

TESIS
KLASIFIKASI PENYAKIT JERUK BERDASARKAN
KOMBINASI FITUR MENGGUNAKAN
MODEL DENSENET-169



disusun oleh

M. HAIKAL FIRDAUS

21.51.2096

Konsentrasi : Business Intelligence

FAKULTAS ILMU KOMPUTER
UNIVERSITAS AMIKOM YOGYAKARTA
YOGYAKARTA

2026

TESIS
KLASIFIKASI PENYAKIT JERUK BERDASARKAN
KOMBINASI FITUR MENGGUNAKAN
MODEL DENSENET-169

CLASSIFICATION OF CITRUS DISEASES BASED ON
FEATURE COMBINATION USING
DENSENET-169 MODEL



disusun oleh

M. HAIKAL FIRDAUS

21.51.2096

Konsentrasi : Business Intelligence

FAKULTAS ILMU KOMPUTER
UNIVERSITAS AMIKOM YOGYAKARTA
YOGYAKARTA

2026

HALAMAN PERSETUJUAN

**KLASIFIKASI PENYAKIT JERUK BERDASARKAN KOMBINASI
FITUR MENGGUNAKAN MODEL DENSENET-169**

**CLASSIFICATION OF CITRUS DISEASES BASED ON FEATURE
COMBINATION USING DENSENET-169 MODEL**

yang disusun dan diajukan oleh

M. Halkal Firdaus

21.51.2096

telah disetujui oleh Dosen Pembimbing Tesis
pada tanggal 20 Januari 2026

Dosen Pembimbing,



Prof. Dr. Ema Utami, S.Si., M.Kom.
NIK. 192302037

HALAMAN PENGESAHAN

**KLASIFIKASI PENYAKIT JERUK BERDASARKAN KOMBINASI
FITUR MENGGUNAKAN MODEL DENSENET-169**

**CLASSIFICATION OF CITRUS DISEASES BASED ON FEATURE
COMBINATION USING DENSENET-169 MODEL**

yang disusun dan diajukan oleh

M. Halkal Firdaus

21.51.2096

Telah dipertahankan di depan Dewan Penguji
pada tanggal 20 Januari 2026

Susunan Dewan Penguji

Nama Penguji

Prof. Dr. Ema Utami, S.Si., M.Kom.
NIK. 192302037

Alva Hendi Muhammad, S.T., M.Eng., Ph.D.
NIK. 190302xxx

Dr. Andi Sunyoto, M.Kom.
NIK. 190302xxx

Tanda Tangan



Tesis ini telah diterima sebagai salah satu persyaratan
untuk memperoleh gelar Magister Komputer
Tanggal 20 Januari 2026

DEKAN FAKULTAS ILMU KOMPUTER



Prof. Dr. Kusriani, M.Kom.
NIK. 190302106

HALAMAN PERNYATAAN KEASLIAN TESIS

Yang bertandatangan di bawah ini,

Nama mahasiswa : M. Haikal Firdaus

NIM : 21.51.2096

Menyatakan bahwa Tesis dengan judul berikut:

Klasifikasi Penyakit Jeruk Berdasarkan Kombinasi Fitur Menggunakan Model Densenet-169

Dosen Pembimbing : Prof. Dr. Ema Utami, S.Si., M.Kom,

1. Karya tulis ini adalah benar-benar ASLI dan BELUMPERNAH diajukan untuk mendapatkan gelar akademik, baik di Universitas AMIKOM Yogyakarta maupun di Perguruan Tinggi lainnya.
2. Karya tulis ini merupakan gagasan, rumusan dan penelitian SAYA sendiri, tanpa bantuan pihak lain kecuali arahan dari Dosen Pembimbing.
3. Dalam karya tulis ini tidak terdapat karya atau pendapat orang lain, kecuali secara tertulis dengan jelas dicantumkan sebagai acuan dalam naskah dengan disebutkan nama pengarang dan disebutkan dalam Daftar Pustaka pada karya tulis ini.
4. Perangkat lunak yang digunakan dalam penelitian ini sepenuhnya menjadi tanggung jawab SAYA, bukan tanggung jawab Universitas AMIKOM Yogyakarta.
5. Pernyataan ini SAYA buat dengan sesungguhnya, apabila di kemudian hari terdapat penyimpangan dan ketidakbenaran dalam pernyataan ini, maka SAYA bersedia menerima SANKSI AKADEMIK dengan pencabutan gelar yang sudah diperoleh, serta sanksi lainnya sesuai dengan norma yang berlaku di Perguruan Tinggi.

Yogyakarta, 20 Januari 2026

Yang Menyatakan,



Nama Mahasiswa

KATA PENGANTAR

Puji syukur penulis panjatkan ke hadirat Allah SWT karena berkat rahmat dan karunia-Nya Tesis berjudul "Klasifikasi Penyakit Jeruk Berdasarkan Kombinasi Fitur Menggunakan Model DenseNet-169" ini dapat diselesaikan sebagai salah satu syarat memperoleh gelar Magister pada Program Studi S2 Teknik Informatika, Program Pascasarjana Universitas AMIKOM Yogyakarta. Penulis menyadari bahwa penyusunan Tesis ini tidak terlepas dari dukungan dan bantuan berbagai pihak, oleh karena itu penulis menyampaikan terima kasih yang sebesar-besarnya kepada:

1. Prof. Dr. Ema Utami, S.Si., M.Kom., selaku Dosen Pembimbing Utama;
2. Dhani Ariatmanto, M.Kom., Ph.D., selaku Dosen Pembimbing Pendamping;
3. Para dosen dan staf Program Pascasarjana Universitas AMIKOM Yogyakarta;
4. Kedua orang tua dan keluarga atas doa dan dukungannya;
5. Teman-teman peneliti serta pihak-pihak yang telah membantu.

Penulis menyadari bahwa Tesis ini masih memiliki keterbatasan. Kritik dan saran yang membangun sangat diharapkan untuk penyempurnaan di masa mendatang. Semoga Tesis ini dapat memberikan manfaat bagi pengembangan ilmu pengetahuan, khususnya dalam bidang pengolahan citra dan klasifikasi penyakit tanaman.

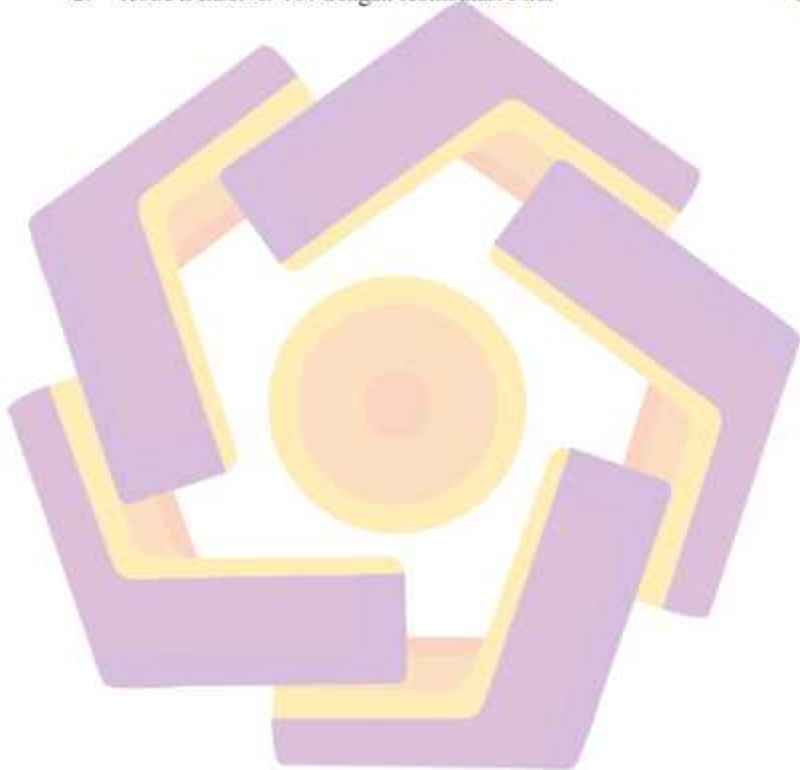
Yogyakarta, <tanggal bulan tahun>

Penulis

DAFTAR ISI

HALAMAN JUDUL	ii
HALAMAN PENGESAHAN	iii
HALAMAN PERSETUJUAN	iv
HALAMAN PERNYATAAN KEASLIAN TESIS	v
KATA PENGANTAR	vi
DAFTAR ISI	vii
DAFTAR TABEL	ix
DAFTAR GAMBAR	x
DAFTAR ISTILAH	xii
INTISARI	xiii
<i>ABSTRACT</i>	xiv
BAB I PENDAHULUAN	15
1.1. Latar Belakang Masalah	15
1.2. Rumusan Masalah	22
1.3. Batasan Masalah	23
1.4. Tujuan Penelitian	24
1.5. Manfaat Penelitian	24
BAB II TINJAUAN PUSTAKA	25
2.1. Tinjauan Pustaka	25
2.2. Keaslian Penelitian	31
2.3. Landasan Teori	35
BAB III METODE PENELITIAN	52
3.1. Jenis, Sifat, dan Pendekatan Penelitian	52
3.2. Metode Pengumpulan Data	53
3.3. Metode Analisis Data	54
3.4. Dataset	55
3.5. Alur Penelitian	56
BAB IV HASIL PENELITIAN DAN PEMBAHASAN	68
4.1. Persiapan Data	68
4.2. Preprocessing Data	73
4.3. Implementasi dan Hasil Model	80
4.4. Analisis dan Pembahasan	175
4.5. Perbandingan Dengan Penelitian Sebelumnya	180

BAB V PENUTUP	183
5.1. Kesimpulan	183
5.2. Saran	184
DAFTAR PUSTAKA	185
LAMPIRAN	190
1. Kode DenseNet-169 tanpa Kombinasi Fitur	190
2. Kode DenseNet-169 dengan Kombinasi Fitur	196



DAFTAR TABEL

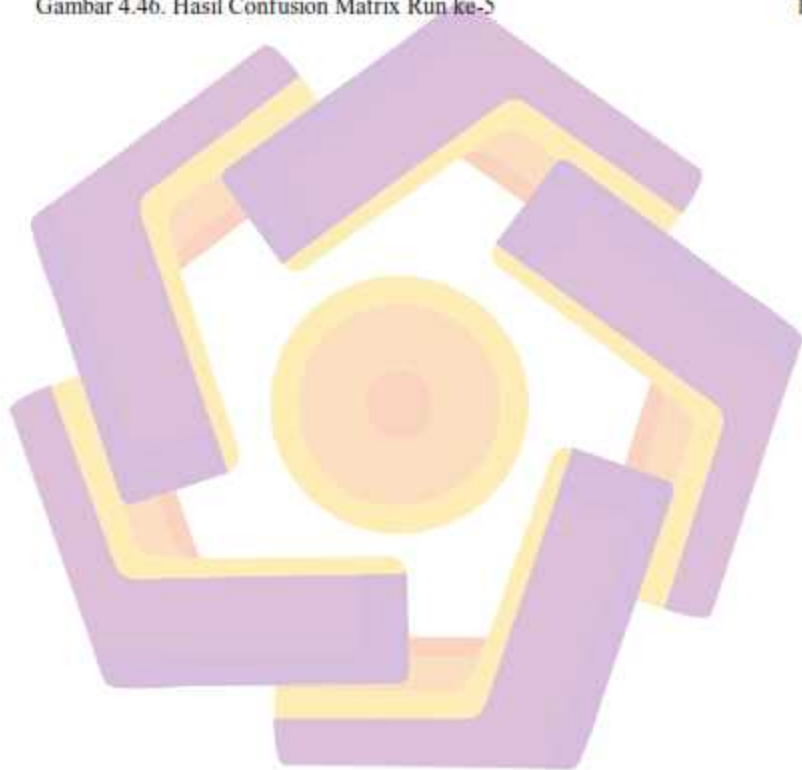
Tabel 2.1. Matriks literatur review dan posisi penelitian Klasifikasi Penyakit Jeruk Berdasarkan Kombinasi Fitur Menggunakan Model DenseNet-169	31
Tabel 4.1. Dataset LDI (<i>Leaf Disease Image</i>)	69
Tabel 4.2. Dataset FDI (<i>Fruit Disease Image</i>)	69
Tabel 4.3. Hasil Matrik Evaluasi DenseNet-169	118
Tabel 4.4. Hasil Akurasi setiap Kelas (DenseNet-169)	127
Tabel 4.5. Hasil Matrik Evaluasi DenseNet-169 Kombinasi Fitur	160
Tabel 4.6. Hasil Akurasi setiap Kelas (DenseNet-169 Kombinasi Fitur)	171
Tabel 4.7. Hasil Pengujian Model Klasifikasi pada kedua Skenario	176
Tabel 4.8. Tabel Perbandingan Dengan Penelitian Terdahulu	180



DAFTAR GAMBAR

Gambar 1.1. Hasil Produksi Jeruk Tahun 2020-2022	15
Gambar 2.1. Cara kerja algoritma Machine Learning	38
Gambar 2.2. Cara kerja Deep Learning	40
Gambar 2.3. Blok Padat dengan 5 Lapisan dan $k = 4$	47
Gambar 2.4. Arsitektur DenseNet-169	48
Gambar 3.1. Alur Penelitian	57
Gambar 3.2. Lapisan Akhir DenseNet-169 dengan Modifikasi Model	64
Gambar 4.1. (a) Citra asli, (b) Citra hasil augmentasi	72
Gambar 4.2. Perubahan jumlah dataset LDI sebelum dan sesudah balancing	73
Gambar 4.3. Perubahan jumlah dataset FDI sebelum dan sesudah balancing	73
Gambar 4.4. Mengubah ukuran citra ke 224x224 piksel	74
Gambar 4.5. Contoh hasil proses peningkatan citra dengan teknik CES	75
Gambar 4.6. Konversi Citra Asli ke HSV, YCbCr, dan LAB	78
Gambar 4.7. Grafik Pelatihan dan Validasi FDI Run 1	89
Gambar 4.8. Grafik Pelatihan dan Validasi LDI Run 1	92
Gambar 4.9. Grafik Pelatihan dan Validasi MDI Run 1	94
Gambar 4.10. Grafik Pelatihan dan Validasi FDI Run 2	97
Gambar 4.11. Grafik Pelatihan dan Validasi LDI Run 2	99
Gambar 4.12. Grafik Pelatihan dan Validasi MDI Run 2	100
Gambar 4.13. Grafik Pelatihan dan Validasi FDI Run 3	102
Gambar 4.14. Grafik Pelatihan dan Validasi LDI Run 3	103
Gambar 4.15. Grafik Pelatihan dan Validasi MDI Run 3	105
Gambar 4.16. Grafik Pelatihan dan Validasi FDI Run 4	106
Gambar 4.17. Grafik Pelatihan dan Validasi LDI Run 4	108
Gambar 4.18. Grafik Pelatihan dan Validasi MDI Run 4	109
Gambar 4.19. Grafik Pelatihan dan Validasi FDI Run 5	111
Gambar 4.20. Grafik Pelatihan dan Validasi LDI Run 5	112
Gambar 4.21. Grafik Pelatihan dan Validasi MDI Run 5	114
Gambar 4.22. Hasil Confusion Matrix Run ke-1	119
Gambar 4.23. Hasil Confusion Matrix Run ke-2	120
Gambar 4.24. Hasil Confusion Matrix Run ke-3	122
Gambar 4.25. Hasil Confusion Matrix Run ke-4	123
Gambar 4.26. Hasil Confusion Matrix Run ke-5	124
Gambar 4.27. Grafik Pelatihan dan Validasi FDI Run 1	130
Gambar 4.28. Grafik Pelatihan dan Validasi LDI Run 1	132
Gambar 4.29. Grafik Pelatihan dan Validasi MDI Run 1	134
Gambar 4.30. Grafik Pelatihan dan Validasi FDI Run 2	136
Gambar 4.31. Grafik Pelatihan dan Validasi LDI Run 2	138
Gambar 4.32. Grafik Pelatihan dan Validasi MDI Run 2	140
Gambar 4.33. Grafik Pelatihan dan Validasi FDI Run 3	142
Gambar 4.34. Grafik Pelatihan dan Validasi LDI Run 3	144
Gambar 4.35. Grafik Pelatihan dan Validasi MDI Run 3	146
Gambar 4.36. Grafik Pelatihan dan Validasi FDI Run 4	148
Gambar 4.37. Grafik Pelatihan dan Validasi LDI Run 4	149

Gambar 4.38. Grafik Pelatihan dan Validasi MDI Run 4	151
Gambar 4.39. Grafik Pelatihan dan Validasi FDI Run 5	153
Gambar 4.40. Grafik Pelatihan dan Validasi LDI Run 5	155
Gambar 4.41. Grafik Pelatihan dan Validasi MDI Run 5	156
Gambar 4.42. Hasil Confusion Matrix Run ke-1	163
Gambar 4.43. Hasil Confusion Matrix Run ke-2	164
Gambar 4.44. Hasil Confusion Matrix Run ke-3	166
Gambar 4.45. Hasil Confusion Matrix Run ke-4	167
Gambar 4.46. Hasil Confusion Matrix Run ke-5	169



DAFTAR ISTILAH

Istilah	Keterangan
CNN	Convolutional Neural Network
DenseNet-169	Arsitektur CNN dengan koneksi padat antar layer
FDI	Fruit Disease Image (dataset citra buah jeruk)
LDI	Leaf Disease Image (dataset citra daun jeruk)
MDI	Mixed Disease Image (gabungan citra daun dan buah)
Akurasi	Persentase prediksi benar terhadap seluruh data
Presisi	Ketepatan model dalam memprediksi kelas tertentu
Recall	Kemampuan model menemukan seluruh data kelas tertentu
F1-Score	Rata-rata harmonik antara presisi dan recall
Confusion Matrix	Matriks evaluasi hasil klasifikasi
Overfitting	Kondisi model terlalu menyesuaikan data latih
Generalisasi	Kemampuan model bekerja pada data baru

INTISARI

Penyakit pada tanaman jeruk merupakan salah satu faktor utama yang menyebabkan penurunan produktivitas dan kualitas hasil panen. Identifikasi penyakit secara dini melalui citra digital menjadi pendekatan yang efektif untuk mendukung sistem pertanian cerdas. Penelitian ini mengusulkan penggunaan model DenseNet-169 untuk klasifikasi penyakit jeruk berbasis citra, dengan dua skenario pengujian, yaitu DenseNet-169 tanpa kombinasi fitur dan DenseNet-169 dengan kombinasi fitur warna.

