Penerapan Teknologi Internet of Things untuk Sistem Deteksi dan Klasifikasi Hama pada Tanaman Anggur Menggunakan YOLO dan SVM

Leo Jaya Sudewo Fakultas
Ilmu Komputer Universitas
Sriwijaya
Sumatera Selatan, Indonesia

Ahmad Zarkasi Fakultas
Ilmu Komputer Universitas
Sriwijaya
Sumatera Selatan, Indonesia

Abstract- Deteksi dan klasifikasi hama tanaman secara akurat menjadi tantangan penting dalam meningkatkan produktivitas pertanian, terutama pada tanaman anggur yang rentan terhadap serangan hama belalang. Penelitian ini mengusulkan sistem cerdas berbasis *Internet of Things* (IoT) yang menggabungkan algoritma YOLOv8 untuk deteksi objek dan *Support Vector Machine* (SVM) untuk klasifikasi dua kelas yaitu belalang dan bukan belalang. Pendekatan ini memanfaatkan kombinasi dua metode berbeda, yaitu deteksi berbasis *deep learning* dan klasifikasi berbasis *machine learning* untuk meningkatkan akurasi serta keandalan sistem dalam kondisi lapangan nyata. Hasil pengujian menunjukkan akurasi 97% dengan nilai *precision, recall*, dan *F1-score* yang seimbang pada kedua kelas. Kebaruan penelitian ini terletak pada penerapan integrasi YOLOv8 dan SVM secara *real- time* di sistem IoT pertanian untuk klasifikasi spesifik hama belalang pada tanaman anggur yang sebelumnya belum banyak dieksplorasi. Pendekatan ini berpotensi digunakan pada sistem pertanian presisi guna mendukung deteksi hama otomatis di lapangan.

Kata Kunci: *Internet of Things*, YOLOv8, *Support Vector Machine*, klasifikasi hama, tanaman anggur, deteksi objek, *machine learning*, *Region of interest*.

I. Pendahuluan

Tanaman anggur termasuk dalam salah satu komoditas hortikultura bernilai ekonomi sangat tinggi yang semakin dibudidayakan secara luas karena memiliki nilai tambah yang tinggi untuk konsumsi langsung maupun pengolahan [1]. Produksi anggur sangat dipengaruhi oleh lingkungan dan keberhasilan pengendalian hama. Hama belalang menjadi salah satu ancaman serius karena dapat merusak daun, batang, dan buah, sehingga menurunkan hasil dan mutu panen [2]. Jika tidak segera ditangani, serangan hama ini dapat menyebabkan kerugian panen hingga lebih dari 15% [3].

Demi menjaga kualitas dan produktivitas tanaman anggur ini diperlukan sistem untuk pengendalian hama yang sangat penting demi menjaga kualitas dan produktivitas tanaman anggur. Namun, metode yang umum digunakan masih bersifat manual, mengandalkan observasi visual petani atau ahli, yang rentan terhadap kesalahan dan memakan waktu, terutama saat hama berada pada posisi atau sudut pandang yang tidak biasa belum lagi faktor eksternal serta keterbatasan jarak pandang [4].

Untuk mengatasi keterbatasan metode manual, *Internet of Things* (IoT) menawarkan solusi pemantauan hama secara *real-time* melalui kamera mikrokontroler dan perangkat seperti Raspberry Pi [5]. Dilengkapi dengan teknologi *machine learning* dengan algoritma YOLOv8 dipilih karena cepat dan mampu mendeteksi objek kecil seperti hama di lingkungan kompleks [6]. Untuk meningkatkan akurasi, hasil deteksi YOLOv8 akan diklasifikasikan ulang menggunakan SVM berdasarkan potongan gambar dari *bounding box*, guna memastikan objek tersebut benar-benar hama belalang dan bukan gangguan visual seperti bayangan atau daun rusak [7]. Kombinasi sudut pandang gambar dan klasifikasi dua tahap membantu mengurangi *false positive* dalam deteksi hama di kondisi lapangan [8]. Model yang hanya difokuskan pada satu jenis hama juga dapat dilatih lebih spesifik dan efisien dibandingkan dengan mengklasifikasikan beberapa jenis hama [9]. Sehingga dengan menggunakan pendekatan ini, sistem tidak hanya dapat mendeteksi keberadaan hama belalang, tetapi juga dapat memverifikasi

kebenarannya dari berbagai sudut untuk menghasilkan deteksi yang lebih akurat dan kontekstual [10].

