

KLASIFIKASI PENYAKIT TANAMAN CABAI RAWIT DILENGKAPI DENGAN SEGMENTASI CITRA DAUN DAN BUAH MENGGUNAKAN YOLO V7

CLASSIFICATION OF CHILI PLANT DISEASES EQUIPPED WITH LEAF AND FRUIT IMAGE SEGMENTATION USING YOLO V7

Masrur Anwar¹, Yosi Kristian², Endang Setyati³

^{1,2,3}Program Pascasarjana Magister Teknologi Informasi, Institut Sains dan Teknologi Terpadu Surabaya

masruranwar19@gmail.com

ABSTRACT

Diseases that attack chili plants can be diagnosed early by observing symptoms or changes that occur in the leaves and fruit of the chili plant. However, diseases or pests that attack chili plants within a single plant can vary. In this study, YOLO v7 was used to perform leaf and chili segmentation on images, and the segmented results were then classified for chili plant disease using Deep Convolutional Neural Network (DCNN) Transfer Learning with the Fine Tuning method. The test results of the constructed model showed that the Yolo v7 segmentation accuracy was 0.970 on mAP50 when performing chili plant leaf and fruit segmentation. For the DCNN model testing with transfer learning method using the EfficientNetV2M based model, an accuracy value of 0.912 was obtained for leaf disease classification and an accuracy of 0.889 was obtained for chili fruit classification.

Keywords: Chili Plant Diseases; Classification; Transfer Learning, Yolo v7 segmentation

ABSTRAK

Penyakit yang menyerang tanaman cabai dapat didiagnosa secara dini dengan mengamati gejala-gejala atau perubahan yang timbul pada daun dan buah cabai. Namun, penyakit atau hama yang menyerang tanaman cabai dalam satu tanaman dapat bervariasi. Pada penelitian ini YOLO v7 digunakan untuk melakukan segmentasi daun dan cabai pada citra atau gambar yang kemudian hasil segmentasi daun serta cabai akan diklasifikasikan penyakit pada tanaman cabai dengan menggunakan Deep Convolutional Neural Network (DCNN) Transfer learning dengan metode Fine Tuning. Dari hasil pengujian terhadap model yang dibangun didapatkan nilai akurasi segmentasi Yolo v7 sebesar 0.970 pada mAP50 dalam melakukan segmentasi daun dan buah tanaman cabai. Untuk pengujian model DCNN dengan metode transfer learning yang menggunakan based model EfficientNetV2M didapatkan nilai akurasi 0.912 untuk klasifikasi penyakit daun dan akurasi 0.9 untuk klasifikasi buah cabai.

Kata Kunci: Penyakit Tanaman Cabai; Classification; Transfer Learning, Segmentasi Yolo v7

PENDAHULUAN

Cabai merupakan salah satu komoditas pertanian atau tanaman sayuran tertinggi dibandingkan dengan komoditas sayuran lainnya (Naully, 2020). Komoditas cabai saat ini memiliki harga yang fluktuatif dan sering kali mengalami peningkatan harga yang signifikan dipasaran yang dipengaruhi karena keterbatasan persediaan cabai (Farid & Subekti, 2012). Dalam tingkatan petani cabai, pada musim penghujan seringkali mengalami gagal panen yang di akibatkan oleh banyaknya penyakit yang menyerang tanaman cabai. Namun, penyakit yang menyerang tanaman cabai dapat didiagnosa

secara dini dengan mengamati gejala-gejala atau perubahan yang timbul pada daun dan buah cabai (Zikra et al., 2021).

Penyakit atau hama yang menyerang tanaman cabai sangat dalam satu tanaman bisa bervariasi. Berikut merupakan penyakit atau hama tanaman cabai (1) Penyakit cabai yang terinfeksi jamur *Fusarium SP* dengan memiliki ciri bercak coklat kering atau kuning pada daun cabai (2) Penyakit virus kuning dengan memiliki ciri daun berwarna kuning (3) Penyakit buah antraknosa dengan memiliki ciri buah cabai mempunyai bercak coklat kering seperti terbakar (4) Penyakit buah busuk dengan memiliki ciri buah akan busuk

berwarna hitam pada bagian dalam namun kulit buah terlihat segar (Meilin, 2014). Semua penyakit yang muncul harus diidentifikasi secara cepat sehingga petani dapat memberikan pestisida agar tanaman cabai memiliki hasil yang diharapkan. Implementasi wawasan penyakit tanaman cabai tidak mudah dan tidak semua petani memahami tentang tanaman cabai dan penyakit atau hama sehingga pada saat terjadi gejala-gejala tidak teridentifikasi secara dini.

