

Sistem Pendeteksi Hama Ulat pada Tanaman Pakcoy untuk Lahan Hidroponik Berbasis Raspberry Pi dengan *Object Detection*

Mutia Sari Zulvi*¹, M. Suhairi², Sukaidah³, Wira Indani⁴

¹Jurusan Teknologi Informasi, Politeknik Caltex Riau, Pekanbaru, 28265, Indonesia

²Jurusan Teknologi Industri, Politeknik Caltex Riau, Pekanbaru, 28265, Indonesia

³Jurusan Teknologi Industri, Politeknik Caltex Riau, Pekanbaru, 28265, Indonesia

⁴Jurusan Teknologi Industri, Politeknik Caltex Riau, Pekanbaru, 28265, Indonesia

*Corresponding Author: mutia@pcr.ac.id

Riwayat Artikel

Diserahkan: 27 Agustus 2024

Direvisi: 13 Februari 2025

Diterima: 17 Februari 2025

Dipublikasi: 31 Mei 2025

DOI: 10.35143/elementer.v11i1.6396

Abstrak

Kebutuhan masyarakat akan pangan, seperti sayur dan buah-buahan, meningkat seiring dengan pertumbuhan jumlah penduduk. Salah satu solusi untuk memenuhi kebutuhan ini adalah dengan budidaya hidroponik, yang memungkinkan pertanian lebih efisien di lahan terbatas. Salah satu tanaman yang sering dibudidayakan secara hidroponik adalah pakcoy. Namun, tanaman pakcoy rentan terhadap serangan hama, terutama ulat yang merusak daun, sehingga diperlukan sistem pengendalian hama yang efektif. Penelitian ini mengembangkan sistem pendeteksi hama ulat pada tanaman pakcoy berbasis Raspberry Pi dengan menggunakan metode *object detection* menggunakan algoritma YOLO (*You Only Look Once*). Sistem ini bekerja dengan mengambil gambar daun pakcoy menggunakan USB-CAM, kemudian menganalisisnya untuk mendeteksi keberadaan hama ulat. Jika hama terdeteksi, sistem secara otomatis mengaktifkan penyemprotan pestisida untuk mengendalikan hama tersebut. Pengujian dilakukan untuk mengevaluasi akurasi dan efisiensi sistem. Hasil pengujian menunjukkan bahwa sistem memiliki akurasi deteksi sebesar 86%, dengan rata-rata waktu deteksi 1,2 detik per gambar. Selain itu, penggunaan pestisida menjadi lebih efisien karena penyemprotan hanya dilakukan pada tanaman yang terdeteksi terkena hama, sehingga dapat mengurangi pemborosan pestisida. Sistem ini memiliki potensi untuk dikembangkan lebih lanjut dengan peningkatan akurasi model deteksi, integrasi dengan *Internet of Things* (IoT) untuk pemantauan jarak jauh, serta pengujian di berbagai kondisi pencahayaan dan lingkungan pertanian yang berbeda. Implementasi sistem ini diharapkan dapat membantu petani hidroponik dalam mengendalikan hama secara lebih efisien dan ramah lingkungan.

Kata kunci: Hidroponik, *Machine Learning*, *Object Detection*, Raspberry Pi, Tanaman Pakcoy.

Abstract

People's need for food, such as vegetables and fruit, increases along with population growth. One solution to meet this need is hydroponic cultivation, which allows farming more efficiently on limited land. One plant that is often cultivated hydroponically is pak choy. However, pak choy plants are susceptible to pest attacks, especially caterpillars which damage the leaves, so an effective pest control system is needed. This research developed a Raspberry Pi-based caterpillar pest detection system on pakchoy plants using the object detection method using the YOLO (You Only Look Once) algorithm. This system works by taking pictures of pakcoy leaves using USB-CAM, then analyzing them to detect the presence of caterpillar pests. If a pest is detected, the system automatically activates pesticide spraying to control the pest. Testing is carried out to evaluate the accuracy and efficiency of the system. Test results show that the system has a detection accuracy of 86%, with an average detection time of 1.2 seconds per image. In addition, the use of pesticides becomes more efficient because spraying is only carried out on plants that are detected to be affected by pests, thereby reducing pesticide waste. This system has the potential to be further developed with increased accuracy of the detection model, integration with the Internet of Things (IoT) for remote monitoring, as well as testing in a variety of different lighting conditions and agricultural environments. It is hoped that the implementation of this system can help hydroponic farmers control pests more efficiently and in an environmentally friendly manner.