Pengujian dilakukan pada tiga dataset, yaitu Fruit Disease Image (FDI), Leaf Disease Image (LDI), dan Mixed Disease Image (MDI), dengan masing-masing skenario dijalankan sebanyak lima kali *running* untuk menguji konsistensi performa model. Evaluasi dilakukan menggunakan metrik akurasi, presisi, recall, dan F1-score.

Hasil eksperimen menunjukkan bahwa DenseNet-169 tanpa kombinasi fitur menghasilkan performa terbaik, dengan rata-rata akurasi mencapai 99,66% pada FDI, 98,66% pada LDI, dan 98,83% pada MDI. Sementara itu, skenario dengan kombinasi fitur tetap menunjukkan performa tinggi, namun dengan akurasi yang relatif lebih rendah, terutama pada dataset LDI. Analisis confusion matrix menunjukkan bahwa model memiliki kemampuan generalisasi yang sangat baik, khususnya pada dataset MDI yang menggabungkan citra daun dan buah jeruk.

Secara keseluruhan, penelitian ini membuktikan bahwa DenseNet-169 efektif digunakan untuk klasifikasi penyakit jeruk, dan bahwa penambahan kombinasi fitur tidak selalu meningkatkan performa model apabila fitur yang digunakan menimbulkan redundansi informasi.

Kata kunci: Convolutional Neural Network (CNN), DenseNet-169, Klasifikasi, Kombinasi Fitur, Penyakit Jeruk

ABSTRACT

Diseases in citrus plants are one of the main factors that cause a decrease in productivity and quality of harvest. Early disease identification through digital images is an effective approach to support smart agricultural systems. This research proposes the use of the DenseNet-169 model for image-based citrus disease classification, with two test scenarios, namely DenseNet-169 without a combination of features and DenseNet-169 with a combination of color features.

Testing was carried out on three datasets, namely Fruit Disease Image (FDI), Leaf Disease Image (LDI), and Mixed Disease Image (MDI), with each scenario run five times to test the consistency of model performance. Evaluation is carried out using accuracy, precision, recall and F1-score metrics.

Experimental results show that DenseNet-169 without feature combination produces the best performance, with average accuracy reaching 99.66% on FDI, 98.66% on LDI, and 98.83% on MDI. Meanwhile, the scenario with a combination of features still shows high performance, but with relatively lower accuracy, especially on the LDI dataset. Confusion matrix analysis shows that the model has very good generalization abilities, especially on the MDI dataset which combines images of leaves and citrus fruits.

Overall, this research proves that DenseNet-169 is effective for citrus disease classification, and that adding a combination of features does not always improve model performance if the features used cause information redundancy.

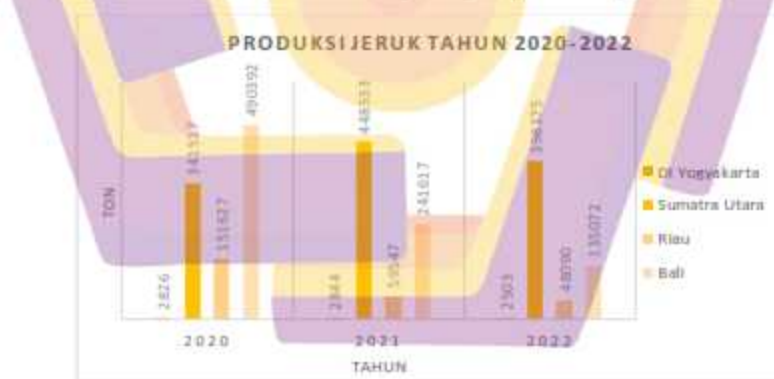
Keyword: Classification, Citrus Disease, Combination of Features, Convolutional Neural Network (CNN), DenseNet-169, Detection

BABI

PENDAHULUAN

1.1. Latar Belakang Masalah

Jeruk merupakan salah satu komoditas buah yang sangat penting dan memiliki nilai ekonomi yang tinggi di berbagai wilayah di dunia, termasuk di Indonesia. Di Indonesia, jeruk merupakan salah satu buah yang sangat populer dan menjadi bagian penting dalam industri pertanian dan perdagangan. Dengan tingginya permintaan pasar baik di dalam negeri maupun ekspor, industri jeruk di Indonesia memberikan kontribusi signifikan terhadap pertumbuhan ekonomi. Namun, ada beberapa daerah di Indonesia yang mengalami penurunan hasil panen dalam satu tahun terakhir. Untuk lebih jelasnya dapat dilihat pada Gambar 1.1.



Gambar 1.1. Hasil Produksi Jeruk Tahun 2020-2022

Menurut data statistik tahun 2023, grafik dari gambar di atas menunjukkan bahwa terjadi penurunan hasil panen di semua daerah yang diamati pada tahun 2022, termasuk tiga daerah yang merupakan bagian dari daerah penghasil jeruk

terbesar di Indonesia. Daerah Istimewa Yogyakarta mengalami penurunan sebesar 341 ton, Sumatra Utara mengalami penurunan sebesar 50.158 ton, Riau mengalami penurunan sebesar 11.457 ton, dan daerah Bali mengalami penurunan sebesar 106.545 ton. Hasil panen yang lebih rendah ini dapat berdampak pada produksi dan pasokan buah jeruk di masing-masing daerah tersebut (BPS, 2023). Menurut (Parananda, 2022) dilaman portal berita balipost pada hari minggu, 18 desember 2022 dengan salah seorang narasumber petani jeruk, I Ketut Mangku Sugata, mengatakan bahwa penyebab terjadinya penurunan hasil panen jeruk tersebut dikarenakan sering diguyur hujan. Hal ini mengakibatkan tanaman jeruk rentan terserang serangan hama dan penyakit. Temuan ini sesuai dengan penelitian yang dilakukan oleh (Lestari et al., 2018), dimana penyebab terjadinya penurunan hasil panen adalah dikarenakan serangan hama dan penyakit. Penelitian ini mengungkapkan bahwa ada dua penyakit yang sering menyerang daun jeruk, yaitu penyakit kanker yang disebabkan oleh patogen "Xanthomonas axonopodispv. Citri" dan penyakit ulat peliang, seperti yang dijelaskan oleh (Ariesdianto et al., 2021). Selain itu, dampak dari penyakit tersebut dapat menghasilkan buah-buahan berkualitas lebih rendah, sehingga penting bagi petani untuk memastikan mereka terbebas dari serangga dan virus penyebab penyakit (Mangla et al., 2022). Umumnya, program pemberantasan penyakit jeruk biasanya sulit dan membutuhkan biaya yang besar karena memerlukan banyak tenaga kerja dan sumber daya, meskipun memberikan keuntungan ekonomi sebagai imbalannya. Sampai saat ini, belum ada cara yang berhasil untuk memberantas penyakit jeruk, dan satu-satunya cara untuk mencegah penyebaran dan mengurangi dampak infeksi

adalah dengan membuang tanaman yang terinfeksi. Berbagai upaya telah dilakukan untuk mengatasi tantangan terkait hama dan penyakit tanaman serta mencegah kerugian panen dengan mengembangkan strategi yang efektif. Meskipun berbagai metode pengendalian penyakit tanaman telah diterapkan, identifikasi dini dan diagnosis yang akurat tetap menjadi kunci untuk mengurangi penyebaran penyakit dan kerusakan tanaman (Syed-Ab-Rahman et al., 2022).

Dalam 5 tahun terakhir, penelitian intensif telah dilakukan dalam menggunakan teknik pemrosesan gambar untuk mengenali penyakit tanaman. Teknik berbasis AI seperti pembelajaran mesin dan pembelajaran mendalam dalam pemrosesan gambar telah menunjukkan ketahanan dan konsistensi yang besar dengan biaya yang murah, presisi tinggi, dan kemampuan untuk mengevaluasi dengan cepat (Dhiman et al., 2023). Salah satu algoritma yang handal dalam memproses gambar adalah Convolutional Neural Network (CNN), seperti yang dijelaskan oleh (Ahmed et al., 2022), CNN mampu menilai data dari berbagai sudut pandang dan mengklasifikasikannya ke dalam salah satu dari beberapa kelas yang telah ditetapkan sebelumnya. CNN memiliki performa tinggi yang membuatnya sangat sesuai untuk pengenalan otomatis penyakit tanaman.

Pada penelitian oleh (Saputra et al., 2023) ini menggunakan pendekatan deep learning untuk mengklasifikasi penyakit pada daun padi dengan algoritma CNN dan arsitektur DenseNet. Algoritma DenseNet yang digunakan adalah DenseNet-121, DenseNet-169, dan DenseNet-201. DenseNet digunakan untuk meningkatkan aliran informasi dan gradien, mengurangi jumlah parameter, meningkatkan akurasi model, dan mengatasi masalah vanishing gradient. Hasil dari

model yang diusulkan menunjukkan bahwa akurasi menggunakan DenseNet-121 mencapai 91,67%, DenseNet-169 mencapai 90%, dan DenseNet-201 mencapai 88,33%. Dengan demikian hasil penelitian ini menunjukkan bahwa akurasi tertinggi dicapai oleh DenseNet-121 dengan 91,67.

Penelitian lain oleh (Albahli & Nawaz, 2022) ini mengusulkan metode DCNet, yaitu model CornerNet yang dimodifikasi dengan menggunakan DenseNet-77 sebagai jaringan utama untuk mendeteksi dan mengklasifikasikan penyakit daun tomat. DCNet menggunakan pendekatan deteksi objek berbasis titik sudut untuk menentukan lokasi dan kelas sudut atas kiri dan bawah kanan dari daerah yang terkena penyakit. DCNet juga menggunakan kombinasi fitur heatmap, embedding, dan offset untuk membedakan pasangan sudut yang termasuk dalam kelas yang sama atau berbeda. Eksperimen pada dataset PlantVillage menunjukkan bahwa DCNet mencapai akurasi rata-rata 99,98%, unggul dari beberapa metode sebelumnya seperti SE-ResNet-50, ResNet-101, VGG19, Xception dan GoogleNet dengan akurasi masing-masing model 96,81%, 90,13%, 90,42%, 88,16%, dan 87,27%. Selain itu, DCNet juga membutuhkan data anotasi yang cukup untuk melatih jaringan secara optimal.

Penelitian yang terkait penyakit jeruk sudah pernah dilakukan peneliti lainnya, (Elaraby et al., 2022) penulis mengusulkan teknik kombinasi fitur dan transfer learning menggunakan model AlexNet dan VGG19 yang telah dilatih sebelumnya pada dataset ImageNet. Selain itu, peneliti juga menggunakan teknik data augmentation untuk meningkatkan jumlah data latihan dan mencegah overfitting. Teknik ini memungkinkan model untuk melihat variasi citra penyakit jeruk yang

lebih banyak dan meningkatkan kemampuan generalisasi model. Hasil penelitian ini juga menunjukkan bahwa model AlexNet dengan optimizer SGDM menghasilkan akurasi klasifikasi tertinggi yaitu 94.3% untuk dataset *leaf disease image* (LDI) dan 93.5% untuk dataset *fruit disease image* (FDI). Model VGG19 dengan optimizer SGDM menghasilkan akurasi klasifikasi sebesar 92.9% untuk dataset LDI dan 92.6% untuk dataset FDI. Namun, hasil dari penelitian ini masih mungkin untuk ditingkatkan akurasi.

Oleh karena itu, diusulkan penggunaan algoritma CNN dengan arsitektur DenseNet untuk meningkatkan akurasi dan efisiensi dari model CNN. Seperti penelitian yang dilakukan oleh (Shireesha & Reddy, 2022), dalam penelitiannya menggunakan model DenseNet-121 untuk melakukan klasifikasi buah dan daun jeruk sehat dengan yang terinfeksi penyakit seperti blackspot, greening, scab, dan canker. Model ini digunakan karena dapat mengekstrak banyak atribut dari berbagai lapisannya, sehingga mampu memberikan hasil akurasi yang tinggi mencapai 96% dengan menggunakan 50 epoch. Akurasi tersebut berhasil dicapai karena model dari DenseNet-121 mampu mengurangi degradasi dan masalah gradien yang sebelumnya telah dilatih dengan menggunakan ImageNet. Selain itu, arsitektur DenseNet yang diusulkan adalah DenseNet-169. DenseNet-169 diusulkan karena menurut penelitian yang dilakukan oleh (Huang et al., 2017), DenseNet-169 memiliki akurasi yang lebih tinggi daripada DenseNet-121 dan lebih efisien daripada DenseNet-201 yang menggunakan set validasi ImageNet. Hal ini karena DenseNet-169 memiliki kedalaman lapisan yang lebih besar dari DenseNet-121 dan juga lebih kecil dari DenseNet-201 sehingga memiliki waktu komputasi yang lebih

besar jika dibandingkan dengan jumlah lapisan yang lebih kecil (Rafif et al., 2022). Berdasarkan 2 hasil pengujian berbeda, DenseNet-201 dengan pengujian satu potong (single-crop) memiliki tingkat kesalahan (error rate) top-1 sebesar 22.58%, DenseNet-169 memiliki tingkat kesalahan top-1 sebesar 23.80%, dan DenseNet-121 memiliki tingkat kesalahan top-1 sebesar 25.02%. Dengan pengujian sepuluh potong (10-crop), tingkat kesalahan top-1 menjadi 21.46% untuk DenseNet-201, 22.08% untuk DenseNet-169 dan 23.61% untuk DenseNet-121. Hal ini memungkinkan DenseNet-169 mampu mengenali objek-objek dalam gambar dengan lebih baik dan cepat daripada DenseNet-121 ataupun DenseNet-201. DenseNet-169 dipilih karena menawarkan trade-off yang optimal antara kedalaman jaringan dan efisiensi parameter. Dibandingkan DenseNet-121, ia memiliki kapasitas representasi yang lebih besar, sedangkan dibandingkan DenseNet-201, ia lebih efisien secara komputasi tanpa penurunan performa yang signifikan pada dataset berukuran sedang seperti yang digunakan dalam penelitian ini.

Berdasarkan beberapa penelitian yang telah dibahas, penggunaan algoritma CNN telah terbukti memiliki potensi yang besar dalam mendeteksi penyakit pada tanaman. Hal ini membuat model CNN mampu mencapai tingkat akurasi yang tinggi, sehingga dapat membantu dalam pengklasifikasian penyakit pada tanaman secara efektif. Namun, penting juga untuk mempertimbangkan beberapa kekurangan potensial yang melekat pada metode ini. Salah satunya adalah kebutuhan akan jumlah variasi data yang besar untuk melatih model CNN yang efektif. Dataset yang besar dan representatif dibutuhkan untuk memastikan model memiliki pemahaman yang baik tentang berbagai variasi penyakit jeruk. Selain itu,

waktu pelatihan yang diperlukan oleh metode CNN cenderung lebih lama dibandingkan dengan metode tradisional karena kompleksitas komputasi yang tinggi.

Oleh karena itu, pemrosesan citra dalam mengklasifikasi penyakit pada tanaman jeruk sangat penting dalam mengatasi masalah dataset yang besar dan representatif. Dengan memanfaatkan teknik augmentasi data dan kombinasi fitur seperti perbaikan citra, segmentasi, dan ekstraksi fitur, kinerja algoritma CNN dapat ditingkatkan. Salah satu penelitian yang memanfaatkan pemrosesan citra yaitu (Safdar et al., 2019) yang mengusulkan sebuah metode otomatis untuk mengenali dan mengklasifikasikan penyakit jeruk menggunakan teknik pengolahan citra komputer. Dengan menggabungkan metode pemrosesan citra untuk ekstraksi fitur dengan algoritma deep learning untuk mengklasifikasikan fitur daun pohon jeruk dan juga menggunakan beberapa teknik dalam ekstraksi fitur seperti filter Gaussian, filter terpadu, transformasi watershed, analisis komponen utama, dan support vector machine (SVM) untuk meningkatkan akurasi dan efisiensi model. Hasil penelitian ini mampu mencapai akurasi klasifikasi tertinggi sebesar 95,5% dengan menggunakan M-SVM.

Selain itu, penelitian (Singh et al., 2020) mengusulkan beberapa teknik pemrosesan gambar untuk mengklasifikasikan penyakit daun jeruk berdasarkan fitur tekstur dan warna. Dengan klasifikasi analisis diskriminan linear, hasil penelitian ini mampu mencapai akurasi rata-rata 84,32% dibandingkan dengan penelitian sebelumnya yang menggunakan HMM (Hidden Markov Model) berdasarkan fitur tekstur saja yang memiliki akurasi rata-rata 82,6%. Hasil

penelitian ini mampu meningkatkan kinerja klasifikasi yang lebih baik untuk penyakit canker dan penyakit jeruk lainnya dengan memanfaatkan fitur warna.

Berdasarkan penjelasan dan permasalahan diatas, penelitian ini bertujuan untuk membandingkan metode klasifikasi penyakit jeruk yang menggunakan arsitektur DenseNet-169 dengan dan tanpa kombinasi fitur. Tujuan penelitian ini adalah untuk menganalisis dan membandingkan hasil dari metode klasifikasi berdasarkan akurasi, presisi, dan recall dari setiap model yang diujikan. Selain itu, penelitian ini juga akan menganalisis keunggulan dan kelemahan dari masing-masing model guna memberikan pemahaman yang lebih mendalam tentang penggunaan teknik kombinasi fitur pada model CNN dan arsitektur DenseNet-169 dalam klasifikasi penyakit jeruk. Dengan adanya sistem ini, diharapkan hasil analisis penyakit jeruk menjadi efisien dan lebih baik dalam mendiagnosa jenis penyakit jeruk. Hal ini diharapkan dapat memberikan kontribusi signifikan dalam pengendalian dan pencegahan penyakit jeruk, serta mengurangi dampak negatifnya terhadap produksi jeruk di Indonesia.

1.2. Rumusan Masalah

Berdasarkan uraian yang telah disampaikan sebelumnya dari latar belakang diatas, dapat diperoleh rumusan masalah sebagai berikut:

- a. Apakah model DenseNet-169 asli dan model DenseNet-169 yang dipadukan dengan teknik kombinasi fitur menghasilkan kinerja yang lebih baik?

- b. Apakah tingkat akurasi dengan kombinasi fitur pada model DenseNet-169 dapat meningkatkan performa kalsifikasi penyakit jeruk dibandingkan dengan model yang lain?