Saat ini dalam computer vision memiliki banyak pengembangan metode dimana pengembangan metode memberikan perubahan terhadap hasil yang lebih akurat serta pemrosesan yang lebih cepat hingga mendekati hasil deteksi *realtime*. *YOLO (Yolo Only Look Once)* merupakan sebuah algoritma *computer vision* untuk melakukan pengenalan obyek yang lebih cepat dari *R-CNN* dan *DPM* dengan hasil deteksi 45 *fps* (Redmon et al., 2016; Erlina & Fikri, 2023). Saat ini *YOLO* memiliki banyak versi dari pengembangan *YOLO v1* yaitu *YOLO v1* (Redmon et al., 2016), *YOLO9000*(Redmon & Farhadi, 2016), *YOLO v3*(Redmon & Farhadi, 2018), *YOLO v4*(Bochkovskiy et al., 2020) dan yang terbaru adalah *YOLO v7* (Wang et al., 2022).

Masing-masing versi memiliki keunggulan serta penyempurnaan algoritma *YOLO* dari versi sebelumnya sehingga didapatkan akurasi lebih baik serta pemrosesan yang lebih cepat dalam melakukan pengenalan obyek. Dalam *YOLO v7* mengusulkan metode trainable bag-of-freebies untuk melakukan peningkatan *real-object-detection* sehingga, didapatkan peningkatan kecepatan deteksi obyek antara 5 sampai 160 *fps* dengan nilai akurasi tertinggi mencapai 56.8 % AP (Wang et al., 2022).

Pada penerapan *Computer Vision* dalam bidang pertanian terdapat beberapa penelitian sebelumnya dengan melakukan klasifikasi penyakit pada daun tanaman dengan menggunakan *Convolutional Neural Network (CNN)* yang telah

diperbaiki dan daun tanaman yang digunakan untuk identifikasi penyakit bervariasi dengan 10 jenis daun tanaman (Hang et al., 2019). Selanjutnya pada penerapan dan penggabungan *YOLO* dengan *CNN* dilakukan penelitian untuk identifikasi leukosit pada leukimia dengan fungsi *YOLO* yang berfokus dengan deteksi obyek leukosit pada leukimia yang kemudian untuk *CNN* berfokus pada klasifikasi leukosit yang diklasifikasikan menjadi kelas *Lymph*, *Mono* dan *Neutro* (Abas et al., 2022).

Pada penelitian yang berfokus untuk mendeteksi penyakit tanaman cabai yang dilakukan menggunakan metode *Gray Level Co-Occurrence Matrix* untuk ekstraksi citra daun cabai yang kemudian dilakukan klasifikasi multiclass dengan menggunakan *Support vector machine* (Zikra et al., 2021). Hasil dari penelitian tersebut didapatkan nilai akurasi 95% dengan waktu komputasi 3 sampai 3.7 untuk mendeteksi penyakit satu citra daun cabai. Pada penelitian sebelumnya terkait deteksi penyakit pada tanaman cabai cukup membantu namun kurang memiliki waktu komputasi yang cepat. Kelas penyakit yang dilakukan deteksi juga hanya berfokus pada citra daun dan belum melakukan deteksi pada citra buah cabai (Meilin, 2014).

Berdasarkan paparan diatas maka, peneliti akan melakukan penelitian dengan topik klasifikasi penyakit pada tanaman cabai rawit (*Capsicum annum L.*) yang dilengkapi dengan segmentasi daun dan buah cabai. Pada penelitian ini *YOLO v7* akan digunakan untuk melakukan segmentasi daun dan cabai pada citra atau gambar yang kemudian hasil segmentasi daun serta cabai akan diklasifikasikan penyakit pada tanaman cabai dengan menggunakan *Deep Convolutional Neural Network Transfer Learning (DCNN)*. Adapun penelitian ini berkontribusi untuk:

- a. Membangun dataset penyakit daun dan buah cabai
- b. Membuat model deteksi penyakit tanaman cabai rawit

- c. Membangun program untuk deteksi penyakit tanaman cabai rawit yang dilengkapi dengan segmentasi daun dan buah cabai