Keberhasilan deteksi dan klasifikasi hama secara otomatis itu tidak hanya berdampak pada peningkatan hasil panen, namun juga pada efisiensi waktu dan tenaga kerja. Sistem cerdas yang memanfaatkan kombinasi dari IoT, *machine learning* dan klasifikasi manual dapat mempercepat pengambilan keputusan dalam pengendalian hama berbasis data mampu mengurangi biaya pemrosesan tanpa mengorbankan akurasi identifikasi [11].

II. Metodologi Penelitian

A. Sumber Data

Dataset yang digunakan dalam penelitian ini dapat diperoleh dan diakses secara publik dari situs kaggle yang berjudul "*Pests Classification using EfficientNetV2-L*". Dataset ini berisi data citra hama-hama tanaman yang akan di lakukan *preprocessing* untuk dibagi menjadi beberapa bagian agar sesuai dengan pendekatan penelitian ini.

B. YOLOv8

YOLOv8 adalah algoritma deteksi objek satu tahap (*one-stage*) yang cepat dan efisien, terdiri dari empat komponen utama meliputi *input*, *backbone*, *neck*, dan *output*. Pada bagian *input*, gambar diproses menggunakan augmentasi *mosaic*, *anchor* adaptif, dan *padding grayscale* [12]. Fitur diekstraksi melalui jaringan *backbone* dan *neck* dengan bantuan modul C2f, pengembangan dari modul C3, yang menggabungkan struktur ELAN dan *Bottleneck* untuk memperkuat aliran informasi. Modul SPPF kemudian menggabungkan fitur dengan ukuran *kernel* yang bervariasi sebelum dikirim ke tahap *output*. Arsitektur ini menjaga bobot model tetap ringan namun tetap akurat dalam mendeteksi objek [13].

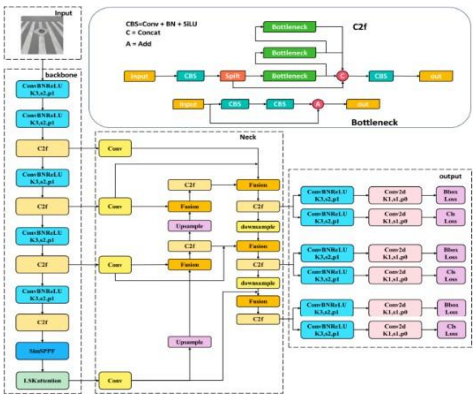


Figure 1 Arsitektur YOLOv8 [13]

C. Support Vector Machine

Metode *Support Vector Machine* (SVM) merupakan algoritma klasifikasi yang andal dan efisien dalam mengidentifikasi pola dan fitur visual dari citra pertanian. Dalam studi terdahulu, SVM berhasil digunakan untuk klasifikasi penyakit tanaman seperti daun jambu biji dengan akurasi tinggi melalui kombinasi fitur tekstur, statistik, dan warna [14]. Mengacu pada keberhasilan tersebut, sistem ini mengintegrasikan deteksi objek menggunakan YOLOv8 dengan klasifikasi menggunakan SVM, guna memastikan identifikasi hama seperti belalang secara lebih presisi dan dapat diimplementasikan secara praktis di lapangan pertanian

D. Perancangan

Perancangan sistem yang akan digunakan pada penelitian ini menggunakan sistem deteksi dengan algoritma YOLOv8 lalu diklasifikasikan dengan sistem klasifikasi menggunakan algoritma *Support Vector Machine* (SVM).

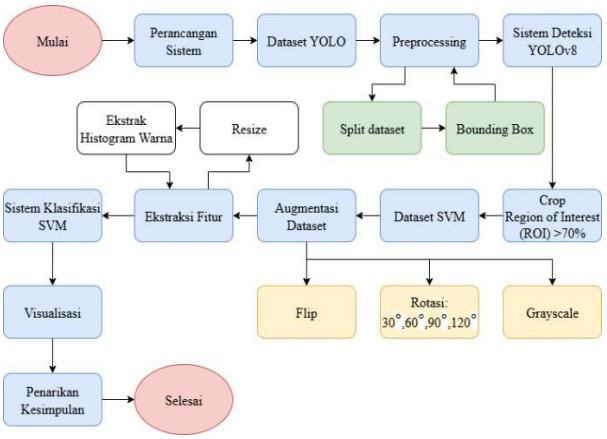


Figure 2 Perancangan Sistem

E. *Preprocessing*

Pertama-tama langkah yang harus dilakukan adalah *preprocessing* untuk deteksi YOLO terlebih dahulu. Ini bertujuan agar dataset bisa di proses dengan baik oleh YOLO. *Preprocessing* pertama yang dilakukan adalah *Split* dataset menjadi beberapa bagian folder yaitu folder *images* dan label. Pada folder *images* berisi lagi subfolder yaitu *train* dan *val* yang berisi hama belalang saja, karena pada YOLO ini akan berfokus untuk mendeteksi hama belalang saja. Dan di folder label juga sama berisi subfolder *train* dan *val*, namun bedanya ini berisi file dalam *format* .txt berfungsi untuk *bounding box* dalam format YOLO.

