian 1:6.12.25-1+rpt1 (2025-04-30) aarch64  
  
The programs included with the Debian GNU/Linux system are free software;  
the exact distribution terms for each program are described in the  
individual files in /usr/share/doc/*/copyright.  
  
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent  
permitted by applicable law.  
Last login: Fri Jun 27 21:42:06 2025  
roman@raspberrypi:~$ sudo raspi-config  
roman@raspberrypi:~$
```

Gambar 3. 2 Cara Access Raspberry Secara Headless

Gambar 3.2 menunjukkan proses akses *Raspberry Pi* secara headless menggunakan SSH dari laptop. User berhasil login ke *Raspberry Pi* pada terminal dengan memasukkan username dan password, lalu menjalankan perintah *sudo raspi-config* untuk mengatur konfigurasi sistem. Cara ini memungkinkan pengguna mengelola *Raspberry Pi* tanpa perlu monitor, keyboard, atau mouse yang langsung terhubung ke perangkat. *Raspberry Pi* berfungsi sebagai pusat pengolahan data sensor dan *image processing* untuk memantau pertumbuhan tanaman hidroponik. Akses SSH memungkinkan seluruh pengelolaan, pemantauan, dan pengaturan sistem dilakukan secara remote dari laptop, sehingga instalasi menjadi lebih praktis, terutama jika perangkat dipasang di lokasi yang sulit dijangkau atau ruang terbatas.

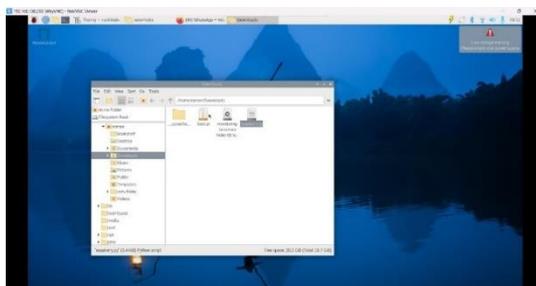
3.4.2 VNC (Virtual Network Computing)



Gambar 3. 3 Tampilan Vnc Membuat Laptop Jadi Monitor Raspberry

Berdasarkan gambar tersebut terlihat aplikasi *RealVNC Viewer* yang menampilkan beberapa alamat IP perangkat di jaringan lokal. Salah satu alamat IP (92.168.138.208) dipilih, yang menandakan perangkat tersebut siap diakses secara remote. Penggunaan *VNC Viewer*, laptop bisa digunakan untuk mengakses dan mengontrol tampilan desktop Raspberry Pi dari jarak jauh, seolah-olah laptop menjadi monitor tambahan untuk Raspberry Pi tersebut. Proyek akhir yang sedang diteliti menggunakan Raspberry Pi sebagai otak sistem untuk mengolah data sensor dan *image processing* guna memantau pertumbuhan tanaman lewat kamera. Penggunaan *VNC* memungkinkan proses pemantauan dan pengelolaan sistem dilakukan dari laptop tanpa perlu monitor fisik di Raspberry Pi

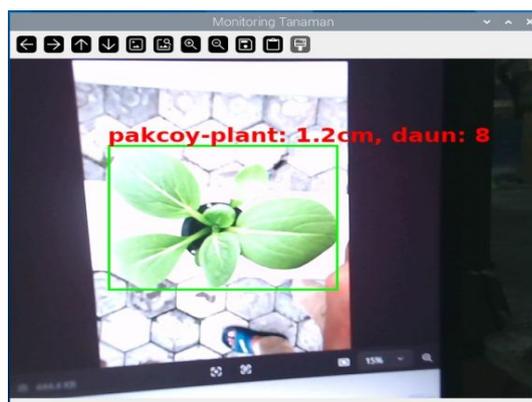
3.4.3 File sudah masuk ke Raspberry pi



Gambar 3. 4 file proyek yang diinput ke raspberry pi

File proyek yang telah dikembangkan di laptop dipindahkan ke Raspberry Pi agar dapat dijalankan langsung pada perangkat tersebut. Proses pemindahan *file* ini penting dalam implementasi sistem monitoring hidroponik, karena Raspberry Pi berperan sebagai pusat pengolahan data sensor dan citra tanaman. *File* seperti “best.pt” merupakan model hasil pelatihan *deep learning*, sedangkan “monitoring-tanaman-hidro...” dan “raspberry.py” adalah script utama untuk pemantauan dan pengolahan data. Langkah memasukkan *file* proyek ke Raspberry Pi merupakan bagian dari tahapan rancang bangun sistem monitoring hidroponik menggunakan *image processing* berbasis *IoT*. Menjalankan script dan model pada Raspberry Pi memungkinkan sistem melakukan pemantauan pertumbuhan tanaman secara otomatis, memproses citra dari kamera, serta mengirimkan data ke platform *IoT* untuk pemantauan jarak jauh. Proses ini mendukung tujuan penelitian, yaitu menciptakan sistem monitoring hidroponik dan *IoT*.

3.4.4 Pengujian di Raspberry Pi



Gambar 3. 5 Pengujian di Raspberry

Pengujian sistem dilakukan langsung di perangkat Raspberry Pi untuk melihat apakah sistem bisa bekerja tanpa bantuan komputer. Hasilnya, sistem berhasil mengenali tanaman dengan baik. Layar menampilkan hasil deteksi tanaman dengan lebar 1,2 cm dan jumlah daun sebanyak 8 helai. Sistem secara otomatis melakukan deteksi menggunakan model *image processing* yang telah dilatih sebelumnya.

Uji coba ini menunjukkan kalau Raspberry *Pi* mampu menjalankan sistem monitoring secara langsung dan menampilkan hasilnya di layar. Artinya, sistem yang dirancang sudah bisa digunakan di lapangan tanpa harus terhubung ke alat yang besar atau mahal. Ini sesuai dengan tujuan utama dari proyek, yaitu membuat alat pemantau tanaman hidroponik yang praktis dan berbasis *IoT*.

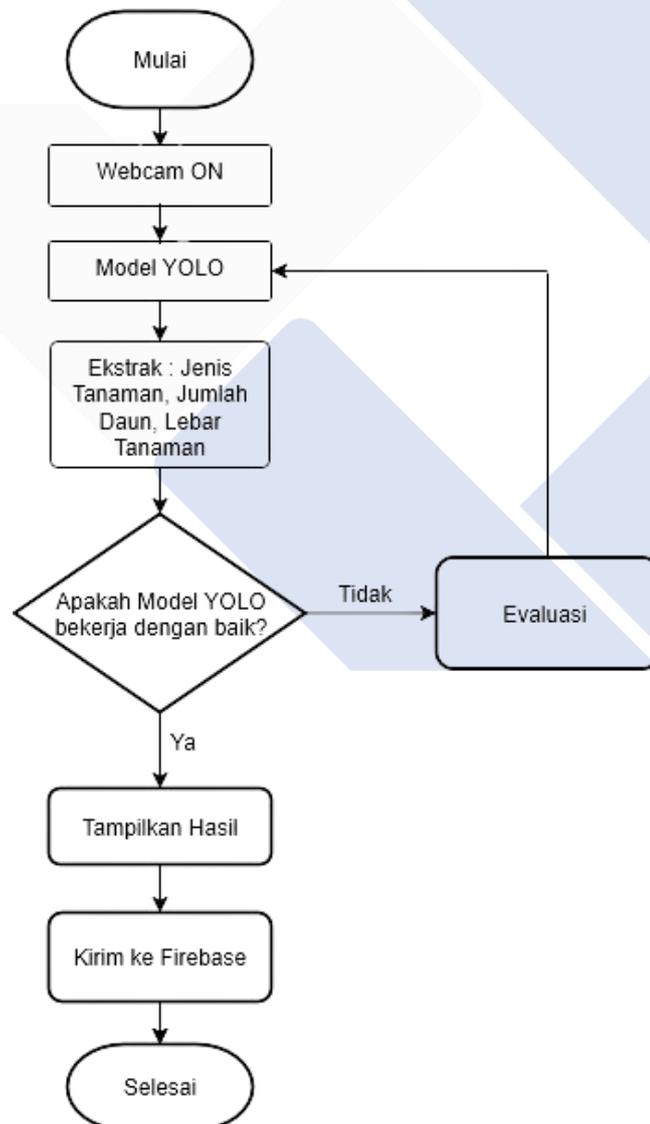
3.5 Pembuatan Laporan Akhir

Tahap terakhir dalam proyek ini adalah pembuatan laporan akhir, yang berfungsi untuk menyajikan ringkasan komprehensif dari seluruh proses pengerjaan. Laporan ini mencakup bagian-bagian penting seperti latar belakang, rumusan masalah, tujuan, batasan masalah, dasar teori, metode penelitian, pembahasan, hasil, kesimpulan dan saran. Selain merangkum hasil kerja, laporan akhir juga bertujuan untuk memberikan informasi yang berguna bagi pembaca, sehingga dapat dijadikan referensi untuk studi atau penelitian di masa mendatang. Oleh karena itu, laporan ini tidak hanya berfungsi sebagai dokumentasi proyek, tetapi juga sebagai kontribusi terhadap pengembangan ilmu pengetahuan

BAB IV PEMBAHASAN

4.1 Sistem Kerja Alat

Bab ini membahas secara mendalam mengenai proses dan mekanisme kerja alat pada proyek akhir ini. Sistem yang dirancang bertujuan untuk memantau pertumbuhan tanaman hidroponik secara otomatis menggunakan teknologi *image processing* yang terintegrasi dengan Internet of Things (IoT). Untuk memberikan pemahaman yang sistematis, disajikan sebuah *flowchart* yang menggambarkan tahapan kerja sistem secara keseluruhan.



Gambar 4. 1 Flowchart

4.2 Penerapan Sistem

Pengembangan sistem dilakukan dengan menggunakan aplikasi *Visual Studio Code* sebagai lingkungan kerja utama, serta bahasa pemrograman Python karena fleksibilitas dan dukungannya terhadap pustaka *image processing*, *machine learning*, dan integrasi dengan sistem *IoT*. Google Colab menjadi tempat untuk melakukan *training dataset* yang sudah ada dengan memakai *library Ultralytics YOLOv8*.

Selanjutnya, Raspberry Pi 4 model B menjadi pusat untuk mengelola sistem yang menjalankan seluruh proses monitoring dan pengiriman data. Untuk penyajian data akan menggunakan platform seperti *Firestore Real-time Database*. *Greenhouse* menjadi lokasi tempat sistem monitoring. Implementasi ini bertujuan untuk menguji integrasi sistem mulai dari pengambilan data, pengolahan informasi, hingga pengiriman data ke platform *IoT* secara *real-time*.

4.2.1 Pengambilan Dataset

Pengumpulan *dataset* merupakan tahapan awal yang sangat krusial dalam pengembangan sistem berbasis *image processing*, seperti deteksi objek maupun segmentasi citra. *Dataset* berfungsi sebagai fondasi utama dalam proses pelatihan model, sehingga kualitas dan keragamannya sangat memengaruhi kinerja sistem secara keseluruhan. Dalam penelitian ini, *dataset* diperoleh melalui pengambilan gambar tanaman hidroponik yang menjadi objek pemantauan.

Proses pengambilan gambar dilakukan secara sistematis dari berbagai sudut, terutama dari sisi atas (*top view*) dan beberapa sisi samping, guna mendapatkan variasi tampilan daun dan struktur tanaman secara menyeluruh. Variasi sudut pandang ini penting untuk memastikan bahwa model pengolahan citra yang dikembangkan mampu mengenali objek dengan presisi tinggi, meskipun dalam kondisi orientasi dan pencahayaan yang berbeda.



Gambar 4. 2 Contoh Gambar Dataset dari Sudut Atas



Gambar 4. 3 Contoh Gambar Dataset dari Sudut Samping

Pengambilan gambar dilaksanakan di area *greenhouse* Politeknik Manufaktur Negeri Bangka Belitung (Polman Babel) dan *rahsaponik greenhouse*, dengan menggunakan *smartphone* karena dapat mengambil gambar beresolusi tinggi sehingga akan memperbagus hasil dari *train dataset* itu sendiri. Proses dokumentasi berlangsung selama satu bulan, dimulai dari tanggal 15 Mei hingga 14 Juni 2025, tujuan dari jangka waktu satu bulan dalam pengambilan *dataset* yaitu untuk menangkap dinamika pertumbuhan tanaman dari waktu ke waktu. Hasil gambar kemudian digunakan sebagai *dataset* dalam proses pelatihan dan evaluasi sistem monitoring berbasis *image processing* yang dikembangkan dalam penelitian ini.

4.2.1.1 Pertumbuhan Tinggi Tanaman