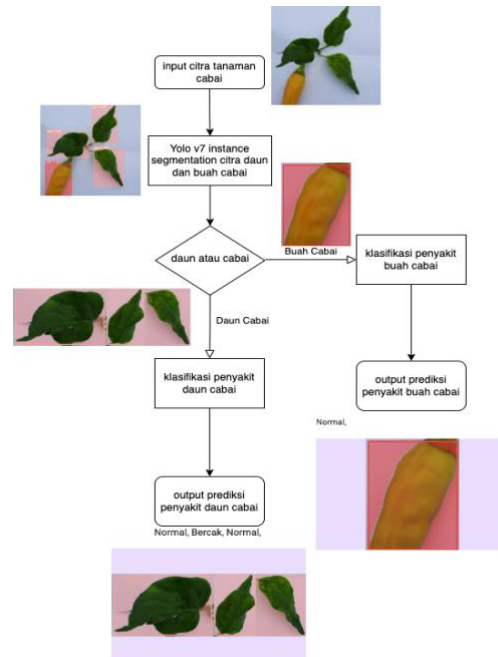
METODE

A. Analisa Dan Perancangan

Analisa dan perancangan program akan menggambarkan *input* dan *output* dari program yang dibangun serta menentukan model yang dibutuhkan. Dalam perancangan yang digambarkan pada flowchat diagram memiliki kebutuhan 2 model *DCNN* yaitu model untuk deteksi penyakit daun dan model untuk deteksi penyakit cabai yang dimana *input* dari citra. Dalam satu input citra yang di upload pada program hasil dari prediksi bisa beberapa dan tergantung pada jumlah citra duan dan cabai. *Yolo v7* segementasi bertugas untuk melakukan klasifikasi dalam satu citra tanaman menjadi beberapa citra daun dan buah cabai. Berikut pada gambar 1 merupakan *flow chart* diagram aplikasi deteksi penyakit daun dan buah cabai.

B. Building Dataset

Dataset yang digunakan pada penelitian ini didapatkan dari pertanian cabai yang berada di kota Malang dengan usia cabai yang bervariasi antar 2 bulan hingga 5 bulan atau cabai siap untuk panen. Dataset pada penelitian ini dibedakan menjadi dua dataset (Abas et al., 2022) yaitu yang pertama dataset untuk *segmentation-instance* yang digunakan untuk segementasi menentukan beberapa citra daun atau buah pada satu citra tanaman cabai. Kemudian dataset yang kedua adalah dataset klasifikasi penyakit yang terdapat pada tanaman daun dan buah cabai.



Gambar 1. Flow chat diagram aplikasi deteksi penyakit daun dan buah cabai



Gambar 2. Labelling dataset segmentasi Yolo v7




1) *Dataset segmentation-instance*: Dataset untuk melakukan train instance segmentation berbeda dengan deteksi obyek biasa. Proses pelabelan untuk dataset instance segmentation harus menentukan area obyek yang presisi dengan obyek yang ada bukan hanya dengan memberikan tanda area kemungkinan terdapat obyek seperti yang ditunjukkan pada Gambar 2. Labelling dataset untuk *segmentation-instance* menggunakan tools dari *Roboflow* dengan keseluruhan citra 1025 yang terdiri dari dua kelas segmentasi yaitu *fruit-chili* dan *leaf-chili*. Dataset *segmentation-instance* dilakukan splitting untuk menentukan jumlah dataset yang akan di gunakan untuk training YOLO v7 sebesar 73%, penggunaan untuk validation 23%, dan

testing dataset sebesar 5% dengan total sebaran kelas untuk fruit-chili 792 serta leaf-chili 595. Dataset yang akan digunakan dilakukan resize ukuran citra dengan ukuran pixel 640 x 640 pixel dan dengan export jenis dataset menjadi format *Yolo v7 PyTorch*(Wang et al., 2022). Dataset secara yang digunakan tersedia pada link berikut <https://universe.roboflow.com/tesis-ttvii/chili-segmentation-dw63r>

2) *Dataset classification:*




Collecting dataset untuk klasifikasi penyakit tanaman cabai didapatkan dari hasil prediksi citra daun dan buah cari menggunakan YOLO v7 segmentation dan kemudian memberikan *masking* pada obyek citra seperti pada penelitian abas et al (Abas et al., 2022). Pemberian *masking* terhadap obyek yang bukan bagian dari obyek daun atau buah bertujuan untuk mengurangi obyek dari daun daun dan cabai sehingga diharapkan nilai akurasi tetap tinggi walaupun jumlah dataset yang tersedia tidak begitu besar. Berikut pada tabel 1 adalah sebaran jumlah dan ciri-ciri setiap kelas penyakit yang terdapat pada tanaman daun cabai datan tabel 2 keas penyakit buah cabai (Meilin, 2014)(Zikra et al., 2021). Dataset penyakit daun tersedia pada link berikut <https://universe.roboflow.com/tesis-ttvii/leaf-disease-e34h5> dan untuk penyakit buah cabai pada <https://app.roboflow.com/tesis-ttvii/chili-t51wv/overview>

Tabel 1. Kelas Penyakit Daun Cabai

Nama kelas	Ciri-ciri	Foto	Jumlah
Bercak daun	Memiliki bercak coklat atau hitam pada daun		358
Kuning Daun	Warna daun berwarna kuning		258
Normal Daun	Memiliki ciri daun yang berwarna hijau dan tanpa ada bercak atau bintik		717