Tahap *preprocessing* selanjutnya yaitu *bounding box*. Ini adalah tahap pemberian label pada gambar di folder *images* sebelumnya. Berbentuk format YOLO yang mencakup kelas objek, titik tengah *bounding box* x, titik tengah *bounding box* y, tinggi *bounding box* dan lebar *bounding box*.

F. Sistem Deteksi YOLO

Untuk mendeteksi keberadaan hama belalang pada gambar tanaman anggur, digunakan algoritma *You Only Look Once version* 8 (YOLOv8). Model dilatih ulang menggunakan dataset anotasi yang telah disesuaikan dalam format YOLO, dengan konfigurasi sebesar 100 *epoch*, ukuran input gambar 640×640 piksel, dan *batch size* sebesar 8. File konfigurasi dataset didefinisikan dalam data.yaml, yang berisi informasi jalur folder dataset serta label kelas.

Setelah proses pelatihan selesai, model terbaik (best.pt) digunakan untuk mendeteksi objek pada gambar uji. Gambar-gambar uji diambil dari folder uji khusus berisi hama belalang. Setiap gambar diproses secara otomatis oleh model, dan hasil deteksi berupa koordinat *bounding box* serta *confidence score* diekstrak. Selanjutnya, *bounding box* digambar di atas gambar asli menggunakan OpenCV dan divisualisasikan dengan matplotlib. Informasi seperti posisi dan tingkat keyakinan model terhadap keberadaan objek ditampilkan untuk setiap hasil deteksi. Proses ini penting untuk memvalidasi performa deteksi sebelum gambar hasil *crop* digunakan dalam tahap klasifikasi menggunakan SVM.

G. *Crop Region of Interest*(ROI)

Setelah proses deteksi dengan YOLOv8, dilakukan pemotongan gambar berdasarkan *bounding box* (ROI) dengan *confidence* lebih dari sama dengan 70%. Bagian gambar yang berisi objek hama dipotong dan disimpan secara otomatis ke folder khusus yaitu folder Dataset SVM yang berisi hama belalang dan bukan belalang. Hasil crop ini digunakan sebagai *input* untuk proses klasifikasi, karena berisi area yang paling relevan untuk dianalisis lebih lanjut oleh algoritma SVM.



Figure 3 Hasil *Crop* ROI

H. Augmentasi Dataset SVM

Dataset SVM ini akan berisi data citra hama Belalang hasil Crop ROI sebelumnya dan data citra hama Bukan Belalang.

1) Flip

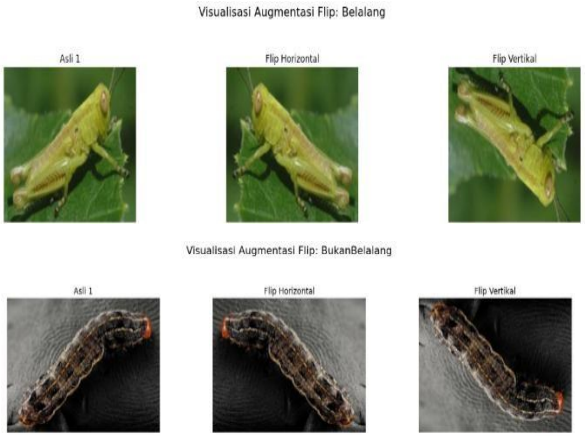


Figure 4 Augmentasi Flip

Untuk meningkatkan performa model klasifikasi serta mengatasi keterbatasan jumlah data, dilakukan proses augmentasi gambar pada Dataset SVM. Teknik augmentasi pertama yang digunakan adalah pembalikan (*flip*) secara horizontal dan vertikal terhadap seluruh gambar dalam masing-masing kelas, yaitu Belalang dan BukanBelalang. Hasil augmentasi disimpan secara terpisah ke dalam folder baru sesuai dengan label kelasnya, yaitu Belalang_augmented dan BukanBelalang_ augmented. Dengan penambahan variasi posisi objek melalui augmentasi ini, diharapkan model dapat mengenali objek hama dari berbagai orientasi dan sudut pandang, sehingga meningkatkan ketahanan (robustness) sistem terhadap kondisi nyata di lapangan.