Tinggi tanaman kailan dan pakcoy merupakan indikator penting dalam menilai pertumbuhan kedua sayuran ini. Kailan umumnya dapat mencapai tinggi antara 30 hingga 60 cm, dipengaruhi oleh varietas, kondisi lingkungan, dan teknik budidaya. Faktor seperti pencahayaan, kelembaban, dan nutrisi tanah berperan signifikan dalam menentukan tinggi tanaman. Pengukuran dilakukan dari permukaan tanah hingga titik tertinggi daun, memberikan informasi tentang tinggi total tanaman yang mencerminkan tingkat pertumbuhan dan perkembangan vegetatifnya. Sementara itu, pakcoy biasanya memiliki tinggi berkisar antara 20 hingga 40 cm. Tinggi pakcoy juga dipengaruhi oleh intensitas cahaya, kelembaban tanah, suhu, dan ketersediaan nutrisi. Penerapan praktik agronomi yang baik dapat meningkatkan pertumbuhan dan menghasilkan tinggi yang optimal. Secara keseluruhan, pemantauan tinggi tanaman kailan dan pakcoy secara berkala sangat penting untuk meningkatkan produktivitas dan kualitas hasil pertanian.

Penelitian ini, data tinggi tanaman asli diperoleh secara manual dan digunakan sebagai *benchmark* atau acuan pembanding terhadap hasil deteksi model. Namun, karena posisi kamera diletakkan menghadap ke atas tanaman (*top-view*), maka parameter yang terdeteksi oleh model *YOLO* berupa dimensi horizontal tanaman. Oleh karena itu, parameter tinggi tanaman secara visual dalam sistem ini direpresentasikan sebagai lebar tanaman, yang diukur dari sebaran objek tanaman dalam citra dari atas.

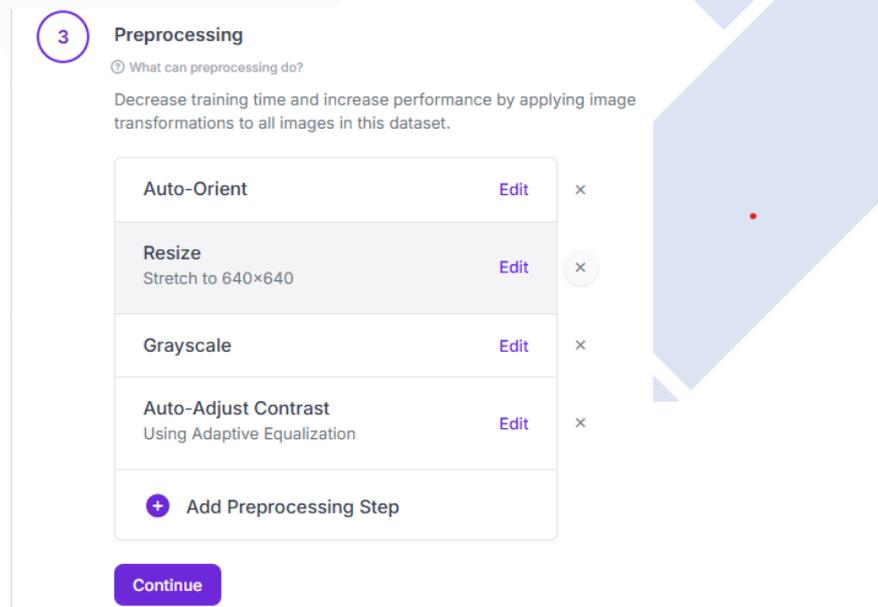
4.2.1.2 Banyak Daun

Tanaman kailan umumnya memiliki jumlah daun yang bervariasi, berkisar antara 8 hingga 15 daun per tanaman, tergantung pada faktor-faktor seperti varietas, kondisi lingkungan, dan teknik budidaya yang diterapkan. Jumlah daun yang optimal tidak hanya berkontribusi pada proses fotosintesis, tetapi juga berpengaruh terhadap kualitas dan kuantitas hasil panen. Peningkatan jumlah daun dapat menunjukkan kesehatan tanaman yang baik dan kemampuan tanaman untuk beradaptasi dengan kondisi lingkungan. Sementara itu, pakcoy biasanya memiliki jumlah daun yang lebih banyak, berkisar antara 10 hingga 20 daun per tanaman. Jumlah daun yang tinggi pada pakcoy juga berperan penting dalam meningkatkan

efisiensi fotosintesis dan pertumbuhan tanaman. Seperti kailan, faktor-faktor seperti pencahayaan, kelembaban tanah, dan nutrisi sangat mempengaruhi jumlah daun yang dihasilkan. Secara keseluruhan, pemantauan jumlah daun pada tanaman kailan dan pakcoy sangat penting dalam upaya meningkatkan produktivitas dan kualitas hasil pertanian, serta untuk memastikan keberhasilan budidaya kedua jenis sayuran ini.

4.2.2 Images/Data Pre-Processing

Gambar-gambar yang telah diambil dan disimpan tersedia dalam bentuk data mentah (*raw data*). Tujuan dari *Pre-Processing* ini yaitu meningkatkan kualitas data dan mempermudah model *machine learning* atau *deep learning* untuk mengetahui pola dari *dataset*. Sebelum digunakan dalam proses pelatihan model berbasis *machine learning* atau *deep learning*, data tersebut harus melalui beberapa tahapan *pre-processing* yang penting untuk memastikan kualitas dan efektivitas sistem yang akan dibangun.



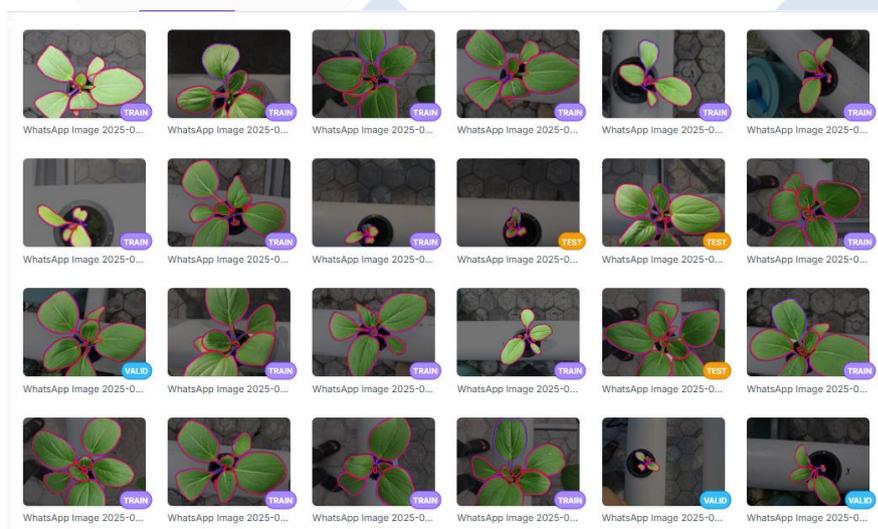
Gambar 4. 4 Pre-Processing dataset

Salah satu tujuan dari *pre-processing* yaitu untuk mengurangi waktu *train dataset* dan meningkatkan performa *training machine learning* atau *deep learning*. Salah satu *pre-processing* yang dilakukan yaitu, *resize* gambar menjadi dimensi 640

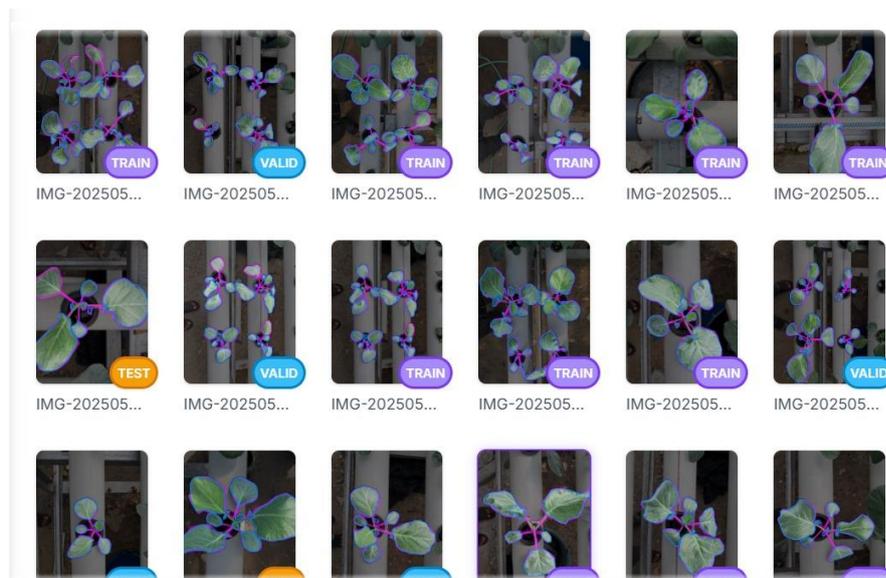
x 640, menambahkan filter *grayscale* dan *auto-adjust contrast*. Hal ini dilakukan untuk menghindari error saat melakukan *training dataset* akibat perbedaan dimensi input, lalu ukuran 640 x 640 menjadi standar pelatihan *dataset* dari model *YOLOv8*, filter *grayscale* mendorong *train dataset* berfokus pada melatih bentuk dan struktur, dan manfaat dari *auto-adjust contrast* yaitu untuk meningkatkan perbedaan antara objek yang ingin dideteksi dengan latar belakangnya.

4.2.3 Pembuatan Label Dataset

Pembuatan label *dataset* merupakan tahap krusial dalam pembuatan sistem *image processing*, agar algoritma *YOLO* dapat mengetahui nama objek yang sedang dideteksi. Langkah awal yang dilakukan adalah proses anotasi gambar, yaitu pemberian label atau penandaan secara manual terhadap objek-objek penting dalam citra, seperti daun dan keseluruhan bagian tanaman. Anotasi ini merupakan komponen kunci dalam pelatihan model segmentasi atau deteksi objek karena memungkinkan sistem untuk memahami pola visual berdasarkan label yang diberikan.



Gambar 4. 5 Hasil Anotasi Tanaman Pakcoy



Gambar 4. 6 Hasil Anotasi Tanaman Kailan

4.2.4 Image Augmentation

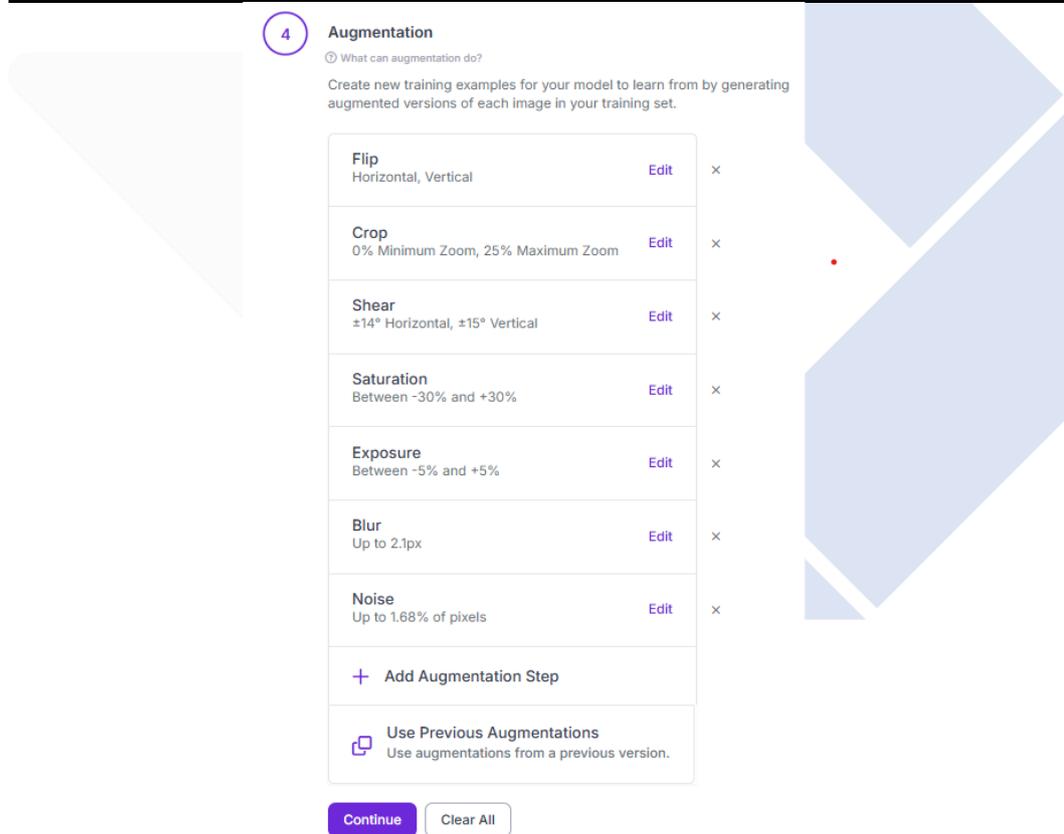
Setelah proses anotasi selesai, langkah selanjutnya adalah melakukan *data augmentation*, yaitu teknik untuk meningkatkan variasi dan jumlah *dataset* tanpa harus menambah gambar baru. *Augmentasi* dilakukan dengan menerapkan berbagai transformasi pada gambar asli, seperti rotasi pada berbagai sudut, *flipping* (membalik horizontal dan vertikal), penyesuaian tingkat pencahayaan (*brightness/contrast*), *zoom in* dan *zoom out*, translasi (pergeseran posisi objek), serta perubahan saturasi warna. Teknik ini tidak hanya menambah jumlah data, tetapi juga memperkaya keragaman visual dalam *dataset*, sehingga model yang dilatih dapat belajar mengenali objek dari berbagai kondisi pencahayaan, sudut pandang, dan orientasi. Hal ini sangat penting, mengingat dalam kondisi lapangan, objek seperti daun dan keseluruhan tanaman dapat tertangkap kamera dalam berbagai posisi yang tidak selalu ideal.

Setelah proses *data augmentation* selesai, *dataset* dibagi menjadi tiga bagian utama secara sistematis, yaitu *training set*, *validation set*, dan *testing set*. *Training set* digunakan untuk melatih model agar dapat mempelajari pola dan karakteristik objek yang ingin dikenali. *Validation set* digunakan untuk mengevaluasi performa model selama proses pelatihan berlangsung, serta membantu dalam penyesuaian parameter guna menghindari *overfitting*. Sementara itu, *testing set* dimanfaatkan untuk menguji akurasi dan performa akhir model

terhadap data yang benar-benar baru dan belum pernah dikenali sebelumnya. Pembagian ini umumnya menggunakan proporsi 87,53% data untuk pelatihan, 8,24% untuk validasi, dan 4,24% untuk pengujian. Namun, proporsi tersebut dapat disesuaikan berdasarkan jumlah total *dataset* yang tersedia dan kompleksitas model yang digunakan. Dengan pembagian ini, sistem yang dikembangkan diharapkan memiliki kemampuan generalisasi yang tinggi, bekerja secara optimal dalam kondisi nyata, serta menghasilkan deteksi yang akurat dan andal.

Tabel 4. 1 Pembagian *Dataset*

Data Training	Data Validasi	Data Test
87,53%	8,24%	4,24%
1.116	105	54



Gambar 4. 7 Proses Augmentasi Dataset

Tujuan utama dari dilakukannya *augmentasi dataset* dalam penelitian ini adalah untuk meningkatkan kemampuan generalisasi model terhadap berbagai

variasi kondisi nyata yang mungkin terjadi di lapangan. Dengan menerapkan teknik *augmentasi* seperti rotasi, *flipping*, perubahan pencahayaan, dan *zoom*, model dilatih untuk mengenali objek tanaman, khususnya daun dan keseluruhan tanaman, dari berbagai sudut pandang, orientasi, dan tingkat pencahayaan yang berbeda. Proses ini sangat penting untuk mengatasi keterbatasan jumlah data asli yang tersedia, serta mengurangi risiko *overfitting* pada saat pelatihan model. *Augmentasi* juga membantu memperkaya keragaman *dataset* tanpa perlu melakukan pengambilan gambar tambahan, sehingga mempercepat dan mengefisienkan proses pengembangan sistem deteksi gambar. Dengan demikian, model yang dihasilkan dapat lebih andal dalam mendeteksi objek secara akurat pada lingkungan nyata yang dinamis dan tidak selalu ideal.

Dengan tahapan ini *dataset* akan menjadi banyak variasi, serta data mentah yang diperoleh dari proses akuisisi dapat diubah menjadi *dataset* yang siap digunakan dalam pelatihan model secara efektif dan efisien, serta menghasilkan sistem pengolahan citra yang akurat dan andal dalam memantau pertumbuhan tanaman hidroponik.

4.2.5 Training Dataset

Setelah seluruh *dataset* berhasil dikumpulkan, dilakukan proses pelatihan model (*training*) menggunakan algoritma *YOLOv8* yang berbasis *deep learning*. Tahapan ini dimulai dengan mengimpor *dataset* yang telah melalui proses anotasi dan *augmentasi* ke dalam lingkungan pelatihan pada platform Google Colab. *YOLOv8* dipilih karena kemampuannya yang tinggi dalam mendeteksi objek secara *real-time* dan akurat, serta kompatibel dengan kebutuhan proyek dalam mengenali daun dan keseluruhan tanaman hidroponik. Selama proses pelatihan, model mempelajari pola visual dari *dataset*, seperti bentuk dan struktur daun serta tinggi batang tanaman, dengan mengoptimalkan bobot jaringan saraf agar dapat mengenali objek tersebut secara otomatis.