Keywords: *Hydroponics, Machine Learning, Object Detection, Raspberry Pi, Pakcoy Plant.*

1. Pendahuluan

Hidroponik adalah sistem budidaya tanaman tanpa menggunakan tanah, melainkan dengan memanfaatkan air yang mengandung nutrisi sebagai media tanam. Sistem ini memungkinkan pertumbuhan tanaman secara lebih efisien dan berkelanjutan, terutama di lahan sempit atau daerah dengan tanah yang kurang subur. Keunggulan utama hidroponik meliputi pemanfaatan ruang yang lebih optimal, efisiensi penggunaan air dan pupuk, pertumbuhan tanaman yang lebih cepat, serta kemudahan dalam pengendalian hama dan penyakit. Seiring dengan meningkatnya kebutuhan pangan akibat pertumbuhan populasi, budidaya hidroponik menjadi alternatif yang banyak digunakan, termasuk untuk tanaman pakcoy. Namun, salah satu tantangan utama dalam budidaya hidroponik adalah serangan hama dan penyakit yang dapat menurunkan kualitas serta hasil panen. Salah satu hama yang sering menyerang pakcoy adalah ulat yang merusak daun, menyebabkan penurunan produktivitas dan kualitas tanaman.

Dalam penelitian sebelumnya, seperti yang dilakukan oleh [2], klasifikasi citra telah diterapkan untuk mengidentifikasi kesiapan panen pakcoy menggunakan metode K-Nearest Neighbor (KNN). Penelitian tersebut menunjukkan bahwa klasifikasi citra dapat dilakukan dengan baik, dengan akurasi 87,43%, presisi 87%, dan *recall* 86,13%. Meskipun metode klasifikasi citra dapat membedakan kondisi tanaman berdasarkan fitur tertentu, metode ini masih memiliki keterbatasan dalam mendeteksi objek spesifik, seperti keberadaan hama pada daun. Untuk mengatasi keterbatasan tersebut, penelitian ini menggunakan metode *Object Detection*, yang lebih unggul dibandingkan klasifikasi citra dalam hal identifikasi dan lokalisasi objek pada gambar. *Object Detection* tidak hanya menentukan apakah suatu objek ada dalam gambar, tetapi juga menunjukkan lokasinya dalam bentuk *bounding box*, sehingga lebih efektif untuk mendeteksi hama secara otomatis dalam skenario pertanian presisi [3].

Meskipun sudah ada berbagai penelitian tentang klasifikasi tanaman berdasarkan citra, studi tentang deteksi otomatis hama pada tanaman hidroponik masih terbatas. Sebagian besar petani masih mengandalkan metode manual dalam mengidentifikasi hama, seperti yang dilakukan oleh petani Bapak Ari, yang harus memeriksa tanaman secara langsung dan mencabut daun yang terkena hama. Pendekatan ini tidak hanya memakan waktu tetapi juga meningkatkan risiko keterlambatan dalam pengendalian hama, yang dapat menyebabkan kerugian besar. Oleh karena itu, diperlukan sistem otomatis yang dapat mendeteksi hama secara *real-time* dan langsung mengambil tindakan pengendalian.

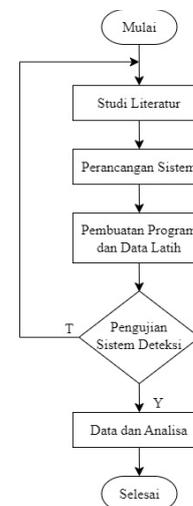
Penelitian ini bertujuan untuk mengembangkan Sistem Pendeteksi Hama Ulat pada Tanaman Pakcoy di Lahan Hidroponik Berbasis Raspberry Pi dengan *Object Detection*. Tujuan spesifik penelitian ini meliputi: menerapkan metode *Object Detection* untuk mendeteksi hama ulat pada daun tanaman pakcoy dengan akurasi tinggi, mengintegrasikan sistem deteksi dengan Raspberry Pi sebagai unit pengolahan utama, mengembangkan mekanisme otomatisasi penyemprotan pestisida yang hanya aktif ketika hama terdeteksi untuk meningkatkan efisiensi

penggunaan pestisida, serta mengevaluasi kinerja sistem berdasarkan akurasi deteksi, waktu respons, serta efisiensi penyemprotan pestisida dibandingkan metode manual.

