

DETEKSI PENYAKIT PADA TANAMAN CABAI BERBASIS DEEP LEARNING MENGGUNAKAN MODEL YOLOv11-L

Candra, Siska Devella, Yohannes

Informatika, Universitas Multi Data Palembang

Jl. Rajawali No.14, 9 Ilir, Kec. Ilir Tim. II, Kota Palembang, Sumatera Selatan 30113, Indonesia

candraandi2003@gmail.com

ABSTRAK

Penyakit pada tanaman cabai sering menyebabkan terjadinya penurunan produktivitas yang signifikan bagi para petani. Deteksi dini merupakan salah satu langkah yang dapat diambil untuk mengantisipasi hal ini, namun jika dilakukan secara manual tentu memerlukan waktu lama. Penelitian ini bertujuan untuk mengembangkan model deteksi dini khususnya pada tanaman cabai menggunakan algoritma YOLOv11-L. Metode penelitian meliputi pengumpulan *dataset* sebanyak 450 citra tanaman cabai yang terdiri dari tanaman cabai sehat, antraknosa, daun bercak cokelat, layu fusarium, virus kuning, dan daun keriting dengan masing-masing data berjumlah 75 citra. Hasil terbaik dari pelatihan model deteksi memperoleh mAP@50 sebesar 64,42% dengan hasil pengujian menggunakan data uji sebanyak 45 citra memperoleh tingkat keberhasilan deteksi sebesar 93,33% dengan kesalahan sebesar 6,67%. Hasil ini menunjukkan bahwa model YOLOv11-L cukup efektif digunakan sebagai alat bantu yang dapat digunakan oleh para petani dalam mengidentifikasi kesehatan tanaman cabai dengan cepat untuk mencegah terjadinya gagal panen.

Kata kunci : Cabai; Deep Learning; Deteksi Penyakit; Hyperparameter; You Only Look Once;

1. PENDAHULUAN

Pertanian adalah suatu kegiatan manusia yang termasuk didalamnya yaitu bercocok tanam, peternakan, perikanan, dan juga kehutanan [1]. Sektor pertanian meliputi beberapa sub sektor yaitu subsektor tanaman bahan makanan, sub sektor hortikultura, sub sektor perikanan, sub sektor peternakan, dan sub sektor kehutanan [2]. Diantara sub sektor tersebut, sub sektor hortikultura merupakan komponen penting dalam pembangunan pertanian yang harus tumbuh dan berkembang dari waktu ke waktu. Produk komoditas tersebut tidak hanya untuk memenuhi kebutuhan pasar di dalam negeri saja, melainkan juga sebagai komoditas ekspor yang menghasilkan devisa negara [3]. Salah satu komoditas hortikultura yang banyak dibudidayakan di Indonesia adalah tanaman cabai, karena mempunyai nilai pemasaran yang tinggi dari segi konsumsi dan ekonomi [4]. Apabila pasokan cabai kurang atau lebih rendah dari konsumsi, maka akan terjadi kenaikan harga [5], sehingga dapat mempengaruhi tingkat inflasi suatu negara.

Salah satu faktor yang mempengaruhi penurunan pasokan cabai yaitu masih banyaknya para petani yang belum mengetahui cara merawat tanaman cabai dengan baik [6]. Selain itu, dalam praktik budidaya tanaman cabai di kalangan petani, terdapat beberapa masalah yaitu sulitnya mendeteksi, mengetahui gejala atau penyebab kerusakan pada tanaman cabai dari jenis hama atau penyakit yang menyerang tanaman cabai, sehingga cabai bisa rusak seperti busuk, terserang hama dan bahkan gagal panen [7]. Hama pada tanaman seperti grayak, kutu daun, thrips, tungau, dan lalat buah dapat menyebabkan penyakit pada tanaman cabai [8]. Berdasarkan wawancara bersama pihak UPTD Balai Perlindungan Tanaman Pangan dan Hortikultura (BTPH) dan Dinas Pertanian

dan Ketahanan Pangan, penyakit umum yang menyerang tanaman cabai diantaranya Antraknosa (*Colletotrichum gloeosporioides*), Bercak Cokelat (*Cercospora* sp.), Layu Fusarium (*Fusarium oxysporum*), Virus Gemini (*Gemini Virus*), Daun Keriting (*Leaf Curl*). Penyakit atau hama menyerang tanaman cabai umumnya dimulai dari tahap persemaian hingga panen [9]. Hal ini tentunya sangat merugikan bagi para petani cabai, apabila penanganan lambat maka kerusakan tanaman akan semakin berat sehingga akan berakibat fatal bagi tanaman itu sendiri [7]. Oleh karena itu, permasalahan ini tentunya harus segera diatasi dengan melakukan pencegahan serta penanganan sejak dini, agar para petani cabai dapat meningkatkan produksi cabai dan menjaga stabilitas harga cabai, dimana pada akhirnya dapat meningkatkan devisa negara.

Salah satu upaya penanggulangan yang umum dilakukan untuk mengatasi permasalahan dalam mendeteksi penyakit pada tanaman cabai adalah bertanya langsung dengan seorang pakar yang ahli dan berpengalaman agar dapat menghasilkan diagnosa yang tepat. Namun, seorang pakar terkadang memiliki keterbatasan waktu sehingga hal tersebut menjadi kendala bagi para petani yang akan melakukan konsultasi untuk menyelesaikan permasalahan penyakit pada tanaman cabai untuk mendapatkan solusi yang terbaik [10].

Saat ini, transformasi digital telah mempengaruhi beberapa industri, termasuk sektor pertanian. Penggunaan teknologi digital di sektor pertanian dapat membantu meningkatkan produktivitas, efisiensi, dan pertumbuhan sektor pertanian serta jaringan pemasaran dan distribusi, serta inovasi dan pertumbuhan [11]. Pada penelitian terdahulu, terdapat berbagai metode model *deep learning* yang dapat

digunakan untuk mendeteksi penyakit pada tanaman seperti *Improved Region Convolutional Neural Network* (R-CNN) dengan VGG16 [12], *Faster Region Convolutional Neural Network* (Faster R-CNN) dengan ResNet50 [13], *You Only Look Once* (YOLO) v11 [14].