```
from google.colab import drive
drive.mount('/content/gdrive')

!ls '/content/gdrive/My Drive/Finalproyek/Data/'

DATA_DIR = '/content/gdrive/My Drive/Finalproyek/Data/'

!pip install ultralytics

import os
from ultralytics import YOLO
model = YOLO('yolov8n-seg.pt')
model.train(data='/content/gdrive/My Drive/Finalproyek/Data/data.yaml', epochs=100, imgsz=640)

!scp -r /content/runs /content/gdrive/My Drive/Finalproyek/
```

Gambar 4. 8 Proses Training Dataset di Google Colab

Gambar 4.7 merupakan program untuk proses pelatihan model dilakukan menggunakan platform Google Colab dengan bantuan pustaka *Ultralytics* untuk menjalankan algoritma *YOLOv8n-segmentation*. Langkah awal dimulai dengan mengimpor dan me-mount *Google Drive* menggunakan perintah *drive.mount('/content/gdrive')*, agar *dataset* yang disimpan dalam *Drive* dapat diakses oleh sistem *Colab*. Selanjutnya, dilakukan pengecekan isi direktori *dataset* yang berada di folder *SegmentationPakcoy/Data/* untuk memastikan bahwa *file* data telah tersedia dan siap untuk digunakan. Direktori tersebut disimpan dalam variabel *DATA_DIR* untuk mempermudah pemanggilan *path* dalam proses berikutnya. Kemudian, pustaka *Ultralytics* diinstal melalui perintah *!pip install ultralytics* agar model *YOLOv8* dapat digunakan.

Setelah semua dependensi tersedia, model *YOLOv8n-seg* dipanggil dengan memanggil *file* bobot awal *YOLOv8n-seg.pt*. Model ini merupakan varian ringan dari *YOLOv8* yang dirancang untuk kebutuhan segmentasi objek. Proses pelatihan dijalankan dengan memanggil metode *.train()* dan menyuplai konfigurasi data dari *file data.yaml* yang berisi *path dataset*, informasi kelas objek, dan format anotasi. Pelatihan dilakukan selama 100 *epochs* dengan ukuran input gambar 640 piksel, yang merupakan standar pelatihan untuk *YOLOv8*. Setelah proses *training* selesai, seluruh hasil pelatihan yang berada dalam folder *runs* disalin kembali ke *Google Drive* menggunakan perintah *scp* agar hasil dapat diakses dan dianalisis lebih lanjut,

termasuk bobot model yang telah dilatih dan visualisasi hasil pelatihan seperti grafik *loss* dan prediksi segmentasi.

Model	size (pixels)	mAP ^{box} ₅₀₋₉₅	mAP ^{mask} ₅₀₋₉₅	Speed CPU ONNX (ms)	Speed A100 TensorRT (ms)	params (M)	FLOPs (B)
YOLOv8n	640	36.7	30.5	96.1	1.21	3.4	12.6
YOLOv8s	640	44.6	36.8	155.7	1.47	11.8	42.6
YOLOv8m	640	49.9	40.8	317.0	2.18	27.3	110.2
YOLOv8l	640	52.3	42.6	572.4	2.79	46.0	220.5
YOLOv8x	640	53.4	43.4	712.1	4.02	71.8	344.1

Gambar 4. 9 Model Segmentasi YOLOv8

Setelah proses *training* selesai, model dievaluasi menggunakan *dataset* validasi dan pengujian untuk mengetahui nilai *accuracy*, *precision*, *recall*, dan *mean Average Precision* (mAP). Hasil pelatihan ini nantinya akan diintegrasikan ke dalam sistem Raspberry Pi sebagai komponen utama pengenalan objek tanaman secara otomatis, yang selanjutnya dikirimkan ke sistem *IoT* untuk ditampilkan kepada pengguna.

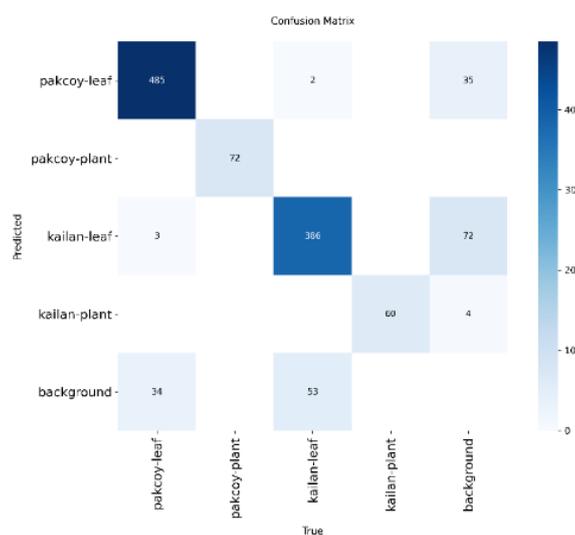
4.2.5.1 Hasil Training Dataset

Proses *training* model telah diselesaikan dengan menggunakan arsitektur *YOLOv8*. Hasil pelatihan dianalisis berdasarkan metrik seperti nilai *loss*, *precision*, *recall*, dan mAP, yang digunakan untuk mengevaluasi kinerja model terhadap *dataset* yang digunakan.

Epoch	GPU_mem	box_loss	seg_loss	cls_loss	dfl_loss	Instances
100/100	6.23G	0.3938	0.5978	0.2661	0.8237	66
	Class	Images	Instances	Box(P	R	mAP50

Gambar 4. 10 Gambar Hasil Training Dataset 100 epochs

Model telah berhasil dilatih selama 100 *epochs* dengan ukuran input gambar sebesar 640×640 piksel. Selama proses pelatihan, model menggunakan memori GPU sebesar 6.23 GB. Berdasarkan hasil akhir *training*, diperoleh nilai *box loss* sebesar 0.3938, *segmentation loss* sebesar 0.5978, *classification loss* sebesar 0.2661, dan *distribution focal loss* sebesar 0.8237. Nilai-nilai *loss* tersebut menunjukkan bahwa model mampu mempelajari pola dari data dengan cukup baik, meskipun masih terdapat ruang untuk peningkatan akurasi khususnya pada segmentasi dan prediksi posisi *bounding box*.



Gambar 4. 11 Confusion Matrix

Berdasarkan *confusion matrix* hasil evaluasi model, dapat dilihat bahwa model memiliki performa yang sangat baik dalam mengklasifikasikan kelas pakcoy-leaf dan kailan-leaf, dengan jumlah prediksi benar masing-masing sebesar 485 dan 386 *instance*. Namun, masih terdapat sejumlah kesalahan prediksi pada kedua kelas tersebut, di mana pakcoy-leaf terkadang keliru diklasifikasikan sebagai kailan-leaf (35 *instance*) maupun sebagai *background* (34 *instance*). Demikian pula, kailan-leaf juga beberapa kali salah dikenali sebagai kailan-plant sebanyak 72 *instance*.

Secara keseluruhan, *confusion matrix* ini menunjukkan bahwa meskipun model sudah cukup akurat dalam mengklasifikasikan beberapa kelas utama, masih diperlukan penyempurnaan terutama dalam membedakan kelas yang serupa dan

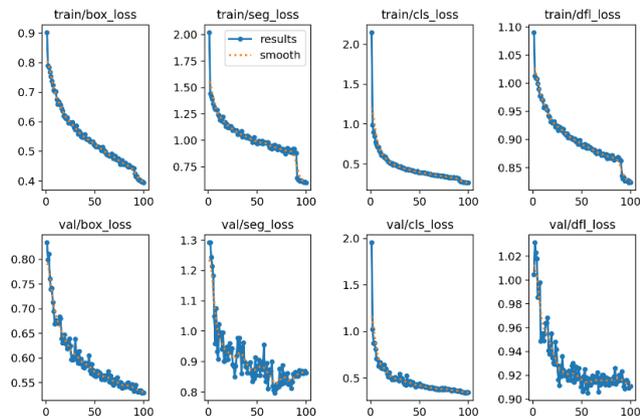
mengurangi kesalahan deteksi pada *background*. Meskipun Masih terdapat masalah yang cukup signifikan terjadi pada kelas pakcoy-plant, di mana seluruh instance dari kelas ini salah diklasifikasikan sebagai kailan-plant. Hal ini mengindikasikan bahwa model belum mampu membedakan secara efektif antara kedua kelas tanaman tersebut, kemungkinan disebabkan oleh kemiripan visual antara objek, kurangnya data representatif, atau inkonsistensi dalam anotasi. Sebaliknya, model menunjukkan performa yang cukup baik dalam mengenali kailan-plant dengan 60 instance yang berhasil dikenali dengan benar dari total keseluruhan.

Selain itu, terdapat sejumlah kasus di mana latar belakang (*background*) secara keliru diprediksi sebagai objek tanaman, seperti pakcoy-leaf (34 instance) dan kailan-leaf (53 instance). Hal ini menunjukkan bahwa model terkadang mendeteksi objek secara berlebihan (*false positive*), yang dapat diminimalkan dengan pelatihan tambahan pada citra latar belakang atau dengan teknik *augmentasi* data yang lebih bervariasi. Setelah diperoleh *confusion matrix*, langkah selanjutnya adalah menghitung nilai akurasi. Berikut ini adalah rumus-rumus yang digunakan:

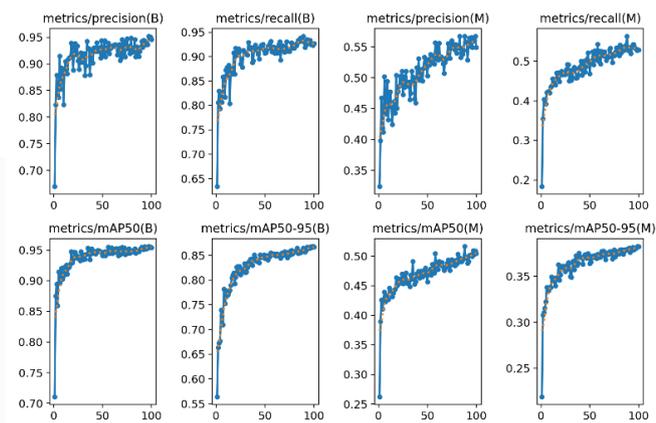
$$\text{Akurasi} = \frac{\text{Jumlah Prediksi Benar}}{\text{Jumlah Seluruh Prediksi}} \times 100\%$$

$$\begin{aligned}\text{Akurasi} &= \frac{931}{1,286} \times 100\% \\ &= 72,39\%\end{aligned}$$

Secara keseluruhan, model mencapai akurasi keseluruhan sebesar 72,39%, dihitung dari total prediksi benar (931 instance) dibandingkan seluruh data pengujian (1286 instance). Meskipun performa model untuk beberapa kelas utama sudah cukup baik, diperlukan peningkatan untuk kelas yang sulit dibedakan serta penanganan terhadap kesalahan klasifikasi latar belakang agar model dapat lebih andal dalam aplikasi nyata. Selain dari *confusion matrix* juga terdapat grafik hasil *train dataset* berdasarkan grafik *loss*, pada gambar 4.9 menunjukkan tren penurunan nilai *loss* selama 100 *epochs*, bahwa model secara konsisten dan mengalami peningkatan performa seiring berjalannya proses *training*.



Gambar 4. 12 Grafik Loss Training dan Validation



Gambar 4. 13 Grafik Metrics Precision dan Recall

Pada gambar 4.11 menunjukkan bahwa precision dan *recall* untuk *bounding box* (B) meningkat secara signifikan dan stabil di atas 0.90, bahwa model memiliki kemampuan sangat baik dalam mendeteksi objek secara tepat dan lengkap. Sedangkan, untuk grafik precision dan *recall* untuk mask (M) tidak setinggi *bounding box* karena berakhir di sekitar angka 0,55 untuk precision dan 0,52 untuk *recall*, namun model masih menunjukkan tren peningkatan yang konsisten, hal ini mencerminkan proses pembelajaran segmentasi yang efektif namun masih bisa ditingkatkan.

Gambar 4.12 juga menjelaskan grafik mAP50-90 untuk *bounding box* (B) meningkat drastis lebih dari 0,95 dan 0,85 yang menandakan deteksi objek secara spasial sangat akurat pada berbagai threshold IoU (Intersection over Union).

Terakhir grafik mAP50-95 untuk mask (M) meningkat hingga sekitar 0,52 dan 0,37, menandakan kemampuan segmentasi cukup baik namun belum seakurat *bounding box*. Model *YOLOv8* menunjukkan kinerja yang baik dalam hal deteksi dan klasifikasi keseluruhan tanaman serta daun, namun masih terdapat kelemahan antara jenis tanaman serta dalam mendeteksi background dengan benar. Meningkatkan performa model dapat dilakukan dengan menambah jumlah *dataset train* pada kelas yang kurang representatif, lalu peningkatan kualitas anotasi dan menambah beberapa *augmentasi* pada *dataset*. Selanjutnya melakukan proses pengujian *training dataset*.

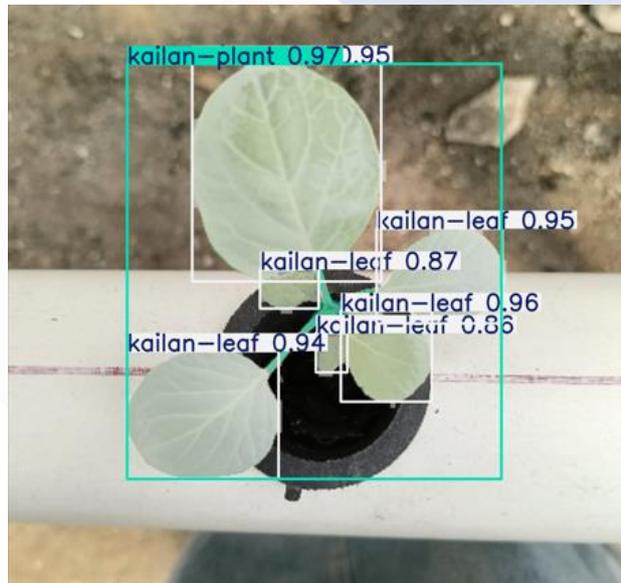
Pengujian model *dataset* bisa dilihat pada gambar 4.12 yang terdapat program untuk mengimport *YOLO* dari library *Ultralytics* dan *import cv2* yang merupakan library dari *OpenCV* untuk proses gambar, seperti membaca dan menampilkan gambar, lalu memanggil model *YOLOv8* hasil *training dataset* (*file best.pt*). Setelah itu, gambar yang akan diprediksi dibaca dari *file contoh_gambar.jpg* menggunakan *OpenCV*. Sebelum melanjutkan, program mengecek apakah gambar berhasil dibaca jika tidak, maka akan muncul error bahwa gambar tidak ditemukan. Jika gambar sudah valid, maka model akan melakukan prediksi terhadap gambar tersebut dan menghasilkan output yang berisi informasi deteksi seperti *bounding box*, label objek, dan *mask* karena ada anotasi menggunakan segmentasi. Hasil deteksi ini kemudian ditampilkan secara langsung dalam jendela pop-up menggunakan perintah *plot(show=True)*. Selain itu, gambar hasil prediksi juga disimpan ke dalam *file* baru bernama *hasil_prediksi.jpg* agar dapat dilihat kembali tanpa perlu menjalankan ulang proses deteksi.