2) Rotasi

Selain augmentasi pembalikan, dilakukan pula augmentasi berupa rotasi gambar untuk menambah keragaman sudut pandang objek hama. Setiap gambar pada kelas Belalang dan BukanBelalang diputar dengan sudut sebesar 30°, 60°, 90°, dan 120° terhadap titik pusat gambar. Teknik rotasi ini menghasilkan empat varian baru dari tiap gambar asli, yang disimpan dalam folder kelas masing- masing. Proses ini bertujuan untuk meningkatkan kemampuan generalisasi model terhadap variasi orientasi objek di lapangan, terutama saat hama muncul dalam posisi miring atau tidak sejajar terhadap bidang kamera. Dengan demikian, sistem

klasifikasi yang dibangun menjadi lebih *robust* dalam mengenali hama dari berbagai arah pandang.

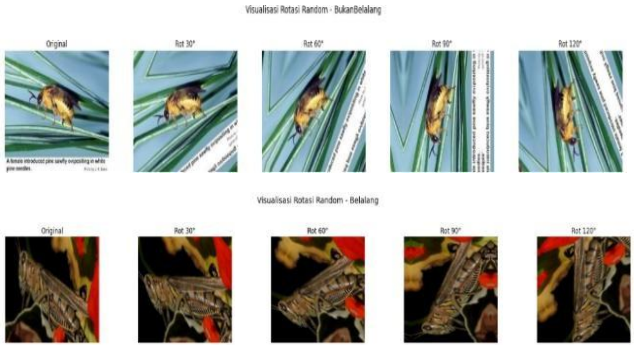


Figure 5 Augmentasi Rotasi

3) Grayscale

Sebagai tambahan dari strategi augmentasi sebelumnya, dilakukan konversi gambar ke *grayscale* untuk meningkatkan variasi data berdasarkan informasi intensitas cahaya. Setiap gambar dari kedua kelas dikonversi menjadi citra hitam-putih, lalu dikembalikan ke format tiga kanal agar tetap kompatibel dengan *pipeline* pengolahan citra berwarna. Hasil konversi disimpan sebagai data augmentasi di folder masing-masing kelas. Augmentasi ini bertujuan untuk membantu model fokus pada bentuk dan tekstur objek hama tanpa bergantung sepenuhnya pada warna, sehingga meningkatkan kemampuan klasifikasi dalam kondisi pencahayaan yang berbeda.



Figure 6 Augmentasi *Grayscale*

I. Ektraksi Fitur

Setelah seluruh gambar dikumpulkan dan diaugmentasi, langkah selanjutnya adalah melakukan ekstraksi fitur untuk digunakan dalam proses klasifikasi. Setiap gambar, baik asli maupun hasil augmentasi, diubah ukurannya menjadi 128×128 piksel dan dikonversi ke format warna RGB. Kemudian, fitur warna diekstraksi menggunakan metode histogram warna tiga kanal (RGB), masing-masing dibagi ke dalam 8 bin sehingga menghasilkan total 512 nilai fitur. Histogram tersebut kemudian dinormalisasi agar skala antar fitur seragam dan direpresentasikan sebagai vektor satu dimensi. Fitur-fitur ini disimpan sebagai representasi numerik dari citra dan dipasangkan dengan label kelasnya (belalang atau bukan belalang) untuk keperluan pelatihan model klasifikasi berbasis *Support Vector Machine* (SVM). Setelah fitur histogram warna berhasil diekstraksi, seluruh data dikonversi ke format *array numerik* untuk diproses lebih lanjut. Label kelas dikodekan secara numerik menggunakan metode Label *Encoding*, di mana setiap kelas direpresentasikan dengan angka agar kompatibel dengan algoritma klasifikasi. Selanjutnya, dilakukan proses normalisasi fitur menggunakan teknik *Standard Scaler*, yang mengubah setiap fitur agar memiliki nilai rata-rata nol dan standar deviasi satu. Normalisasi ini penting untuk memastikan semua fitur berada dalam skala yang sebanding dan mencegah bias terhadap fitur tertentu. Setelah data dinormalisasi, dataset dibagi menjadi dua bagian, yaitu data latih sebesar 80% dan data uji sebesar 20%. Pembagian dilakukan secara *stratified* agar distribusi kelas tetap seimbang, dan nilai acak tetap (random_state=42) digunakan untuk memastikan reproducibilitas eksperimen.