Oleh karena itu, permasalahan penyakit pada tanaman cabai akan diatasi dengan mengembangkan sebuah model *deep learning* yang dapat mendeteksi penyakit pada tanaman cabai. Dengan demikian, solusi ini diharapkan dapat membantu para petani dalam mendeteksi dan mengetahui gejala penyakit pada tanaman cabai sejak dini dengan mudah.

2. TINJAUAN PUSTAKA

2.1. Penyakit Cabai

Berikut adalah beberapa penyakit pada tanaman cabai yang umumnya sering ditemukan.

- a. Antraknosa (*Colletotrichum spp.*), merupakan salah satu penyakit yang menyebabkan turunnya hasil pertumbuhan tanaman cabai menjadi rendah. Penyakit ini berbentuk bintik hitam kecil yang kelamaan menjadi besar dan membentuk lekukan, lalu menjadi kering dan berkerut [15].
- b. Daun Bercak Cokelat (*Cercospora sp.*), merupakan penyakit pada tanaman cabai menyerang tanaman cabai terutama di daerah tropis dengan kondisi panas dan lembap yang disebabkan oleh jamur *Cercospora* [16]. Penyakit ini terlihat pada daun, di mana munculnya bercak bulat berukuran kecil, kebasahan, dan berubah warna menjadi cokelat kehitaman ketika terkena serangan bercak yang berat [17].
- c. Layu Fusarium (*Fusarium Oxysporum*), merupakan penyakit pada tanaman cabai yang disebabkan oleh jamur *Fusarium sp* [18]. Penyakit ini dapat menyebabkan matinya tanaman cabai hingga gagal panen, di mana penularan penyakit ini berlangsung dengan sangat cepat, terutama pada lahan yang memiliki topografi lereng, karena penyakit ini ditularkan melalui aliran air [19].
- d. Virus Kuning (*Virus Gemini*), merupakan penyakit pada tanaman cabai yang disebabkan oleh patogen kelompok *Begomovirus*. Penyakit ini biasanya ditandai dengan bercak kuning di sekitar tulang daun dan tulang daun tampak lebih hijau dan daun berkembang menjadi warna kuning yang sangat jelas [20].
- e. Daun Keriting (*Leaf Curl*), merupakan penyakit pada tanaman cabai yang disebabkan oleh virus dari kelompok *Geminiviridae* (*Begomovirus*) dan ditularkan melalui kutu kebul (*Bemisia Tabaci*), di mana gejala awalnya ditandai dengan daun yang mengeriting pada sisi atas, daun tampak berkerut, pengurangan luas daun, melepuh di area interveinal, pita vena, pemendekan tangkai daun dan stunting parah pada tanaman cabai serta apabila penyakit ini berlanjut hingga tahap

pertumbuhan akhir tanaman cabai, maka kuncup bunga menunjukkan absisif dan kepala sari steril dapat terbenam atau butiran serbuk sari yang tidak normal, yang pada akhirnya menghasilkan buah yang berkualitas buruk dan kurang berkembang [21].

2.2. Deep Learning

Deep Learning merupakan sebuah cabang dari pembelajaran mesin (*machine learning*) yang menggunakan jaringan saraf (*neural network*) sebagai fungsi perkiraan (*approximators*) dengan menerapkan penekanan khusus pada penumpukan banyak lapisan komponen yang secara struktural serupa [22].

2.3. Hyperparameter

Hyperparameter merupakan suatu variabel yang mempengaruhi output dari sebuah model [23]. Penyesuaian *hyperparameter* bertujuan dalam mengoptimalkan dan meningkatkan kinerja dari suatu model [24].

2.4. Penelitian Terdahulu

Pada penelitian sebelumnya, terdapat berbagai model dan arsitektur yang telah dikembangkan untuk melakukan tugas deteksi dalam bidang pertanian.

a. Improved R-CNN dengan VGG16

Pada penelitian yang berjudul "*Plant Disease Detection using Region Based Convolutional Neural Network*" yang dilakukan oleh Rehana, dkk. (2023) [12] mengembangkan sistem deteksi penyakit pada sembilan jenis gangguan daun tomat menggunakan metode *Improved R-CNN* dengan VGG16 sebagai *feature extractor*. *Dataset* yang digunakan sebanyak 934 gambar di Kaggle yang di anotasi melalui *LabelImg* dan diubah ukurannya menjadi 256×256 px, di mana model dilatih dengan konfigurasi 300 *epoch*, *learning rate* 0.00001 serta *optimizer* Adam. Hasil evaluasi menunjukkan performa yang sangat impresif dengan mAP 96.31%, *recall* 98.07% dan F2-Score 97.71%.

b. Faster R-CNN dengan ResNet50

Pada penelitian yang berjudul "*Detection of Disease in Maize Plant Using Deep Learning*" yang dilakukan oleh Reddy dkk. (2021) [13] mengembangkan sistem deteksi penyakit tanaman jagung yang terdiri dari 4 *class* (*Northern Leaf Blight*, *Common Rust*, dan *Cercospora/Bacterial Spot*) menggunakan arsitektur *Faster R-CNN* dengan *ResNet-50* sebagai *backbone* nya. *Dataset* yang digunakan mencakup 1269 gambar yang dianotasi menggunakan *LabelImg*. Hasil pengujian model menunjukkan performa akurasi 91% dengan *error rate* dibawah 0.1 dengan total mAP sebesar 94.06% pada *threshold* IoU 0.5.

c. You Only Look Once (YOLO) v11

Pada penelitian yang berjudul "*Comparative performance of YOLOv8, YOLOv9, YOLOv10, YOLOv11 and Faster R-CNN Models for Detection*"

of Multiple Weed Species” yang dilakukan oleh Sharma dkk. (2024) [14] melakukan studi komparatif antara model YOLOv8, YOLOv9, YOLOv10, YOLOv11, dan *Faster R-CNN* untuk mendeteksi berbagai spesies gulma dengan menggunakan 2348 gambar yang diambil melalui kamera *GoPro* dan dianotasi melalui *Roboflow*. Data tersebut diproporsikan menjadi 85% data *training*, 10% data *validation*, dan 5% data *testing* dengan resolusi 640×640 px. Hasil komparatif tersebut menunjukkan bahwa model YOLOv11 merupakan model yang seimbang dengan *inference time* tercepat (13.5 ms) dan mAP 92.1%. Walaupun YOLOv9 mendapatkan mAP tertinggi 93.5%, namun pada proses-nya *inference time* nya jauh lebih lambat (54.2 ms). Sementara *Faster R-CNN* mampu memberikan hasil terendah dengan mAP 86.5% dengan *inference time* (63.8 ms).