Tabel 2. Kelas Penyakit Buah Cabai

Nama Kelas	Ciri-ciri	Foto	jumlah
------------	-----------	------	--------

Cabai Busuk	Memiliki bintik hitam atau berwarna hitam		229
Cabai Normal	Memiliki warna hijau atau merah dan tanpa ada bintik hitam atau coklat		463
Cabai Patek	Memiliki bagian cabai yang terlihat kering seperti terbakar		278

C. *Implementation dan uji coba*

Proses implementasi program deteksi penyakit pada tanaman cabai dilakukan train terhadap beberapa model yaitu Yolo v7 dengan dataset instance segmentation, model DCNN klasifikasi daun cabai dengan dataset cabai, dan yang terakhir DCNN klasifikasi buah cabai dengan dataset cabai. Pada model DCNN dengan transfer learning untuk daun dan buah cabai menggunakan teknik fine tuning dimana, fine tuning terdiri dari *unfreezing* beberapa lapisan atas (*top layer*) yang dibekukan (*freeze*) dan bersama-sama melatih pengklasifikasi yang baru ditambahkan dan lapisan atau *layer* model yang tidak dibekukan. *Fine-tuning* mengoptimalkan keduanya bobot pengklasifikasi serta bobot beberapa atau semua lapisan dasar model pra-terlatih.

Dalam melakukan pengujian untuk Yolo v7 dilakukan dengan menggunakan metode mAP (mean Average Precision) untuk mengukur akurasi deteksi dari obyek yang telah diprediksi oleh Yolo. Nilai dari Average Precision lebih besar dari 0 hingga 1 yang nantinya akan digunakan untuk melakukan perhitungan mAP dengan rumus sebagai berikut

$$mAp = \frac{1}{N} \sum_{i=0}^n AP_i$$

Dimana nilai *N* adalah jumlah data *AP* (Average Precision). Nilai dari *AP* didapatkan dari persamaan berikut:

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

Berikut merupakan penjelasan rumus pada bagian 2 dan 3 antara lain: (1) *TP* (True Positif) merupakan kondisi ketika model mengklasifikasikan sebagai benar (True) dan nilai aktual nya adalah benar (True), (2) *TN* (True Negatif) kondisi ketika model mengklasifikasikan sebagai benar (True) dan nilai aktual nya adalah salah(False), (3) *FP* (False Positif) kondisi ketika model mengklasifikasikan sebagai salah (False) dan nilai aktualnya adalah benar(True), *FN* (False Negatif) kondisi Ketika model mengkalsifikasikan sebagai salah (False) dan nilai aktualnya adalah salah (False).

Ketika menentukan Average Precision harus didapatkan deteksi obyek dengan pencarian prediski dengan perhitungan IoU (intersection of union) berikut merupakan rincian persamaan IoU :

$$IoU = \frac{\text{area of overlap}}{\text{area of union}}$$

Dimana *area of overlap* dapat diistilahkan dengan area tumpang tindih yang di prediksi dari bouding box. Sedangkan *area of union* adalah area penyatuan atau cakupan yang diprediksi dari bouding box. Kemudian nilai yang didapatkan $IoU > 0.5$ maka didapatkan True Positif sedangkan $IoU < 0.5$ maka didapatkan True Negatif.

HASIL DAN PEMBAHASAN

A. *Yolo v7 instance-segmentation*

Yolo v7 yang ditulis oleh Chien-yao Wang melakukan peningkatan pada versi sebelumnya dengan beberapa metode yang diusulkan antara lain : (1) Merancang metode *trainable bag-of-freebies* yang digunakan untuk meningkatkan akurasi deteksi obyek tanpa meningkatkan biaya inferensi (*cost inference*) (2) Untuk evolusi metode deteksi objek, ditemukan dua masalah baru yaitu bagaimana modul yang diparameterisasi (*re-parameterized*) ulang menggantikan modul asli, dan bagaimana strategi penetapan label dinamis menangani

penugasan ke lapisan keluaran yang berbeda (3) Dengan mengusulkan metode "*extend*" dan "*compound scaling*" untuk detektor objek waktu nyata (*real object detector*) yang dapat secara efektif memanfaatkan parameter dan komputasi (4) Metode yang diusulkan dalam *YOLO v7* dapat secara efektif mengurangi sekitar 40% parameter dan 50% komputasi detektor objek *real-time* mutakhir, dan memiliki kecepatan inferensi yang lebih cepat dan akurasi deteksi yang lebih tinggi(Wang et al., 2022).