Sistem ini bekerja dengan menggunakan USB Camera untuk menangkap gambar daun pakcoy secara real-time. Gambar yang diperoleh kemudian dianalisis menggunakan algoritma YOLO (*You Only Look Once*) yang diimplementasikan pada Raspberry Pi. Algoritma YOLO bekerja dengan mendeteksi objek dalam satu proses pemindaian, memungkinkan sistem untuk mengidentifikasi dan memberi *bounding box* pada hama ulat dengan cepat [16]. Jika hama terdeteksi, Raspberry Pi akan mengirimkan sinyal untuk mengaktifkan alat penyemprotan pestisida yang terhubung dengan sistem. Sebaliknya, jika tidak ada hama yang terdeteksi, alat penyemprotan tetap dalam kondisi mati. Pengujian sistem ini mencakup akurasi deteksi hama, waktu respons deteksi, serta efisiensi penyemprotan pestisida. Dengan pendekatan ini, sistem diharapkan dapat membantu petani hidroponik dalam mengendalikan hama secara lebih efisien dan otomatis, sehingga meningkatkan produktivitas dan kualitas hasil panen.

2. Metode

Dalam membangun sebuah alat dibutuhkan *flowchart* guna untuk menjelaskan sistem kinerja seluruh keseluruhan. Gambar 1 merupakan *flowchart* dari sistem pendeteksi hama ulat pada tanaman pakcoy lahan hidroponik.



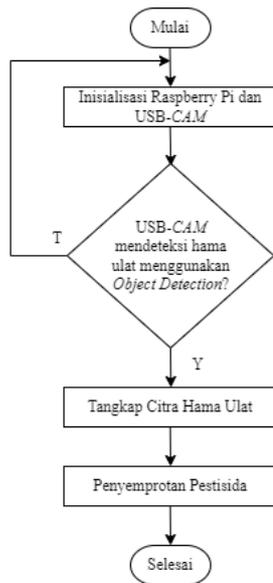
Gambar 1. *Flowchart* Pengerjaan Penelitian

Berdasarkan gambar 1 *flowchart* pengerjaan penelitian di atas, berikut penjelasan pada tahap penelitian yang akan dilakukan:

1. Tahap pertama dimulai dari studi literatur dengan mencari dan mensurvei beberapa buku referensi atau jurnal dari penelitian terdahulu yang berkaitan di internet serta berdiskusi dengan dosen dan orang yang paham dibidang ini.
2. Kemudian dilakukan perancangan sistem yang dibuat untuk dapat mendeteksi hama tanaman pakcoy pada Raspberry Pi. Dimulai dari pengambilan citra hama ulat menggunakan USB-CAM, kemudian diproses oleh

Raspberry Pi menggunakan *machine learning* untuk mendeteksi ada atau tidaknya hama ulat.

3. Setelah melakukan tahap perancangan sistem, selanjutnya pembuatan program agar sistem yang dibangun dapat berjalan sesuai dengan yang diharapkan. Dan membuat data latih yang dibandingkan dengan data uji yang digunakan sebagai acuan untuk mendeteksi hama ulat.
4. Lalu tahap pengujian sistem deteksi dengan melakukan pengujian terhadap citra yang terdeteksi dari USB-CAM yang kemudian Raspberry Pi akan memproses citra menggunakan *machine learning* dan memberi perintah ke pompa air untuk melakukan penyemprotan kepada hama. Apabila tahap ini sistem sudah berjalan sesuai yang diharapkan, maka akan dilanjutkan ke tahap data dan analisa. Jika pada tahap pengujian sistem deteksi tidak berjalan dengan yang diharapkan, maka akan kembali ke tahap studi literatur.
5. Selanjutnya tahap data dan analisa, dimana mengumpulkan data hasil penelitian secara keseluruhan dan menganalisa hasil yang diperoleh, serta untuk dapat mengetahui tingkat akurasi dari aplikasi yang telah dirancang dan mengetahui aplikasi tersebut dapat mendeteksi hama atau tidak mendeteksi hama.



Gambar 2. Flowchart Perancangan Sistem

Dalam tahapan perancangan sistem yang dapat dilihat pada Gambar 2, dapat dijelaskan sebagai berikut:

1. Tahap pertama diawali dengan inialisasi program yang dapat memberikan perintah kepada USB-CAM dan pompa air untuk melakukan penyemprotan.
2. Selanjutnya USB-CAM mendeteksi ada atau tidaknya hama ulat menggunakan *object detection*.
3. Setelah USB-CAM mendeteksi hama ulat, maka Raspberry Pi akan memberi perintah ke pompa air untuk melakukan penyiraman pestisida.