1.3. Batasan Masalah

Batasan masalah dari penelitian ini adalah sebagai berikut:

- a. Penelitian ini melakukan analisis untuk membandingkan hasil klasifikasi penyakit jeruk yang berisi gambar buah dan daun, masing-masing terdiri dari 4 kelas dimana 3 kelas merupakan penyakit jeruk dan 1 kelas merupakan jeruk yang sehat.
- b. Dataset yang digunakan adalah LDI (*Leaf Disease Image*) dan FDI (*Fruit Disease Image*), yang terdiri dari 596 dan 1789 data gambar. Data tersebut berasal dari kaggle yang sebelumnya pernah digunakan oleh (Syed-Ab-Rahman et al., 2022) dan (Silva et al., 2021) dalam penelitiannya tentang penyakit jeruk.
- c. Model yang digunakan dalam penelitian ini yaitu DenseNet-169 dengan dan tanpa kombinasi fitur untuk melakukan klasifikasi.
- d. Penelitian ini berfokus pada hasil klasifikasi penyakit jeruk yang terbagi menjadi empat kelas yaitu, blackspot, canker, huanglongbing dan healthy.
- e. Indikator yang digunakan dalam mengevaluasi hasil klasifikasi adalah Confusion Matrix.
- f. Model yang dikembangkan merupakan sistem klasifikasi citra, bukan sistem deteksi objek atau segmentasi, sehingga tidak menghasilkan lokalisasi spasial dari gejala penyakit.

1.4. Tujuan Penelitian

Penelitian ini memiliki tujuan sebagai berikut:

- a. Membandingkan hasil kinerja dari kedua metode klasifikasi yang diujikan, yaitu model DenseNet-169 dengan kombinasi fitur dan tanpa kombinasi fitur.
- b. Mengetahui hasil klasifikasi berdasarkan akurasi, presisi, dan recall dari kedua model klasifikasi.

1.5. Manfaat Penelitian

Penelitian ini memiliki manfaat sebagai berikut:

- a. Bagi peneliti, penelitian ini memberikan kesempatan untuk memperoleh pemahaman yang lebih baik tentang perbandingan kinerja antara model klasifikasi DenseNet-169 dengan dan tanpa kombinasi fitur dalam mengklasifikasikan jenis penyakit jeruk.
- b. Menambah pengetahuan dan pemahaman tentang model yang efektif dalam mengklasifikasikan jenis penyakit jeruk.
- c. Memberikan kontribusi dalam pengembangan sistem klasifikasi penyakit jeruk secara langsung yang dapat membantu dalam diagnosis awal penyakit tanaman secara cepat dan akurat.
- d. Menjadi referensi penting bagi penelitian selanjutnya dalam bidang yang sama, serta memberikan wawasan dan arah baru untuk pengembangan penelitian di masa depan.

BAB II

TINJAUAN PUSTAKA

2.1. Tinjauan Pustaka

Beberapa tahun terakhir, klasifikasi penyakit jeruk menggunakan teknik deep learning telah menjadi perhatian utama. Penelitian-penelitian terkait telah berfokus pada pengembangan model yang dapat mengklasifikasikan berbagai jenis penyakit jeruk secara akurat. Dalam tinjauan pustaka ini, akan dibahas temuan dan kontribusi studi penelitian yang relevan dalam bidang ini.

Pada penelitian yang dilakukan oleh (Kukreja & Dhiman, 2020) menyatakan dampak penyakit jeruk dapat menurunkan kualitas dan kuantitas buah, mengurangi daya saing pasar, dan meningkatkan biaya produksi. Deteksi dini otomatis pada penyakit jeruk adalah langkah untuk mencegah penyebaran penyakit, mengurangi kerugian ekonomi, dan meningkatkan efisiensi manajemen tanaman. Metode yang digunakan adalah deep convolutional neural network (CNN) dengan data augmentation dan pre-processing. Data augmentation dan pre-processing dapat meningkatkan kualitas dan konsistensi data citra yang digunakan. Metode dan teknik tersebut digunakan karena memiliki keunggulan dalam hal performa, kecepatan, dan fleksibilitas dibandingkan dengan metode konvensional yang bergantung pada fitur manual dan keahlian manusia. Model yang diusulkan dapat mencapai akurasi 89,1%, sedangkan model deep convolutional neural network (CNN) hanya mencapai akurasi 67%. Hal ini menunjukkan bahwa data augmentation dan pre-processing dapat meningkatkan performa model dalam

mendeteksi dan mengklasifikasikan penyakit jeruk. Oleh karena itu, metode ini dapat memberikan solusi yang efektif dan efisien untuk deteksi otomatis penyakit jeruk menggunakan teknik deep learning dengan data augmentation dan pre-processing, sehingga mampu mengungguli metode deep CNN yang tidak menggunakan data augmentation dan pre-processing.

Pada penelitian yang dilakukan oleh (Soini et al., 2019), mengatasi penyakit jeruk dapat mencegah dampak buruk dari penyakit pada bentuk, warna, dan rasa buah jeruk yang tidak normal, serta kematian pohon jeruk dalam beberapa tahun setelah terinfeksi. Deteksi dini otomatis pada penyakit jeruk penting untuk mencegah penyebaran penyakit dan mengambil tindakan pencegahan seperti penggunaan insektisida dan vaksinasi pohon. Metode yang digunakan dalam penelitian ini adalah teknik computer vision dan deep learning. Teknik computer vision digunakan untuk mengekstrak sub-gambar buah dari gambar pohon sebagai input untuk proses deep learning. Teknik deep learning digunakan untuk mengklasifikasikan sub-gambar buah menjadi terinfeksi atau tidak terinfeksi dengan menggunakan model inception yang telah dilatih ulang dengan 4000 iterasi. Hasil model menunjukkan bahwa metode yang diusulkan dengan model inception memiliki akurasi validasi 93,3% dan mengungguli model klasifikasi lainnya dalam mendeteksi penyakit jeruk pada tanaman jeruk.

Dalam penelitian yang dilakukan oleh (Syed-Ab-Rahman et al., 2022) mereka mengusulkan sebuah model klasifikasi penyakit jeruk berbasis pembelajaran mendalam dua tahap yang menggunakan Faster R-CNN sebagai kerangka kerja utama. Model ini dapat mengekstrak fitur gambar secara otomatis

dengan menggunakan ResNet101 sebagai ekstraktor fitur, mengusulkan wilayah potensial yang berisi objek target (daun jeruk yang terinfeksi) dengan menggunakan RPN (*Region Proposal Network*), memilih 50 ROI teratas dengan skor objektivitas tertinggi dengan menggunakan ROI pooling, dan melakukan klasifikasi akhir ke salah satu dari empat kelas yaitu bercak hitam, kanker bakteri, HLB, atau sehat dengan menggunakan klasifikator. Model ini dapat mengklasifikasikan penyakit jeruk dengan akurasi tinggi 86,2%, 97,2%, dan 94,6% untuk bercak hitam, kanker bakteri, dan HLB, masing-masing. Model ini juga dapat mengenali daun jeruk sehat dengan akurasi 97%, serta memberikan akurasi klasifikasi 94,37% dan presisi rata-rata 95,8%. Model ini menunjukkan hasil yang stabil dengan deviasi standar rendah. Model ini juga mengungguli metode lainnya dalam hal F1-score, presisi, recall, dan akurasi untuk ketiga jenis penyakit jeruk.

Sedangkan pada penelitian yang dilakukan oleh (Elaraby et al., 2022), mereka telah mengusulkan sebuah kerangka kerja untuk mengenali dan mengklasifikasikan penyakit jeruk menggunakan dua model jaringan saraf konvolusional (CNN) yang telah dilatih sebelumnya, yaitu AlexNet dan VGG19. Penelitian ini menggunakan teknik transfer learning untuk memanfaatkan bobot dari model yang telah dilatih sebelumnya pada dataset yang berbeda. Penelitian ini juga menggunakan augmentasi data untuk meningkatkan jumlah data latih dan menghindari masalah overfitting yang dapat terjadi jika menggunakan data yang sedikit. Augmentasi data dilakukan dengan menggunakan rotasi, cerminan, dan pergeseran warna. Penelitian ini menggunakan dua dataset citra penyakit jeruk, yaitu *fruit disease image* (FDI) dan *leaf disease image* (LDI), yang berisi enam kelas

penyakit jeruk, yaitu bercak hitam, canker, greening, sehat, scab, dan melanose. Penulis mengevaluasi kinerja model yang diusulkan dengan menggunakan metrik seperti akurasi, presisi, sensitivitas, spesifisitas, dan F-score. Hasil menunjukkan bahwa model AlexNet dengan optimasi *stochastic gradient descent with momentum* (SGDM) memberikan hasil terbaik dengan akurasi 94,3%, sedangkan model VGG19 dengan optimasi Adam memberikan akurasi 93,5%. Penelitian ini juga membandingkan hasilnya dengan penelitian sebelumnya yang menggunakan metode CNN lainnya dan menunjukkan bahwa model yang diusulkan mengungguli metode yang sudah ada. Penulis menyimpulkan bahwa metode pembelajaran mendalam berbasis transfer learning adalah metode yang matang dan efektif untuk mengklasifikasikan penyakit jeruk dengan performa yang tinggi.

Pendekatan deep learning juga digunakan oleh (Ganesh et al., 2019) untuk deteksi dan segmentasi jeruk berdasarkan kerangka kerja segmentasi instan Mask R-CNN. Pendekatan ini menggunakan data masukan multi-modal yang terdiri dari gambar RGB dan HSV dari pohon jeruk. Deteksi dini otomatis pada jeruk penting untuk mencegah penyebaran penyakit jeruk, seperti citrus greening, yang dapat mengurangi hasil panen dan kualitas buah. Peneliti menggunakan augmentasi data, seperti rotasi, pemotongan, dan perubahan skala, untuk meningkatkan variasi data latih dan mengurangi overfitting. Mereka juga membandingkan kinerja algoritma menggunakan gambar RGB dan RGB+HSV. Hasil menunjukkan bahwa penambahan data HSV meningkatkan presisi menjadi 97,53% dari 89,47%, ketika menggunakan data RGB saja. Skor akurasi rata-rata sebesar 97,53% untuk deteksi jeruk dan 88,99% untuk segmentasi jeruk. Model ini mengungguli model yang

menggunakan data RGB saja, yang memiliki akurasi rata-rata sebesar 89,47% untuk deteksi jeruk dan 78,98% untuk segmentasi jeruk. Berdasarkan hasil yang diusulkan menggunakan pendekatan multi-modal, dapat meningkatkan deteksi dan segmentasi jeruk secara signifikan dibandingkan dengan pendekatan berbasis RGB saja.

Pada penelitian (Dhiman et al., 2023) menjelaskan penyakit jeruk dapat menurunkan kualitas dan produktivitas buah jeruk, serta merugikan perekonomian petani dan industri jeruk. Oleh karena itu, deteksi citra penyakit jeruk sangat penting dilakukan dengan menggunakan teknik pengolahan citra dan pembelajaran mendalam. Penelitian ini mengusulkan sebuah kerangka kerja untuk klasifikasi penyakit jeruk menggunakan gabungan antara CNN (Convolutional Neural Network) dan LSTM (Long Short-Term Memory) dengan edge computing. Kerangka kerja ini terdiri dari tiga fase, yaitu pra-pemrosesan, augmentasi data, dan segmentasi citra, pelatihan model CNN-LSTM dengan teknik pemangkasan (*pruning*) dan kuantisasi (*quantization*), serta pengujian dan klasifikasi citra. Augmentasi data digunakan untuk meningkatkan variasi dan jumlah data latih dengan melakukan transformasi seperti pemutaran, pergeseran, cerah, dan sebagainya. Metode CNN-LSTM digunakan untuk mengekstraksi fitur spasial dan temporal dari citra, serta mengurangi kompleksitas model dengan membuang parameter yang kurang penting. Edge computing digunakan untuk mempercepat proses inferensi dan menghemat sumber daya komputasi. Hasil eksperimen menunjukkan bahwa kerangka kerja yang diusulkan dapat mencapai akurasi 98,25% dengan pemangkasan dan kuantisasi, yang lebih baik daripada model CNN

dasar tanpa teknik kompresi. Berdasarkan hasil dari model yang diusulkan, mampu mengklasifikasikan penyakit jeruk secara efisien dan akurat dengan menggunakan CNN-LSTM dan edge computing.

Selain itu, penelitian yang dilakukan oleh (Xing et al., 2019) membahas tentang model Weakly Dense Connected Convolution Network (Weakly DenseNet) untuk mengenali dan mengklasifikasikan hama dan penyakit jeruk. Model ini dirancang untuk meningkatkan efisiensi parameter dengan menggunakan dua lapisan konvolusi 1x1 untuk menyempurnakan fitur map, dan menggunakan metode konkatenasi untuk penggunaan kembali fitur. Model ini juga menggunakan SGD dengan momentum Nesterov untuk melatih modelnya, dan menggunakan learning rate schedule untuk menyesuaikan learning rate berdasarkan validasi loss. Pendekatan dari model ini menggunakan dataset gambar hama dan penyakit jeruk yang terdiri dari 12.561 citra dari 17 spesies hama jeruk dan tujuh jenis penyakit jeruk. Hasil pendekatan ini mencapai akurasi klasifikasi tertinggi dengan parameter lebih sedikit, yaitu 93,42% pada data validasi dan 93,33% pada data uji. Menariknya, meskipun Weakly DenseNet-16 memiliki jumlah parameter yang lebih sedikit, ia berhasil mengungguli mereka dalam hal performa.

2.2. Keaslian Penelitian

Tabel 2.1. Matriks literatur review dan posisi penelitian
Klasifikasi Penyakit Jeruk Berdasarkan Kombinasi Fitur Menggunakan Model DenseNet-169

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
1	A Deep Neural Network based disease detection scheme for Citrus fruits	Vinay Kukreja, Poonam Dhiman International Conference on Smart Electronics and Communication (ICOSEC), 2020	Mengembangkan algoritma dense CNN untuk melakukan deteksi dan memberikan metode yang efektif untuk mengena li cacat yang tampak pada buah jeruk.	Metode data augmentation dan pre-processing dapat meningkatkan kuantitas dan kualitas dataset, yang pada akhirnya akan meningkatkan akurasi klasifikasi. Hasil menunjukkan bahwa model yang diusulkan mencapai akurasi keseluruhan sebesar 89,1% pada dataset buah jeruk.	Memperluas dataset buah jeruk sehingga sistem yang diusulkan dapat dioptimalkan untuk mencapai hasil yang lebih baik. Selain itu, dataset yang digunakan masih terbatas dan tidak mencakup semua jenis penyakit.	Penelitian ini menggunakan preprocessing citra dengan teknik kombinasi fitur, selain itu dataset yang digunakan tidak terbatas pada penyakit buah jeruk, tetapi juga pada daun jeruk.
2	Citrus Greening Infection Detection (CiGID) by Computer Vision and Deep Learning	Charles T. Soni, Sofiane Fellah, Muhammad Rizwan Abid ACM International Conference Proceeding Series, 2019	Mengembangkan algoritma deteksi infeksi penyakit greening jeruk (CiGID) yang menggabungkan teknik visi komputer dan pembelajaran mendalam.	Algoritma ini mengklasifikasikan buah jeruk terinfeksi atau tidak oleh penyakit greening jeruk yang disebut Huanglongbing (HLB) dan gambar pohon. Model inception dilatih dengan 4000 iterasi dan mencapai akurasi validasi 93,3%. Hal ini menunjukkan bahwa algoritma CiGID dapat menjadi alat bantu	Pengembangan lebih lanjut untuk mendeteksi penyakit lain pada tanaman jeruk atau tanaman lainnya, karena algoritma CiGID hanya dapat mendeteksi infeksi HLB pada buah jeruk yang memiliki gejala visual yang jelas dan bergantung pada	Penelitian ini menggunakan model CNN dengan arsitektur DenseNet-169, selain itu menggunakan dataset dengan tiga kelas penyakit jeruk dan satu kelas jeruk sehat.

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
				keputusan yang berguna untuk petani jeruk dalam mencegah dan mengendalikan penyebaran HLB di kebun jeruk.	kuaitas gambar pohon jeruk yang diambil.	
3	Citrus disease detection and classification using end-to-end anchor-based deep learning model	Sharifah Farhana, Syed-Ab-Rahman, Mohammad Hesam Hesamian, Mukesh Prasad Applied Intelligence, 2021	Mengembangkan metode deteksi dan klasifikasi penyakit jeruk berbasis deep learning yang dapat menganalisis gambar daun jeruk yang sehat dan terinfeksi dengan canker, black spot, dan Huanglongbing. Metode yang diusulkan adalah model jaringan saraf konvolusional dua tahap berdasarkan Faster R-CNN.	Metode yang diusulkan dapat melakukan deteksi dan klasifikasi penyakit jeruk dengan akurasi tinggi, yaitu 86,2%, 97,2%, dan 94,6% untuk black spot, canker, dan Huanglongbing, masing-masing dengan melakukan pelatihan pada jumlah sampel yang sangat minimal.	Meningkatkan ketangguhan model dan mengurangi standar deviasi, serta memperluas dataset buah jeruk.	Menggunakan teknik preprocessing seperti peningkatan kualitas citra dan konversi warna untuk mengekstraksi fitur, selain itu menggunakan model DenseNet-169.
4	Classification of Citrus Diseases Using Optimization Deep Learning Approach	Ahmed Elaraby, Walid Hamdy, Saad Alanazi Computational Intelligence and Neuroscience, 2022	Mengembangkan deep learning untuk mendeteksi dan mengklasifikasikan penyakit pada tanaman jeruk menggunakan pengolahan citra	Metode yang digunakan dapat menghasilkan akurasi klasifikasi yang tinggi (94,3%) dalam mengidentifikasi penyakit jeruk seperti anthracnose, black spot, canker, scab, greening, dan melanose.	Meningkatkan ketersediaan data citra penyakit jeruk dengan berbagai kondisi dan situasi, yang dapat meningkatkan generalisasi dan robustness model deep	Penelitian ini menggunakan model DenseNet-169 dengan teknik optimasi SGD dan menggunakan teknik preprocessing untuk meningkatkan kualitas citra dari dataset.