```

1  from ultralytics import YOLO
2  import cv2
3
4  model = YOLO("best.pt")
5
6  image_path = "IMG20250527095810.jpg"
7  image = cv2.imread(image_path)
8
9  if image is None:
10 |     raise ValueError(f"Gambar tidak ditemukan: {image_path}")
11
12  results = model(image)
13
14  results[0].plot(show=True)
15
16  cv2.imwrite("hasil_prediksi.jpg", results[0].plot())

```

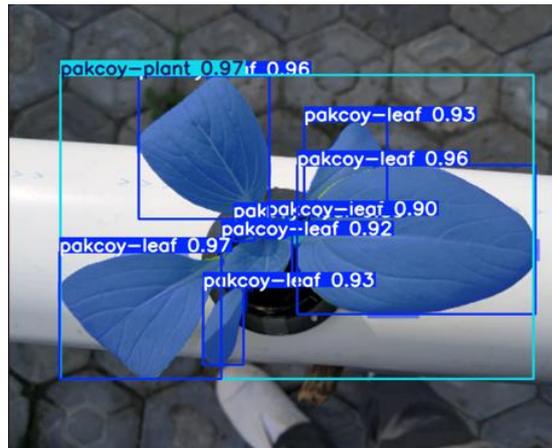
Gambar 4. 14 Program Prediksi Model Training Dataset



Gambar 4. 15 Hasil Prediksi Hidroponik Kailan

Gambar 4.14 merupakan hasil prediksi dari model *YOLOv8* yang sudah di *train* untuk mendeteksi objek daun dan tanaman kailan. Berdasarkan prediksi, model berhasil mengidentifikasi beberapa objek dengan label *kailan-leaf* dan satunya sebagai *kailan-plant*. Setiap objek yang terdeteksi dengan tingkat kepercayaan model terhadap klasifikasi yang diberikan. Misalnya, objek yang terdeteksi sebagai *kailan-plant* memiliki *confidence* sebesar 0,97, sedangkan beberapa objek *kailan-leaf* terdeteksi dengan *confidence* antara 0,86 hingga 0,96. Hal ini menunjukkan bahwa model memiliki tingkat keyakinan tinggi dalam

membedakan antara daun dan tanaman kailan secara visual. Program juga akan di tes pada tanaman pakcoy, hasilnya seperti berikut.

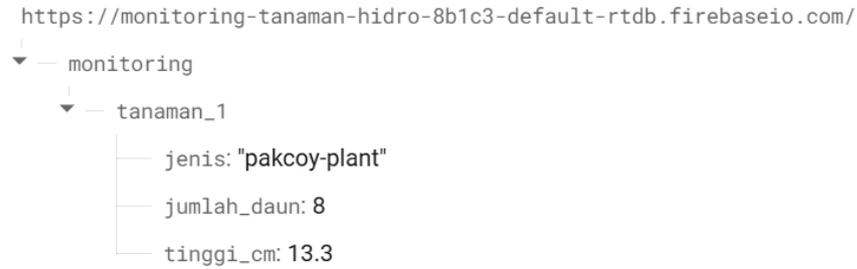


Gambar 4. 16 Hasil Prediksi Pakcoy

Gambar tersebut merupakan hasil deteksi objek menggunakan model *YOLOv8* yang telah dilatih untuk mengenali tanaman pakcoy dan bagiannya, khususnya daun dan keseluruhan tanaman. Terlihat bahwa sistem berhasil mendeteksi objek *pakcoy-plant* dengan tingkat *confidence* sangat tinggi, yaitu 0.97 yang menunjukkan model sangat yakin bahwa objek tersebut adalah tanaman pakcoy. Selain itu, model juga mendeteksi beberapa *pakcoy-leaf* secara akurat dengan *confidence score* berkisar antara 0.90 hingga 0.97. Setiap deteksi ditandai dengan kotak (*bounding box*) berwarna biru untuk daun dan biru muda untuk tanaman secara keseluruhan, serta dilengkapi dengan label dan nilai *confidence*. Gambar ini menunjukkan bahwa model tidak hanya mampu mengenali satu objek secara keseluruhan, namun bisa mendeteksi objek yang lebih spesifik seperti daun pada tanaman pakcoy.

4.3 Pengujian Sistem

Proses selanjutnya yang akan dilakukan yaitu, pengujian sistem monitoring pertumbuhan tanaman dengan sistem *image processing* dan berbasis *IoT*. Pengujian dilakukan untuk mengetahui apakah sistem dapat mendeteksi objek tanaman dengan akurat, mengukur parameter yang diperlukan seperti lebar tanaman dan jumlah daun, serta mengirimkan data ke *platform cloud* (Firebase) secara *real-time* sesuai yang diharapkan. Berikut hasil pengujiannya di Firebase, yaitu:



Gambar 4. 17 Integrasi dengan Firebase

Gambar 4.16 menunjukkan hasil pengujian sistem pada integrasi Firebase *Real-time Database*. Data yang dikirim mencakup informasi jenis tanaman (jenis), jumlah daun (jumlah_daun), dan lebar tanaman dalam satuan sentimeter (cm). Dalam kasus ini, sistem berhasil mendeteksi bahwa tanaman yang diamati adalah *pakcoy-plant*, dengan total 8 daun dan lebar tanaman sebesar 13,3 cm. Hasil ini menunjukkan bahwa sistem telah berhasil menjalankan proses inferensi menggunakan model *YOLOv8* dan mengirimkan hasil deteksi secara otomatis ke Firebase tanpa kesalahan.



Gambar 4. 18 Hasil Pengujian Model

Gambar di atas menunjukkan hasil deteksi sistem terhadap tanaman pakcoy menggunakan model *YOLOv8* yang telah dilatih sebelumnya. Berdasarkan hasil inferensi, sistem berhasil mengklasifikasikan objek sebagai *pakcoy-plant* dengan jumlah daun sebanyak 8 helai dan estimasi lebar tanaman sebesar 13,3 cm. Untuk

memverifikasi akurasi sistem, dilakukan perbandingan terhadap hasil pengukuran manual menggunakan penggaris, yang menunjukkan bahwa tinggi tanaman sebenarnya adalah 13,0 cm. Selisih antara hasil sistem dan pengukuran manual adalah sebesar 0,3 cm, yang berarti menghasilkan persentase error sekitar 2,31% terhadap nilai sebenarnya. Nilai error yang sangat kecil ini menunjukkan bahwa sistem memiliki tingkat akurasi yang tinggi dalam mengestimasi tinggi tanaman, serta dapat diandalkan untuk proses monitoring pertumbuhan tanaman secara otomatis tanpa memerlukan intervensi manusia secara langsung. Selain itu, visualisasi *bounding box* dan informasi teks pada gambar juga memberikan kejelasan terhadap interpretasi hasil yang diperoleh sistem.

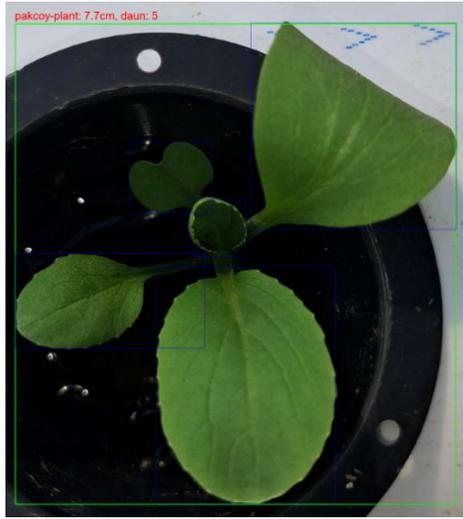
4.3.1 Hasil Pengujian Sistem

Pengujian sistem monitoring hidroponik ini dilakukan pada 5 hari dengan masing masing 5 sampel tanaman pakcoy dan 5 tanaman kailan. Hal ini untuk mengetahui pertumbuhan dari masing masing tanaman hidroponik dalam beberapa rentang waktu. Berikut adalah pengujian sistem pada hari pertama.

Tabel 4. 2 Hasil Pengujian Tanaman Pakcoy Hari Ke-1

No	Hasil Monitoring	Banyak Daun	Lebar Tanaman
1		6	8,5 cm

2



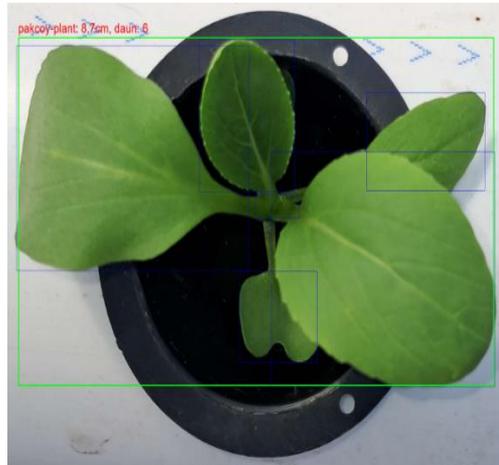
5 7,7 cm

3



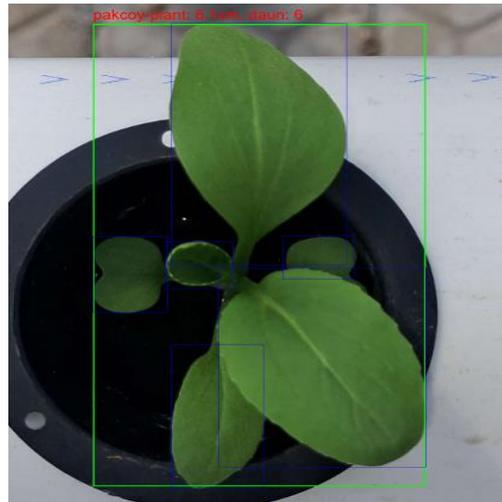
6 8,2 cm

4



6 8,7 cm

5



6 8,1 cm

Tabel 4. 3 Pengambilan Data 1 Tanaman Pakcoy Secara Manual

No	Banyak Daun	Tinggi Tanaman
1	6	8,5 cm
2	5	7,5 cm
3	6	8,2 cm
4	6	8,6 cm
5	6	8,2 cm

Berdasarkan perbandingan tabel 4.2 dan 4.3, dapat dilihat bahwa hasil deteksi menggunakan *YOLOv8* menunjukkan tingkat akurasi yang sangat baik dengan memiliki output yang hampir sempurna. Pada parameter jumlah daun, model *YOLOv8* mampu mendeteksi daun pakcoy dengan sempurna. Untuk parameter tinggi tanaman akan menjadi *benchmark* untuk kalibrasi lebar tanaman, dimana hasil lebar tanaman tidak jauh dengan data tinggi tanaman yang sudah diukur secara manual, perbedaan nilai antara hasil otomatis dan manual sangat kecil, berada pada kisaran 0,1–0,2 cm. Lalu di lanjutkan dengan pengujian di hari berikutnya:

Tabel 4. 4 Hasil Pengujian Pakcoy Hari Ke-2

No	Hasil Monitoring	Banyak Daun	Lebar Tanaman
1		7	10,4 cm
2		7	9,7 cm
3		7	10,3 cm



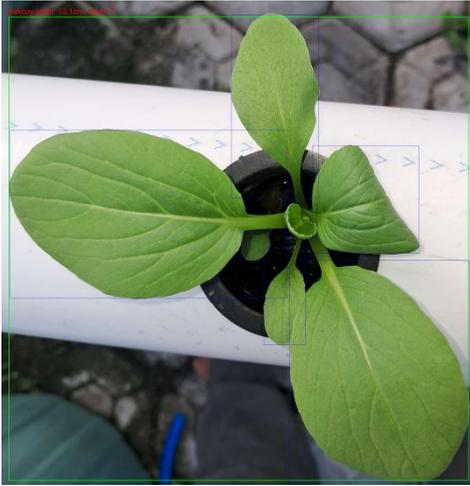
Tabel 4. 5 Pengambilan Data 2 Tanaman Pakcoy Secara Manual

No	Banyak Daun	Tinggi Tanaman
1	7	10,5 cm
2	7	9,5 cm
3	7	10,2 cm
4	7	10,5 cm
5	7	10,2 cm

Tabel 4.4 dan 4.5 menunjukkan model *YOLOv8* memiliki akurasi yang tinggi jika dibandingkan dengan hasil pengukuran manual. Model memiliki konsistensi dalam mendeteksi jumlah daun, yakni 7 helai daun, baik melalui deteksi

otomatis maupun melakukan verifikasi manual. Selain itu rata selisih hanya sekitar 0,16 cm. Hal ini menunjukkan bahwa model memiliki keakuratan yang andal pada saat pengambilan data tanaman pakcoy pada hari ke-2.

Tabel 4. 6 Hasil Pengujian Pakcoy Hari Ke-3

No	Hasil Monitoring	Banyak Daun	Lebar Tanaman
1		7	13,1 cm
2		7	12,7 cm

3



8 13,3 cm

4



8 13,8 cm

5



7 13,2 cm

Tabel 4. 7 Pengambilan Data 3 Tanaman Pakcoy Secara Manual

No	Banyak Daun	Tinggi Tanaman
1	8	13 cm
2	8	12,5 cm
3	8	13,2 cm
4	8	13,5 cm
5	8	13,2 cm