J. Sistem Klasifikasi SVM

Setelah proses ekstraksi fitur dan pembagian data selesai, dilakukan pelatihan model klasifikasi menggunakan algoritma *Support Vector Machine* (SVM). Model SVM dipilih karena kemampuannya yang baik dalam menangani klasifikasi dua kelas dan efisien untuk data berdimensi tinggi. Dalam penelitian ini digunakan kernel linear dengan parameter regularisasi C=1.0, yang memberikan keseimbangan antara margin pemisah yang lebar dan jumlah kesalahan klasifikasi. Model kemudian dilatih menggunakan data latih dan diuji dengan data uji yang telah dipisahkan sebelumnya. Hasil dari proses ini berupa label prediksi dari setiap sampel data uji, yang akan dievaluasi lebih lanjut untuk mengetahui performa klasifikasi.

III. Hasil dan Pembahasan

A. Proses *Epoch* Model YOLOv8s

Model YOLOv8s yang digunakan untuk melakukan pelatihan deteksi objek terhadap dataset hama belalang. Dataset yang digunakan terdiri dari 218 gambar latih dan 62 gambar validasi, yang telah dianotasi secara manual. Pelatihan dilakukan selama 100 *epoch* dengan ukuran gambar 640x640 dan *batch size* sebesar 8. Berdasarkan hasil pelatihan,

performa model pada *epoch* awal menunjukkan nilai mAP (*mean Average Precision*) yang rendah, yaitu 0.134 untuk mAP50. Namun seiring proses pelatihan berjalan, performa model meningkat secara signifikan. Pada epoch ke-96, model berhasil mencapai precision sebesar 0.916 dan recall sebesar 0.919, dengan nilai mAP50 mencapai 0.931. Setelah pelatihan selesai, model terbaik (best.pt) divalidasi ulang dan menghasilkan hasil evaluasi yang konsisten, memperkuat bahwa model telah belajar mendeteksi objek target (hama belalang) dengan baik.

Epoch	Box Loss	Class Loss	DFL Loss	Precision	Recall	mAP50
1	1.749	3.308	2.063	0.177	0.403	0.134
2	1.803	2.415	2.045	0.114	0.29	0.104
3	1.978	2.381	2.201	0.279	0.0484	0.0425
4	2.075	2.547	2.29	0.0312	0.0968	0.0111
5	2.242	2.44	2.34	0.0191	0.0806	0.00542
⋮	⋮	⋮	⋮	⋮	⋮	⋮
96	1.091	0.7487	1.567	0.916	0.919	0.931
97	1.092	0.7242	1.555	0.895	0.903	0.908
98	1.1	0.741	1.58	0.888	0.896	0.904
99	1.082	0.7198	1.601	0.888	0.895	0.909
100	1.098	0.7502	1.6	0.903	0.903	0.906

mendeteksi hama secara otomatis dalam konteks pertanian, khususnya untuk pemantauan tanaman anggur. Model ini selanjutnya digunakan sebagai tahap awal dalam sistem klasifikasi hama menggunakan metode *Support Vector Machine* (SVM) untuk mengklasifikasikan objek hasil deteksi sebagai belalang atau bukan.

B. Hasil Deteksi YOLOv8s

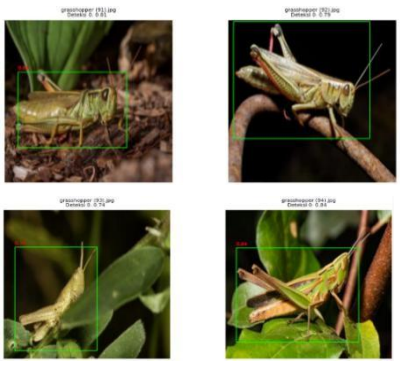


Figure 8 Hasil Deteksi YOLO

Gambar tersebut menampilkan hasil deteksi objek menggunakan model YOLOv8s pada gambar- gambar hama belalang. Setiap belalang yang terdeteksi ditandai dengan kotak hijau (*bounding box*) yang menunjukkan lokasi objek dalam gambar. Di atas kotak tersebut tercantum nilai *confidence score*, seperti 0.81, 0.73, hingga 0.84, yang menunjukkan tingkat keyakinan model terhadap hasil deteksi. Semakin tinggi nilai *confidence*, semakin yakin model bahwa objek tersebut adalah belalang. Keempat gambar yang ditampilkan memperlihatkan deteksi belalang dari berbagai sudut pandang dan kondisi lingkungan, seperti belalang yang terlihat dari samping, atas, dan berada di atas daun atau batang. Hal ini menunjukkan bahwa model YOLOv8s mampu mengenali objek dengan cukup baik meskipun terdapat variasi posisi dan latar belakang. Keberhasilan deteksi ini juga menandakan bahwa proses training dan penyusunan dataset sudah dilakukan dengan cukup representatif. Hasil ini dapat digunakan untuk tahap selanjutnya, yaitu pemotongan (*crop*) *region of interest* (ROI) dari hasil deteksi untuk proses klasifikasi lebih lanjut menggunakan metode *Support Vector Machine* (SVM).