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
			dan transfer learning.	Selain itu, metode yang digunakan juga mengungguli metode-metode sebelumnya dalam hal waktu komputasi dan parameter jaringan.	learning. Selain itu, data yang digunakan masih terbatas dan tidak mencakup semua jenis penyakit jeruk yang ada.	
5	Deep Orange: Mask R-CNN based Orange Detection and Segmentation	P. Ganesh, K. Volle, T. F. Burks, S.S. Mehta IFAC-PapersOnLine, 2019	Mengembangkan pendekatan deep learning yang dinamakan Deep Orange, berdasarkan kerangka kerja dari Mask R-CNN.	Pendekatan ini menggunakan data input multi-modal yang terdiri dari gambar RGB dan HSV, yang diproses oleh jaringan Mask R-CNN yang telah dilatih sebelumnya. Pendekatan ini dapat mencapai presisi deteksi dan segmentasi yang tinggi, yaitu 97,53% dan 89%.	Mengevaluasi pendekatan Deep Orange pada kondisi pencahayaan dan latar belakang yang berbeda. Selain itu, data pelatihan dan pengujian kurang bervariasi.	Penelitian ini menggunakan model DenseNet-169 dengan teknik kombinasi fitur, selain itu data set yang digunakan tidak terbatas pada penyakit buah jeruk, tetapi juga pada daun jeruk.
6	Smart Disease Detection System for Citrus Fruits Using Deep Learning with Edge Computing	Poonam Dhiman, Amandeep Kaur, Yasir Hamid, Eatedal Alabdulkreem, Hela Elmanna, Nedal Ababneh Sustainability, 2023	Mengembangkan model yang efisien untuk prediksi penyakit buah sitrus menggunakan teknik deep learning CNN dan LSTM dengan edge computing.	Model CNN-LSTM yang digunakan mampu berfungsi pada platform dengan sumber daya terbatas, dan mengurangi ukuran model secara signifikan. Model CNN-LSTM mencapai akurasi sebesar 98,5% pada dataset Kaggle untuk penyakit buah sitrus.	Menerapkan model yang digunakan pada gambar pertanian waktu nyata dan menguji efektivitasnya dalam kondisi lingkungan yang bervariasi.	Penelitian ini menggunakan pendekatan transfer learning pada model DenseNet-169 dengan memanfaatkan teknik preprocessing dalam kombinasi fitur citra.
7	Citrus Pests and Diseases	Shuli Xing, Marelly Lee,	Mengembangkan model jaringan saraf	Weakly DenseNet-16, didesain dengan prinsip	Menerapkan model yang digunakan pada	Penelitian ini menggunakan dataset

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
	Recognition Model Using Weakly Dense Connected Convolution Network	Keun-kwang Lee Sensors, 2019	konvolusi (CNN) yang sederhana namun efektif untuk mengidentifikasi hama dan penyakit pada buah jeruk berdasarkan dataset gambar yang mereka miliki.	peningkatan efisiensi parameter. Mereka menggunakan dua lapisan konvolusi 1×1 untuk menyaring fitur-fitur yang tidak perlu dan mengabungkan fitur-fitur dari lapisan-lapisan sebelumnya dengan metode konkatenasi. Akurasi klasifikasi tertinggi yang dicapai oleh Weakly DenseNet-16 adalah 97,5% pada dataset validasi.	gambar pertanian waktu nyata dan menguji efektivitasnya dalam kondisi lingkungan yang bervariasi.	yang berfokus pada penyakit jeruk yang terdapat pada buah dan daun jeruk, sehingga tidak diperlukannya identifikasi pada bagian lain terlebih dahulu seperti hama penyebab penyakit.

2.3. Landasan Teori

2.3.1. Klasifikasi

Dalam penelitian ini, istilah 'deteksi' mengacu pada proses identifikasi ada atau tidak adanya penyakit pada citra daun atau buah jeruk. Sementara 'klasifikasi' adalah proses penentuan jenis penyakit (blackspot, canker, huanglongbing) atau status sehat. Kedua proses ini berjalan secara integral dalam model klasifikasi multi-kelas yang dibangun. Deteksi dan klasifikasi penting dalam berbagai bidang seperti pengolahan citra, analisis data, dan pengenalan pola. Deteksi adalah mengidentifikasi objek atau fenomena dalam data atau citra, sedangkan klasifikasi melibatkan pengelompokan objek ke dalam kategori yang telah ditentukan. Dalam pengolahan citra, deteksi dan klasifikasi digunakan untuk mengidentifikasi dan mengklasifikasikan objek dalam citra. Metode deteksi melibatkan segmentasi, ekstraksi fitur, dan pemrosesan statistik. Sementara itu klasifikasi melibatkan pengelompokan objek atau fitur ke dalam kategori atau kelas yang telah ditentukan sebelumnya. Metode klasifikasi dapat melibatkan penggunaan algoritma Machine Learning seperti Support Vector Machine (SVM), Naive Bayes, K-Nearest Neighbor (KNN), ataupun algoritma seperti Jaringan Saraf Tiruan (JST), Deep Learning/Neural Network, dan berbagai metode lainnya untuk mempelajari pola dari data latih dan mengklasifikasikan data uji berdasarkan pola yang telah dipelajari. Penelitian ini dikategorikan sebagai klasifikasi citra karena sistem yang dikembangkan bertujuan untuk mengenali dan mengelompokkan citra daun dan buah jeruk ke dalam kelas penyakit tertentu atau kondisi sehat berdasarkan ciri visual yang diekstraksi dari citra. Model melakukan analisis secara global terhadap

pola warna, tekstur, dan bentuk pada citra tanpa melakukan pemisahan area tertentu. Oleh karena itu, sistem tidak menghasilkan segmentasi maupun deteksi spasial terhadap bagian penyakit, melainkan hanya memberikan hasil berupa label kelas, sehingga pendekatan yang digunakan termasuk dalam klasifikasi citra multikelas.

2.3.2. Image Preprocessing

Image preprocessing atau pemrosesan citra merupakan langkah yang sangat penting dalam pengolahan citra. Pemrosesan citra mempunyai berbagai teknik dan metode yang bertujuan untuk meningkatkan dan mempersiapkan citra untuk analisis atau pemrosesan lanjutan. Preprocessing citra memainkan peran penting dalam meningkatkan kualitas citra, mengekstrak fitur yang relevan, dan meningkatkan kinerja algoritma analisis citra selanjutnya.

1) Image Enhancement

Peningkatan citra merupakan tugas penting dalam pemrosesan citra yang bertujuan untuk meningkatkan kualitas visual dari citra dengan memanipulasi nilai piksel atau karakteristiknya. Salah satu teknik yang umum digunakan untuk peningkatan warna adalah CES (Color Enhancement by Scaling). Pendekatan CES memiliki fitur unik yang menggunakan komponen kromatik dan pencahayaan untuk meningkatkan tampilan visual secara signifikan (Gavhale et al., 2014). Teknik CES menawarkan fleksibilitas dalam peningkatan warna dengan memungkinkan penyesuaian selektif pada komponen kromatik, CES dapat meningkatkan keseimbangan warna secara

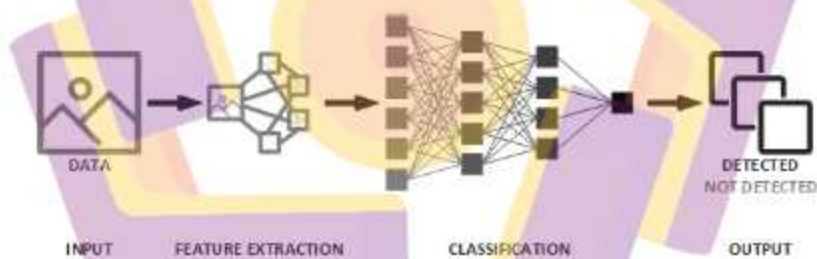
keseluruhan, memperbaiki kesalahan warna, dan menonjolkan detail-detail halus dalam citra.

2) Color Conversion

Konversi warna adalah proses mengubah representasi citra dari satu ruang warna ke ruang warna lainnya. Ruang warna adalah sistem yang digunakan untuk menggambarkan dan mengukur warna dalam citra. Terdapat berbagai jenis ruang warna yang digunakan dalam konversi warna, seperti RGB, CIELAB, YcbCr, dan HSV. RGB (Red, Green, Blue), yang merupakan model standar untuk representasi warna dalam citra digital. Model ini berdasarkan kombinasi intensitas dari tiga komponen warna dasar, yaitu merah, hijau, dan biru. Dalam model HSV (Hue, Saturation, Value), hue menggambarkan warna murni seperti yang dilihat oleh pengamat, sedangkan saturation merujuk pada tingkat keaslian relatif atau jumlah cahaya putih yang ditambahkan ke hue. Value (nilai) mengacu pada amplitudo cahaya. Selain itu, terdapat ruang warna lainnya seperti CIELAB dan YCbCr yang digunakan dalam konteks tertentu. CIELAB ($L^*a^*b^*$) merupakan ruang warna yang memiliki komponen luminans (L) untuk mewakili kecerahan dan dua komponen kromatis (a dan b) untuk mewakili warna. YCbCr digunakan secara luas dalam sistem video dan kompresi gambar, dengan komponen Y yang mewakili luminans dan komponen Cb dan Cr yang mewakili kromatisitas (Gavhale et al., 2014). Konversi warna memungkinkan pemisahan komponen warna yang signifikan, sehingga memudahkan analisis berdasarkan atribut warna tertentu.

2.3.3. Machine Learning

Machine learning adalah cabang dari ilmu komputer yang mempelajari bagaimana membuat sistem komputer dapat belajar dari data dan pengalaman tanpa diprogram secara eksplisit. Dasar teori machine learning memberikan kerangka kerja untuk memahami dan mengembangkan algoritma yang dapat belajar pola dan hubungan dari data. Algoritma machine learning bertujuan untuk membuat prediksi atau keputusan berdasarkan data yang diamati dengan mempelajari pola dan hubungan dari data tersebut. Inferensi statistik memberikan kerangka teoritis untuk mengestimasi parameter yang tidak diketahui, membuat prediksi, dan mengkuantifikasi ketidakpastian dalam model machine learning (Martino & Elvira, 2021). Berikut cara kerja Machine Learning seperti pada Gambar 2.1.



Gambar 2.1. Cara kerja algoritma Machine Learning

Umumnya, machine learning mengklasifikasikan tugas-tugasnya ke dalam beberapa skenario yang sering digunakan. Berikut adalah beberapa contoh skenario tersebut:

3) Supervised Learning

Dalam supervised learning, sistem komputer diajarkan menggunakan data yang telah diberi label, yang berisi informasi tentang kelas atau nilai target yang

diinginkan. Tujuan dari supervised learning adalah mengembangkan sistem komputer yang dapat memprediksi kelas atau nilai target dari data baru yang tidak memiliki label. Misalnya klasifikasi, di mana sistem diberi tugas untuk mengategorikan data ke dalam kelas-kelas yang telah ditentukan.

4) Unsupervised Learning

Dalam unsupervised learning, sistem komputer belajar menggunakan data yang tidak memiliki label, yang berarti data tersebut tidak memiliki informasi tentang kelas atau nilai target. Tujuan dari unsupervised learning adalah mengembangkan sistem komputer yang dapat mengidentifikasi pola, struktur, atau hubungan yang tersembunyi dalam data tersebut. Misalnya teknik clustering, di mana sistem berupaya mengelompokkan data ke dalam kelompok-kelompok yang memiliki kesamaan.

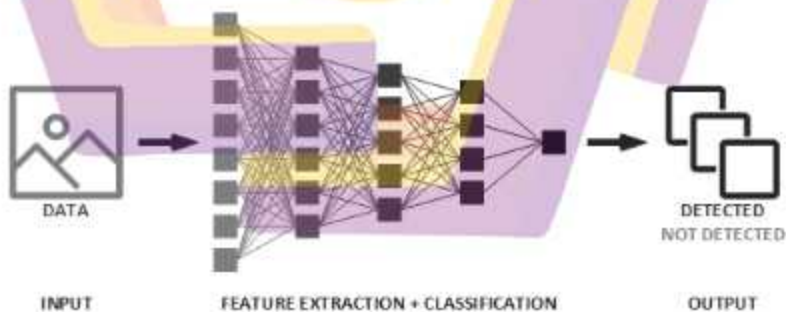
5) Reinforcement Learning

Dalam reinforcement learning, sistem komputer belajar melalui umpan balik yang diberikan berupa reward atau punishment berdasarkan aksi yang dilakukan dalam lingkungan tertentu. Tujuan dari reinforcement learning adalah mengembangkan sistem komputer yang dapat secara efektif memaksimalkan reward atau meminimalkan punishment dalam jangka panjang melalui pembelajaran dari pengalaman.

Deep Learning

Deep learning adalah cabang dari machine learning yang mempelajari bagaimana membuat sistem komputer dapat belajar dari data yang memiliki tingkat abstraksi yang tinggi dan kompleks. Deep learning dapat didefinisikan sebagai

proses pembelajaran yang menggunakan model matematika yang terdiri dari banyak lapisan pemrosesan non-linear yang dapat merepresentasikan data dengan tingkat abstraksi yang berbeda (Lecun et al., 2015). Deep learning memiliki banyak keunggulan dibandingkan dengan machine learning tradisional, seperti kemampuan untuk belajar fitur-fitur otomatis dari data mentah tanpa perlu ekstraksi fitur manual, kemampuan untuk menangani data yang memiliki dimensi tinggi dan kompleksitas tinggi, kemampuan untuk memodelkan fungsi-fungsi non-linear yang kompleks, dan kemampuan untuk memanfaatkan perkembangan teknologi komputasi paralel dan hardware khusus seperti GPU (Graphics Processing Unit). Deep learning juga memiliki banyak aplikasi dalam bidang-bidang seperti pengenalan pola, pengolahan citra, pengolahan bahasa alami, bioinformatika, robotika, kecerdasan buatan, dan lain-lain. Deep learning juga dapat digunakan dalam mendeteksi dan mengklasifikasikan penyakit tanaman untuk memprediksi kelas atau label yang sesuai seperti pada Gambar 2.2 berikut.



Gambar 2.2. Cara kerja Deep Learning

2.3.4. Convolutional Neural Network

Convolutional Neural Network (CNN) adalah jenis deep learning yang terdiri dari lapisan-lapisan konvolusi dan pooling yang dapat mengekstrak fitur-fitur lokal dan invarian dari data citra. CNN dapat didefinisikan sebagai jaringan saraf tiruan yang terdiri dari banyak lapisan pemrosesan yang dapat merepresentasikan data citra dengan tingkat abstraksi yang berbeda (LeCun et al., 1998). Lapisan konvolusi pada CNN bertugas untuk mengekstrak fitur-fitur lokal dari data citra dengan cara melakukan operasi konvolusi antara data citra dengan filter-filter yang memiliki bobot-bobot yang dapat dipelajari. Lapisan pooling pada CNN bertugas untuk mereduksi dimensi spasial dari data citra dengan cara melakukan operasi subsampling atau downsampling pada data citra. Lapisan-lapisan konvolusi dan pooling pada CNN biasanya diikuti oleh lapisan-lapisan fully connected yang bertugas untuk melakukan klasifikasi atau regresi pada fitur-fitur yang telah diekstrak oleh lapisan-lapisan konvolusi dan pooling.

CNN memiliki banyak keunggulan dibandingkan dengan model deep learning lainnya, seperti kemampuan untuk menangani data citra yang memiliki dimensi tinggi dan kompleksitas tinggi, kemampuan untuk mengekstrak fitur-fitur lokal dan invarian secara otomatis dari data citra tanpa perlu ekstraksi fitur manual, kemampuan untuk memodelkan fungsi-fungsi non-linear yang kompleks, dan kemampuan untuk memanfaatkan perkembangan teknologi komputasi paralel dan hardware khusus seperti GPU (Graphics Processing Unit). CNN juga memiliki banyak aplikasi dalam bidang-bidang seperti pengenalan pola, pengolahan citra, pengolahan video, pengenalan suara, pengenalan teks, dan lain-lain. CNN juga

dapat digunakan untuk mengenali karakter tulisan tangan seperti pada penelitian di halaman web saat ini.

2.3.4.1. Layer Ekstraksi Fitur

Layer Ekstraksi Fitur adalah lapisan pada jaringan saraf tiruan yang bertugas untuk mengekstrak fitur-fitur penting dari data masukan. Layer Ekstraksi Fitur dapat ditemukan pada berbagai jenis jaringan saraf tiruan, seperti deep neural network, convolutional neural network, dan lain-lain. Pada convolutional neural network (CNN), Layer Ekstraksi Fitur terdiri dari lapisan-lapisan konvolusi dan pooling. Lapisan konvolusi bertugas untuk mengekstrak fitur-fitur lokal dari data citra dengan cara melakukan operasi konvolusi antara data citra dengan filter-filter yang memiliki bobot-bobot yang dapat dipelajari. Lapisan pooling bertugas untuk mereduksi dimensi spasial dari data citra dengan cara melakukan operasi subsampling atau downsampling pada data citra.

Layer Ekstraksi Fitur memiliki peran yang sangat penting dalam pembelajaran mesin dan deep learning karena fitur-fitur yang diekstrak oleh Layer Ekstraksi Fitur dapat mempengaruhi kinerja model dalam melakukan klasifikasi atau regresi. Oleh karena itu, desain dan pelatihan Layer Ekstraksi Fitur harus dilakukan dengan hati-hati agar model dapat mengekstrak fitur-fitur yang relevan dan diskriminatif dari data masukan.

a) Lapisan Konvolusional

Lapisan Konvolusional adalah lapisan pada convolutional neural network (CNN) yang bertugas untuk mengekstrak fitur-fitur lokal dari data citra dengan cara melakukan operasi konvolusi antara data citra dengan filter-filter yang memiliki

bobot-bobot yang dapat dipelajari. Lapisan Konvolusional dapat didefinisikan sebagai lapisan yang terdiri dari banyak unit konvolusi yang masing-masing memiliki filter yang berbeda dan menghasilkan feature map yang berbeda.

Operasi konvolusi pada Lapisan Konvolusional adalah operasi matematika yang menghitung nilai output dari suatu unit konvolusi dengan cara mengalikan nilai input dari suatu daerah lokal pada data citra dengan nilai bobot dari filter, kemudian menjumlahkan hasil perkalian tersebut. Operasi konvolusi dapat dilakukan dengan berbagai cara, seperti menggunakan stride, padding, atau dilation.