Hasil tabel 4.6 dan 4.7 model masih memiliki akurasi yang cukup tinggi dalam mendeteksi jumlah daun dan mengestimasi lebar tanaman, namun beberapa dari model tidak memberikan nilai jumlah daun yang sama dengan verifikasi manual dikarenakan helai daun yang terlihat dari model hanya 7 saja, sedangkan pada sampel nomor 1,2 dan 5 terdapat helai daun yang tertutup oleh daun pakcoy yang besar.

Tabel 4. 8 Hasil Pengujian Pakcoy Hari Ke-4

No	Hasil Monitoring	Banyak Daun	Lebar Tanaman
1		8	15,7 cm

2



8 14,9 cm

3

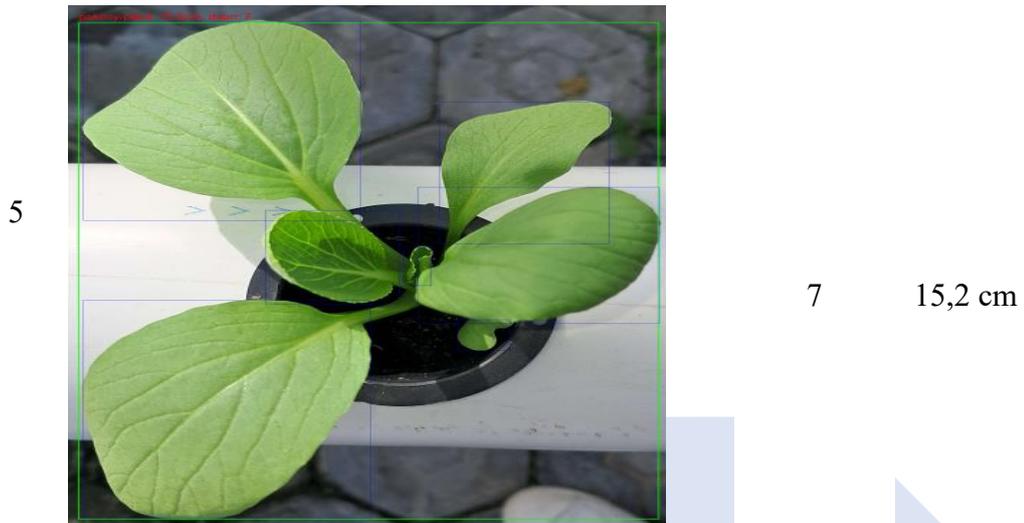


8 15,4 cm

4



7 16,1 cm

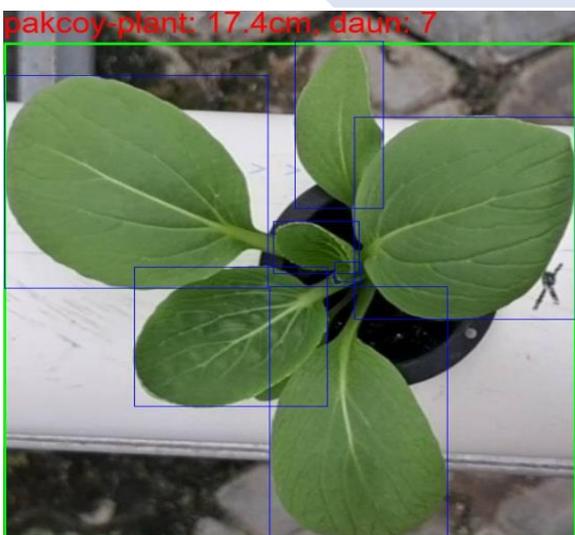


Tabel 4. 9 Pengambilan Data 4 Tanaman Pakcoy Secara Manual

No	Banyak Daun	Tinggi Tanaman
1	9	15,8 cm
2	9	15 cm
3	9	15,2 cm
4	9	16,1 cm
5	9	15,4 cm

Tabel 4.8 dan 4.9 menunjukkan hasil perbandingan pengambilan data dengan model *YOLOv8* dan metode manual, yaitu hasil monitoring tanaman pakcoy menggunakan model *YOLOv8*, dapat disimpulkan bahwa model telah mampu mendeteksi jumlah daun dan lebar tanaman dengan cukup akurat. Meskipun jumlah daun yang terdeteksi oleh model *YOLOv8* sedikit lebih rendah dibandingkan dengan hasil pengukuran manual (7–8 daun terdeteksi vs. 9 daun manual), hal ini disebabkan oleh adanya tumpang tindih dan posisi daun yang saling menutupi, terutama daun bagian bawah yang tersembunyi di balik daun yang lebih besar. Kondisi ini menyulitkan model untuk mendeteksi seluruh daun secara visual dari sudut pandang kamera.

Tabel 4. 10 Hasil Pengujian Pakcoy Hari Ke-5

No	Hasil Monitoring	Banyak Daun	Lebar Tanaman
1		7	18,2 cm
2		9	17,2 cm
3		7	17,4 cm



Tabel 4. 11 Pengambilan Data 5 Tanaman Pakcoy Secara Manual

No	Banyak Daun	Tinggi Tanaman
1	10	18,25 cm
2	10	17,2 cm
3	10	17,4 cm
4	10	18,1 cm
5	10	17,1 cm

Menurut tabel 4.10 dan 4.11 hasil monitoring pertumbuhan tanaman pakcoy menggunakan model *YOLOv8*, dapat disimpulkan bahwa model telah berhasil mendeteksi jumlah daun dan lebar tanaman dengan tingkat akurasi yang cukup

baik. Hasil deteksi menunjukkan jumlah daun yang sedikit lebih rendah dibandingkan dengan pengukuran manual, yaitu berkisar antara 7 hingga 8 daun, sementara pengukuran manual menunjukkan jumlah daun sebanyak 10 pada setiap sampel. Perbedaan ini disebabkan oleh posisi beberapa daun yang tertutup oleh daun yang lebih besar atau saling tumpang tindih, sehingga tidak seluruhnya terdeteksi oleh model *YOLOv8*.

Tabel 4. 12 Hasil Pengujian Kailan Hari Ke-1

No	Hasil Monitoring	Banyak Daun	Lebar Tanaman
1	 <p>Kailan-plant: 7.0cm, daun: 5</p>	5	7 cm
2	 <p>kailan-plant: 7.1cm, daun: 4</p>	4	7,1 cm

3



6 7,5 cm

4



4 7,1 cm

5



2 5,8 cm

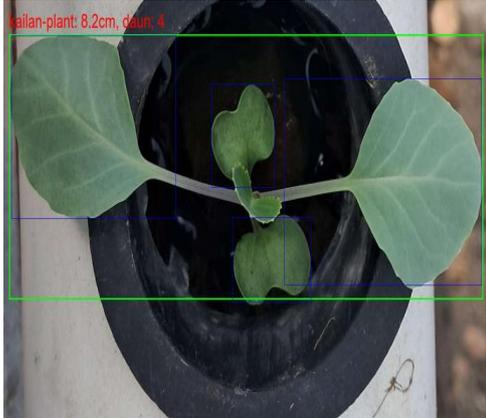
Tabel 4. 13 Pengambilan Data 1 Tanaman Kailan Secara Manual

No	Banyak Daun	Tinggi Tanaman
1	5	7,2 cm
2	5	7,1 cm
3	5	7,8 cm
4	4	7,1 cm
5	4	5,9 cm

Berdasarkan tabel 4.12 dan 4.13 hasil pengujian model *YOLOv8* dalam memantau pertumbuhan tanaman kailan, dapat disimpulkan bahwa model secara umum mampu memberikan hasil prediksi yang cukup akurat terhadap jumlah daun dan tinggi tanaman. Sebagai contoh, pada sampel nomor 1 hingga 4, hasil deteksi *YOLOv8* menunjukkan kesesuaian yang cukup baik dengan hasil pengukuran manual, dengan selisih yang relatif kecil pada lebar tanaman (selisih maksimum 0,3 cm) dan jumlah daun yang hanya berbeda pada beberapa kasus.

Namun, pada sampel nomor 5, model menunjukkan adanya error yang cukup signifikan. Pertama, terjadi kesalahan dalam pendeteksian jumlah daun; model hanya mendeteksi 2 daun, sedangkan hasil manual mencatat 4 daun. Kedua, model mendeteksi dua kelas tanaman secara bersamaan, yaitu *pakcoy-plant* dan *kailan-plant*, padahal tanaman pada sampel tersebut seluruhnya adalah kailan. Kesalahan klasifikasi ini menunjukkan bahwa model masih memiliki kelemahan dalam membedakan antara dua jenis tanaman yang secara visual mungkin memiliki kemiripan bentuk pada fase awal pertumbuhan pada tanaman kailan dan tanaman pakcoy.

Tabel 4. 14 Hasil Pengujian Kailan Hari Ke-2

No	Hasil Monitoring	Banyak Daun	Lebar Tanaman
1	 <p>kailan-plant: 8.0cm, daun: 10</p>	10	8,0 cm
2	 <p>kailan-plant: 7.3cm, daun: 7</p>	7	7,3 cm
3	 <p>kailan-plant: 8.2cm, daun: 4</p>	4	8,2 cm

4



4 7,7 cm

5



4 6,6 cm

Tabel 4. 15 Pengambilan Data 2 Tanaman Kailan Secara Manual

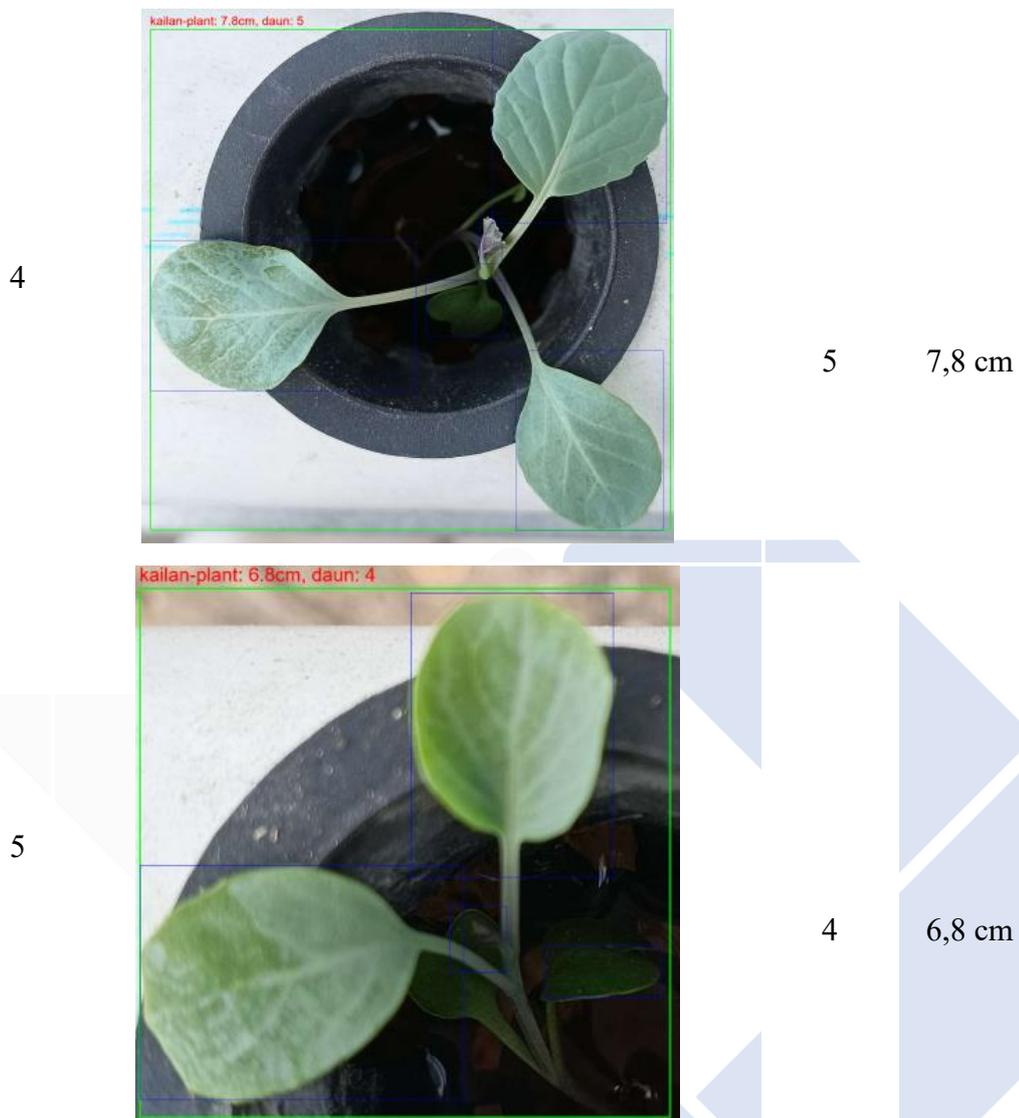
No	Banyak Daun	Tinggi Tanaman
1	5	7,9 cm
2	5	7,3 cm
3	5	8,3 cm
4	5	7,7 cm
5	4	6,4 cm

Berdasarkan tabel 4.14 dan 4.15 hasil perbandingan antara sistem monitoring tanaman kailan menggunakan model *YOLOv8* dan pengukuran manual, ditemukan adanya perbedaan khususnya pada jumlah daun yang terdeteksi. Model *YOLOv8* menunjukkan hasil pendeteksian daun yang sedikit lebih tinggi dibandingkan dengan pengamatan manual, terutama pada tanaman nomor 1 dan 2 yang masing-masing terdeteksi memiliki 7 daun oleh model, sedangkan hasil manual menunjukkan 5 daun. Hal ini disebabkan oleh adanya daun yang tertimpa oleh batang atau objek lain dalam citra sehingga model mendeteksinya sebagai dua objek terpisah.

Dengan demikian, dapat disimpulkan bahwa meskipun model *YOLOv8* sudah cukup baik dalam mendeteksi lebar tanaman (dengan selisih rata-rata yang kecil terhadap pengukuran manual), namun masih terdapat kesalahan minor pada segmentasi daun. Hal ini menjadi dasar bahwa hasil deteksi jumlah daun memerlukan verifikasi lebih lanjut, dengan memperbanyak *dataset* tanaman kailan dan memberikan anotasi yang lebih baik.

Tabel 4. 16 Hasil Pengujian Kailan Hari Ke-3

No	Hasil Monitoring	Banyak Daun	Lebar Tanaman
1		5	8,4 cm
2		6	7,8 cm
3		5	8,9 cm



Tabel 4. 17 Pengambilan Data 3 Tanaman Kailan Secara Manual

No	Banyak Daun	Tinggi Tanaman
1	5	8,4 cm
2	5	7,8 cm
3	5	8,8 cm
4	5	7,9 cm
5	4	6,7 cm

Berdasarkan tabel 4.16 dan 4.17 hasil monitoring tanaman kailan menggunakan model *YOLOv8*, diperoleh data jumlah daun dan lebar tanaman yang secara umum mendekati hasil pengukuran manual. Namun, terdapat perbedaan

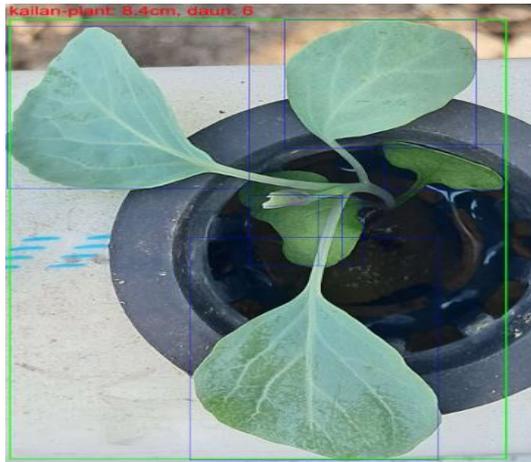
kecil terutama pada deteksi jumlah daun. Sebagai contoh, pada sampel kedua, model *YOLOv8* mendeteksi 6 daun, sementara hasil pengukuran manual menunjukkan 5 daun. Setelah dilakukan peninjauan terhadap citra hasil deteksi, diketahui bahwa terdapat satu daun yang posisinya saling tumpang tindih dengan batang tanaman, sehingga terdeteksi sebagai dua objek daun oleh model.

Hal ini menunjukkan bahwa meskipun model *YOLOv8* mampu melakukan deteksi secara akurat untuk lebar tanaman (dengan error sangat kecil, ± 0.1 cm), deteksi jumlah daun masih memiliki potensi error akibat kondisi visual seperti tumpang tindih daun atau pencahayaan. Oleh karena itu, akurasi model dalam mendeteksi jumlah daun dapat ditingkatkan dengan pengolahan data latih yang lebih beragam atau teknik pasca-pemrosesan tambahan untuk menyaring duplikasi objek deteksi.

Tabel 4. 18 Hasil Pengujian Kailan Hari Ke-4

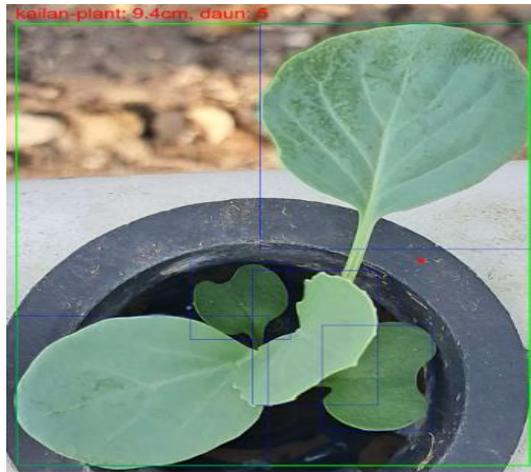
No	Hasil Monitoring	Banyak Daun	Lebar Tanaman
1		4	8,5 cm

2



6 8,4 cm

3



5 9,4 cm

4



5 8,4 cm

5



3

5,3 cm

Tabel 4. 19 Pengambilan Data 4 Tanaman Kailan Secara Manual

No	Banyak Daun	Tinggi Tanaman