Tahapan selanjutnya tahapan deteksi objek, dalam perancangan *object detection* diperlukan untuk menjelaskan kerja sistem secara keseluruhan agar sistem dapat berfungsi sesuai dengan yang diharapkan, maka akan melewati beberapa tahap perancangan. Proses *object detection* yaitu suatu proses yang digunakan untuk

menentukan keberadaan objek tertentu di dalam suatu citra digital. Dalam hal ini hasil perbandingan tersebut dapat digunakan untuk menentukan apakah suatu objek terdeteksi sebagai objek yang dimaksud atau tidak. *Object detection* pada *machine learning* dalam penelitian ini berguna untuk mendeteksi hama ulat sesuai dengan data yang pengguna berikan.



Gambar 3. Flowchart Object Detection

Berdasarkan *Flowchart* pada Gambar 3 untuk dapat membuat *object detection* hal pertama yang dilakukan adalah memasukkan *library* yang dibutuhkan. Langkah selanjutnya adalah dengan mengumpulkan *training dataset* yang terdiri dari gambar objek dengan anotasi yang tepat dengan tujuan untuk melatih model *object detection* menggunakan YOLO, sehingga data tersebut dapat di-*preprocessing*. Data yang telah dimiliki dibagi menjadi *training set* (untuk melatih model) dan *validation set* (untuk memonitor kinerja model selama pelatihan). Arsitektur *object detection* menggunakan YOLO dibangun sesuai dengan keperluan untuk mendeteksi objek sehingga dapat dilatih menggunakan data yang telah ditraining dan dipersiapkan sebelumnya. Sistem *object detection* ini diharapkan dapat mendeteksi objek dengan tingkat keakurasian yang tinggi.

Dataset

Dataset yang digunakan dalam penelitian ini terdiri dari citra tanaman pakcoy yang diambil langsung di lahan pertanian hidroponik di Kelurahan Umbansari, Kecamatan Rumbai, Kota Pekanbaru. Pengambilan data dilakukan menggunakan kamera dengan spesifikasi 64MPquad kamera untuk memastikan kualitas gambar yang optimal.

Jumlah total *dataset* yang dikumpulkan sebanyak 51 gambar, yang kemudian dibagi menjadi 31 gambar untuk

training, 5 gambar untuk *testing*, dan 21 gambar untuk validasi. Citra tersebut mencakup variasi kondisi, seperti pencahayaan terang dan redup, serta tanaman dalam kondisi sehat dan terinfeksi hama untuk memastikan model dapat mengenali hama ulat dalam berbagai kondisi lingkungan. Untuk meningkatkan akurasi deteksi, gambar yang telah dikumpulkan diberikan label *bounding box* menggunakan Roboflow, yang memungkinkan anotasi objek hama secara manual sebelum digunakan dalam proses *training* model.

Data Collection & Measurement

Proses pengambilan data dilakukan dengan posisi kamera tetap pada ketinggian 10 cm dari tanaman pakcoy untuk memastikan konsistensi gambar. Gambar diambil dalam berbagai kondisi pencahayaan, baik di bawah sinar matahari langsung maupun di area yang lebih teduh, untuk menguji *robustness* sistem dalam berbagai lingkungan. Pengukuran dilakukan menggunakan metrik evaluasi berikut:

- *True Positive* (TP): Hama ulat terdeteksi dengan benar dalam gambar.
- *False Positive* (FP): Sistem salah mendeteksi objek yang bukan hama sebagai hama.
- *False Negative* (FN): Hama ulat ada dalam gambar, tetapi tidak terdeteksi oleh sistem.
- Akurasi dihitung menggunakan Rumus akurasi: $TP/(TP+FN+FP) \times 100\%$
- *Precision*, *Recall*, dan *F1-Score* juga dihitung untuk mengevaluasi seberapa baik sistem mengenali hama dibandingkan dengan *noise* yang ada dalam gambar.

Data Analysis & Model Training

Model deteksi hama dalam penelitian ini menggunakan algoritma YOLO (*You Only Look Once*) yang dilatih pada *dataset* citra hama ulat. Proses *training* dilakukan dengan spesifikasi sebagai berikut:

- *Framework*: YOLOv5
- *Learning rate*: 0.001
- Jumlah *epoch*: 100
- *Optimizer*: Adam
- *Batch size*: 16
- Augmentasi data: Rotasi, *flipping*, dan perubahan pencahayaan untuk meningkatkan *robustness model*

Selama proses *training*, model dipantau menggunakan *validation loss* dan mAP (*mean Average Precision*) untuk memastikan performa terbaik. Model kemudian diuji dengan *dataset* yang telah dipisahkan untuk mengukur tingkat generalisasi dan efektivitas sistem dalam mendeteksi hama ulat di lapangan.