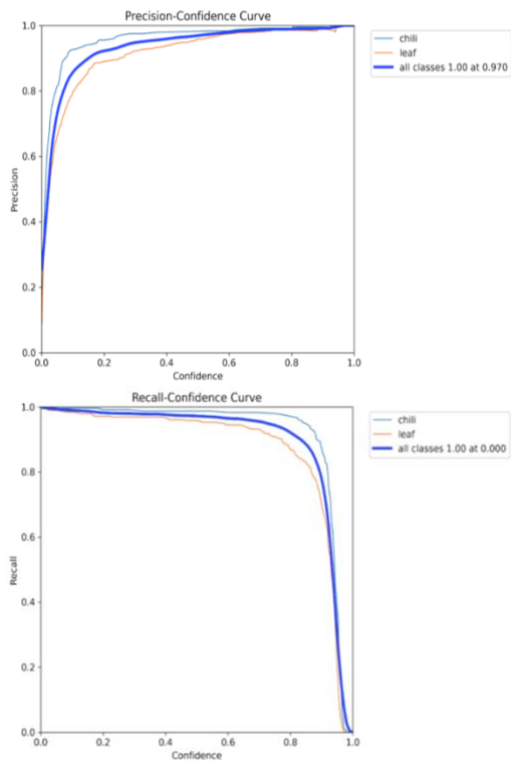
Dalam modul implementasi metode *Yolo v7* memiliki beberapa modul klasifikasi antara lain: (1) *object detection*, (2) *image classification*, (3) *pose estimation*, dan (4) *instance segmentation*. Pada penelitian ini implementasi *Yolo v7* menggunakan instance segmentation atau masking terhadap obyek yang terdeteksi sebagai daun atau buah cabai.

Proses training model *Yolo v7* instance segmentation menggunakan dataset citra daun dan buah tanaman cabai dengan ukuran 640 x 640 pixel pada *Google Colab* dengan jenis runtime GPU 15 *Gigabyte*. Pada tabel 3 merupakan hasil percobaan untuk menentukan model *Yolo v7* terbaik untuk deteksi daun dan buah cabai dengan 100 *epoch* dengan 16 *bacth* citra.

Tabel 3. Hasil Ujicoba Model Yolo V7

Class	Instances	Box precision	Box recall	Box mAP50
All class	617	0.972	0.964	0.986
Chili	235	0.984	0.983	0.994
Leaf	382	0.975	0.945	0.979

Pada gambar 3 merupakan hasil dari grafik perhitungan *precision-recall curve* dengan menunjukkan perbandingan keseimbangan hasil akurasi antar kelas dengan all class. Nilai akurasi 0.986 adalah nilai akurasi model terbaik dari beberapa kali epoch yang dilakukan saat training.



Gambar 3. Precision Dan Recall Model Segmentasi Yolo v7

Dari hasil model *Yolo v7* untuk segmentasi dilakukan modifikasi prediksi pada saat *masking* citra dengan melakukan *invert* hasil *masking* pada area *box prediction* yang bertujuan untuk menutup area yang bukan terdeteksi sebagai obyek. Setelah hasil *box prediction* didapatkan maka, dilakukan *cropping* setiap hasil prediksi yang digunakan untuk proses klasifikasi penyakit tanaman pada daun atau buah cabai. Pada gambar 4 merupakan hasil dari prediksi dengan model *Yolo v7* yang sesudah di lakukan modifikasi dan sebelum dilakukan modifikasi.



Hasil Yolo v7
segmentation

Hasil modifikasi
Yolo v7
segmentasi

Gambar 4. Modifikasi hasil output masking Yolo v7

Seluruh dataset yang telah di *training* menggunakan *Yolo v7* dilakukan *predict* menggunakan model *Yolo v7* sebagai dasar pembuatan dataset penyakit daun dan buah cabai. Hasil *predict* tersebut merupakan hasil *cropping* segmentasi citra daun dan buah cabai. Tidak semua obyek yang terdeteksi sebagai citra daun dan buah cabai dijadikan pembuatan dataset karena terdapat beberapa citra yang memiliki ukuran pixel terlalu kecil dan tidak sesuai dengan kebutuhan dari model DCNN yaitu 224 x 224 pixel.