Lapisan Konvolusional memiliki beberapa keunggulan dibandingkan dengan lapisan fully connected, seperti kemampuan untuk mengekstrak fitur-fitur lokal dan invarian secara otomatis dari data citra tanpa perlu ekstraksi fitur manual, kemampuan untuk mengurangi jumlah parameter dan komputasi yang dibutuhkan, dan kemampuan untuk memanfaatkan perkembangan teknologi komputasi paralel dan hardware khusus seperti GPU (Graphics Processing Unit). Lapisan Konvolusional juga memiliki beberapa tantangan, seperti pemilihan ukuran filter, jumlah filter, stride, padding, dan fungsi aktivasi yang sesuai dengan data masukan dan tujuan model.

b) Lapisan Pooling

Lapisan Pooling adalah lapisan pada convolutional neural network (CNN) yang bertugas untuk mereduksi dimensi spasial dari data citra dengan cara melakukan operasi subsampling atau downsampling pada data citra. Lapisan Pooling dapat didefinisikan sebagai lapisan yang terdiri dari banyak unit pooling yang masing-masing memiliki ukuran dan stride yang sama dan menghasilkan

feature map yang lebih kecil. Operasi subsampling atau downsampling pada Lapisan Pooling adalah operasi matematika yang menghitung nilai output dari suatu unit pooling dengan cara mengambil nilai statistik tertentu dari suatu daerah lokal pada data citra. Operasi subsampling atau downsampling dapat dilakukan dengan berbagai cara, seperti menggunakan max pooling, average pooling, atau min pooling.

Lapisan Pooling memiliki beberapa keunggulan, seperti kemampuan untuk mengurangi jumlah parameter dan komputasi yang dibutuhkan, kemampuan untuk meningkatkan invariansi terhadap translasi, rotasi, dan skala, dan kemampuan untuk mencegah overfitting. Lapisan Pooling juga memiliki beberapa tantangan, seperti pemilihan ukuran dan stride yang sesuai dengan data masukan dan tujuan model.

c) **Activation Function**

Activation Function adalah fungsi matematika yang digunakan untuk menentukan nilai output dari suatu unit atau neuron pada jaringan saraf tiruan. Activation Function dapat didefinisikan sebagai fungsi yang memetakan nilai input dari suatu unit atau neuron ke nilai output yang biasanya memiliki rentang tertentu. Activation Function memiliki peran yang sangat penting dalam jaringan saraf tiruan karena dapat menentukan apakah suatu unit atau neuron aktif atau tidak, dapat menambahkan non-linearitas pada model, dan dapat mempengaruhi proses pembelajaran dan konvergensi model. Activation Function juga dapat mempengaruhi kinerja model dalam melakukan klasifikasi atau regresi.

Terdapat berbagai jenis Activation Function yang dapat digunakan pada jaringan saraf tiruan, seperti linear function, sigmoid function, tanh function, ReLU function, softmax function, dan lain-lain. Masing-masing Activation Function memiliki karakteristik, keunggulan, dan kelemahan yang berbeda. Pemilihan Activation Function yang sesuai dengan data masukan dan tujuan model adalah salah satu faktor yang dapat mempengaruhi kualitas model.

2.3.4.2. Lapisan Klasifikasi

Lapisan Klasifikasi adalah lapisan pada jaringan saraf tiruan yang bertugas untuk melakukan klasifikasi atau pengelompokan data masukan ke dalam kelas-kelas yang telah ditentukan. Lapisan Klasifikasi biasanya merupakan lapisan terakhir atau output pada model jaringan saraf tiruan. Lapisan Klasifikasi terdiri dari unit-unit atau neuron yang mewakili kelas-kelas yang ada. Jumlah unit atau neuron pada Lapisan Klasifikasi harus sesuai dengan jumlah kelas yang ada. Nilai output dari unit atau neuron pada Lapisan Klasifikasi dapat diinterpretasikan sebagai probabilitas atau kepercayaan bahwa data masukan termasuk ke dalam kelas yang bersangkutan. Lapisan Klasifikasi biasanya menggunakan Activation Function yang sesuai dengan jenis klasifikasi yang dilakukan. Untuk klasifikasi biner, Activation Function yang sering digunakan adalah sigmoid function, yang dapat menghasilkan nilai output antara 0 dan 1. Untuk klasifikasi multikelas, Activation Function yang sering digunakan adalah softmax function, yang dapat menghasilkan nilai output antara 0 dan 1 dan menjumlahkan menjadi 1. Lapisan Klasifikasi juga membutuhkan fungsi loss atau fungsi biaya yang dapat mengukur seberapa baik model dapat melakukan klasifikasi. Fungsi loss yang sering

digunakan adalah cross-entropy loss, yang dapat mengukur seberapa dekat nilai output dengan nilai target.

2.3.5. Transfer Learning

Transfer learning adalah teknik dalam pembelajaran mesin (machine learning) di mana pengetahuan yang dipelajari dari sebuah tugas digunakan kembali untuk meningkatkan kinerja pada tugas yang terkait. Misalnya, untuk klasifikasi gambar, pengetahuan yang diperoleh saat belajar mengenali mobil dapat diterapkan saat mencoba mengenali truk. Dengan menggunakan/memindahkan informasi dari tugas yang telah dipelajari sebelumnya ke tugas baru, transfer learning berpotensi meningkatkan efisiensi pembelajaran secara signifikan (Al-rimy et al., 2023).

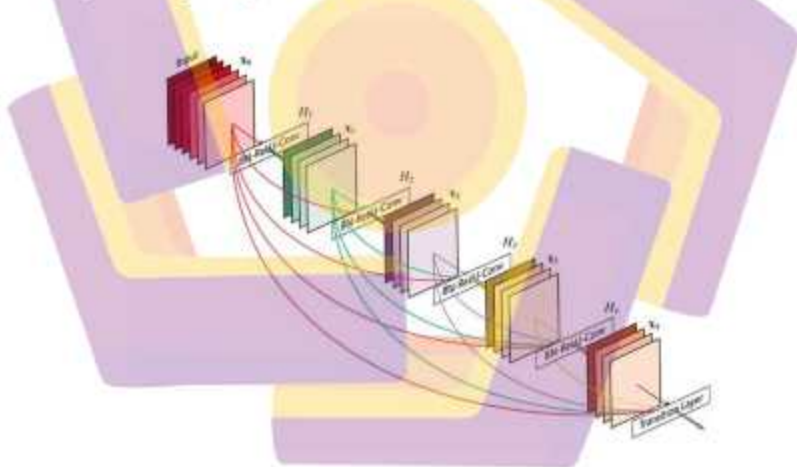
2.3.6. Kombinasi Fitur

Kombinasi fitur dalam computer vision mengacu pada proses menggabungkan atau memadukan berbagai jenis fitur atau representasi visual yang diekstraksi dari data citra atau video untuk meningkatkan performa dalam berbagai tugas computer vision (Azaza et al., 2021). Selain itu, kombinasi fitur digunakan untuk memadukan fitur-fitur yang diekstraksi secara manual (hand-crafted) dengan fitur-fitur yang dipelajari secara otomatis menggunakan deep learning untuk meningkatkan akurasi dalam tugas-tugas seperti deteksi objek, pengenalan ekspresi wajah, dan segmentasi citra (Cao, 2022).

2.3.7. Densenet-169

DenseNet-169 adalah salah satu arsitektur jaringan saraf konvolusional (CNN) yang dikembangkan oleh (Huang et al., 2017) untuk meningkatkan kinerja dan efisiensi CNN dalam tugas pengenalan objek. Gambar 7.3 mengilustrasikan

skema dari blok padat (dense block) yang merupakan komponen utama dari arsitektur DenseNet. Blok padat terdiri dari beberapa lapisan konvolusi yang saling terhubung secara langsung, sehingga setiap lapisan menerima semua peta ciri (feature-map) dari lapisan-lapisan sebelumnya sebagai masukan, dan menghasilkan k peta ciri baru sebagai keluaran. Parameter k disebut sebagai laju pertumbuhan (growth rate) karena menentukan seberapa banyak informasi baru yang ditambahkan oleh setiap lapisan. Dengan demikian, blok padat memfasilitasi aliran informasi dan gradien antara lapisan-lapisan dalam jaringan, dan mendorong penggunaan kembali fitur yang telah dipelajari. Gambar 2.3 menunjukkan contoh blok padat dengan 5 lapisan dan $k = 4$.



Gambar 2.3. Blok Padat dengan 5 Lapisan dan $k = 4$

Arsitektur DenseNet-169 memiliki parameter yang dapat dilatih jauh lebih sedikit dibandingkan dengan arsitektur lain. Selain itu, menurut (Huang et al., 2017) DenseNet-169 adalah salah satu arsitektur DenseNet yang memiliki kinerja terbaik pada dataset ImageNet, yang terdiri dari 1,2 juta gambar dari 1.000 kelas.

DenseNet-169 memiliki 169 lapisan dan 14,3 juta parameter, dan mencapai tingkat kesalahan top-1 sebesar 23,80% dan top-5 sebesar 6,85% pada validasi ImageNet. Ini lebih baik daripada DenseNet-121 yang memiliki 121 lapisan dan 8 juta parameter, dan mencapai tingkat kesalahan top-1 sebesar 25,02% dan top-5 sebesar 7,71%. DenseNet-169 juga lebih baik daripada DenseNet-201 yang memiliki 201 lapisan dan 20 juta parameter, dan mencapai tingkat kesalahan top-1 sebesar 22,58% dan top-5 sebesar 6,34%. Oleh karena itu, DenseNet-169 membantu meningkatkan kedalaman CNN yang dalam sambil menghindari hilangnya informasi, yang terjadi ketika jalur antara lapisan input dan output menjadi terlalu besar. Dengan mengurangi jumlah parameter, DenseNet-169 menghilangkan peta fitur yang berlebihan sehingga mengurangi jumlah filter (Rubin et al., 2018).

Gambar 2.4 menunjukkan arsitektur DenseNet-169.



Gambar 2.4. Arsitektur DenseNet-169

2.3.7.1. Relevansi DenseNet-169 untuk Deteksi Penyakit Tanaman

DenseNet-169 merupakan arsitektur Convolutional Neural Network (CNN) yang sangat sesuai untuk tugas klasifikasi citra penyakit tanaman, khususnya pada tanaman jeruk. Keunggulan utama dari arsitektur ini terletak pada mekanisme koneksi padat (dense connections) yang memungkinkan setiap lapisan menerima feature maps dari semua lapisan sebelumnya (Huang et al., 2017). Berikut adalah analisis mendalam mengenai relevansi DenseNet-169 dalam konteks penelitian ini:

a) Preservasi dan Propagasi Fitur Multi-Skala

Gejala penyakit tanaman jeruk, seperti blackspot, canker, dan huanglongbing, muncul dalam berbagai skala dan tingkat kompleksitas visual. Blackspot bermanifestasi sebagai bercak hitam lokal, canker sebagai lesi berbentuk lingkaran dengan tepi berminyak, sedangkan huanglongbing menyebabkan perubahan warna asimetris dan pola blotchy mottle yang menyebar (Syed-Ab-Rahman et al., 2022). DenseNet-169, melalui mekanisme feature reuse, mempertahankan dan menggabungkan fitur tingkat rendah (tepi, tekstur dasar, variasi warna lokal) dengan fitur tingkat tinggi (pola kompleks, konteks global). Hal ini memungkinkan model untuk secara simultan mendeteksi lesi lokal seperti bercak blackspot sekaligus mengenali pola global seperti perubahan warna akibat greening, sehingga meningkatkan akurasi klasifikasi pada variasi gejala yang beragam.

b) Efisiensi Parameter dan Mitigasi Overfitting

Dataset citra penyakit tanaman seperti LDI dan FDI, meskipun telah melalui augmentasi, tetap memiliki ukuran terbatas dibandingkan dataset skala besar seperti ImageNet. DenseNet-169 dirancang dengan prinsip efisiensi parameter melalui growth rate yang rendah, menghasilkan model dengan parameter yang lebih sedikit dibandingkan arsitektur dengan kedalaman serupa seperti ResNet (Huang et al., 2017). Efisiensi ini mengurangi risiko overfitting dan meningkatkan kemampuan generalisasi model pada data baru atau variasi kondisi lingkungan yang tidak terdapat dalam data pelatihan. Menurut penelitian (Xing et al., 2019), arsitektur

berbasis koneksi padat seperti DenseNet menunjukkan performa yang lebih stabil pada dataset citra pertanian dengan jumlah sampel terbatas.

c) Aliran Gradien yang Diperkuat untuk Pembelajaran Optimal

Masalah vanishing gradient sering menghambat pelatihan model CNN yang dalam pada dataset spesifik domain. Mekanisme dense connections dalam DenseNet-169 menciptakan jalur alternatif yang lebih pendek untuk aliran gradien selama backpropagation, memastikan bahwa lapisan awal tetap menerima sinyal pembelajaran yang kuat (Huang et al., 2017). Dalam konteks deteksi penyakit tanaman, di mana fitur pembeda halus (seperti tekstur permukaan daun atau gradasi warna) sangat krusial untuk diagnosis dini, kemampuan untuk melatih seluruh lapisan jaringan secara efektif menjadi faktor penentu akurasi model.

d) Kesesuaian dengan Karakteristik Citra Pertanian

Citra pertanian di lapangan sering kali mengandung variasi pencahayaan, sudut pengambilan, oklusi, dan latar belakang yang kompleks. DenseNet-169, dengan kapasitasnya untuk membangun representasi fitur yang kaya dan hierarkis, menunjukkan ketahanan (robustness) yang baik terhadap variasi-variasi non-esensial tersebut. Fitur warna dari ruang HSV yang diekstrak pada tahap pra-pemrosesan dapat dikombinasikan secara efektif dengan fitur tekstur dan bentuk yang dipelajari oleh lapisan konvolusi DenseNet, menghasilkan representasi yang lebih diskriminatif untuk membedakan penyakit dengan gejala visual yang mirip.

e) Dukungan Empiris dari Penelitian Terdahulu

Beberapa penelitian dalam bidang precision agriculture telah mengadopsi dan memvalidasi keefektifan arsitektur DenseNet. (Albahli & Nawaz, 2022)

berhasil mengimplementasikan DenseNet-77 sebagai backbone dalam model DCNet untuk deteksi penyakit daun tomat, mencapai akurasi 99,98%. Shireesha & Reddy (2022) juga melaporkan kinerja DenseNet-121 yang sangat baik (96% akurasi) dalam klasifikasi penyakit jeruk. Pemilihan DenseNet-169 dalam penelitian ini didasarkan pada pertimbangan keseimbangan antara kapasitas model (lebih dalam dari DenseNet-121) dan efisiensi komputasi (lebih ringan dari DenseNet-201), sehingga optimal untuk eksperimen komparatif dengan dan tanpa teknik kombinasi fitur tambahan.

Dengan demikian, pemilihan DenseNet-169 tidak hanya didasarkan pada kinerja empirisnya pada dataset umum, tetapi terutama pada kesesuaian karakteristik arsitekturalnya dengan tantangan spesifik dalam domain computer vision untuk pertanian, yaitu kebutuhan akan analisis multi-skala, efisiensi parameter, aliran gradien yang stabil, dan ketahanan terhadap variasi citra di lapangan.

BAB III

METODE PENELITIAN

3.1. Jenis, Sifat, dan Pendekatan Penelitian

Penelitian ini merupakan sebuah penelitian eksperimental yang bertujuan untuk menguji dan membandingkan kinerja model Convolutional Neural Network (CNN) DenseNet-169, baik dalam bentuk aslinya maupun dengan penambahan kombinasi fitur, dalam melakukan klasifikasi jenis penyakit pada tanaman jeruk. Penelitian ini juga bersifat kuantitatif karena melibatkan penggunaan data yang dapat diukur secara kuantitatif, seperti akurasi, presisi, dan recall, untuk mengevaluasi hasil kinerja model klasifikasi tersebut. Model yang dikembangkan berfungsi sebagai sistem deteksi dan klasifikasi terintegrasi, dimana output utama adalah label kelas yang sekaligus mengindikasikan terdeteksinya penyakit tertentu atau kondisi sehat.

Penelitian ini menggunakan 2 pendekatan. Pertama, menggunakan model CNN beserta arsitektur DenseNet-169 untuk melakukan ekstraksi fitur dari gambar penyakit jeruk. DenseNet-169 relatif lebih efisien dalam pelatihan dibandingkan dengan model lain dan memiliki kemampuan untuk mengurangi masalah gradien menghilang, memperkuat propagasi fitur, mendorong penggunaan fitur kembali, dan secara substansial mengurangi jumlah parameter. Kedua, setelah ekstraksi fitur dengan model DenseNet-169, dilakukan pendekatan dalam deteksi dan klasifikasi menggunakan lapisan terakhir dari model tersebut sebagai input untuk model CNN baru yang akan digunakan untuk melakukan klasifikasi jenis penyakit jeruk.

Dengan menggunakan DenseNet-169, model dapat membedakan dan mengklasifikasikan penyakit jeruk ke dalam kategori penyakit yang tepat berdasarkan fitur-fitur yang teridentifikasi melalui proses ekstraksi. Dengan demikian, meskipun istilah deteksi digunakan dalam konteks identifikasi penyakit, model yang dikembangkan secara metodologis merupakan sistem klasifikasi citra, karena tidak menghasilkan informasi lokasi atau batas area penyakit pada citra.

3.2. Metode Pengumpulan Data

Dataset yang digunakan dalam penelitian ini adalah dataset penyakit tanaman jeruk yang terbagi menjadi dua jenis kumpulan data yang berbeda, yang pertama LDI (*Leaf Disease Image*) dan kedua FDI (*Fruit Disease Image*) dataset ini masing-masing terdiri dari 596 dan 1789 gambar. LDI adalah dataset yang digunakan untuk melakukan identifikasi penyakit pada daun jeruk. Dataset ini berisi gambar daun yang terinfeksi penyakit dan sehat. Dataset ini digunakan untuk melatih model *deep learning* dalam melakukan klasifikasi jenis penyakit pada tanaman. Dataset ini terdiri dari 4 kelas, dimana terdapat 3 jenis penyakit daun jeruk yaitu *Blackspot*, *Canker*, dan HLB, serta terdapat 1 kelas daun sehat atau *Healthy*. Sedangkan FDI adalah dataset yang digunakan untuk melakukan identifikasi penyakit pada buah jeruk. Dataset ini berisi gambar buah yang terinfeksi penyakit dan sehat. Sama seperti sebelumnya, dataset ini digunakan untuk melatih model *deep learning*, selain itu dataset ini juga terdiri dari 4 kelas yang sama seperti dataset LDI. Dataset tersebut di unduh dan bersumber dari kaggle, dan juga sudah pernah digunakan oleh (Syed-Ab-Rahman et al., 2022) dan (Silva et al., 2021) dalam penelitiannya tentang penyakit jeruk.