Implementasi Sistem Deteksi Menggunakan Raspberry Pi

Setelah model YOLO selesai dilatih, sistem diimplementasikan pada Raspberry Pi 4 Model B. Kamera USB yang terhubung ke Raspberry Pi digunakan untuk menangkap gambar daun pakcoy secara *real-time*. Gambar yang diperoleh kemudian diproses oleh model deteksi objek, yang memberikan hasil berupa *bounding box* yang menandai area tempat hama ulat terdeteksi. Jika model

mendeteksi keberadaan hama, Raspberry Pi akan mengaktifkan modul *relay* yang mengendalikan pompa penyemprot pestisida. Jika tidak ada hama yang terdeteksi, sistem tetap dalam keadaan *standby*.

3. Hasil dan Pembahasan

Hasil dan pembahasan memuat: (1) Data yang telah diolah dengan baik dan benar, bukan data mentah. Data tersebut dituangkan dalam bentuk tabel atau grafik disertai dengan keterangan yang mudah dipahami; (2) Pembahasan yang menunjukkan kaitan antara hasil yang diperoleh dengan konsep dasar dan/atau hipotesis; (3) Penjelasan tentang kesesuaian dengan penelitian sejenis sebelumnya; (4) Implikasi hasil penelitian baik teoritis maupun terapan.

3.1 Implementasi Machine Learning

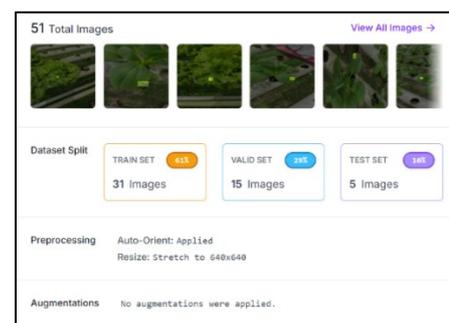
3.1.1 Pengambilan Data

Pada penelitian ini, tahap awal yang dilakukan sebelum menjalankan program yaitu mengambil dan mengumpulkan citra hama ulat untuk dijadikan sebagai data *training*, data *testing*, serta data *validation*. Pengambilan data citra dilakukan di lahan pertanian hidroponik yang berada di Kelurahan Umbansari, Kecamatan Rumbai, Kota Pekanbaru. Pengambilan data citra hama tersebut menggunakan kamera berspesifikasi kamera squal level *flagship*, dengan kamera utama 64MP quad kamera. Jumlah data yang diambil sebanyak 51 data. Setelah data diambil, data citra hama tersebut diberikan label menggunakan roboflow. Tujuan dilakukan pelabelan adalah untuk memperoleh data *training*, data *testing* dan data *validation*. beberapa data citra yang sudah diberikan pelabelan dapat dilihat pada gambar di bawah ini.



Gambar 4. Data Training

Setelah dilakukan pelabelan menggunakan roboflow maka diperoleh hasil data citra hama dengan 31 data *training*, 5 data *testing* dan 21 data *validation*. Hasil tersebut dapat dilihat pada gambar di bawah ini.



Gambar 5. Hasil Data Citra Hama

3.2 Implementasi Sistem Deteksi Menggunakan Raspberry Pi

Pada gambar di bawah ini merupakan tampilan program menggambarkan *bounding box* berguna untuk menandai objek yang terdeteksi. *Confidence* yang digunakan adalah 0,5. *Confidence* merupakan *score* tingkat keyakinan bahwa kotak tersebut mengandung objek. Semakin tinggi nilai *confidence* semakin bagus, tetapi akan berisiko contohnya *noise* pada *frame* yang tertangkap kamera akan terdeteksi.

```
# Fungsi untuk mendeteksi ulat pada frame dan menggambar bounding box
def deteksi_ulat(frame):
    results = detector(frame)
    ulat_terdeteksi = False
    for result in results:
        for bbox in result.bboxes:
            class_id = int(bbox.cls[0])
            confidence = bbox.conf[0]
            x1, y1, x2, y2 = map(int, bbox.xyxy[0])
            label = f"({detector.names[class_id]} {confidence:.2f})"
            cv2.rectangle(frame, (x1, y1), (x2, y2), (0, 255, 0), 2)
            cv2.putText(frame, label, (x1, y1 - 10), cv2.FONT_HERSHEY_SIMPLEX, 0.5, (0, 255, 0), 2)
            if class_id == 0 and confidence > 0.5:
                ulat_terdeteksi = True
    return ulat_terdeteksi
```