C. Hasil Train Prediksi SVM



Figure 9 Visualisasi Train Prediksi SVM Gambar ini menunjukkan hasil visualisasi prediksi model klasifikasi terhadap data latih dalam sistem klasifikasi dua kelas yaitu Belalang dan BukanBelalang. Dari lima gambar yang ditampilkan prediksi model menunjukkan hasil yang tepat di mana label sebenarnya dan hasil prediksi sesuai. Tiga gambar pertama merupakan contoh dari kelas BukanBelalang yang diprediksi dengan benar, sedangkan dua gambar terakhir merupakan contoh dari kelas Belalang yang juga diklasifikasikan dengan benar oleh model. Akurasi tinggi pada data latih ini mengindikasikan bahwa model mampu mengenali ciri khas visual dari masing-masing kelas dalam proses pelatihan. Namun, hasil ini juga harus disandingkan dengan performa pada data uji untuk memastikan bahwa model tidak mengalami *overfitting*, yaitu terlalu menghafal data latih dan kurang mampu menggeneralisasi pada data baru.

D. Hasil *Test* Prediksi SVM



Figure 10 Visualisasi *Test* Prediksi SVM

Gambar di atas memvisualisasi hasil prediksi model pada data uji untuk klasifikasi dua kelas yaitu Belalang dan BukanBelalang. Terdapat enam gambar yang masing-masing menunjukkan label sebenarnya dan hasil prediksi oleh model. Hasil visualisasi menunjukkan bahwa seluruh prediksi yang dilakukan oleh model telah sesuai dengan label aslinya, baik untuk gambar belalang maupun bukan belalang. Gambar yang merupakan belalang yang berhasil diklasifikasikan dengan tepat sebagai "Belalang", sedangkan sisa gambar berikutnya adalah objek bukan belalang yang juga diklasifikasikan dengan benar. Hal ini menunjukkan bahwa model memiliki performa yang baik dan mampu membedakan kedua kelas dengan presisi tinggi pada data uji. Keakuratan penuh ini mencerminkan bahwa fitur yang digunakan dan algoritma klasifikasi yang diterapkan telah bekerja dengan efektif.

E. Hasil dan Visualisasi Klasifikasi

	Precision	Recall	F1-score	Support
Belalang	0.96	0.97	0.97	151
BukanBelalang	0.97	0.96	0.97	150
Akurasi			**0.97**	**301**
Macro avg	0.97	0.97	0.97	301
Weighted avg	0.97	0.97	0.97	301

Figure 11 Classification Report

Gambar di atas menunjukkan tabel hasil evaluasi performa model klasifikasi SVM dalam membedakan antara kelas Belalang dan BukanBelalang. Model menunjukkan nilai *precision* sebesar 0.96 untuk kelas Belalang dan 0.97 untuk BukanBelalang, yang berarti model mampu meminimalkan kesalahan prediksi *false positive*. Nilai *recall* sebesar 0.97 untuk Belalang dan 0.96 untuk BukanBelalang menunjukkan kemampuan model dalam mendeteksi objek yang sebenarnya termasuk dalam kelas tersebut. Sementara itu, nilai F1-score sebesar 0.97 untuk kedua kelas menandakan keseimbangan yang sangat baik antara *precision* dan *recall*. Total *support* untuk setiap kelas hampir seimbang, yaitu 151 dan 150 sampel. Rata- rata makro (*macro avg*) dan rata-rata berbobot (*weighted avg*) keduanya menunjukkan nilai 0.97 pada semua metrik, menandakan konsistensi performa model di seluruh kelas. Secara keseluruhan, model mencapai akurasi 97%, menunjukkan bahwa sistem klasifikasi ini sangat efektif dan andal dalam mengidentifikasi hama belalang pada tanaman anggur.