3.3. Metode Analisis Data

Penelitian ini menggunakan metode analisis data yang mengombinasikan teknik kombinasi fitur pada preprocessing citra gambar dengan model DenseNet-169 untuk deteksi dan klasifikasi empat jenis penyakit jeruk. Pada tahap pertama, teknik kombinasi fitur digunakan. Tahap ini, citra input ditingkatkan kualitasnya dengan teknik *Color Enhancement by Scaling (CES enhancement)* untuk memperbaiki komponen *chromatic* dan *luminance* agar deteksi dan klasifikasi lebih baik. Selanjutnya, fitur diekstraksi dengan menggunakan model warna HSV, YCbCr, dan LAB untuk meningkatkan akurasi klasifikasi.

Tahap kedua, model DenseNet-169 dilatih menggunakan data gambar penyakit jeruk hasil pemrosesan gambar dengan teknik kombinasi fitur untuk mengidentifikasi pola-pola penting dalam gambar tersebut. Bersamaan dengan ekstraksi fitur, kemampuan model CNN dalam seleksi fitur dimanfaatkan dengan menggunakan teknik seperti L1 regularization untuk mendapatkan fitur yang lebih efektif dalam tahap klasifikasi. Model DenseNet-169 digunakan sebagai algoritma untuk mendeteksi dan mengklasifikasi empat kelas jeruk yang berbeda dengan tepat berdasarkan fitur-fitur yang teridentifikasi melalui proses ekstraksi. Dengan menggunakan kombinasi fitur sebagai teknik preprocessing dan DenseNet-169, penelitian ini bertujuan untuk mencapai tingkat akurasi yang tinggi dalam mengklasifikasikan jenis penyakit jeruk.

3.4. Dataset

Dalam penelitian ini, digunakan dataset LDI (*Leaf Disease Image*) yang dikembangkan oleh (Syed-Ab-Rahman et al., 2022) dan FDI (*Fruit Disease Image*) yang dikembangkan oleh (Silva et al., 2021). Dataset ini juga digunakan oleh (Elaraby et al., 2022) dan (Faisal et al., 2023). Dataset pertama adalah LDI, (Syed-Ab-Rahman et al., 2022) mengembangkan dataset yang berisi gambar daun jeruk yang sehat dan terinfeksi dengan tiga jenis penyakit (blackspot, canker, dan greening). Data gambar daun jeruk dikumpulkan dari situs web kaggle. Selain itu terdapat beberapa langkah pra-pemrosesan pada gambar, seperti mengubahnya menjadi skala abu-abu, melakukan histogram equalization, dan mengkonversinya menjadi gambar biner. Gambar biner digunakan untuk mendefinisikan kotak pembatas di sekitar daun dan memisahkannya dari latar belakang. Kemudian, mereka menerapkan informasi kotak pembatas pada gambar asli untuk mengekstrak gambar daun dari latar belakang. Dataset ke-dua adalah FDI, (Silva et al., 2021) mengembangkan dataset yang berisi gambar daun jeruk yang sehat dan terinfeksi dengan tiga jenis penyakit (blackspot, canker, dan greening). Data gambar daun jeruk dikumpulkan dari situs web kaggle. Dalam penelitian ini, dataset FDI (*Fruit Disease Image*) dan LDI (*Leaf Disease Image*) digunakan untuk mendeteksi dan mengklasifikasikan penyakit pada tanaman jeruk menggunakan pendekatan deep learning dan pengolahan citra. Dataset FDI dan LDI berisi gambar buah jeruk yang terinfeksi penyakit seperti blackspot, canker, dan greening (huanglongbing) serta dataset jeruk yang sehat. Selain itu, dataset daun jeruk dapat digunakan sebagai tahap awal diagnosis penyakit jeruk (Singh et al., 2020). Begitupun juga dengan

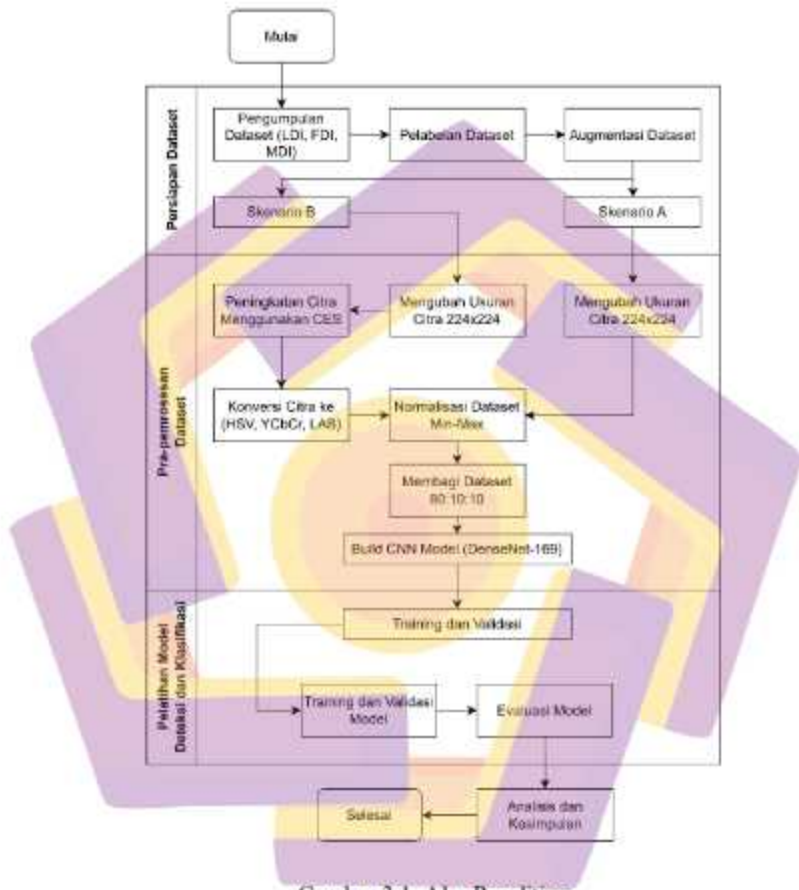
dataset buah jeruk, daun atau buah jeruk yang terdeteksi penyakit akan membantu petani untuk menetapkan kualitas dan kuantitas dari buah jeruk. Penyakit pada tanaman jeruk memiliki gejala yang berbeda-beda pada buah dan daun, sehingga perlu dilakukan deteksi dan klasifikasi secara terpisah. Akan tetapi, dataset yang digunakan dalam penelitian ini memiliki nama penyakit yang sama pada penyakit tanaman jeruk, sehingga eksperimen menggabungkan kedua dataset tersebut perlu dilakukan untuk mengetahui efektivitas model deteksi dan klasifikasi pada penyakit jeruk.

Dataset LDI dan FDI dicirikan oleh latar belakang yang relatif terkontrol dan objek yang terpusat. Meskipun augmentasi telah dilakukan, variasi tekstur dan kondisi lingkungan dalam gambar terbatas. Karakteristik ini memungkinkan model seperti DenseNet-169, yang dirancang untuk ekstraksi fitur bertingkat dan efisien, untuk mencapai akurasi tinggi. Namun, homogenitas ini juga menjadi peringatan untuk potensi overfitting, dimana model mungkin terlalu spesifik mempelajari artefak dataset daripada fitur general penyakit jeruk yang robust (Dhiman et al., 2023).

3.5. Alur Penelitian

Penelitian ini menggunakan pendekatan eksperimental dengan menerapkan arsitektur Deep Convolutional Neural Network (DCNN) berbasis DenseNet-169 untuk klasifikasi penyakit pada tanaman jeruk. Secara keseluruhan, alur penelitian terbagi ke dalam lima tahap utama, yaitu persiapan dataset, pra-pemrosesan dataset, pemrosesan dataset, pelatihan model deteksi dan klasifikasi, serta analisis

dan kesimpulan. Diagram alur penelitian secara visual disajikan pada Gambar 3.1 di bawah ini.



Gambar 3.1. Alur Penelitian

Pada Gambar diatas menunjukkan empat tahapan utama yang dilakukan secara sistematis, yaitu persiapan dataset, pra-pemrosesan data, pelatihan model dan klasifikasi, serta analisis dan kesimpulan. Berikut adalah penjelasan mendalam untuk setiap tahapan.

1) Persiapan Dataset

Tahap persiapan dataset merupakan tahap awal yang sangat penting dalam penelitian ini karena kualitas data masukan sangat memengaruhi performa model deep learning yang dikembangkan. Pada tahap ini dilakukan tiga proses utama, yaitu pengumpulan dataset, pelabelan dataset, dan augmentasi dataset.

a) Pengumpulan Dataset (Data Collection)

Dataset yang digunakan dalam penelitian ini merupakan data sekunder yang diperoleh dari repositori publik Kaggle. Dataset tersebut terbagi menjadi dua jenis utama, yaitu Leaf Disease Image (LDI) yang berisi citra daun jeruk dan Fruit Disease Image (FDI) yang berisi citra buah jeruk. Dataset LDI terdiri dari 596 citra, sedangkan dataset FDI terdiri dari 1.789 citra, sehingga total dataset awal yang digunakan berjumlah 2.385 citra. Pemilihan dataset ini didasarkan pada pertimbangan bahwa dataset tersebut telah digunakan pada beberapa penelitian terdahulu yang relevan dalam bidang deteksi dan klasifikasi penyakit tanaman jeruk. Dengan demikian, validitas dan relevansi dataset terhadap permasalahan penelitian telah teruji. Selain itu, penggunaan dataset publik memungkinkan penelitian ini untuk direplikasi dan dibandingkan dengan penelitian lain yang sejenis. Dataset LDI dan FDI masing-masing memiliki karakteristik visual yang berbeda. Citra daun jeruk umumnya menampilkan pola tekstur dan perubahan warna pada permukaan daun, sedangkan citra buah jeruk lebih menonjolkan variasi warna, bercak, dan deformasi pada kulit buah.

Perbedaan karakteristik ini menjadi dasar pertimbangan dalam melakukan pengujian model secara terpisah maupun gabungan.

b) Pelabelan Dataset (Data Labeling)

Setelah dataset dikumpulkan, tahap selanjutnya adalah pelabelan dataset. Pelabelan dilakukan dengan mengelompokkan setiap citra ke dalam kelas tertentu berdasarkan informasi penyakit yang menyertai dataset tersebut. Dalam penelitian ini, digunakan empat kelas kategori, yaitu tiga kelas penyakit jeruk yang terdiri dari blackspot, canker, dan huanglongbing (greening), serta satu kelas non-penyakit yaitu healthy. Proses pelabelan dilakukan dengan menyusun struktur direktori folder terpisah untuk setiap kelas pada masing-masing dataset LDI dan FDI. Setiap citra ditempatkan pada folder yang sesuai dengan label kelasnya. Struktur folder ini memudahkan proses pembacaan data secara otomatis oleh sistem deep learning dan memastikan bahwa data masukan sesuai dengan label target selama proses pelatihan dan evaluasi model.

c) Augmentasi Dataset (Data Augmentation)

Jumlah dataset awal yang tersedia relatif terbatas dan memiliki distribusi data yang tidak seimbang antar kelas. Kondisi ini berpotensi menyebabkan model mengalami overfitting dan memiliki kemampuan generalisasi yang rendah. Oleh karena itu, diterapkan teknik augmentasi dataset untuk meningkatkan variasi data dan menyeimbangkan jumlah sampel pada setiap kelas. Teknik augmentasi yang digunakan meliputi rotasi, pergeseran (shifting), pembalikan (flipping), zooming, perubahan skala,

pengisian piksel kosong, serta distorsi citra. Melalui proses augmentasi ini, variasi citra diperbanyak tanpa mengubah label kelas, sehingga model dapat mempelajari pola visual yang lebih beragam. Hasil dari proses augmentasi ini adalah setiap kelas pada dataset LDI dan FDI memiliki jumlah citra yang sama, yaitu 750 citra per kelas. Selain itu, dataset LDI dan FDI juga digabungkan untuk membentuk dataset gabungan yang disebut Mixed Disease Image (MDI). Penggabungan ini dilakukan untuk menguji kemampuan model dalam melakukan klasifikasi penyakit jeruk pada citra yang berasal dari organ tanaman yang berbeda, yaitu daun dan buah, dalam satu model yang sama.

2) Pra-pemrosesan Dataset (Data Pre-processing)

Tahap pra-pemrosesan dataset bertujuan untuk menstandarisasi data masukan dan meningkatkan kualitas citra sebelum digunakan dalam proses pelatihan model. Tahapan ini sangat penting karena citra yang tidak seragam dan memiliki kualitas rendah dapat menurunkan performa model deep learning.

a) Penyesuaian Ukuran Citra

Citra yang diperoleh dari dataset memiliki resolusi dan ukuran yang bervariasi. Untuk mengatasi perbedaan tersebut, seluruh citra diubah ukurannya menjadi 224×224 piksel. Ukuran ini dipilih agar sesuai dengan ukuran input standar pada arsitektur DenseNet-169 yang digunakan dalam penelitian ini. Penyesuaian ukuran citra juga bertujuan untuk menjaga konsistensi dimensi data serta mengurangi beban komputasi selama proses pelatihan.

b) Peningkatan Kualitas Citra

Setelah proses resizing, dilakukan peningkatan kualitas citra menggunakan teknik Contrast Enhancement System (CES). Teknik ini difokuskan pada perbaikan komponen kromatik dan luminans citra agar pola visual penyakit jeruk dapat terlihat lebih jelas. Pada tahap ini dilakukan reduksi noise menggunakan filter Median dan Gaussian untuk menghilangkan derau yang dapat mengganggu proses ekstraksi fitur. Selain itu, penyesuaian kontras dan kecerahan juga dilakukan untuk menonjolkan perbedaan visual antara area sehat dan area yang terinfeksi penyakit. Proses peningkatan kualitas citra ini diterapkan pada seluruh jenis dataset, yaitu LDI, FDI, dan MDI.

c) Konversi Ruang Warna dan Kombinasi Fitur

Selain menggunakan citra dalam ruang warna RGB, penelitian ini juga mengeksplorasi penggunaan ruang warna lain, yaitu HSV, YCbCr, dan LAB. Konversi ruang warna ini bertujuan untuk mengekstraksi fitur tambahan yang tidak sepenuhnya terwakili dalam ruang warna RGB. Ruang warna HSV digunakan untuk memisahkan informasi warna dari intensitas cahaya, sehingga efektif dalam menangkap perubahan pigmen warna pada daun dan buah jeruk. Ruang warna YCbCr digunakan untuk memisahkan komponen luminans dan krominans, yang bermanfaat dalam ekstraksi fitur tekstur. Sementara itu, ruang warna LAB digunakan untuk merepresentasikan warna mendekati persepsi visual manusia dan mampu menangkap variasi warna serta tekstur yang lebih halus. Pengujian

dilakukan dalam dua skenario, yaitu penggunaan citra hasil peningkatan kualitas tanpa konversi ruang warna dan penggunaan citra dengan kombinasi fitur hasil konversi ruang warna.

d) Normalisasi Dataset

Untuk mempercepat proses pelatihan dan menjaga stabilitas pembaruan bobot, dilakukan normalisasi dataset menggunakan metode Min-Max. Normalisasi ini mengubah rentang nilai piksel dari 0–255 menjadi 0–1. Dengan normalisasi, perbedaan skala antar fitur dapat diminimalkan sehingga proses optimasi berjalan lebih stabil.

e) Pembagian Data

Dataset yang telah melalui tahap pra-pemrosesan kemudian dibagi menjadi tiga bagian, yaitu data latih sebesar 80%, data validasi sebesar 10%, dan data uji sebesar 10%. Pembagian dilakukan menggunakan teknik stratified sampling agar distribusi kelas pada setiap subset tetap seimbang. Data latih digunakan untuk melatih model, data validasi untuk memantau performa selama pelatihan, dan data uji untuk mengevaluasi kinerja model secara objektif.

3) Pelatihan Model dan Klasifikasi

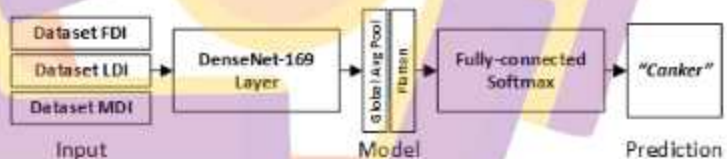
Tahap pelatihan model dan klasifikasi merupakan inti dari penelitian ini. Pada tahap ini digunakan arsitektur Convolutional Neural Network DenseNet-169 sebagai ekstraktor fitur utama. DenseNet-169 dipilih karena memiliki mekanisme dense connection yang menghubungkan setiap layer dengan seluruh layer sebelumnya, sehingga informasi fitur dapat digunakan kembali

dan aliran gradien tetap terjaga pada jaringan yang dalam. Penelitian ini menggunakan pendekatan transfer learning dengan memanfaatkan bobot awal DenseNet-169 yang telah dilatih pada dataset ImageNet. Pendekatan ini memungkinkan model untuk memanfaatkan pengetahuan awal dari dataset berskala besar sehingga mempercepat proses konvergensi dan meningkatkan performa klasifikasi pada dataset yang relatif terbatas. Modifikasi dilakukan pada lapisan akhir DenseNet-169 dengan menambahkan dua lapisan Dense yang menggunakan inisialisasi bobot He Normal, fungsi aktivasi ReLU, serta regularisasi L1 untuk mengurangi kompleksitas model dan mencegah terjadinya overfitting. Lapisan output disesuaikan dengan jumlah kelas yang digunakan dalam penelitian ini, yaitu empat kelas, dengan menggunakan fungsi aktivasi Softmax untuk klasifikasi multikelas. Model yang telah dilatih akan diterapkan pada data pengujian yang tidak pernah dilihat sebelumnya. Hasil klasifikasinya akan dievaluasi untuk mengukur tingkat keberhasilan dalam mengenali dan mengklasifikasikan jenis penyakit jeruk tersebut.

a) DenseNet-169

Pada tahap pertama, citra berukuran 224x224 piksel yang telah diproses sebelumnya akan di inputkan pada lapisan konvolusi. Model DenseNet-169 sebelumnya akan dilatih dengan menggunakan dataset ImageNet dan memodifikasi lapisan terakhir. Fitur-fitur yang relevan dari gambar daun jeruk diekstraksi menggunakan model DenseNet-169. Fitur-fitur ini mencakup karakteristik visual seperti tekstur, warna, dan bentuk yang dapat mengindikasikan keberadaan penyakit pada tanaman jeruk. Lapisan

akhir dari DenseNet-169 terdapat lapisan fully-connected dengan dua layer dense yang dimodifikasi. Lapisan-lapisan ini menggunakan metode He Normal Initialization untuk inisialisasi bobot kernel dalam lapisan konvolusi, yang membantu dalam propagasi gradien yang lebih baik selama pelatihan, dan juga menggunakan fungsi aktivasi ReLU dan regularisasi L1 untuk meningkatkan kinerja dan mencegah overfitting. Sedangkan layer dense paling akhir digunakan sebagai lapisan deteksi dan klasifikasi. Selain itu, terdapat lapisan flatten yang digunakan dalam mengubah representasi data dari matriks multidimensi menjadi vektor satu dimensi. Setelah model selesai dibuat, langkah selanjutnya adalah melakukan klasifikasi penyakit jeruk berdasarkan fitur-fitur yang telah dibangun. Struktur lapisan akhir model yang dimodifikasi ditunjukkan pada Gambar 3.2.