1	6	9,1 cm
2	6	8,5 cm
3	5	9,5 cm
4	6	8,2 cm
5	4	7,4 cm

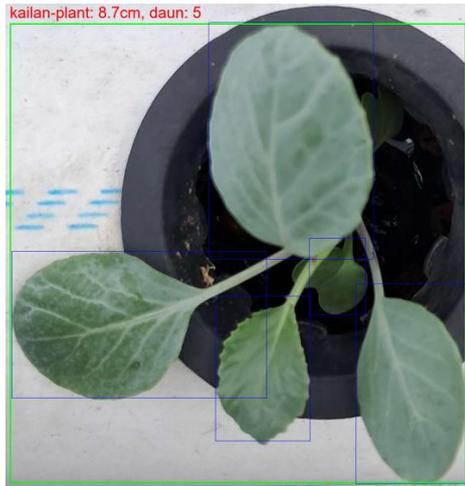
Berdasarkan tabel 4.18 dan 4.19 hasil monitoring tanaman kailan menggunakan model *YOLOv8*, secara umum sistem telah mampu melakukan deteksi jumlah daun dan lebar tanaman dengan cukup baik. Namun, terdapat beberapa perbedaan hasil dibandingkan pengukuran manual yang menunjukkan adanya error dalam proses deteksi.

Pada sampel ke-1 dan ke-4, model *YOLOv8* mengalami kesalahan dalam mendeteksi jumlah daun. Hal ini disebabkan oleh adanya tumpang tindih antar daun, yang menyebabkan sebagian daun tidak terdeteksi dengan baik oleh model. Selain itu, pada sampel ke-5, terjadi error yang tidak diketahui penyebab pastinya dalam deteksi jumlah daun. Di sisi lain, terdapat pula kesalahan signifikan dalam pengukuran lebar tanaman pada sampel ke-5, yang disebabkan oleh posisi *rockwool* yang lebih rendah dari *netpot*, sehingga membuat sistem pengukuran visual mendeteksi lebar tanaman yang lebih kecil dari sebenarnya.

Tabel 4. 20 Hasil Pengujian Kailan Hari Ke-5

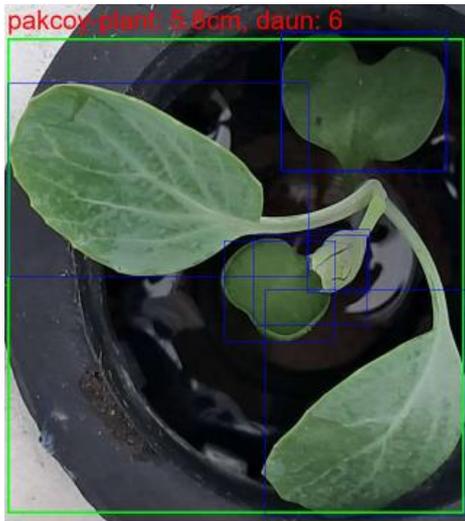
No	Hasil Monitoring	Banyak Daun	Lebar Tanaman
1		7	9,9 cm
2		6	9,2 cm
3		6	10 cm

4



5 8,7 cm

5



6 5,8 cm

Tabel 4. 21 Pengambilan Data 5 Tanaman Kailan Secara Manual

No	Banyak Daun	Tinggi Tanaman
1	6	9,7 cm
2	6	9,1 cm
3	5	10 cm
4	6	8,7 cm
5	5	8 cm

Berdasarkan tabel 4.20 dan 4.21 hasil monitoring menggunakan model *YOLOv8*, ditemukan bahwa sistem mampu mendeteksi jumlah daun dan lebar tanaman kailan dengan tingkat akurasi yang cukup baik, meskipun terdapat beberapa perbedaan dibandingkan dengan hasil pengukuran manual. Perbedaan ini dapat dijelaskan melalui beberapa kondisi teknis pada citra tanaman.

Pada sampel ke-1 dan ke-3, model *YOLOv8* mendeteksi jumlah daun lebih banyak dibandingkan pengamatan manual. Hal ini disebabkan oleh adanya tumpang tindih antara daun dan batang, yang menyebabkan satu helai daun dihitung sebagai dua objek terpisah oleh model. Sementara itu, pada sampel ke-4, terjadi kesalahan deteksi di mana daun tertutup sepenuhnya akibat sudut pengambilan gambar, sehingga model gagal mengenali salah satu daun. Selain itu, pengukuran tinggi pada sampel ke-5 menunjukkan hasil signifikan karena posisi media tanam (*rockwool*) berada jauh di bawah permukaan *netpot*, menyebabkan referensi titik awal pengukuran oleh model menjadi tidak sesuai dengan tinggi sebenarnya yang diukur secara manual.

Berdasarkan hasil uji coba yang telah dilakukan, sistem monitoring pertumbuhan tanaman hidroponik menggunakan *image processing* sudah mampu memprediksi lebar tanaman dan jumlah daun dengan cukup baik pada sebagian besar data uji. Dari tabel hasil prediksi dan akurasi, rata-rata hasil model mendekati data asli, meskipun masih ada beberapa prediksi yang meleset satu atau dua daun, atau berbeda beberapa sentimeter pada lebar tanaman. Model *YOLO* yang digunakan dalam sistem ini berhasil mencapai akurasi deteksi sekitar 90% ke atas, berdasarkan hasil pengujian pada *dataset* uji yang telah dilakukan. Hal ini dapat

dilihat dari nilai akurasi, presisi, dan *recall* yang tercantum pada tabel evaluasi dan *confusion matrix*, yang menunjukkan bahwa model mampu mengenali objek tanaman dengan tingkat kesalahan yang rendah. Akurasi tertinggi diperoleh pada gambar dengan kualitas baik dan tanaman yang tidak tertutup daun lain. Sebaliknya, pada gambar buram atau saat daun saling menutupi, prediksi jumlah daun cenderung kurang tepat. Kesalahan ini umumnya disebabkan oleh pencahayaan, posisi kamera, dan adanya objek lain di sekitar tanaman. Selain itu, selama proses pengambilan gambar, terdapat daun yang tidak tertangkap kamera sehingga tidak terhitung saat pemrosesan. Hal ini terjadi karena jumlah daun yang mulai banyak dan besar, serta beberapa daun tertutup oleh daun lain, sehingga sulit dideteksi secara visual oleh sistem.

Secara keseluruhan, sistem sudah berjalan sesuai harapan untuk tahap awal dengan tingkat error yang masih dapat diterima. Namun, perbaikan pada kualitas gambar, variasi data latih, dan pengaturan parameter model sangat diperlukan agar hasil prediksi menjadi lebih akurat dan konsisten di berbagai kondisi gambar.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil penelitian dan pengujian yang telah dilakukan, maka dapat diambil kesimpulan sebagai berikut:

1. Sistem monitoring pertumbuhan tanaman hidroponik berbasis *image processing* dengan model YOLO berhasil dikembangkan dan berjalan secara *real-time*, memudahkan pemantauan secara langsung dan efisien.
2. Sistem mampu mendeteksi lebar tanaman dan jumlah daun secara otomatis dengan tingkat presisi dan *recall* yang cukup tinggi berdasarkan evaluasi *confusion matrix*.
3. Kualitas hasil monitoring dipengaruhi oleh pencahayaan, posisi kamera, kualitas gambar, dan tumpang tindih daun, serta pengaturan parameter dan *preprocessing* gambar.
4. Pemilihan sensor kamera dan perangkat keras yang sesuai sangat berpengaruh terhadap kinerja sistem, khususnya saat proses *training*.
5. Sistem telah terintegrasi dengan teknologi IoT melalui Firebase, memungkinkan akses data secara *real-time* dari jarak jauh.
6. Data latih sudah cukup membantu model belajar, namun peningkatan variasi data tetap diperlukan untuk akurasi yang lebih konsisten.

5.2 Saran

Berdasarkan hasil penelitian dan kesimpulan yang diperoleh, berikut beberapa saran yang dapat diberikan untuk pengembangan dan perbaikan sistem ke depannya:

1. Perbanyak variasi data latih dengan menambahkan gambar tanaman dalam berbagai kondisi, seperti ukuran daun yang berbeda, sudut pengambilan gambar yang beragam.

2. Lakukan pengujian lanjutan dengan data yang lebih banyak dan beragam untuk memastikan sistem dapat bekerja dengan baik dalam berbagai situasi nyata.
3. Pertimbangkan pengembangan sistem monitoring secara *real-time* yang terintegrasi dengan perangkat keras seperti kamera dan Raspberry Pi untuk aplikasi langsung di lapangan.



DAFTAR PUSTAKA

- [1] a. B. Kaswar, “sistem pendeteksi objek tanaman selada hidroponik dalam netpot menggunakan metode segmentasi otsu thresholding disertai operasi morfologi,” vol. 6, no. 1, 2023.
- [2] a. & meliana, “rancang bangun sistem kontrol nutrisi tanaman hidroponik berbasis internet of things skripsi,” 2023.
- [3] m. Rio *et al.*, “review: mobile-based hydroponic plant monitoring with iot mode and image processing-based plant pest detection,” 2021. [online]. Available: www.psychologyandeducation.net
- [4] r. I. Ramadhan, h. Fitriyah, and e. R. Widasari, “sistem deteksi daun busuk pada pakcoy hidroponik menggunakan metode thresholding pada warna hue dan saturasi berbasis raspberry pi,” 2023. [online]. Available: <http://j-ptiik.ub.ac.id>
- [5] y. Suharto, h. Suhardiyanto, and a. Susila, “pengembangan sistem hidroponik untuk budidaya tanaman kentang (*solanum tuberosum* l.),” *j. Keteknikan pertan.*, vol. 04, no. 2, pp. 1–8, oct. 2016, doi: 10.19028/jtep.04.2.211-218.