Hasil akurasi sebesar 97 persen pada sistem ini menunjukkan peningkatan kinerja dibandingkan dengan penelitian terdahulu yang dilakukan oleh Fuad Mahrus Fathoni et al [15]. Penelitian tersebut menggunakan metode K-Nearest Neighbor (KNN) dengan ekstraksi fitur *Gray Level Co-occurrence Matrix* (GLCM) untuk mengklasifikasikan penyakit daun anggur dan menghasilkan akurasi tertinggi sebesar 88,6 persen [15]. Dibandingkan dengan pendekatan tersebut, sistem pada penelitian ini yang menggabungkan algoritma YOLOv8 dan *Support Vector Machine* (SVM) menunjukkan peningkatan akurasi sebesar 8,4 persen. Hal ini disebabkan oleh kemampuan YOLOv8 dalam mendeteksi objek secara presisi pada berbagai kondisi pencahayaan dan sudut pandang, serta keunggulan SVM dalam memisahkan dua kelas dengan *margin* yang optimal. Perbedaan hasil ini menegaskan bahwa pendekatan berbasis kombinasi deteksi dan klasifikasi memiliki keunggulan signifikan dalam meningkatkan performa pengenalan hama pada tanaman anggur berbasis *Internet of Things* (IoT).

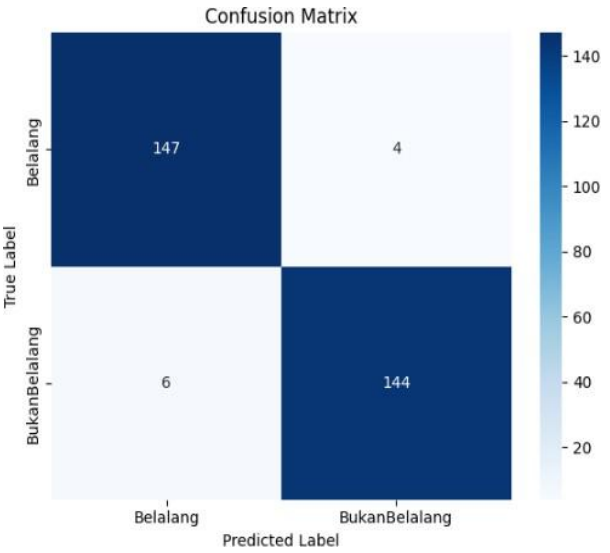


Figure 12 Confusion Matrix

Pada gambar diatas menampilkan viusalisasi dari confusion matrix yang menunjukkan kinerja model

Support Vector Machine (SVM) dalam mengklasifikasikan dua kelas yaitu *Belalang* dan BukanBelalang. Dari total data uji, sebanyak 147 gambar yang benar-benar merupakan belalang berhasil diklasifikasikan dengan tepat sebagai *Belalang*, sementara 144 gambar yang memang bukan belalang juga dikenali dengan benar sebagai BukanBelalang. Namun, terdapat sedikit kesalahan klasifikasi, di mana 4 gambar belalang salah dikenali sebagai bukan belalang, dan 6 gambar bukan belalang salah diklasifikasikan sebagai belalang. Secara keseluruhan, distribusi prediksi menunjukkan performa model yang sangat baik dengan jumlah prediksi benar yang dominan dan hanya sedikit kesalahan. Hasil ini selaras dengan nilai akurasi dan metrik evaluasi lainnya yang sebelumnya ditampilkan, yang menunjukkan bahwa model mampu membedakan kedua kelas dengan presisi dan konsistensi yang tinggi.

IV. Kesimpulan dan Saran

Penelitian ini telah mengimplementasikan sistem deteksi dan klasifikasi hama belalang pada tanaman anggur berbasis *Internet of Things* dengan pendekatan *machine learning*. Sistem ini menggabungkan algoritma YOLOv8 untuk deteksi awal objek hama dengan tingkat *precision* dan *recall* yang tinggi, serta algoritma *Support Vector Machine* (SVM) untuk klasifikasi dua kelas belalang dan bukan belalang. Proses augmentasi data, ekstraksi fitur menggunakan histogram warna RGB, serta normalisasi dan pelatihan model SVM terbukti efektif dalam meningkatkan akurasi sistem. Hasil evaluasi menunjukkan bahwa sistem mampu mencapai akurasi hingga 97%, dengan nilai *precision* dan *recall* yang seimbang pada kedua kelas. Hal ini membuktikan bahwa sistem yang dikembangkan dapat memberikan hasil yang andal, efisien, dan praktis untuk diterapkan di lapangan dalam mendeteksi dan mengidentifikasi hama secara otomatis.

Untuk pengembangan selanjutnya, sistem ini dapat diperluas untuk mengenali lebih banyak jenis hama dan penyakit tanaman lainnya dengan menambah jumlah kelas dalam proses klasifikasi. Selain itu, integrasi sistem ke dalam Disarankan pula penggunaan teknik ekstraksi dan fitur *deep learning* lanjutan agar akurasi dapat ditingkatkan lebih dalam. Baik pada kondisi pencahayaan minim dan latar belakang yang lebih kompleks. Pengujian sistem di lingkungan lapangan secara langsung juga diperlukan untuk memastikan keandalan sistem dalam kondisi nyata yang dinamis.