Gambar 3.2. Lapisan Akhir DenseNet-169 dengan Modifikasi Model

b) Training, Validasi, dan Evaluasi

Pada tahap training, validasi, dan evaluasi terdapat enam skenario pengujian model, dimana skenario pertama, kedua, dan ketiga menguji model DenseNet-169 tanpa kombinasi fitur terhadap dataset FDI, LDI, dan gabungan kedua dataset yaitu FDI dan LDI. Sedangkan skenario ke-empat, ke-lima, dan ke-enam menguji model DenseNet-169 dengan kombinasi

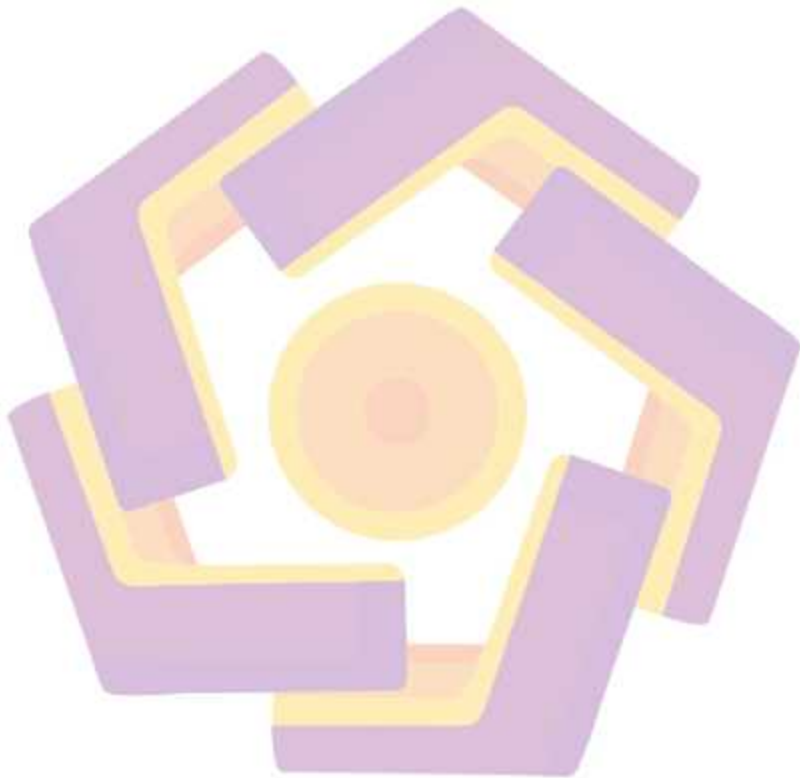
fitur terhadap dataset FDI, LDI, dan gabungan kedua dataset yaitu FDI dan LDI. Berdasarkan skenario diatas, terdapat dua model DenseNet-169 yang dibangun, yang mana masing-masing dari setiap model DenseNet-169 yang dibangun terdapat tiga skenario berdasarkan dataset. Setelah model berhasil di latih dan di validasi, ke enam skenario di evaluasi menggunakan data testing untuk dilakukan deteksi dan klasifikasi. Setelah proses deteksi dan klasifikasi, semua skenario yang menggunakan model DenseNet-169 dan DenseNet-169 dengan kombinasi fitur akan menghasilkan prediksi untuk setiap gambar penyakit jeruk. Prediksi ini akan dievaluasi menggunakan metrik evaluasi seperti akurasi, presisi, dan recall. Metrik-metrik ini sangat penting dalam mengevaluasi performa model dalam mengklasifikasikan gambar penyakit jeruk dengan benar. Akurasi memberikan informasi tentang seberapa akurat model dalam melakukan klasifikasi secara keseluruhan, sementara presisi menunjukkan sejauh mana prediksi positif yang dibuat oleh model benar. Recall menggambarkan sejauh mana model mampu mengenali kasus positif yang sebenarnya. Dengan menggunakan metrik evaluasi ini, dapat mengukur dan menilai kualitas dan kehandalan model klasifikasi DenseNet-169 baik dengan kombinasi fitur maupun tanpa kombinasi fitur dalam mengklasifikasikan gambar psenyakit jeruk secara akurat. Eksperimen dilakukan dalam enam skenario pengujian. Tiga skenario pertama menggunakan input citra RGB tanpa kombinasi fitur pada dataset LDI, FDI, dan MDI. Tiga skenario berikutnya menggunakan kombinasi fitur

hasil konversi ruang warna HSV, YCbCr, dan LAB pada ketiga jenis dataset tersebut. Setiap skenario pengujian dijalankan sebanyak lima kali, dan hasil yang dilaporkan merupakan nilai rata-rata dari seluruh pengujian untuk memperoleh performa yang lebih stabil dan objektif.

4) Analisis dan Kesimpulan

Setelah evaluasi model selesai, langkah berikutnya adalah melakukan analisis terhadap hasil evaluasi yang telah diperoleh. Analisis ini bertujuan untuk mendapatkan pemahaman yang lebih mendalam tentang kinerja model, mengidentifikasi kelebihan dan kekurangan model, serta menemukan aspek-aspek yang perlu diperbaiki atau dioptimalkan. Dalam analisis ini, kita dapat membandingkan hasil evaluasi antara model DenseNet-169 dengan dan tanpa kombinasi fitur, serta membandingkannya dengan tujuan awal penelitian. Kita juga dapat menganalisis kinerja model dalam mengklasifikasikan setiap jenis penyakit kulit, mengevaluasi metrik evaluasi untuk setiap kelas penyakit, dan mengidentifikasi kelas mana yang memiliki kinerja yang lebih baik atau lebih buruk. Evaluasi kinerja model dilakukan menggunakan confusion matrix untuk menghitung metrik akurasi, presisi, recall, dan F1-score. Metrik-metrik ini digunakan untuk menilai kemampuan model dalam mengklasifikasikan penyakit jeruk secara tepat serta untuk mengidentifikasi kesalahan klasifikasi yang terjadi pada setiap kelas. Berdasarkan analisis tersebut, kita dapat membuat kesimpulan mengenai keberhasilan atau kegagalan model dalam mengklasifikasikan jenis penyakit jeruk. Kesimpulan ini akan memberikan informasi penting dalam mengevaluasi apakah model yang telah

dikembangkan telah mencapai tujuan penelitian atau masih memerlukan perbaikan. Kesimpulan ini juga dapat digunakan sebagai dasar untuk mengambil keputusan atau memberikan rekomendasi terkait penerapan model dalam praktik klinis atau penelitian lanjutan.



BAB IV

HASIL PENELITIAN DAN PEMBAHASAN





4.1. Persiapan Data

Dalam mempersiapkan sebuah dataset, akan dijelaskan beberapa tahapan mengenai proses mempersiapkan dataset didalamnya, yang meliputi pengumpulan data, pelabelan data, dan augmentasi data. Berikut adalah langkah-langkah yang akan dilakukan sebagai berikut:




4.1.1. Pengumpulan Data

Untuk mengumpulkan dataset LDI (*Leaf Disease Image*) dan FDI (*Fruit Disease Image*), peneliti mendapatkan dataset dari situs website yang bersifat open data public. Dataset dengan kategori LDI merupakan sebuah dataset yang dikembangkan oleh (Syed-Ab-Rahman et al., 2022) yang memuat data citra daun jeruk sehat dan daun yang terinfeksi penyakit. Sedangkan dataset dengan kategori FDI dikembangkan oleh (Silva et al., 2021) memuat data citra yang terdiri dari buah jeruk sehat dan buah jeruk yang terinfeksi penyakit. Data dari kedua skema dataset tersebut bersumber dari kaggle. Sumber tersebut sudah menyediakan dataset yang dibutuhkan oleh peneliti dengan jumlah data citra sebanyak 596 untuk dataset LDI, dan sebanyak 1789 data citra untuk dataset FDI. Selain itu, terdapat 3 jenis penyakit jeruk dan 1 jenis untuk jeruk sehat, dengan rincian seperti Tabel 4.1 untuk dataset LDI dan Tabel 4.2 untuk dataset FDI sebagai berikut.

Tabel 4.1. Dataset LDI (*Leaf Disease Image*)

No	Kategori	Jumlah	Gambar
1	Blackspot	171	
2	Canker	163	
3	Greening / HLB (Huanglongbing)	204	
4	Healthy	58	

Tabel 4.2. Dataset FDI (*Fruit Disease Image*)

No	Kategori	Jumlah	Gambar
1	Blackspot	344	
2	Canker	349	
3	Greening / HLB (Huanglongbing)	545	
4	Healthy	551	

4.1.2. Labeling Data

Setelah dua skema dataset LDI dan FDI berhasil dikumpulkan, pada skema dataset LDI dan FDI masing-masing memiliki empat kelas yang sama. Dari empat kelas tersebut tiga diantaranya merupakan kelas dari penyakit jeruk, dan satu kelas untuk jeruk yang sehat. Kemudian kelas-kelas tersebut diberikan label sesuai nama penyakit jeruk dan satu kelas untuk jeruk sehat. Nama-nama penyakit tersebut

anantara lain *blackspot*, *canker*, dan *greening* atau disebut juga huanglongbing (HLB). Untuk informasi lebih lanjut mengenai data penelitian dapat ditemukan pada Tabel 4.1 untuk dataset LDI dan Tabel 4.2 untuk dataset FDI.

Pada tahap ini, setiap gambar dalam dataset telah disusun dengan cermat ke dalam folder yang diberi label sesuai dengan kategori jenis penyakit jeruk yang terdapat di dalamnya. Proses penempatan gambar ke dalam folder-folder terkait dilakukan setelah melalui proses penandaan atau labeling yang teliti.

4.1.3. Augmentasi Data

Augmentasi data merupakan suatu metode yang bertujuan untuk meningkatkan keberagaman dataset, mencegah terjadinya overfitting, serta memperkaya variasi dalam dataset guna meningkatkan kualitas model yang dibangun. Selain itu, proses augmentasi juga bertujuan untuk melakukan penyeimbangan data agar setiap kelas penyakit memiliki jumlah sampel yang seimbang, memastikan representasi yang adil dalam pembentukan model. Metode augmentasi yang diterapkan meliputi rotasi, pergeseran, flipping, resize gambar (perbesaran/pegecilan), distorsi gambar, dan penanganan piksel kosong. Proses augmentasi ini diimplementasikan menggunakan bahasa pemrograman Python melalui serangkaian perintah yang telah disusun dengan cermat. Berikut kode perintah dalam bahasa pemrograman python:

```

python.py
def distort_image(image):
    distortion_range = 10
    random_distortion = random.randint(-distortion_range, distortion_range)
    distorted_image = image + random_distortion
    distorted_image = np.clip(distorted_image, 0, 255).astype(np.uint8)
    return distorted_image
datagen = ImageDataGenerator(rotation_range=30,
                             width_shift_range=0.1,
                             height_shift_range=0.1,
                             zoom_range=0.1,
                             horizontal_flip=True,
                             preprocessing_function=distort_image)

```

Kode tersebut merupakan implementasi dari augmentasi data pada gambar menggunakan Python dengan menggunakan modul 'ImageDataGenerator' dari Keras. Fungsi 'distort_image' digunakan untuk menerapkan distorsi acak pada gambar yang dimasukkan sebagai input. Distorsi tersebut dihasilkan dengan mengubah nilai piksel gambar secara acak dalam rentang tertentu. Pada bagian berikutnya, dilakukan inisialisasi objek 'ImageDataGenerator' yang mengatur berbagai jenis augmentasi yang akan diterapkan, seperti rotasi gambar sebesar 30 derajat, pergeseran horizontal dan vertikal sebesar 10% dari lebar dan tinggi gambar, serta pembesaran atau pengecilan gambar hingga 10%. Selain itu, dilakukan juga flipping horizontal pada gambar. Fungsi 'distort_image' juga dimasukkan sebagai fungsi pra-pemrosesan (preprocessing function).

Selanjutnya, dilakukan iterasi pada setiap kelas (direktori) dalam dataset yang telah ditentukan sebelumnya. Setiap gambar dalam setiap kelas di-load, diubah menjadi array, dan di-expand dimensi array-nya untuk sesuai dengan format yang diperlukan oleh 'datagen.flow'. Pada proses augmentasi, dilakukan pengulangan untuk setiap gambar dengan menggunakan 'datagen.flow', di mana

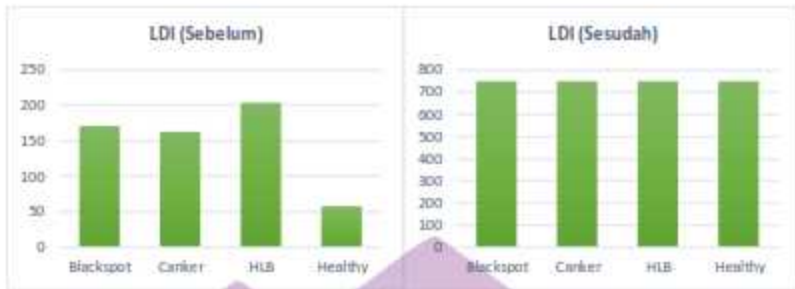
akan dihasilkan versi-versi baru dari gambar tersebut sesuai dengan augmentasi yang telah ditentukan sebelumnya. Gambar-gambar baru ini disimpan ke dalam folder output yang sesuai dengan kelas asalnya. Proses ini berhenti setelah menghasilkan beberapa gambar baru dari setiap gambar asli. Berikut contoh hasil augmentasi yang dilakukan dapat dilihat pada Gambar 4.1.



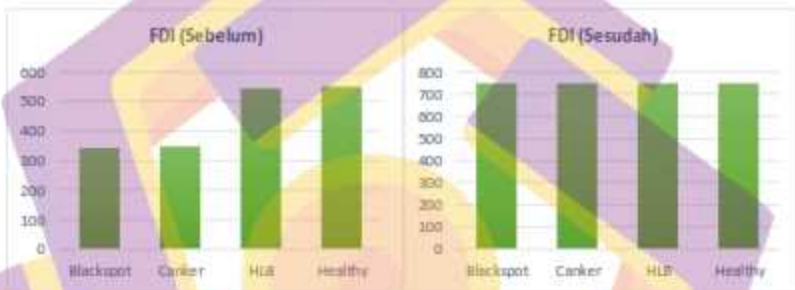
Gambar 4.1. (a) Citra asli, (b) Citra hasil augmentasi

Gambar diatas merupakan hasil contoh dari augmentasi data yang dilakukan pada salah satu data gambar, dimana (a) merupakan gambar sebelum di augmentasi, dan (b) merupakan data sesudah dilakukan augmentasi.

Balancing data dalam dataset dilakukan melalui proses augmentasi yang bertujuan untuk menyeimbangkan jumlah sampel pada setiap jenis penyakit jeruk. Dalam konteks ini, tujuannya adalah meningkatkan jumlah sampel menjadi 750 gambar per kelas, sehingga secara total akan ada 3000 gambar untuk kategori LDI dan 3000 gambar untuk kategori FDI. Untuk mencapai tujuan ini, teknik oversampling digunakan untuk menambah sampel pada kelas minoritas, sedangkan teknik undersampling digunakan untuk mengurangi jumlah sampel pada kelas mayoritas. Perubahan jumlah dataset sebelum dan setelah proses balancing dapat dilihat pada Gambar 4.2 dan 4.3.



Gambar 4.2. Perubahan jumlah dataset LDI sebelum dan sesudah balancing



Gambar 4.3. Perubahan jumlah dataset FDI sebelum dan sesudah balancing

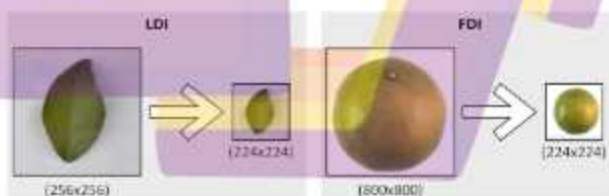
4.2. Preprocessing Data

Sebelum data gambar jeruk dimasukkan ke dalam model klasifikasi, rangkaian langkah preprocessing data merupakan aspek yang krusial dalam penelitian ini. Preprocessing data tak hanya penting untuk pengembangan model dengan kombinasi fitur yang tepat untuk arsitektur yang akan diadopsi, tetapi juga bertujuan untuk menyiapkan data secara optimal guna proses pelatihan model serta meningkatkan kualitas citra demi hasil yang lebih baik dari model yang digunakan. Signifikansinya terletak pada kebersihan dan relevansi data, menjadi fondasi penting dalam memastikan model yang tercipta mampu memberikan hasil

klasifikasi yang akurat dan dapat diandalkan. Tahapan-tahapan yang dilakukan dalam proses ini merupakan landasan vital yang menjamin data yang siap dan berkualitas sebelum diaplikasikan dalam model klasifikasi. Berikut adalah rangkaian tahapan yang dilakukan dalam proses preprocessing data:

4.2.1. Mengubah Ukuran Citra

Dataset yang terkumpul menunjukkan adanya variasi ukuran gambar yang diunduh, dimulai dari 256x256 piksel hingga 800x800 piksel. Untuk memastikan konsistensi dalam penerapan arsitektur DenseNet-169, semua gambar dalam dataset disesuaikan menjadi ukuran yang seragam, yaitu 224x224 piksel, sebagaimana yang ditunjukkan pada Gambar 4.4. Tindakan ini diambil dengan tujuan memenuhi persyaratan ukuran yang cocok dengan arsitektur model yang digunakan serta untuk meningkatkan efisiensi komputasi. Dengan menyamakan ukuran gambar, setiap gambar dalam dataset dipastikan memiliki dimensi yang seragam, hal ini bertujuan untuk mengoptimalkan kesiapan dalam tahap pelatihan model menggunakan arsitektur DenseNet-169.