- [6] d. R. Susilawati and m. Si, *dasar-dasar bertanam secara hidroponik*. [online]. Available: www.unsri.unsripress.ac.id
- [7] heru agus hendra & agus andoko, *bertanam sayuran ala paktani hydrofoam*. 2014. [online]. Available: [https://books.google.co.id/books?id=e6fmbgaaqbaj&lpg=pa2&ots=l9htnh_c8i&dq=bertanam sayuran hidroponik ala paktani hydrofarm&lr&pg=pr2#v=onepage&q=bertanam sayuran hidroponik ala paktani hydrofarm&f=false](https://books.google.co.id/books?id=e6fmbgaaqbaj&lpg=pa2&ots=l9htnh_c8i&dq=bertanam%20sayuran%20hidroponik%20ala%20paktani%20hydrofarm&lr&pg=pr2#v=onepage&q=bertanam%20sayuran%20hidroponik%20ala%20paktani%20hydrofarm&f=false)
- [8] s. Tembe, s. Khan, and r. Acharekar, “iot based automated hydroponics system,” *int. J. Sci. Eng. Res.*, 2018, [online]. Available: <http://www.ijser.org>
- [9] a. J. Rozaqi, a. Sunyoto, and r. Arief, “deteksi penyakit pada daun kentang

menggunakan pengolahan citra dengan metode convolutional neural network detection of potato leaves disease using image processing with convolutional neural network methods”.

- [10] dan r. S. I wayan suartika e. P, arya yudhi wijaya and teknik, “□,” *j. Tek. Its*, vol. 5, 2016.
- [11] a. Ahmad, “mengenal artificial intelligence, machine learning, neural network, dan deep learning.” [online]. Available: www.teknoindonesia.com
- [12] m. Sobron *et al.*, *implementasi artificial intelligence pada system manufaktur terpadu*.
- [13] b. A. Khalaf, s. A. Mostafa, a. Mustapha, m. A. Mohammed, and w. M. Abdulllah, “comprehensive review of artificial intelligence and statistical approaches in distributed denial of service attack and defense methods,” *iee access*, vol. 7, pp. 51691–51713, 2019, doi: 10.1109/access.2019.2908998.
- [14] j. Homepage, m. A. Hidayat, n. Latifah husni, and f. Damsi, “malcom: indonesian journal of machine learning and computer science image processing based flood detector using convolutional neural network (cnn) within surveillance camera pendeteksi banjir dengan image processing berbasis convolutional neural network (cnn) pada kamera pengawas,” vol. 2, pp. 10–18, 2022.
- [15] a. Raup, w. Ridwan, y. Khoeriyah, q. Yuliati zaqiah, and u. Islam negeri sunan gunung djati bandung, “deep learning dan penerapannya dalam pembelajaran.” [online]. Available: <http://jiip.stkipyapisdompu.ac.id>
- [16] s. Tufail, h. Riggs, m. Tariq, and a. I. Sarwat, “advancements and challenges in machine learning: a comprehensive review of models, libraries, applications, and algorithms,” apr. 01, 2023, *mdpi*. Doi: 10.3390/electronics12081789.
- [17] s. Ilahiyah and a. Nilogiri, “implementasi deep learning pada identifikasi jenis tumbuhan berdasarkan citra daun menggunakan convolutional neural network”.

- [18] t. Perumal, n. Mustapha, r. Mohamed, and f. M. Shiri, “a comprehensive overview and comparative analysis on deep learning models,” *j. Artif. Intell.*, vol. 6, no. 1, pp. 301–360, 2024, doi: 10.32604/jai.2024.054314.
- [19] l. Alzubaidi *et al.*, “review of deep learning: concepts, cnn architectures, challenges, applications, future directions,” *j. Big data*, vol. 8, no. 1, dec. 2021, doi: 10.1186/s40537-021-00444-8.
- [20] x. Zhao, l. Wang, y. Zhang, x. Han, m. Deveci, and m. Parmar, “a review of convolutional neural networks in computer vision,” *artif. Intell. Rev.*, vol. 57, no. 4, apr. 2024, doi: 10.1007/s10462-024-10721-6.
- [21] j. Lee and k. Il hwang, “yolo with adaptive frame control for real-time object detection applications,” *multimed. Tools appl.*, vol. 81, no. 25, pp. 36375–36396, oct. 2022, doi: 10.1007/s11042-021-11480-0.
- [22] r. Bree and g. Gallagher, “using microsoft excel to code and thematically analyse qualitative data: a simple, cost-effective approach. * all ireland journal of teaching and learning in higher education (aishe-j) creative commons attribution-noncommercial-sharealike 3.0,” 2016. [online]. Available: [http://ojs.aishe.org/index.php/aishe-j/article/view/\[281\]](http://ojs.aishe.org/index.php/aishe-j/article/view/[281])
- [23] m. A. Kabir, f. Ahmed, m. M. Islam, and m. R. Ahmed, “python for data analytics: a systematic literature review of tools, techniques, and applications,” *acad. J. Sci. Technol. Eng. Math. Educ.*, vol. 4, no. 04, pp. 134–154, nov. 2024, doi: 10.69593/ajsteme.v4i04.146.
- [24] v. K. Patil, g. Suresh birajdar, s. M. Chaudhari, and a. M. Gandhamal, “comparative approach for face detection in python, opencv and hardware,” 2023. [online]. Available: www.jetir.org
- [25] s. E. Mathe, h. K. Kondaveeti, s. Vappangi, s. D. Vanambathina, and n. K. Kumaravelu, “a comprehensive review on applications of g,” may 01, 2024, *elsevier ireland ltd*. Doi: 10.1016/j.cosrev.2024.100636.
- [26] f. B. Setiawan, h. W. Kusuma, s. Riyadi, and leonardus heru pratomo, “penerapan pi cam menggunakan program berbasis raspberry pi 4,” *cyclotr*.

J. Tek. Elektro, vol. 5, no. 2, pp. 51–56, 2022.

- [27] “laporan proyek akhir pendeteksi jentik nyamuk dalam penampungan air dengan image processing scanning berbasis iot.pdf.crdownload.”
- [28] r. Gelar guntara, “pemanfaatan google colab untuk aplikasi pendeteksian masker wajah menggunakan algoritma deep learning yolov7,” *j. Teknol. Dan sist. Inf. Bisnis*, vol. 5, no. 1, pp. 55–60, feb. 2023, doi: 10.47233/jteksis.v5i1.750.
- [29] a. Khanna and s. Kaur, “internet of things (iot), applications and challenges: a comprehensive review,” sep. 01, 2020, *springer*. Doi: 10.1007/s11277-020-07446-4.
- [30] d. S. Patel and h. Shastri, “automatic hydroponics farming system with image processing based smart nutrients system,” *int. Res. J. Eng. Technol.*, no. July, 2023, doi: 10.13140/rg.2.2.35273.31840.



Lampiran 1
Daftar Riwayat Hidup

DAFTAR RIWAYAT HIDUP

1. Data Pribadi

Nama Lengkap : Roman Sabda Agung
Tempat & tanggal lahir : Belitung, 25 Desember 2001
Alamat rumah : Dsn Lipat Kajang II,
Kel/Ds. Baru, Kec. Manggar,
Kab. Bangka, Provinsi Kep. Bangka
Belitung
Telp : -
Hp: 081993619951
Email: Romansabdagung@gmail.com

Jenis Kelamin : Laki-laki
Agama : Islam



2. Riwayat Pendidikan

- SD Negeri 2 Manggar 2007 - 2013
- SMP Negeri 1 Manggar 2013 - 2016
- SMA Negeri 1 Manggar 2016 - 2019

3. Pendidikan Non-Formal

Magang Mandiri (MBKM) di PT Varotech Indonesia 2024.

Sungailiat, 15 Juli 2025

Roman Sabda Agung

DAFTAR RIWAYAT HIDUP

1. Data Pribadi

Nama Lengkap : Sabihisma Rabika Wahada
Tempat & tanggal lahir : Belinyu, 15 Juli 2004
Alamat rumah : Jl. Kp. Bukit Tani,
Kel. Belinyu, Kec. Belinyu,
Kab. Bangka, Provinsi Kep. Bangka
Belitung
Telp. : -
Hp : 081248103815
Email : Sabihismarahada@gmail.com
Jenis Kelamin : Perempuan
Agama : Islam



2. Riwayat Pendidikan

- SD Negeri 12 Belinyu 2010 - 2016`
- SMP Negeri 1 Belinyu 2016 - 2019
- SMA Negeri 1 Belinyu 2019 – 2022

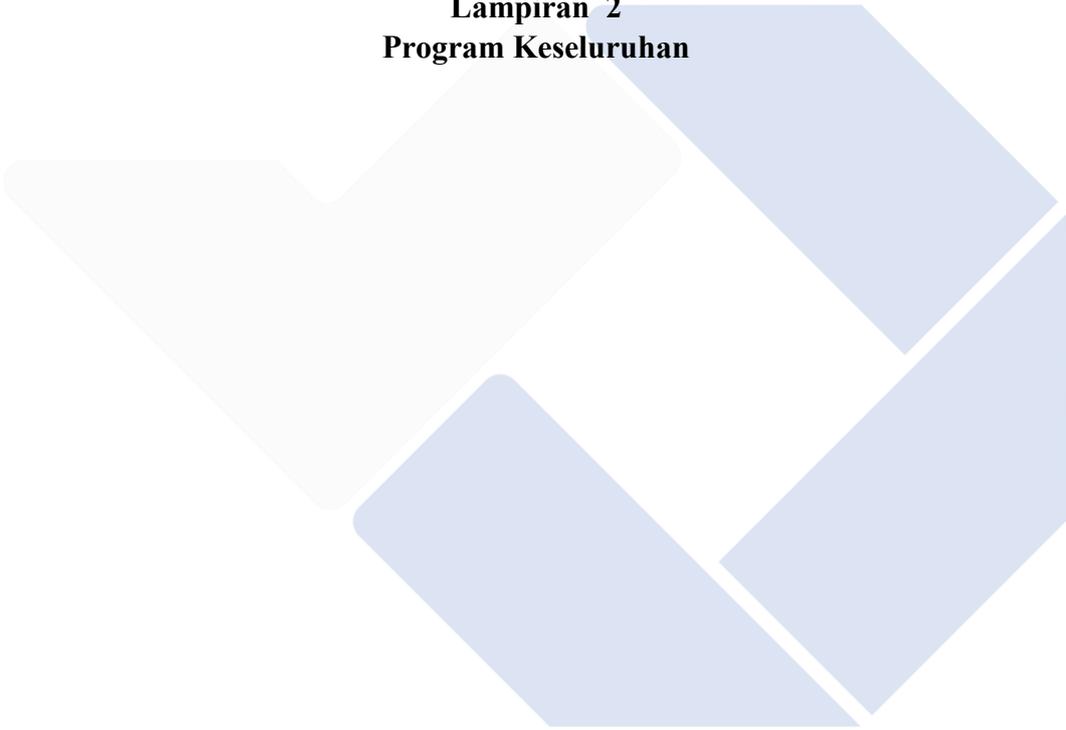
3. Pendidikan Non-Formal

Magang Mandiri (MBKM) di PT PLN (Persero) ULP Sungailiat 2024.

Sungailiat, 15 Juli 2025

Sabihisma Rabika Wahada

Lampiran 2
Program Keseluruhan



Kode Program Training Dataset Model YOLOv8 Google Colab