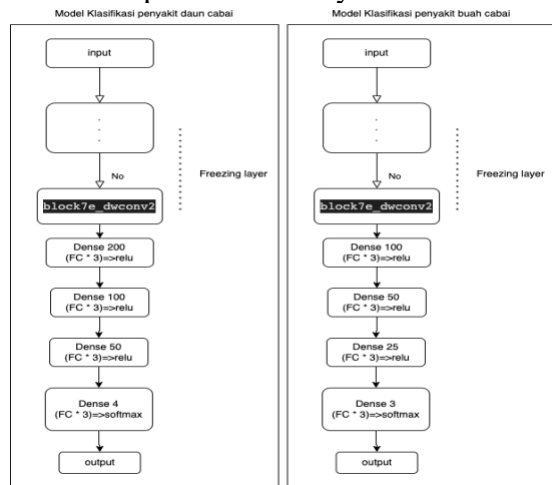
B. DCNN transfer learning

DCNN model pada penelitian ini menggunakan model dari *EfficientNetV2* yang kemudian dilakukan *transfer learning* dengan teknik *fine tuning*. Model *EfficientNetV2* memiliki keunggulan lebih cepat dalam melakukan training dan perbaikan efisiensi parameter dari model sebelumnya. Arsitektur *EfficientNetV2* memiliki beberapa pengembangan antara lain (1) menambahkan *MBCov* dan *fused-MBCov* pada lapisan awal (2) memilih rasio ekspansi yang lebih kecil untuk *MBCov* karena rasio ekspansi yang lebih kecil cenderung memiliki *overhead* akses memori yang lebih sedikit. (3) memilih kernel 3x3 yang lebih kecil, tetapi menambahkan lebih banyak lapisan untuk mengkompensasi berkurangnya bidang reseptif akibat ukuran kernel yang lebih kecil. (4) menghapus tahap *stride-1* terakhir di *EfficientNet* asli, karena ukuran parameternya yang besar dan *overhead* akses memori (Tan & Le, 2021). Implementasi metode *EfficientNetV2* memiliki tiga versi yaitu S, M, dan L yang diinisialisasi sesuai dengan ukuran dari *model*. Merujuk pada Analisa dan perancangan program pada penelitian ini memiliki 2 model DCNN *transfer learning* yaitu model DCNN deteksi penyakit daun cabai dan model DCNN deteksi penyakit buah cabai.

Model DCNN yang dibangun membutuhkan dataset *single classification* pada citra dengan ukuran 224 x 224 pixel

sesuai dengan kebutuhan input dari model *EfficientNetV2*. Model *EfficientNetV2* yang digunakan adalah versi M (medium) dengan ukuran model yang sedang dan tidak terlalu besar serta memiliki based model akurasi yang cukup tinggi dibandingkan versi sebelumnya. Model *EfficientNetV2* menggunakan *fine tuning* pada penelitian ini dan melakukan *freezing* pada *early layer* sampai dengan layer *block7e_dwconv2*.

Pada model deteksi penyakit daun cabai dan buah cabai ditambahkan beberapa layer layer dengan activation *relu* sesuai dengan tabel 4 untuk mencari nilai akurasi model terbaik. Setelah itu ditambahkan layer terakhir dengan 3 *dense* berdasarkan *output* kelas prediksi dan *activation softmax* karena pada penelitian ini *single classification*. Pada gambar 5 menunjukkan gambaran arsitektur model bagian-bagian yang dilakukan *freezing* dan bagian yang dilakukan penambahan layer.



Gambar 5. Gambaran Model Effcientnetv2 Menerapkan *Fine-Tuning*

Dari kedua konfigurasi model tersebut, dilakukan penyesuaian, penambahan, dan perubahan pada layer tertentu untuk mencari nilai akurasi dan loss yang terbaik agar model tidak overfitting atau pun underfitting. Tabel 4 menunjukkan beberapa konfigurasi model yang telah dimodifikasi menggunakan metode *fine tuning*, yang nantinya akan dilakukan pelatihan (training) dan pengujian (testing).

Tabel 4. Modifikasi Penambahan Layer Pada Model

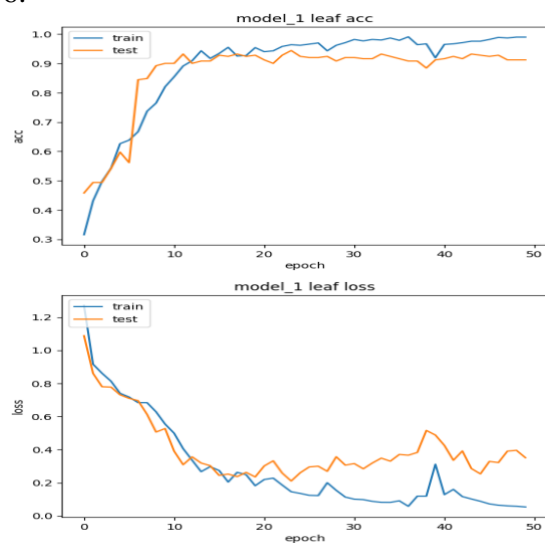
Layer	Model1	Model2	Model3
1	Dense 32 (FC*3)=>relu	Dense 64 (FC*3)=>relu	Dense 128 (FC*3)=>relu
2	Batch normalitation	Batch normalitation	Batch normalitation
3	Dropout	Dropout	Dropout
4	Dense 16 (FC*3)=>relu	Dense 32 (FC*3)=>relu	Dense 64 (FC*3)=>relu
5	Dropout	Dropout	Dropout
6	Dense 8 (FC*3)=>relu	Dense 16 (FC*3)=>relu	Dense 32 (FC*3)=>relu
7	Dense 3 => softmax	Dense 3 => softmax	Dense 3 => softmax

Jumlah training yang digunakan setatis sebanyak 50 *epochs*, 10 *batch* untuk masing masing model yang di buat. Untuk *training* menggunakan pemrosesan GPU serta lama waktu *training* tiap model tidak dimasukan kedalam hasil tiap model. Berikut merupakan hasil nilai akurasi dan loss model yang telah training yang ditunjukkan pada tabel 5.