3. METODE PENELITIAN

3.1. Pengumpulan Data Penelitian

Penelitian ini menggunakan *dataset* sebanyak 450 gambar tanaman cabai yang telah divalidasi oleh pakar, di mana *dataset* terdiri dari 6 *class* dengan distribusi masing-masing 75 gambar. *Dataset* yang digunakan mencakup 5 kategori penyakit tanaman cabai serta 1 kategori tanaman cabai sehat guna melatih model agar mampu melakukan diferensiasi fitur secara akurat antara tanaman yang terinfeksi patogen dengan tanaman normal, sehingga meminimalisir terjadinya kesalahan deteksi.

Dataset dikumpulkan secara *hybird* melalui platform digital (*Kaggle* dan *Roboflow*) serta pengambilan sampel secara langsung di lapangan wilayah Gandus, Palembang. Dalam keperluan pengembangan model, *dataset* dibagi menjadi 3 proporsi yaitu 80:10:10, di mana 80% (360) data *training*, 10% (45) data *validation* dan 10% (45) data *testing*. Terkait detail dari sampel data gambar tanaman cabai yang terkumpul dan digunakan untuk penelitian ini dapat terlihat pada Tabel 1.

Tabel 1. *Dataset* Penelitian

Nama Sampel	Sampel Gambar	Jumlah Data
Tanaman Cabai Sehat		75
Antraknosa		75
Daun Bercak Cokelat		75
Layu Fusarium		75

Nama Sampel	Sampel Gambar	Jumlah Data
Virus Kuning		75
Daun Keriting		75

3.2. Rancangan Penelitian

Penelitian ini menggunakan model *deep learning* *You Only Look Once* (YOLO) sebagai model kerangka deteksi karena memiliki keunggulan dalam menyeimbangkan kecepatan dan akurasi [25]. Secara mekanisme, YOLO bekerja dengan membagi gambar input menjadi bentuk *grid* sebesar $S \times S$ yang diproses melalui jaringan saraf (*neural network*) untuk melakukan *feature extraction* [26]. Setiap sel pada *grid* kemudian memprediksi sejumlah *B bounding box*, *confidence score* dan *class prediction* secara simultan [27].

Varian YOLO yang digunakan dalam penelitian ini adalah YOLOv11-L (*Large*). Pemilihan model ini didasarkan pada pertimbangan efisiensi komputasi model, di mana YOLOv11 secara umum memiliki jumlah parameter yang jauh lebih sedikit (2.6 – 56.9 juta parameter) dibandingkan *Improved R-CNN* dengan VGG16 dengan 138 juta parameter [12], [28] dan *Faster R-CNN* dengan *ResNet50* dengan 26 juta parameter [13], [28]. Meskipun pada penelitian terdahulu nilai mAP 0.5-IoU YOLOv11 tercatat sebesar 92.1% [14] yang sedikit lebih rendah dibawah *Improved R-CNN* dengan VGG16 dan *Faster R-CNN* dengan *ResNet50* (94.06%), namun dengan performa tersebut sistem dapat dikatakan bekerja dengan baik dan andal dalam mendeteksi objek karena berada di atas ambang batas 90% [29].

Dalam penelitian ini, penggunaan YOLOv11-L dianggap ideal dalam memberikan keseimbangan antara ketepatan dan keberlanjutan sistem. Dengan jumlah parameter 25.3 juta, varian model ini memiliki ukuran yang lebih ringan dan proses komputasi yang lebih cepat dibandingkan dengan arsitektur R-CNN dan *Faster R-CNN* [24]. Jumlah parameter yang lebih kecil dapat mengurangi resiko terjadinya *overfitting* serta dapat meningkatkan efisiensi daya dan waktu komputasi [30]. Selain itu, YOLOv11-L menunjukkan kinerja mAPval 50-90 sebesar 53.4 dengan kecepatan inferensi yang kompetitif terhadap varian tertinggi yaitu YOLOv11-X, sehingga memenuhi kriteria sistem deteksi yang presisi namun tetap responsif jika diimplementasikan pada lapangan. Adapun rancangan pada penelitian ini dibagi menjadi 6 tahapan.

Tahapan pertama dalam perancangan yaitu melakukan persiapan *dataset* dengan mengumpulkan *dataset* penyakit tanaman cabai secara *hybird* melalui platform digital (*Kaggle* dan *Roboflow*) serta pengambilan sampel secara langsung di lapangan wilayah Gandus, Palembang.

Tahapan kedua yaitu *pre-processing dataset*, di mana semua data yang telah divalidasi oleh pakar dilakukan pemrosesan lebih lanjut melalui platform *Roboflow* sebelum dilatih dengan model. Tahapan *pre-processing dataset* pada penelitian ini dibedakan menjadi 3 bagian, yaitu:

- Tahapan penyesuaian ukuran gambar (*resizing*), di mana semua ukuran gambar pada *dataset* dilakukan penyesuaian dengan *input* model yaitu 640×640 px agar pemrosesan citra dapat berlangsung lebih cepat dan tidak banyak dalam menghabiskan memori [31] serta dapat mencapai nilai yang optimal [32].
- Tahapan anotasi gambar (*annotation image*), di mana semua data yang terkumpul diberikan label beserta *bounding box* untuk memandai lokasi suatu objek agar model dapat lebih mudah mengenali objek yang diinginkan [33].
- Tahapan augmentasi gambar (*augmentation image*), di mana gambar pada *class dataset* dengan jumlah sampel sedikit yang digunakan untuk proses *training* dilakukan modifikasi untuk menciptakan jumlah dan variasi baru melalui metode *rotating*, *flipping*, *scaling*, *cropping* dan *blurring* [33].

Tahapan ketiga yaitu *dataset splitting*, di mana tahap ini merupakan metode yang digunakan untuk membagi *dataset* [34] ke dalam 3 bagian dengan proporsi 80:10:10.