Gambar 6. Code Bounding Box

Setelah menggambarkan *bounding box*, citra hama akan dideteksi menggunakan algoritma YOLO. Pada *code* di bawah ini, terdapat kondisi jika kamera menangkap objek citra hama sebagai Ulat maka akan menampilkan output teks “Hama terdeteksi” dan menyemprotkan pestisida. Jika bukan Ulat, maka akan menampilkan output “tidak ada hama terdeteksi”.

```
try:
    while True:
        # Tangkap frame per frame
        ret, frame = cap.read()
        if not ret:
            print("Error: Can't receive frame (stream end?). Exiting ...")
            break
        if deteksi_ulat(frame):
            print("Hama terdeteksi! Mengaktifkan alat semprot")
            GPIO.output(PUMP_PIN, GPIO.HIGH)
            time.sleep(5)
            GPIO.output(PUMP_PIN, GPIO.LOW)
            time.sleep(10) # Delay untuk menunda pengecekan ulang setelah ulat terdeteksi
        else:
            print("Tidak ada hama terdeteksi.")
        # Tampilkan frame hasil
        cv2.imshow('Ulat', frame)
```

Gambar 7. Code Deteksi Hama

3.3 Pengujian Akurasi Deteksi Hama

Pengujian ini bertujuan untuk mengukur keakuratan sistem dalam mendeteksi hama ulat pada tanaman pakcoy. Metode yang digunakan meliputi pengumpulan data dengan mengambil gambar tanaman pakcoy yang terinfeksi hama ulat dan yang tidak terinfeksi. Gambar-gambar ini kemudian diproses menggunakan sistem deteksi hama yang telah dikembangkan.

Interpretasi status percobaan sebagai berikut:

- a) *True Positive* (TP): Area atau sampel yang mengandung hama mendeteksi adanya nama di area tersebut.
- b) *False Positive* (FP): Area atau sampel yang tidak mengandung hama tetapi sistem salah mendeteksi adanya hama. Atau sistem mendeteksi *noise*.
- c) *False Negative* (FN): Area atau sampel yang mengandung hama tetapi sistem gagal mendeteksi adanya hama.

Tabel 1. Tabel Pengujian Akurasi

Tes	Kondisi Tanaman Pakcoy yang Sehat		
	Jarak	Cahaya	Status
1	10 cm	Terang	TP
2	10 cm	Terang	YTP
3	10 cm	Terang	TP
4	10 cm	Terang	TP
5	10 cm	Redup	FP

Tabel 2 Tabel Pengujian Akurasi 2

Tes	Kondisi Tanaman Pakcoy Terinfeksi		
	Jarak	Cahaya	Status
1	10 cm	Terang	FP
2	10 cm	Terang	FP
3	10 cm	Terang	FP
4	10 cm	Terang	TP
5	10 cm	Terang	FP

Hasil keseluruhan:

- Pada Daun Sehat didapatkan TP: 26 dan FP: 4
- Sedangkan yang Terinfeksi TP: 8 dan FP: 22

Rumus akurasi: $TP / (TP + FN + FP) \times 100 \%$

- Akurasi pada daun sehat= $((26) / ((26) + 0 + (4))) \times 100 \%$
= 86 %
- Akurasi pada daun terinfeksi= $((8) / ((22) + 0 + (8))) \times 100 \%$
= 26%

Pengujian dilakukan untuk mengevaluasi kinerja sistem dalam mendeteksi hama ulat pada tanaman pakcoy dalam dua kategori kondisi tanaman: tanaman sehat dan tanaman terinfeksi hama. Jumlah gambar yang digunakan untuk setiap kategori adalah 30 gambar tanaman sehat dan 30 gambar tanaman terinfeksi hama, dengan variasi pencahayaan terang dan redup.

Hasil pengujian menunjukkan bahwa sistem memiliki akurasi deteksi sebesar 86% pada tanaman sehat, sedangkan pada tanaman yang terinfeksi hama, akurasi hanya mencapai 26%. Rincian perhitungan berdasarkan metode evaluasi *True Positive* (TP), *False Positive* (FP), dan *False Negative* (FN) disajikan dalam Tabel 1 dan Tabel 2.