References

[1] D. Pitaloka, “HORTIKULTURA: POTENSI, PENGEMBANGAN DAN TANTANGAN.”

[2] A. Y. Darmawan, Y. M. P. Tanga, and J. Unjung, “Grape leaf disease classification using efficientnet feature extraction and catboostclassifier,” *Journal of Soft Computing Exploration*, vol. 6, no. 1, pp. 1–8, Mar. 2025, doi: 10.52465/joscex.v6i1.507.

[3] Q. Zhou, S. Li, and Y. Zhang, “Combined Impacts of COVID-19 Pandemic and Locust Plague on Grain Production and Trade Patterns in South Asia,” *Agriculture (Switzerland)*, vol. 13, no. 6, Jun. 2023, doi: 10.3390/agriculture13061212.

[4] J. Wei *et al.*, “Improving the Accuracy of Agricultural Pest Identification: Application of AEC-YOLOv8n to Large- Scale Pest Datasets,” *Agronomy*, vol. 14, no. 8, Aug. 2024, doi: 10.3390/agronomy14081640.

[5] M. Zarboubi, A. Bellout, S. Chabaa, A. Dliou, and A. Zeroual, “IoT-Based Pest Detection in Agriculture Using Raspberry Pi and YOLOv10m for Precision Farming,” *ITM Web of Conferences*, vol. 69, p. 04017, 2024, doi: 10.1051/itmconf/20246904017.

[6] M. Sohan, T. Sai Ram, and Ch. V. Rami Reddy, “A Review on YOLOv8 and Its Advancements,” 2024, pp. 529–545. doi: 10.1007/978-981-99-7962-2_39.

[7] L. Zhang, S. Sun, H. Zhao, Z. Li, and D. Li, “Multi-species insect recognition method based on computer visions: Sustainable agricultural development,” *Ecol Inform*, vol. 88, Sep. 2025, doi: 10.1016/j.ecoinf.2025.103125.

[8] J. Mu *et al.*, “TFEMRNet: A Two-Stage Multi-Feature Fusion Model for Efficient Small Pest Detection on Edge Platforms,” *AgriEngineering*, vol. 6, no. 4, pp. 4688–

4703, Dec. 2024, doi: 10.3390/agriengineering6040268.

[9] S. M. Venkateswara and J. Padmanabhan, "Deep learning based agricultural pest monitoring and classification," *Sci Rep*, vol. 15, no. 1, Dec. 2025, doi: 10.1038/s41598-025-92659-5.

[10] L. Jiao *et al.*, "Global Context-Aware- Based Deformable Residual Network Module for Precise Pest Recognition and Detection," *Front Plant Sci*, vol. 13, Jun. 2022, doi: 10.3389/fpls.2022.895944.

[11] C. M. Badgujar, A. Poulose, and H. Gan, "Agricultural object detection with You Only Look Once (YOLO) Algorithm: A bibliometric and systematic literature review," *Comput Electron Agric*, vol. 223, Aug. 2024, doi: 10.1016/j.compag.2024.109090.

[12] M. Sohan, T. Sai Ram, and Ch. V. Rami Reddy, "A Review on YOLOv8 and Its Advancements," 2024, pp. 529–545. doi: 10.1007/978-981-99-7962-2_39.

[13] X. Wang, H. Gao, Z. Jia, and Z. Li, "BL-YOLOv8: An Improved Road Defect Detection Model Based on YOLOv8," *Sensors (Basel)*, vol. 23, no. 20, Oct. 2023, doi: 10.3390/s23208361.

[14] K. K. Ray *et al.*, "Guava leaf disease detection using support vector machine (SVM)," *Smart Agricultural Technology*, vol. 12, Dec. 2025, doi: 10.1016/j.atech.2025.101190.

[15] F. Mahrus Fathoni, C. Aji Putra, A. Lina Nurlaili, F. Ilmu Komputer, U. Pembangunan Nasional, and J. Timur, "KLASIFIKASI PENYAKIT DAUN ANGGUR MENGGUNAKAN METODE K-NEAREST NEIGHBOR BERDASARKAN GRAY LEVEL CO- OCCURRENCE MATRIX," vol. 3, no. 1, 2024, [Online]. Available: https://ojs.unsiq.ac.id/index.php/biner