Tahapan keempat yaitu melakukan pelatihan model YOLOv11-L melalui *platform Kaggle Notebook* (Python 3.11.11) dengan memanfaatkan *accelerator GPU T4* $\times 2$. Pelatihan ini bertujuan agar model dapat mendeteksi penyakit tanaman cabai melalui 54 skenario pelatihan yang menggunakan 2 strategi utama yaitu membekukan lapisan *backbone* (*freeze backbone layer*) dan tanpa membekukan lapisan *backbone* (*doesn't freeze backbone layer*).

Tahapan kelima yaitu melakukan proses pengujian deteksi terhadap model YOLOv11-L yang telah dilatih menggunakan data uji (*test set*) sebesar 10% (450 gambar). Pada tahap ini, model dilakukan evaluasi guna mengukur kemampuan generalisasi dalam mendeteksi penyakit pada tanaman cabai yang belum pernah dilihat sebelumnya oleh model.

Tahapan keenam yaitu menyimpan model terbaik (*save best model*) berdasarkan nilai performa tertinggi melalui metrik evaluasi pada persamaan 1, 2, 3, 4, dan 5. Hal ini bertujuan agar model dengan kemampuan deteksi yang paling optimal dapat digunakan kembali pada tahapan implementasi dan pengujian lanjutan secara konsisten tanpa perlu melakukan pelatihan ulang.

3.3. Pengujian dan Evaluasi Hasil Pengujian

Pengujian pada penelitian ini bertujuan untuk memastikan sistem model memiliki kualitas yang baik [35] dalam mendeteksi penyakit pada tanaman cabai melalui penyesuaian *hyperparameter* pada model

YOLOv11-L. Konfigurasi *hyperparameter* yang diuji meliputi *learning rate* (0.01, 0.001, 0.0001), *batch size* (8, 16, 32), *epoch* (30, 50, 100) dan *optimizer* (Adam).

Evaluasi hasil pengujian model dilakukan untuk menilai kinerja dari sistem menggunakan *mean Average Precision* (mAP) sebagai metrik utama [36], [37], [38]. Selain itu, secara *de facto*, metrik mAP 0.5-IoU (mAP@50) menjadi standar umum yang digunakan dalam evaluasi deteksi objek [39]. Adapun metrik pendukung yang digunakan pada penelitian ini dijabarkan sebagai berikut:

- Precision*, merupakan metrik yang berfokus pada kemampuan model dalam memprediksi kelas positif secara akurat untuk meminimalkan kesalahan prediksi [40].

$$Precision = \frac{TP}{TP + FP} \quad (1)$$

- Recall*, merupakan metrik yang mengukur persentase kejadian positif aktual yang diidentifikasi model dengan benar sebagai positif benar [40].

$$Recall = \frac{TP}{TP + FN} \quad (2)$$

- Average Precision* (AP), merupakan nilai yang mencerminkan rata-rata akurasi model dengan menggambarkan nilai *precision* dan *recall* secara keseluruhan [41].

$$AP = \int_0^1 p(r)dr \quad (3)$$

- Mean Average Precision* (mAP), merupakan nilai yang dihitung dari nilai rata-rata AP setiap kelas dengan rentang 0 hingga 1 [40].

$$mAP = \frac{1}{n} \sum_{k=1}^{k=n} AP_k \quad (4)$$

- Intersection of Union* (IoU), merupakan nilai yang digunakan untuk mengukur tingkat tumpang tindih (*overlap*) antara *bounding box* prediksi model dengan *ground-truth* [40].

$$IoU = \frac{TP}{TP + FP + FN} \quad (5)$$

4. HASIL DAN PEMBAHASAN

Penelitian ini menguji 54 skenario pelatihan yang terbagi menjadi 2 strategi utama masing-masing 27 skenario yang menggunakan model YOLOv11-L dengan variasi *learning rate* (0.01, 0.001, 0.0001), *batch size* (8, 16, 32), *epoch* (30, 50, 100) dan *optimizer* (Adam).

4.1. Analisis Strategi 1 : Freeze Backbone Layer

Tabel 2. Hasil Pengujian Strategi 1

Hyperparameter	Precision	Recall	mAP@50
0.01;8;30;Adam	0,85250	0,45515	0,58581
0.01;8;50;Adam	0,83277	0,51856	0,61274
0.01;8;100;Adam	0,81868	0,54746	0,61741
0.01;16;30;Adam	0,81538	0,48161	0,58342
0.01;16;50;Adam	0,83348	0,55601	0,59877
0.01;16;100;Adam	0,81326	0,54043	0,59743
0.01;32;30;Adam	0,88703	0,48311	0,57982

Hyperparameter	Precision	Recall	mAP@50
0.01;32;50;Adam	0,91088	0,52950	0,57358
0.01;32;100;Adam	0,87767	0,56878	0,59849
0.001;8;30;Adam	0,78889	0,55269	0,61246
0.001;8;50;Adam	0,84213	0,53558	0,59568
0.001;8;100;Adam	0,79459	0,53608	0,59122
0.001;16;30;Adam	0,77929	0,56291	0,63149
0.001;16;50;Adam	0,77734	0,59154	0,61037
0.001;16;100;Adam	0,76852	0,56220	0,60952
0.001;32;30;Adam	0,77596	0,64259	0,63713
0.001;32;50;Adam	0,73871	0,57133	0,61135
0.001;32;100;Adam	0,74748	0,60150	0,60749
0.0001;8;30;Adam	0,75556	0,47799	0,59695
0.0001;8;50;Adam	0,78227	0,50877	0,60127
0.0001;8;100;Adam	0,82556	0,52489	0,61527
0.0001;16;30;Adam	0,78292	0,49513	0,60377
0.0001;16;50;Adam	0,79467	0,52924	0,59902
0.0001;16;100;Adam	0,80221	0,55897	0,60697
0.0001;32;30;Adam	0,80979	0,51547	0,61314
0.0001;32;50;Adam	0,80443	0,51640	0,60227
0.0001;32;100;Adam	0,80915	0,54701	0,60085

Berdasarkan hasil pengujian 27 skenario pada strategi 1 (Tabel 2), penggunaan *learning rate* 0.01 dapat menghasilkan precision yang tinggi mencapai > 0.9 pada *batch size* 32 dan *epoch* 50, namun memiliki *recall* yang rendah, di mana hal ini menunjukkan banyak objek penyakit tanaman cabai yang tidak terdeteksi oleh model deteksi YOLOv11-L. Konfigurasi terbaik pada strategi 1 ini dicapai oleh *learning rate* 0.001, *batch size* 32, *epoch* 30, dan *optimizer* Adam dengan nilai *precision* 0.77596, *recall* 0.64159 dan mAP@50 sebesar 0.63713. Hal ini menunjukkan bahwa *learning rate* 0.001 mampu memberikan keseimbangan performa yang lebih stabil pada kondisi *freeze backbone layer*.