Tabel 5. Hasil Pengujian Model Deteksi Penyakit Daun Cabai

Model	Akurasi	Loss	Valid akurasi	Valid Loss
Model1	0.990	0.053	0.912	0.351
Model2	0.998	0.008	0.932	0.443
Model3	0.986	0.0469	0.920	0.383

Dari tabel hasil pengujian untuk model deteksi penyakit daun cabai didapatkan nilai akurasi terbaik pada model Model1. Grafik perbandingan untuk validasi akurasi dengan akurasi serta validasi loss dengan loss menunjukkan model tidak terlalu overfitting atau pun underfitting yang ditunjukkan pada gambar 6.

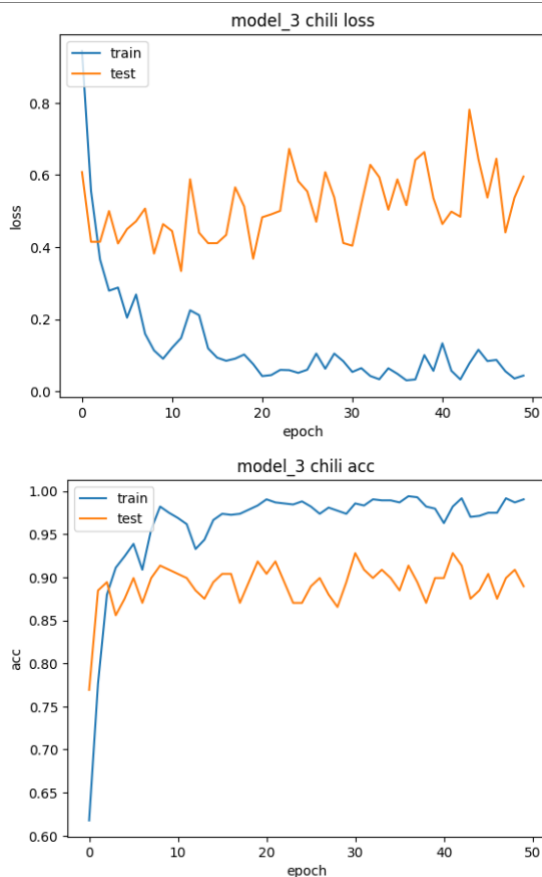


Gambar 6. Grafik Perbandingan Akurasi Dan Loss Model Daun

Pada tabel 6 merupakan hasil nilai akurasi dan loss model yang telah training dimana hasil terbaik untuk deteksi citra buah cabai pada model3. Grafik perbandingan untuk validasi akurasi dengan akurasi serta validasi loss dengan loss menunjukkan model tidak terlalu overfitting atau pun underfitting yang ditunjukkan pada gambar 7.

Tabel 6. Hasil Pengujian Model Deteksi Penyakit Daun Cabai

Model	Akurasi	Loss	Valid Akurasi	Valid Loss
Model1	0.993	0.03	0.826	0.815
Model2	0.998	0.009	0.894	0.664
Model 3	0.990	0.04	0.889	0.595

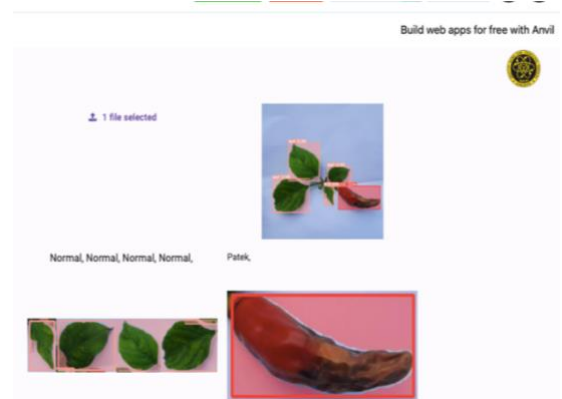


Gambar 7. Grafik Perbandingan Akurasi Dan Loss Model Cabai

C. Implementasi Yolo v7 dan DCNN

Implementasi model Yolo v7 dan DCNN agar program yang dibangun memiliki interface maka, peneliti menggunakan tools anvile. Anvile merupakan sebuah tools server website yang berbasis Bahasa pemrograman python yang dapat terhubung dengan google colab sehingga model yang dibangun dapat dijalan.