```
# ----- MOUNT GOOGLE DRIVE -----  
from google.colab import drive  
drive.mount('/content/gdrive')  
  
# ----- LIST DATASET -----  
!ls '/content/gdrive/My Drive/Finalproyek/Data/'  
  
# ----- SET PATH DATA -----  
DATA_DIR = '/content/gdrive/My Drive/Finalproyek/Data/'  
  
# ----- INSTALL ULTRALYTICS -----  
!pip install ultralytics  
  
# ----- IMPORT DAN TRAINING -----  
import os  
from ultralytics import YOLO  
model = YOLO('yolov8n-seg.pt')  
model.train(data='/content/gdrive/My Drive/Finalproyek/Data/data.yaml', epochs=200,  
imgsz=640)  
  
# ----- SALIN HASIL TRAINING -----  
!scp -r /content/runs '/content/gdrive/My Drive/Finalproyek/'  
-----
```

Kode Monitoring Tanaman di Raspberry Pi 4 Model B

```
import cv2  
from ultralytics import YOLO  
import numpy as np  
import firebase_admin  
from firebase_admin import credentials, db  
from PIL import Image, ImageDraw, ImageFont  
from time import sleep  
  
# === Firebase setup ===  
cred = credentials.Certificate("monitoring-tanaman-hidro-8b1c3-firebase-adminsdk-fbsvc.json")  
firebase_admin.initialize_app(cred, {  
    'databaseURL': 'https://monitoring-tanaman-hidro-8b1c3-default-rtdb.firebaseio.com/'  
})  
  
# === Load YOLOv8 model ===  
# ----- LOAD MODEL -----  
model = YOLO("best.pt")  
names = model.names
```

```

PLANT_CLASS_IDS = [1, 3]
LEAF_CLASS_IDS = [0, 2]

# === Kalibrasi pixel per cm ===
pixel_per_cm = 50.00

# === Load font untuk PIL ===
font_path = "/usr/share/fonts/truetype/dejavu/DejaVuSans-Bold.ttf"
font = ImageFont.truetype(font_path, 28)

# === Webcam setup ===
cap = cv2.VideoCapture(0)
if not cap.isOpened():
    raise RuntimeError("Webcam tidak terdeteksi!")

print("Mulai monitoring... Tekan 'q' untuk keluar.")

while True:
    ret, frame = cap.read()
    if not ret:
        print("Gagal mengambil gambar.")
        break

    results = model(frame, imsz=640)[0]
    boxes = results.boxes
    plants, leaves = [], []

    for box in boxes:
        cls_id = int(box.cls[0])
        x1, y1, x2, y2 = map(int, box.xyxy[0])
        if cls_id in PLANT_CLASS_IDS:
            plants.append((cls_id, (x1, y1, x2, y2)))
        elif cls_id in LEAF_CLASS_IDS:
            leaves.append((cls_id, (x1, y1, x2, y2)))

    # ----- KONVERSI KE PIL -----
    image_pil = Image.fromarray(cv2.cvtColor(frame, cv2.COLOR_BGR2RGB))
    draw = ImageDraw.Draw(image_pil)

    # === Fungsi IoU ===
    def compute_iou(boxA, boxB):
        ax1, ay1, ax2, ay2 = boxA
        bx1, by1, bx2, by2 = boxB
        x_left = max(ax1, bx1)
        y_top = max(ay1, by1)
        x_right = min(ax2, bx2)
        y_bottom = min(ay2, by2)
        if x_right < x_left or y_bottom < y_top:
            return 0.0
        inter_area = (x_right - x_left) * (y_bottom - y_top)
        union_area = ((ax2 - ax1) * (ay2 - ay1)) + ((bx2 - bx1) * (by2 - by1)) - inter_area
        return inter_area / union_area

```

```

for idx, (plant_cls, plant_box) in enumerate(plants, start=1):
    x1, y1, x2, y2 = plant_box
    height_px = y2 - y1
    height_cm = height_px / pixel_per_cm
    leaf_count = 0

    for leaf_cls, leaf_box in leaves:
        iou = compute_iou(leaf_box, plant_box)
        if iou > 0.0:
            if (plant_cls == 1 and leaf_cls == 0) or (plant_cls == 3 and leaf_cls == 2):
                leaf_count += 1

    label = f"{names[plant_cls]}: {height_cm:.1f}cm, daun: {leaf_count}"
    draw.rectangle([x1, y1, x2, y2], outline=(0, 255, 0), width=3)
    draw.text((x1, y1 - 30), label, font=font, fill=(255, 0, 0))

    data = {
        "jenis": names[plant_cls],
        "tinggi_cm": round(height_cm, 2),
        "jumlah_daun": leaf_count
    }
    db.reference(f'monitoring/tanaman_{idx}').set(data)

# ----- TAMPILKAN HASIL -----
frame_annotated = cv2.cvtColor(np.array(image_pil), cv2.COLOR_RGB2BGR)
cv2.imshow("Monitoring Tanaman", frame_annotated)

if cv2.waitKey(1) & 0xFF == ord('q'):
    break

sleep(0.2)

cap.release()
cv2.destroyAllWindows()
-----

```

Top Sources

- 12% Internet sources
- 6% Publications
- 6% Submitted works (Student Papers)

Top Sources

The sources with the highest number of matches within the submission. Overlapping sources will not be displayed.

1	Internet	journal.um-surabaya.ac.id	<1%
2	Internet	eprints3.upgris.ac.id	<1%
3	Internet	ojs.unida.ac.id	<1%
4	Internet	manuals.plus	<1%
5	Publication	Bagus Prasetyo Nugroho, Yani Prihati, Sinta Tridian Galih. "Implementasi Algorit...	<1%
6	Internet	www.researchgate.net	<1%
7	Publication	Ananda Rizki Dani, Irma Handayani. "Classification of Yogyakarta Batik Motifs Usi...	<1%
8	Student papers	Universitas Maritim Raja Ali Haji	<1%
9	Internet	kc.umh.ac.id	<1%
10	Internet	www.perioptix.com	<1%
11	Internet	www.arduino.biz.id	<1%

12	Student papers	Telkom University	<1%
13	Internet	repositori.usu.ac.id	<1%
14	Internet	ejournal.gunadarma.ac.id	<1%
15	Internet	ejournal.methodist.ac.id	<1%
16	Internet	claudkurtz.blogspot.com	<1%
17	Internet	ejournal.unma.ac.id	<1%
18	Internet	ejournal.unuja.ac.id	<1%
19	Internet	ecampus.iainbatusangkar.ac.id	<1%
20	Student papers	Sriwijaya University	<1%
21	Internet	www.scilit.net	<1%
22	Internet	www.coursehero.com	<1%
23	Student papers	Konsorsium Turnitin Relawan Jurnal Indonesia	<1%
24	Internet	ejournal.itn.ac.id	<1%
25	Internet	eprints.poltektegal.ac.id	<1%

26	Publication	Widyawati Widyawati, Rafli Sidik, Ely Nuryani, Persis Haryo Winasis. "RANCANG B...	<1%
27	Internet	ojs.unh.ac.id	<1%
28	Student papers	itera	<1%
29	Student papers	Universitas Bengkulu	<1%
30	Internet	prosidingfrima.digitechuniversity.ac.id	<1%
31	Internet	tvsatelite.net	<1%
32	Internet	www.neoteknologi.com	<1%
33	Publication	Wahyu Arini. "TINGKAT DAYA KAPILARITAS JENIS SUMBU PADA HIDROPONIK SIST...	<1%
34	Internet	www.liputan6.com	<1%
35	Student papers	Forum Perpustakaan Perguruan Tinggi Indonesia Jawa Timur	<1%
36	Student papers	Institut Bisnis dan Teknologi Indonesia (INSTIKI)	<1%
37	Student papers	Universitas Merdeka Malang	<1%
38	Student papers	Ursuline Academy High School	<1%
39	Internet	picgarut.id	<1%

40	Student papers	Mount Saint Dominic Academy	<1%
41	Publication	Taufiqurrahman Taufiqurrahman, Aji Prasetya Hadi, Rully Emirza Siregar. "Evalua...	<1%
42	Student papers	Universitas Sumatera Utara	<1%
43	Student papers	Universiti Malaysia Pahang	<1%
44	Internet	id.howtodogood.com	<1%
45	Internet	suhuandroid.id	<1%
46	Internet	telematicswire.net	<1%
47	Internet	www.termpaperwarehouse.com	<1%
48	Internet	artikelpendidikan.id	<1%
49	Internet	cybersulutnews.com	<1%
50	Internet	judul-skripsi-perpustakaan.blogspot.com	<1%
51	Internet	jurnal.untan.ac.id	<1%
52	Internet	ojs.amikomsolo.ac.id	<1%
53	Internet	repository.pnj.ac.id	<1%

54	Publication	Benedika Ferdian Hutabarat, Mardian Peslinof, M. Ficky Afrianto, Yoza Fendriani. ...	<1%
55	Publication	Juanto Simangunsong, Nurmala Dewi Simanjuntak, Aprima Anugerah Matondan...	<1%
56	Internet	be.shaip.com	<1%
57	Internet	ejournal.atmajaya.ac.id	<1%
58	Internet	journal.budiluhur.ac.id	<1%
59	Internet	tekno.kompas.com	<1%
60	Internet	www.scribd.com	<1%
61	Internet	www.sherlocked.org	<1%
62	Publication	Elfira Aulia Septrian, Erna Zuni Astuti. "Application of Random Forest Method for ...	<1%
63	Internet	berkahkhair.com	<1%
64	Internet	binus.ac.id	<1%
65	Internet	digilib.uinsby.ac.id	<1%
66	Internet	eprints.undip.ac.id	<1%
67	Internet	fuadh.gitbooks.io	<1%

68	Internet	id.123dok.com	<1%
69	Internet	journal.nurulfikri.ac.id	<1%
70	Internet	repository.upi.edu	<1%
71	Internet	sipeg.unj.ac.id	<1%
72	Internet	sistemasi.ftik.unisi.ac.id	<1%
73	Internet	teknologipintar.org	<1%
74	Internet	text-id.123dok.com	<1%
75	Internet	www.frontiersin.org	<1%
76	Internet	www.matthies-sport.de	<1%
77	Publication	Afriani Afriani, Herry Sujaini, Niken Candraningrum. "Analisis Perbandingan Met...	<1%
78	Publication	Catharina Andriyani Prasetyawati, Heri Suryanto. "Kualitas tanah dan pertumbuh...	<1%
79	Student papers	INTI Universal Holdings SDM BHD	<1%
80	Publication	Rindi Kusumawardani, Putu Dana Karningsih. "Detection and Classification of Ca...	<1%
81	Internet	chipset.fti.unand.ac.id	<1%

82	Internet	conference.upnvj.ac.id	<1%
83	Internet	core.ac.uk	<1%
84	Internet	eprints.utm.my	<1%
85	Internet	etheses.uin-malang.ac.id	<1%
86	Internet	halosis.asia	<1%
87	Internet	jurnal.wicida.ac.id	<1%
88	Internet	kkn.unnes.ac.id	<1%
89	Internet	linuxerindonesia.org	<1%
90	Internet	media.neliti.com	<1%
91	Internet	repository.its.ac.id	<1%
92	Internet	tambahpinter.com	<1%
93	Internet	toffeev.com	<1%
94	Internet	widuri.raharja.info	<1%
95	Internet	www.it-newbie.com	<1%

96	Internet	www.slideshare.net	<1%
97	Publication	Nurul Tazkiyah Adam, Zahra Arwananing Tyas, Tikaridha Hardiani. "Deteksi Gest..."	<1%
98	Internet	ejurnal.its.ac.id	<1%
99	Publication	Lilian Safitri, Sophia Angelina Pakpahan, Yunindah Lestari Lapihu. "Analisis Usah..."	<1%
100	Publication	Niken Zeliana Putri, Martanto Martanto, Arif Rinaldi Dikananda, Ahmad Rifa'i. "Se..."	<1%
101	Internet	doku.pub	<1%
102	Internet	id.scribd.com	<1%

SURAT PERNYATAAN

Saya/Kami yang bertandatangan dibawah ini telah menyelesaikan Proyek Akhir yang berjudul:
Rancang Bangun Monitoring Hidroponik dengan Sistem Image Processing Berbasis IoT

Oleh :

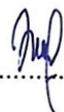
1. Roman Sabda Agung /NPM 1052255
2. Sabihisma Rabika Wahada /NPM 1052256

Dengan ini menyatakan bahwa isi laporan akhir proyek akhir sama dengan *hardcopy*.

Demikian surat pernyataan ini dibuat dengan sebenar-benarnya.

Sungailiat, 15 September 2025

1. Roman Sabda Agung ()

2. Sabihisma Rabika Wahada ()

Mengetahui,

Pembimbing 1,



(Aan Febriansyah, S.ST., M.T.)

Pembimbing 2,



(Badriyah, S.P., M.Si.)