4.2. Analisis Strategi 2 : Doesn't Freeze Backbone Layer

Tabel 3 Hasil Pengujian Strategi 2

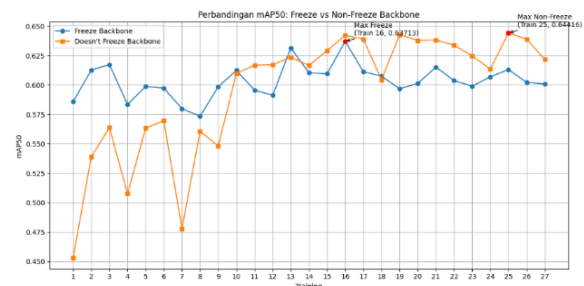
Hyperparameter	Precision	Recall	mAP@50
0.01;8;30;Adam	0,76028	0,20566	0,45327
0.01;8;50;Adam	0,81092	0,32031	0,53892
0.01;8;100;Adam	0,87533	0,42022	0,56385
0.01;16;30;Adam	0,76580	0,26717	0,50757
0.01;16;50;Adam	0,84578	0,37593	0,56338
0.01;16;100;Adam	0,85188	0,46435	0,56970
0.01;32;30;Adam	0,71311	0,24598	0,47776
0.01;32;50;Adam	0,87708	0,33744	0,56075
0.01;32;100;Adam	0,90569	0,46352	0,54824
0.001;8;30;Adam	0,81908	0,49917	0,60987
0.001;8;50;Adam	0,83533	0,55469	0,61686
0.001;8;100;Adam	0,79796	0,58657	0,61726
0.001;16;30;Adam	0,79512	0,54102	0,62365
0.001;16;50;Adam	0,77568	0,57597	0,61662
0.001;16;100;Adam	0,92356	0,56143	0,62926
0.001;32;30;Adam	0,80612	0,56699	0,64222
0.001;32;50;Adam	0,78809	0,56570	0,63904
0.001;32;100;Adam	0,76657	0,56489	0,60449
0.0001;8;30;Adam	0,75819	0,58971	0,64274
0.0001;8;50;Adam	0,80058	0,56633	0,63781
0.0001;8;100;Adam	0,82677	0,56897	0,63818

Hyperparameter	Precision	Recall	mAP@50
0.0001;16;30;Adam	0,79791	0,56265	0,63388
0.0001;16;50;Adam	0,82718	0,57106	0,62476
0.0001;16;100;Adam	0,80013	0,56586	0,61351
0.0001;32;30;Adam	0,82739	0,54792	0,64416
0.0001;32;50;Adam	0,80193	0,55838	0,63902
0.0001;32;100;Adam	0,76224	0,55577	0,62165

Pada strategi 2 (Tabel 3), pelatihan secara *end-to-end* dapat menghasilkan peningkatan performa yang lebih konsisten pada *learning rate* rendah. Pada konfigurasi *learning rate* yang cukup tinggi 0.01 menyebabkan ketidakstabilan pembaruan bobot dengan mAP@50 rendah (0.45 – 0.56). Namun, konfigurasi *learning rate* 0.0001, *batch size* 32, *epoch* 30 dengan *optimizer* Adam berhasil mencapai performa yang lebih optimal dengan nilai mAP@50 tertinggi sebesar 0.64416.

4.3. Perbandingan Strategi dan Evaluasi Visual

Perbandingan kedua strategi menunjukkan bahwa model dengan strategi 2 : *doesn't freeze backbone layer* (mAP@50 : 0.64416) lebih unggul dibandingkan *freeze backbone layer* (mAP@50 : 0.63713). Pelatihan secara *end-to-end* memungkinkan model menyesuaikan fitur tingkat rendah hingga tinggi secara spesifik terhadap karakteristik visual 6 *class dataset*. Kompleksitas visual yang mirip antar kelas berhasil diekstraksi oleh model deteksi YOLOv11-L dengan lebih optimal melalui strategi 2. Untuk memberikan gambaran visual yang lebih jelas mengenai pengaruh variasi *hyperparameter* terhadap kinerja model YOLOv11-L, hasil pengujian dengan membandingkan nilai mAP@50 ini divisualisasikan pada Gambar 1.



Gambar 1. Visualisasi Perbandingan Strategi 1 dan 2

4.4. Pengujian Data Uji (Test Set)

Setelah diperoleh konfigurasi model terbaik dengan nilai mAP@50 sebesar 0.64416 (64.42%) pada strategi 2 : *doesn't freeze backbone layer*, model tersebut akan dilakukan pengujian menggunakan data uji sebanyak 10% (45 gambar) untuk mengetahui kinerja model YOLOv11-L dalam kondisi nyata. Proses uji coba deteksi dilakukan dengan menetapkan nilai ambang batas (*threshold*) *Intersection of Union* (IoU) sebesar 0.5. Kemudian, evaluasi terhadap hasil uji coba dilakukan dengan membandingkan hasil deteksi model terhadap diagnosis pakar sebagai referensi kebenaran (*ground truth*), di mana hasil pengujian disajikan pada Tabel 4.