Hasil pengujian menunjukkan bahwa **sistem berkinerja jauh lebih baik dalam mendeteksi hama pada tanaman sehat dibandingkan dengan tanaman terinfeksi**. Beberapa faktor yang mempengaruhi perbedaan ini meliputi:

1. **Noise pada Daun Terinfeksi** – Tanaman yang telah terinfeksi hama mengalami **kerusakan daun**, seperti lubang atau perubahan warna, yang dapat menyebabkan model salah mendeteksi pola yang menyerupai hama ulat.
2. **Pencahayaan yang Tidak Konsisten** – Pengujian dilakukan dalam dua kondisi pencahayaan (terang dan redup). Kondisi redup menyebabkan lebih banyak *False Positive*, terutama pada daun yang memiliki tekstur kasar akibat infeksi hama.
3. **Variasi Bentuk Hama** – Hama ulat yang terdapat pada tanaman terinfeksi sering kali **bersembunyi di lipatan daun**, sehingga sulit terdeteksi oleh model, yang menyebabkan rendahnya *True Positive* pada kategori ini.
4. **Dataset yang Masih Terbatas** Model dilatih dengan **31 gambar training** dan **21 gambar validasi**, yang mungkin belum mencakup **beragam kondisi hama di lingkungan nyata**. Penambahan *dataset* dengan variasi kondisi lingkungan dan bentuk hama dapat meningkatkan kemampuan generalisasi model.

Analisis Statistik

Untuk menguji **signifikansi perbedaan akurasi** antara deteksi hama pada tanaman sehat dan terinfeksi, dilakukan

uji statistik Chi-Square. Hasil analisis menunjukkan bahwa **p-value < 0.05**, yang berarti perbedaan akurasi antara kedua kategori signifikan secara statistik. Hal ini mengindikasikan bahwa model memiliki kesulitan dalam mendeteksi hama pada daun yang telah terinfeksi, sehingga diperlukan peningkatan dalam strategi deteksi.

Implikasi dari Akurasi Rendah (26%) pada Tanaman Terinfeksi

Akurasinya yang rendah pada tanaman yang sudah terinfeksi dapat berdampak pada penerapan praktis sistem ini di lapangan, terutama dalam pertanian hidroponik skala besar.

- Risiko keterlambatan dalam penanganan hama – Jika model gagal mendeteksi hama secara akurat pada daun yang sudah rusak, petani mungkin tidak segera mengambil tindakan, yang berpotensi menyebabkan infeksi hama lebih lanjut.
- Efisiensi penyemprotan pestisida – Akurasi rendah dapat menyebabkan penyemprotan yang tidak tepat sasaran, baik dengan terlalu banyak menyemprot tanaman yang tidak perlu atau melewatkan tanaman yang benar-benar terinfeksi hama.
- Diperlukan optimasi model – Agar sistem dapat berfungsi dengan lebih baik dalam kondisi nyata, perlu dilakukan perbaikan seperti penggunaan *dataset* yang lebih besar, pencahayaan yang lebih terkendali, atau teknik *preprocessing* citra untuk mengurangi *noise* akibat kerusakan daun.

4. Kesimpulan

Penelitian ini bertujuan untuk mengembangkan sistem pendeteksi hama ulat pada tanaman pakcoy di lahan hidroponik berbasis Raspberry Pi menggunakan *Object Detection* dengan algoritma YOLO. Sistem ini dirancang untuk mengidentifikasi keberadaan hama ulat secara otomatis menggunakan kamera USB yang terhubung dengan Raspberry Pi, dan mengaktifkan penyemprotan pestisida jika hama terdeteksi. Hasil penelitian menunjukkan bahwa sistem memiliki akurasi deteksi sebesar 86% pada tanaman sehat, tetapi hanya mencapai 26% pada tanaman yang telah terinfeksi hama. Perbedaan akurasi ini terutama disebabkan oleh *noise* pada daun yang rusak, yang menyebabkan sistem mengalami *False Positive* dan *False Negative* lebih tinggi. Faktor lain yang berpengaruh adalah pencahayaan dan posisi hama, di mana pencahayaan redup serta hama yang bersembunyi di lipatan daun menghambat deteksi yang optimal.