Website yang dibangun untuk diharapkan memudahkan dalam penggunaan program. Pada gambar 8 menunjukkan hasil program berbasis website yang dibangun. Input dari satu citra tanaman cabai dilakukan segmentasi dengan model Yolo V7 antara daun dan cabai. Setelah itu hasil dari segementasi dilakukan klasifikasi penyakit daun dan buah cabai pada setiap citra.



Gambar 8. Implementasi Interface Yolo V7 Dan DCNN

SIMPULAN

Berdasarkan implementasi metode terhadap program yang dibangun untuk deteksi penyakit pada tanaman cabai, peneliti menarik kesimpulan sebagai berikut:

1. Dataset Yolo v7 Segementasi tanaman cabai yang dibangun memiliki variasi yang seimbang sehingga metode Yolo v7 dapat mendeteksi daun dan buah tanaman cabai dengan nilai akurasi yang tinggi.
2. Membangun model dengan metode transfer learning yang menggunakan based model EfficientnetV2M untuk klasifikasi penyakit pada daun cabai tidak memerlukan tambahan neuron yang terlalu besar pada layer yang ditambahkan untuk pelatihan (training). Semakin kecil nilai neuron maka semakin bagus nilai akurasi dan loss akurasi. Jika ditambahkan neuron yang lebih tinggi maka model akan mengalami overfitting dan memiliki aktual loss yang melebar dari dari nilai loss. Sedangkan dalam membangun

model klasifikasi penyakit pada buah cabai memerlukan tambahan jumlah neuron yang lebih besar dibandingkan dengan model untuk klasifikasi penyakit pada daun cabai.

3. Metode klasifikasi untuk deteksi penyakit tanaman cabai yang digabungkan dengan *masking* dengan Yolo v7 dapat melakukan deteksi penyakit yang berbeda bervariasi pada tanaman cabai dimana, pada tanaman cabai dapat terserang penyakit lebih dari satu penyakit dalam satu pohon.

DAFTAR PUSTAKA

- Abas, S. M., Abdulazeez, A. M., & Zeebaree, D. Q. (2022). A YOLO and convolutional neural network for the detection and classification of leukocytes in leukemia. *Indonesian Journal of Electrical Engineering and Computer Science*, 25(1), 200. <https://doi.org/10.11591/ijeecs.v25.i1.pp200-213>
- Bochkovskiy, A., Wang, C.-Y., & Liao, H.-Y. M. (2020). *YOLOv4: Optimal Speed and Accuracy of Object Detection* (arXiv:2004.10934). arXiv. <http://arxiv.org/abs/2004.10934>
- Erlina, T., & Fikri, M. (2023). A YOLO Algorithm-based Visitor Detection System for Small Retail Stores using Single Board Computer. *Journal of Applied Engineering and Technological Science (JAETS)*, 4(2), 908–920. <https://doi.org/10.37385/jaets.v4i2.1872>
- Farid, M., & Subekti, N. A. (2012). Tinjauan Terhadap Produksi, Konsumsi, Distribusi Dan Dinamika Harga Cabe Di Indonesia. *Pusat Kebijakan Perdagangan Dalam Negeri*, 24.
- Hang, Zhang, Chen, Zhang, & Wang. (2019). Classification of Plant Leaf Diseases Based on Improved Convolutional Neural Network. *Sensors*, 19(19), 4161. <https://doi.org/10.3390/s19194161>
- Meilin, A. (2014). Hama Dan Penyakit Pada Tanaman Cabai Serta Pengendaliannya. *Balai Pengkajian Teknologi Pertanian Jambi*, 26.
- Naully, D. (2020). Fluktuasi Dan Disparitas Harga Cabai Di Indonesia. *Jurnal Agrosains dan Teknologi*, 14.
- Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). *You Only Look Once: Unified, Real-Time Object Detection* (arXiv:1506.02640). arXiv. <http://arxiv.org/abs/1506.02640>
- Redmon, J., & Farhadi, A. (2016). *YOLO9000: Better, Faster, Stronger* (arXiv:1612.08242). arXiv. <http://arxiv.org/abs/1612.08242>
- Redmon, J., & Farhadi, A. (2018). *YOLOv3: An Incremental Improvement* (arXiv:1804.02767). arXiv. <http://arxiv.org/abs/1804.02767>
- Tan, M., & Le, Q. V. (2021). *EfficientNetV2: Smaller Models and Faster Training* (arXiv:2104.00298). arXiv. <http://arxiv.org/abs/2104.00298>
- Wang, C.-Y., Bochkovskiy, A., & Liao, H.-Y. M. (2022). *YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors* (arXiv:2207.02696). arXiv. <http://arxiv.org/abs/2207.02696>
- Zikra, F., Usman, K., & Patmasari, R. (2021). *Deteksi Penyakit Cabai Berdasarkan Citra Daun Menggunakan Metode Gray Level Co-Occurrence Matrix Dan Support Vector Machine*. 9.