Tabel 4 Hasil Pengujian Data Uji (Test Set)

Hasil Deteksi Model	Hasil Deteksi Pakar	Keterangan
Layu Fusarium	Layu Fusarium	✓
Virus Kuning	Virus Kuning	✓
Daun Bercak Cokelat	Daun Bercak Cokelat	✓
Layu Fusarium	Layu Fusarium	✓
Daun Bercak Cokelat	Daun Bercak Cokelat	✓
Cabai Sehat	Cabai Sehat	✓
Daun Bercak Cokelat	Daun Bercak Cokelat	✓
Virus Kuning	Virus Kuning	✓
Daun Bercak Cokelat	Daun Bercak Cokelat	✓
Virus Kuning	Virus Kuning	✓
Daun Bercak Cokelat	Daun Bercak Cokelat	✓
Virus Kuning	Virus Kuning	✓
Cabai Sehat	Cabai Sehat	✓
Layu Fusarium	Layu Fusarium	✓
Cabai Sehat	Cabai Sehat	✓
Daun Bercak Cokelat dan Bounding Box Ranting Kayu	Daun Bercak Cokelat	½
Daun Keriting	Daun Keriting	✓
Cabai Sehat	Cabai Sehat	✓
Virus Kuning	Virus Kuning	✓
Layu Fusarium	Layu Fusarium	✓
Daun Bercak Cokelat	Daun Bercak Cokelat	✓
Daun Keriting	Daun Keriting & Cabai Sehat	½
Daun Keriting	Daun Keriting & Cabai Sehat	½
Layu Fusarium	Layu Fusarium	✓
Daun Keriting	Daun Keriting & Cabai Sehat	½
Virus Kuning	Virus Kuning	✓
Daun Bercak Cokelat	Daun Keriting	✗
Daun Keriting	Daun Keriting	✓
Layu Fusarium	Layu Fusarium	✓
Cabai Sehat	Cabai Sehat	✓
Daun Keriting	Daun Keriting	✓
Layu Fusarium	Layu Fusarium	✓
Daun Keriting	Daun Keriting	✓
Virus Kuning	Virus Kuning	✓
Daun Bercak Cokelat	Daun Bercak Cokelat	✓
Layu Fusarium	Layu Fusarium	✓
Antraknosa	Antraknosa	✓
Antraknosa	Antraknosa	✓
Antraknosa	Antraknosa	✓
Antraknosa	Antraknosa	✓
Antraknosa	Antraknosa	✓
Antraknosa	Antraknosa	✓
Cabai Sehat	Cabai Sehat	✓
Cabai Sehat	Cabai Sehat	✓

Berdasarkan analisis terhadap data uji sebanyak 45 gambar, didapatkan bahwa sebanyak 40 pengujian dikategorikan sebagai deteksi benar, 1 pengujian

sebagai deteksi salah dan 4 pengujian menghasilkan deteksi sebagian. Secara persentase, hasil deteksi kebenaran model YOLOv11-L mampu mencapai 93.33% dengan tingkat deteksi kesalahan dan sebagian sebesar 6.67%. Hasil ini menunjukkan bahwa model YOLOv11-L memiliki tingkat akurasi yang tinggi dalam mendeteksi penyakit pada tanaman cabai pada *threshold* IoU 0.5. Hal ini dikarenakan model telah mampu menangkap karakteristik visual esensial yang sesuai dengan klasifikasi aslinya, sehingga identifikasi dinyatakan benar apabila objek dominan dalam citra berhasil terdeteksi sesuai dengan kelas penyakit yang sesungguhnya, meskipun tidak seluruh objek dalam satu citra terdeteksi secara menyeluruh oleh *bounding box*.

5. KESIMPULAN

Berdasarkan hasil dan pembahasan yang telah dibahas terkait deteksi penyakit pada tanaman cabai menggunakan YOLOv11-L, dapat disimpulkan bahwa model deteksi yang dilatih (*training*) menggunakan 450 data gambar pada 6 *class* mampu memberikan kinerja deteksi yang cukup baik. Hasil terbaik dari pelatihan model menunjukkan nilai *mean Average Precision* pada *Intersection of Union* 0.50 (mAP@50) sebesar 64,42% pada strategi kedua dengan konfigurasi *hyperparameter training learning rate* 0.0001, *batch size* 32 dan *epoch* 30 dengan *optimizer* Adam tanpa membekukan lapisan *backbone* (*doesn't freeze backbone layer*). Pada tahap pengujian menggunakan data uji (*test set*) sebesar 10% dari total *dataset* (45 citra), model mampu menunjukkan tingkat akurasi deteksi sebesar 93,33% dengan tingkat kesalahan sebesar 6,67%. Hasil tersebut menunjukkan bahwa model YOLOv11-L memiliki potensi yang baik untuk digunakan dalam sistem deteksi penyakit tanaman cabai. Untuk penelitian selanjutnya, pengembangan dapat difokuskan pada peningkatan jumlah dan variasi *dataset*, optimasi *hyperparameter*, serta implementasi model secara *realtime* dengan perangkat *Internet of Things* (IoT) guna meningkatkan performa dan keandalan sistem di lapangan.

DAFTAR PUSTAKA

- [1] P. Lasaksi, "Analisis Peran Sektor Pertanian Pemerintah Terhadap Perekonomian," *Lentera Multidiscip. Stud.*, vol. 1, no. 3, pp. 165–171, 2023, [Online]. Available: <https://lentera.publikasiku.id/index.php/ln/article/view/23/78>
- [2] R. E. Prang, F. Tulusan, and V. Londa, "Implementasi Program Sentra Hortikultura di Desa Wulurmaatus Kecamatan Modinding Kabupaten Minahasa Selatan," *J. Adm. Publik*, vol. VIII, no. 4, pp. 282–290, 2023, doi: <https://doi.org/10.35797/jap.v8i4>.
- [3] B. P. dan P. P. K. Pertanian, "Rencana Strategis Pusat Penelitian dan Pengembangan Hortikultura 2020 - 2024," 2020.
- [4] N. Inaya, S. Meriem, and Masriany, "Identifikasi