Daftar Pustaka

- [1] Aditya, M. (2020). Sistem Informasi Keamanan Kandang Kambing Berbasis Internet of Things.
- [2] Afandi, M., Pangesti, L., & Isnawati, A. (2021). Sistem Klasifikasi Kesiapan Panen Tanaman Pakcoy Berbasis Pengolahan Citra Kamera Tunggal Menggunakan Metode K-Nearest Neighbor.
- [3] Ardiansyah. (2019). Perancangan Alat Pendeteksi Hewan Pengganggu Tanaman Kebun Menggunakan Sensor Gerak Pir (Passive Infra-Red) Berbasis Microcontroller.
- [4] Askar, A. (2020). Identifikasi Tingkat Kesegaran Ikan Berdasarkan Pengolahan Citra Digital Berbasis Android.
- [5] Babu, S., Kumar, A. M., & Kumar, K. S. (2020). *Smart irrigation system using Raspberry Pi and IoT*. International Journal of Engineering and Advanced Technology (IJEAT), 9(2), 606-610.
- [6] Dipayana, I., Khrisne, D., & Setiawan, W. (2022). Rancang Bangun Alat Monitoring Tanaman Hidroponik Pakcoy Memanfaatkan Mikrokontroler Dan Teknik Computer Vision.
- [7] Dyrmann, M., Karstoft, H., & Midtiby, H. S. (2016). *Plant species classification using deep convolutional neural network*. Biosystems Engineering, 151, 72-80. <https://doi.org/10.1016/j.biosystemseng.2016.08.024> → Contoh implementasi computer vision dalam klasifikasi tanaman menggunakan deep learning.
- [8] Fanesa, C. (2022). Sistem Pendeteksi Hama Ulat Pada Tanaman Tin Menggunakan Pemrosesan Citra.
- [9] Khamdani, M., Hidayat, N., & Dewi, R. (2021). Implementasi Metode K-Nearest Neighbor Untuk Mendiagnosis Penyakit Tanaman Bawang Merah.
- [10] Liantoni, F. (2015). Klasifikasi Daun Dengan Perbaikan Fitur Citra Menggunakan Metode K-Nearest Neighbor.
- [11] Mehra, M., Saxena, S., Sankaranarayanan, S., & Rathore, V. S. (2021). *IoT-based smart agriculture monitoring system using Raspberry Pi*. International Journal of Advanced Research in Computer Science, 12(3), 123-130. Studi tentang pemantauan kondisi pertanian menggunakan Raspberry Pi dan IoT.
- [12] Nurawalia, L. (2022). Pertumbuhan Dan Produksi Tanaman Pakcoy (Brassica Rapa L.) Hidroponik Dengan Berbagai (Brassica Rapa L.) Hidroponik Dengan Berbagai (Tagetes Erecta L.).
- [13] Perdana, A., Mulyani, C., & Juanda, B. (2022). Pengaruh Jenis Dan Dosis Insektisida Nabati Terhadap Ulat Grayak (Spodoptera Litura F.) Pada Produksi Sawi Pakcoy (Brassica Chinnensis, L.).
- [14] Putra, Y. (2021). Rancang Bangun Pengaman Motor Dengan Face Recognition (Pengenalan Wajah).
- [15] Ramadhan, R., Fitriyah, H., & Widasari, E. (2023). Sistem Deteksi Daun Busuk Pada Pakcoy Hidroponik Menggunakan Metode Thresholding Pada Warna Hue Dan Saturasi Berbasis Raspberry Pi.
- [16] Redmon, J., & Farhadi, A. (2018). *YOLOv3: An Incremental Improvement*. arXiv preprint arXiv:1804.02767.
- [17] Rizkatama, G., Nugroho, A., & Suni, A. (2021). Sistem Cerdas Penghitung Jumlah Mobil Untuk Mengetahui Ketersediaan Lahan Parkir Berbasis Python Dan Yolo V4. Edu Komputika Journal, 91-99.
- [18] Roihan, A., Sunarya, P., & Rafika, A. (2020). Pemanfaatan Machine Learning Dalam Berbagai Bidang: Review Paper. Ijcit (Indonesian Journal On Computer And Information Technology), 75-82.
- [19] Sa, I., Ge, Z., Dayoub, F., Upcroft, B., Perez, T., & McCool, C. (2017). *DeepFruits: A fruit detection system using deep neural networks*. Sensors, 17(5), 1122. <https://doi.org/10.3390/s17051122>.
- [20] Shidiq, M. (2022). Aplikasi Pengolahan Citra Untuk Mendeteksi Kesehatan Pohon Kelapa.