- morfologi penyakit tanaman cabai (*Capsicum* sp.) yang disebabkan oleh patogen dan serangan hama lingkup kampus UIN Alauddin Makassar,” vol. 2, no. 1, pp. 8–15, 2022, doi: <https://doi.org/10.24252/filogeni.v2i1.27092>.
- [5] I. Wibisonya, “Analisis Risiko Harga Cabai Merah Keriting Di Kabupaten Cianjur Provinsi Jawa Barat,” *J. Agribus. Sci. Rural Dev.*, vol. 1, no. 2, pp. 23–29, 2022, doi: <https://doi.org/10.32639/jasrd.v1i2.111>.
- [6] K. Khairunnas, H. Gemasih, and Hendri Syahputra, “Rancang Bangun Sistem Pakar Penyakit Tanaman Cabai Menggunakan Metode Naïve Bayes Berbasis Web,” *Ocean Eng. J. Ilmu Tek. dan Teknol. Marit.*, vol. 1, no. 3, pp. 74–88, 2022, doi: <https://doi.org/10.58192/ocean.v1i3.374>.
- [7] Darmansah and N. W. Wardani, “Analisa Penyebab Kerusakan Tanaman Cabai Menggunakan Metode K-Means,” *JATISI (Jurnal Tek. Inform. dan Sist. Informasi)*, vol. 7, no. 2, pp. 126–134, 2020, doi: <https://doi.org/10.35957/jatisi.v7i2>.
- [8] H. Sudibyo, M. B. Ulum, and R. Efendi, “Sistem Pakar Mengidentifikasi Penyakit pada Tanaman Cabai,” *Innov. J. Soc. Sci. Res.*, vol. 3, no. 4, pp. 5922–5934, 2023, doi: <https://doi.org/10.31004/innovative.v3i4>.
- [9] M. Laely, I. G. P. S. Wijaya, and A. Aranta, “Sistem Pakar Diagnosis Tanaman Cabai dengan Metode Forward Chaining dan Dempster Shafer,” *J. Teknol. Informasi, Komputer, dan Apl.*, vol. 2, no. 2, pp. 268–279, 2020, doi: <https://doi.org/10.29303/jtika.v2i2.118>.
- [10] P. A. Wangga, F. E. Neno, and T. Kurra, “Sistem Pakar untuk Mendiagnosa Penyakit Tanaman Cabai Menggunakan Metode Forward Chaining,” *Mars J. Tek. Mesin, Ind. Elektro dan Ilmu Komput.*, vol. 2, no. 5, pp. 148–156, 2024.
- [11] L. M. Harahap, T. G. Pakpahan, R. A. Wijaya, and A. Z. Nasution, “Dampak Transformasi Digital pada Agribisnis: Tantangan dan Peluang bagi Petani di Indonesia,” *Publ. Ilmu Tanam. dan Agribisnis*, vol. 1, no. 2, pp. 99–108, 2024, doi: <https://doi.org/10.62951/botani.v1i2.55>.
- [12] H. Rehana, M. Ibrahim, and M. H. Ali, “Plant Disease Detection using Region-Based Convolutional Neural Network,” *Elsevier*, vol. 2, pp. 1–19, 2023, doi: <https://doi.org/10.56294/dm2023135>.
- [13] D. B. R. S. Reddy, D. G. B. Madhavi, C. H. S. Lakshmi, D. K. V. Nagendra, and D. R. Sridevi, “Detection of Disease in Maize Plant Using Deep Learning,” *Alinteri J. Agric. Sci.*, vol. 36, no. 2, pp. 82–88, 2021, doi: <https://doi.org/10.47059/alinteri/v36i2/ajas21118>.
- [14] A. Sharma, V. Kumar, and L. Longchamps, “Comparative performance of YOLOv8, YOLOv9, YOLOv10, YOLOv11 and Faster R-CNN models for detection of multiple weed species,” *Smart Agric. Technol.*, vol. 9, no. 100648, 2024, doi: <https://doi.org/10.1016/j.atech.2024.100648>.
- [15] W. A. N. Zai, P. K. T. Mendrofa, A. B. S. Waruwu, P. H. Telaumbanua, and A. B. Ndraha, “Hama dan Penyakit pada Tanaman Cabai Rawit yang Dibudidayakan di dalam Polybag,” *Flora J. Kaji. Ilmu Pertan. dan Perkeb.*, vol. 2, no. 1, pp. 139–151, 2025, doi: <https://doi.org/10.62951/flora.v2i1.254>.
- [16] Y. Devianto, S. Dwiasnati, B. Sukowo, A. Fauzi, and K. A. Baihaqi, “Penerapan Technique for Order Performance by Similarity to Ideal Solution (TOPSIS) untuk Mendiagnosa Penyakit Bercak Daun Cabai,” *MALCOM Indones. J. Mach. Learn. Comput. Sci.*, vol. 3, no. 2, pp. 136–142, 2023, doi: <https://doi.org/10.57152/malcom.v3i2.850>.
- [17] H. Hamidson et al., “Pengelolaan Tanaman Terpadu Pada Cabai Merah (*Capsicum annum* L.) Terhadap Perkembangan Penyakit Bercak Daun Di Kecamatan Indralaya Utara Kabupaten Ogan Ilir Sumatera Selatan,” *Pros. Semin. Nas. Perlindungan Tanam.*, vol. 2, pp. 27–42, 2024, [Online]. Available: <https://semnas.bppf-unib.com/index.php/perlintan/article/view/7>.
- [18] G. Djajakirana and P. H. Sijabat, “Pengaruh Media Tanam Terhadap Pertumbuhan Bibit Tanaman Cabai (*Capsicum annum* L.) dan Intensitas Serangan Layu Fusarium (*Fusarium oxysporum* Schlecht),” *J. Ilmu Tanah dan Lingkung.*, vol. 24, no. 2, pp. 62–66, 2022, doi: <https://doi.org/10.29244/jitl.24.2.62-66>.
- [19] Heriyanto, “Kajian Pengendalian Penyakit Layu Fusarium Dengan Trichoderma Pada Tanaman Tomat,” *J. Trit.*, vol. 10, no. 1, pp. 45–58, 2019.
- [20] Al-Hadiat, M. Taufik, R. M. G. HS, and Syair, “Efisiensi Penularan Pepper yellow leaf curl Indonesia virus (PepYLCIV) dengan Kutukebul, Kejadian Penyakit dan Pertumbuhan Tanaman cabai,” *Berk. Penelit. Agron. (Journal Agron. Res.)*, vol. 10, no. 2, pp. 106–117, 2022.
- [21] P. R. Shingote et al., “An Overview of Chili Leaf Curl Disease: Molecular Mechanisms, Impact, Challenges, and Disease Management Strategies in Indian Subcontinent,” *Front. Microbiol.*, vol. 13, no. June, pp. 1–18, 2022, doi: <https://doi.org/10.3389/fmicb.2022.899512>.
- [22] D. A. Roberts and S. Yaida, “The Principles of Deep Learning Theory An Effective Theory Approach to Understanding Neural Networks,” *arXiv*, 2022, doi: <https://doi.org/10.1017/9781009023405>.
- [23] W. Nugraha and A. Sasongko, “Hyperparameter Tuning on Classification Algorithm with Grid Search,” *Sist. J. Sist. Inf.*, vol. 11, no. 2, pp. 391–401, 2022, [Online]. Available: <http://sistemasi.ftik.unisi.ac.id>.
- [24] A. Julianto, A. Sunyoto, and F. W. Wibowo, “Optimasi Hyperparameter Convolutional Neural Network Untuk Klasifikasi Penyakit Tanaman Padi,” *Tek. Teknol. Inf. dan Multimed.*, vol. 3, no. 2, pp. 98–105, 2022, doi: <https://doi.org/10.46764/teknimedia.v3i2.77>.
- [25] C. Li et al., “YOLOv6 v3.0: A Full-Scale

- Reloading,” 2023. [Online]. Available: <https://arxiv.org/abs/2301.05586>
- [26] D. N. Alfarizi, R. A. Pangestu, D. Aditya, M. A. Setiawan, and P. Rosyani, “Penggunaan Metode YOLO Pada Deteksi Objek: Sebuah Tinjauan Literatur Sistematis,” *J. Artif. Intel. dan Sist. Penunjang Keputusan*, vol. 1, no. 1, pp. 54–63, 2023, [Online]. Available: <https://jurnalmahasiswa.com/index.php/aidanspk>
- [27] F. Ramadah, P. D. Wibawa, and A. Rizal, “Fire Detection System Using Image Processing on Webcam with Yolov3 Method,” *e-Proceeding Eng.*, vol. 9, no. 2, pp. 226–231, 2022.
- [28] O. Elharrouss, Y. Akbari, N. Almaadeed, and S. Al-maadeed, “Backbones-Review: Feature Extraction Networks for Deep Learning and Deep Reinforcement Learning Approaches,” 2022. [Online]. Available: <https://arxiv.org/abs/2206.08016>
- [29] S. Yuliyanto, N. F. Amani, F. Akhyar, and K. Usman, “Sistem Inspeksi Permukaan Baja Berbasis Deep Learning Menggunakan Metode Anchor-Free,” *J. Ilm. Tek. Mesin, Elektro dan Komput.*, vol. 2, no. 3, pp. 184–190, 2022, doi: 10.51903/juritek.v2i3.364.
- [30] C. Y. Wijaya, “Are We Undervaluing Simple Models?,” kdnuggets. Accessed: Nov. 03, 2024. [Online]. Available: <https://www.kdnuggets.com/are-we-undervaluing-simple-models>
- [31] T. Hidayat, “Penerapan Algoritma Convolutional Neural Network Dalam Identifikasi Citra Daging Merah Konsumsi Jenis Lokal,” Universitas Nusa Mandiri, 2021. [Online]. Available: https://repository.nusamandiri.ac.id/repo/files/242673/download/01-tesis-202111303-taopik-hidayat-dikompresi_Compressed.pdf#page=65.10
- [32] A. Yudamson, S. R. Sulistiyanti, S. Alam, F. A. Setyawan, and T. Yulianti, “Substraksi RGB untuk Identifikasi Jenis Daging Konsumsi Berbasis Pengolahan Citra,” *Electr. – J. Rekayasa dan Teknol. Elektro Substraksi*, vol. 14, no. 2, pp. 52–55, 2020, doi: 10.23960/elc.v14n2.2148.
- [33] N. Fatkhin and A. Fadjeri, “Pembelajaran Mesin Untuk Deteksi Helm Keselamatan Menggunakan Algoritma YOLOv8,” *J. Ilm. Sinus*, vol. 22, no. 2, pp. 77–86, 2024, doi: 10.30646/sinus.v22i2.843.
- [34] V. Ariyani, P. Putri, A. B. Prasetijo, and D. Eridani, “Perbandingan Kinerja Algoritme Naïve Bayes Dan K-Nearest Neighbor (Knn) Untuk Prediksi Harga Rumah,” *J. Ilm. Tek. Elektro*, vol. 24, no. 2, pp. 162–171, 2022.
- [35] D. Debiyanti, S. Sutrisna, B. Budrio, A. K. Kamal, and Y. Yulianti, “Pengujian Black Box pada Perangkat Lunak Sistem Penilaian Mahasiswa Menggunakan Teknik Boundary Value Analysis,” *J. Inform. Univ. Pamulang*, vol. 5, no. 2, pp. 162–166, 2020, doi: 10.32493/informatika.v5i2.5446.
- [36] Firdaus, M. Wibowo, R. Tullah, and W. Ricesa, “Studi Perbandingan Algoritma YOLO dan FOMO untuk Object Detection pada Perangkat ESP32-CAM,” *INSECT Informatics Secur.*, vol. 11, no. 1, pp. 44–54, 2025, doi: <https://doi.org/10.33506/insect.v11i1.4289>.
- [37] R. D. Saputra and D. H. Fudholi, “Model Mobile untuk Deteksi Objek pada On-Shelf Availability Produk Retail,” *Automata*, vol. 4, no. 2, 2023, [Online]. Available: <https://journal.uui.ac.id/AUTOMATA/article/view/28610>
- [38] R. Padilla, W. L. Passos, T. L. B. Dias, S. L. Netto, and E. A. B. Da Silva, “A comparative analysis of object detection metrics with a companion open-source toolkit,” *Electron.*, vol. 10, no. 3, pp. 1–28, 2021, doi: 10.3390/electronics10030279.
- [39] Z. Zou, K. Chen, Z. Shi, Y. Guo, and J. Ye, “Object Detection in 20 Years: A Survey,” *Proc. IEEE*, vol. 111, no. 3, pp. 257–276, 2023, doi: 10.1109/JPROC.2023.3238524.
- [40] Z. Zhao, L. Alzubaidi, J. Zhang, Y. Duan, and Y. Gu, “A comparison review of transfer learning and self-supervised learning: Definitions, applications, advantages and limitations,” *Expert Syst. Appl.*, vol. 242, pp. 1–38, 2024, doi: 10.1016/j.eswa.2023.122807.
- [41] Z. S. Hidayat, Y. A. Wijaya, and R. Kurniawan, “Optimizing YOLOv8 For Autonomous Driving: Batch Size For Best Mean Average Precision (mAP),” *J. Tek. Inform.*, vol. 5, no. 4, pp. 1147–1153, 2024, [Online]. Available: <https://jutif.if.unsoed.ac.id/index.php/jurnal/article/download/1626/541/9955>