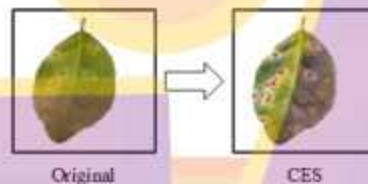


Gambar 4.4. Mengubah ukuran citra ke 224x224 piksel

4.2.2. Peningkatan Citra

Teknik peningkatan citra dalam pemrosesan gambar dapat membantu memperbaiki kejelasan, detail, dan karakteristik penting pada gambar yang

diperlukan oleh model CNN. Teknik peningkatan citra membantu model CNN untuk memahami informasi yang tersembunyi dalam gambar dengan lebih baik, memungkinkan model untuk membuat prediksi yang lebih akurat dan hasil yang lebih handal dalam pemrosesan data gambar. Dengan demikian, teknik Color Enhancement by Scaling (CES) digunakan untuk meningkatkan kualitas visual dari gambar dengan meningkatkan komponen chromatic dan luminance secara proporsional. Dengan memanfaatkan perubahan skala pada komponen warna dan kecerahan gambar, CES meningkatkan kontras dan kejelasan gambar, menjadikannya lebih tajam dan terperinci. Pendekatan ini melibatkan eliminasi noise pada gambar menggunakan filter median dan filter Gaussian. Selain itu, teknik CES enhancement juga mencakup penyesuaian kontras dan kecerahan gambar guna mempertajam detailnya. Contoh hasil proses peningkatan citra dengan teknik ini dapat diamati pada Gambar 4.5.



Gambar 4.5. Contoh hasil proses peningkatan citra dengan teknik CES

Selain itu, di bawah ini merupakan implementasi kode program dari teknik Color Enhancement by Scaling (CES) menggunakan algoritma CLAHE (Contrast Limited Adaptive Histogram Equalization). Berikut script atau kode program dari teknik CES.



```
clahe = cv2.createCLAHE(clipLimit=3.0, tileGridSize=(8, 8))
r, g, b = cv2.split(image)
r = clahe.apply(r)
g = clahe.apply(g)
b = clahe.apply(b)

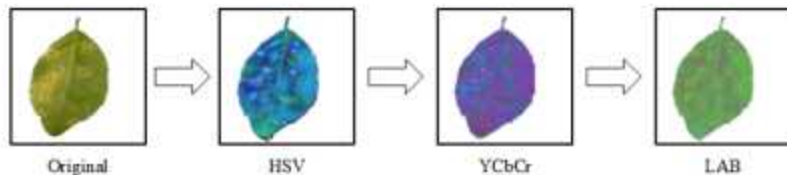
image = cv2.merge([r, g, b])
```

Melalui kode program ini, gambar dipisahkan menjadi tiga saluran warna, yaitu merah (red), hijau (green), dan biru (blue), mengizinkan pemrosesan yang lebih terperinci pada setiap komponen warna. Proses CLAHE diterapkan pada setiap saluran warna secara terpisah, di mana nilai kontras dari setiap blok pada histogram dibatasi agar tidak melampaui ambang batas tertentu ($\text{clipLimit}=3.0$). Langkah ini membantu menghindari peningkatan kontras yang berlebihan pada gambar, memastikan hasil yang lebih seimbang. Selanjutnya, teknik CLAHE diterapkan pada setiap saluran warna guna meningkatkan kejelasan gambar dan menyoroti detail yang signifikan. Proses ini juga melibatkan penggunaan filter median dan filter Gaussian untuk mengurangi noise pada gambar, yang kemudian diikuti dengan penyesuaian kontras dan kecerahan untuk mempertajam detail gambar. Gambar-gambar yang telah ditingkatkan kualitasnya melalui teknik ini digabungkan kembali menjadi gambar RGB yang lebih tajam dan terperinci. Implementasi ini memperjelas detail dan karakteristik penting pada gambar, yang akan memberikan kontribusi signifikan pada pemahaman dan klasifikasi citra oleh model Convolutional Neural Network (CNN).

4.2.3. Konversi Citra

Konversi citra berfungsi mengubah model warna dari RGB ke model warna lain (HSV, YCbCr, dan LAB) untuk mempermudah ekstraksi fitur. Model warna HSV digunakan untuk mengekstrak fitur warna pada citra daun jeruk, sedangkan model warna YCbCr untuk mengekstrak fitur tekstur. Selain itu, model warna LAB digunakan untuk mengekstrak fitur tekstur dan warna pada citra daun jeruk. Melalui konversi citra, penanganan noise pada citra dilakukan dengan filter median dan Gaussian, memperbaiki deteksi serta klasifikasi citra. Teknik ini akan diuji coba pada tiga jenis dataset: LDI, FDI, dan gabungan LDI dan FDI.

Konversi ke model warna HSV membantu dalam mengatasi masalah kecerahan dan kontras dalam gambar, serta memperbolehkan pemrosesan yang lebih efektif terhadap warna. Penggunaan model warna YCbCr memiliki keunggulan dalam memisahkan informasi kecerahan dari komponen warna, memberikan kontrol yang lebih baik terhadap kejelasan dan warna dalam gambar. Sedangkan konversi ke model warna LAB memisahkan komponen warna dari kecerahan, memungkinkan manipulasi gambar yang lebih tepat untuk peningkatan kualitasnya, membantu menghasilkan detail yang lebih tajam dan akurat pada tahap deteksi dan klasifikasi gambar. Dengan berbagai teknik ini, penggunaan peningkatan citra dengan beragam konversi ke model warna memberikan kemampuan untuk meningkatkan kualitas gambar serta memfasilitasi proses analisis yang lebih efisien dan akurat dalam pengolahan gambar. Gambar 4.6 dibawah ini merupakan hasil konversi citra gambar original, hsv, YCbCr, dan LAB.



Gambar 4.6. Konversi Citra Asli ke HSV, YCbCr, dan LAB

Untuk mengkonversi warna citra seperti gambar diatas, maka dibuat sebuah script atau kode program sebagai berikut.

```
DenseNet-168.py
# ambil komponen V (value) dari ruang warna HSV
image_hsv = cv2.cvtColor(image, cv2.COLOR_RGB2HSV)

# ubah citra warna ke skema warna YCbCr
image = cv2.cvtColor(image_hsv, cv2.COLOR_RGB2YCbCr)

# ubah citra warna ke skema warna LAB
image = cv2.cvtColor(image, cv2.COLOR_RGB2LAB)
```

Script diatas menampilkan transformasi citra dari model warna RGB ke model warna yang berbeda. Pertama, dilakukan konversi citra dari RGB ke ruang warna HSV menggunakan fungsi `cv2.cvtColor()`, yang memungkinkan pemisahan komponen hue, saturation, dan value pada gambar. Model warna ini membantu dalam mengekstrak fitur warna pada citra daun jeruk, memperbaiki kontras dan kecerahan, serta memungkinkan manipulasi warna yang lebih efektif. Selanjutnya, gambar dikonversi ke model warna YCbCr untuk mengisolasi informasi kecerahan dan komponen warna, memberikan kontrol yang lebih baik terhadap kejelasan dan warna dalam gambar. Terakhir, dilakukan konversi ke model warna LAB, yang memisahkan komponen warna dari kecerahan, memungkinkan peningkatan

kualitas gambar dengan manipulasi yang lebih akurat, dan menghasilkan detail yang lebih tajam. Proses ini membantu dalam pemrosesan citra untuk mendukung deteksi dan klasifikasi gambar yang lebih efisien. Dengan berbagai teknik konversi citra yang diterapkan, terbukti bahwa peningkatan kualitas gambar melalui variasi model warna dapat meningkatkan kemampuan analisis gambar serta membantu dalam mendapatkan informasi yang lebih akurat dalam pengolahan citra. Gambar di bawah ini merupakan hasil dari proses konversi citra menggunakan model warna HSV, YCbCr, dan LAB dari citra asli.

4.2.4. Normalisasi Dataset

Normalisasi data berguna untuk mengubah rentang nilai data menjadi antara 0 hingga 1. Proses normalisasi dilakukan setelah serangkaian transformasi citra seperti konversi warna dan peningkatan citra menggunakan berbagai teknik seperti CES enhancement. Proses ini membantu dalam mempersiapkan data untuk proses deteksi dan klasifikasi dengan memastikan bahwa semua nilai dalam dataset memiliki skala yang seragam.

Normalisasi data dilakukan setelah citra diubah ke skema warna LAB dan dikonversi menjadi nilai float32 dalam rentang 0 hingga 1. Proses ini dilakukan dengan mengurangi nilai minimum dari setiap data dan membaginya dengan selisih nilai maksimum dan minimum dari setiap data. Normalisasi data ini penting karena dapat membantu dalam mengurangi perbedaan rentang nilai yang ada di dalam dataset, sehingga mempermudah proses pelatihan model dan analisis lebih lanjut. Dengan menggunakan normalisasi data, meminimalkan kemungkinan bias yang mungkin muncul akibat perbedaan rentang nilai pada setiap fitur dalam dataset,

yang pada gilirannya dapat meningkatkan kinerja model dalam proses klasifikasi dan deteksi.

4.2.5. Membagi Dataset

Dalam penelitian ini ada dua kategori dataset yaitu FDI dan LDI, yang mana ada 3000 citra FDI dan 3000 citra LDI. Masing-masing kategori dataset dibagi menjadi tiga bagian dengan proporsi yang telah ditentukan, yaitu 80% untuk data pelatihan (2400 citra), 10% untuk data validasi (300 citra), dan 10% untuk data pengujian (300 citra). Data pelatihan digunakan sebagai dataset utama untuk melatih model klasifikasi, yang membantu model untuk belajar pola dan fitur yang ada dalam dataset. Data validasi diperlukan untuk mengevaluasi model selama proses pelatihan dengan tujuan memonitor performa dan mencegah overfitting. Sedangkan data pengujian digunakan untuk menguji kinerja model secara independen setelah proses pelatihan selesai dan validasi telah dilakukan.

4.3. Implementasi dan Hasil Model

Dalam implementasi model Convolutional Neural Network (CNN) dengan menggunakan arsitektur DenseNet-169, berikut merupakan langkah-langkah yang diambil dalam pengembangan model arsitektur DenseNet-169 untuk melakukan klasifikasi pada dataset Leaf Disease Image (LDI), Fruit Disease Image (FDI), dan gabungan kedua dataset tersebut.

4.3.1. Rincian Implementasi Model

Dalam implementasi model CNN dengan arsitektur DenseNet-169 ini, terdapat tiga skenario yang sama untuk dataset yang berbeda yaitu dataset Leaf Disease Image (LDI), Fruit Disease Image (FDI), dan gabungan kedua dataset

tersebut yang diberi nama sebagai Mixed Disease Image (MDI). Berikut langkah-langkah yang diambil untuk implementasi model CNN.

a) Membangun Model

Pada pengembangan CNN, arsitektur DenseNet-169 digunakan sebagai basisnya. Arsitektur DenseNet-169 memiliki parameter yang dapat dilatih jauh lebih sedikit dibandingkan dengan arsitektur lain. Karena model yang dibangun menggunakan transfer learning dari dataset ImageNet, seperti menurut (Huang et al., 2017) DenseNet-169 adalah salah satu arsitektur DenseNet yang memiliki kinerja terbaik pada dataset ImageNet. Penggunaan DenseNet-169 dari pustaka Keras memungkinkan pemakaian model yang telah di-pretrain sebelumnya, memperoleh pemahaman mendalam terhadap fitur-fitur tingkat tinggi dari berbagai dataset gambar umum. Dengan pemanfaatan representasi fitur yang kuat dan abstrak yang telah tersedia, penelitian ini memusatkan perhatian pada deteksi dan klasifikasi spesifik terhadap penyakit pada tanaman jeruk. Implementasi kode di bawah ini memberikan langkah-langkah konkret penggunaan DenseNet-169.



```

import os
import cv2
import random
import numpy as np
import pandas as pd
import seaborn as sns
import tensorflow as tf
import matplotlib.pyplot as plt
from keras import layers
from tensorflow.keras import regularizers
from sklearn.metrics import confusion_matrix
from sklearn.metrics import KerasClassifier
from sklearn.preprocessing import LabelEncoder
from sklearn.model_selection import train_test_split
from tensorflow.keras.applications import DenseNet169
from tensorflow.keras.callbacks import ModelCheckpoint
from sklearn.metrics import classification_report, confusion_matrix, accuracy_score
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score

```

Pada kode program, modul dan pustaka yang dibutuhkan diimport untuk membangun dan melatih model Convolutional Neural Network (CNN) menggunakan arsitektur DenseNet-169 dengan TensorFlow dan Keras. Modul-modul seperti `os`, `cv2`, `random`, `numpy`, `pandas`, `seaborn`, `matplotlib.pyplot`, serta fungsi-fungsi dari `tensorflow`, `keras`, dan `sklearn` diimport untuk pengolahan data, visualisasi, pembangunan model, dan evaluasi klasifikasi. Beberapa modul seperti `os` dan `cv2` digunakan untuk pengelolaan file dan manipulasi citra. Kemudian, terdapat penggunaan modul `numpy` dan `pandas` untuk manipulasi data dan perhitungan numerik. `Seaborn` dan `matplotlib.pyplot` digunakan untuk visualisasi data, termasuk pembuatan grafik dan plot. Selanjutnya, modul-modul dan fungsi-fungsi dari TensorFlow dan Keras, seperti `tensorflow.keras`, `layers`, `callbacks`, dan `applications`, diimport untuk pembangunan, pelatihan, dan evaluasi model CNN menggunakan arsitektur DenseNet-169. Terdapat juga fungsi-fungsi dari `sklearn` yang digunakan untuk pemrosesan data, evaluasi klasifikasi seperti

confusion_matrix, dan pengukuran metrik evaluasi seperti accuracy_score, precision_score, recall_score, dan f1_score. Beberapa fungsi dan kelas dari sklearn.cluster, serta scikeras.wrappers juga diimpor untuk pembungkusan model Keras agar dapat digunakan dengan fungsi-fungsi dari scikit-learn. Berikut ini merupakan script dari base model beserta konfigurasi layer-layer arsitektur DenseNet-169.



```

model = tf.keras.models.Sequential()
model.add(tf.keras.layers.Rescaling(1./255, input_shape=(IMG_HEIGHT, IMG_WIDTH, 3)))
base_model = DenseNet169(include_top=False, weights='imagenet', input_shape=(IMG_HEIGHT,
                                                                    IMG_WIDTH, 3))
model.add(base_model)

model.add(tf.keras.layers.GlobalAveragePooling2D())
model.add(tf.keras.layers.Flatten())
model.add(tf.keras.layers.Dense(100, activation='relu', kernel_initializer='he_normal',
                                kernel_regularizer=tf.keras.regularizers.L1(0.01)))
model.add(tf.keras.layers.Dense(64, activation='relu', kernel_regularizer=tf.keras.regularizers.L1(0.01)))
model.add(tf.keras.layers.Dense(4, activation='softmax'))

```

Base model DenseNet-169 digunakan sebagai bagian krusial dari arsitektur CNN. Pertama, Sequential model dari TensorFlow Keras digunakan untuk menyusun lapisan-lapisan secara berurutan. Lapisan Rescaling ditambahkan untuk normalisasi nilai piksel gambar ke rentang antara 0 dan 1. DenseNet-169 diimplementasikan sebagai base model dengan konfigurasi tertentu, seperti include_top=False untuk menghilangkan lapisan fully connected di bagian akhir, weights='imagenet' untuk memuat bobot pra-terlatih dari dataset ImageNet, dan input_shape=(IMG_HEIGHT, IMG_WIDTH, 3) untuk menentukan dimensi input dari model (tinggi 224 piksel, lebar 224 piksel, dan saluran warna gambar). Setelah itu, lapisan-lapisan tambahan ditambahkan setelah base model DenseNet169,

seperti `GlobalAveragePooling2D` yang digunakan untuk mengubah keluaran spasial menjadi satu vektor rata-rata, dan `Flatten` untuk mengubah representasi data menjadi vektor satu dimensi. Dilanjutkan dengan penambahan lapisan Dense (fully connected layer) dengan 128 dan 64 neuron masing-masing, yang menggunakan fungsi aktivasi `ReLU` untuk mempelajari representasi yang lebih kompleks dari data. Penggunaan kernel initializer `"he_normal"` membantu dalam inialisasi bobot secara efisien. Terakhir, lapisan output Dense dengan 4 neuron dan aktivasi softmax digunakan untuk melakukan klasifikasi multi-kelas dengan menghasilkan probabilitas untuk masing-masing kelas yang ditargetkan. Konfigurasi ini bertujuan untuk mengekstrak fitur-fitur relevan dari gambar dan menghasilkan prediksi kelas yang akurat dalam konteks klasifikasi untuk 4 kelas yang ditentukan.

b) Pelatihan dan Validasi Model

Proses pelatihan model menggunakan data yang telah di-preprocessing sebelumnya. Model yang dikompilasi dengan fungsi loss `"sparse_categorical_crossentropy"` umumnya digunakan untuk klasifikasi multi-kelas saat label diberikan dalam bentuk bilangan bulat (dalam kasus ini, untuk nilai label dari 0 hingga 3 yang merepresentasikan 4 kelas yang berbeda). Pelatihan ini juga menggunakan optimizer dari `SGD` (Stochastic Gradient Descent). Berikut merupakan script pelatihan model `DenseNet-169` beserta detail tentang langkah-langkahnya.



```

optimizer = tf.keras.optimizers.SGD(learning_rate=0.001)
model.compile(loss="sparse_categorical_crossentropy", optimizer=optimizer,
              metrics=["accuracy"])

# Melatih model dengan menggunakan data pelatihan dan validasi
history = model.fit(X_train, y_train, batch_size=64, epochs=100,
                   validation_data=(X_val, y_val), callbacks=[checkpoint])

```

Optimizer SGD (Stochastic Gradient Descent) digunakan untuk mengoptimalkan proses pembelajaran jaringan. Ketika model tersebut dilatih dengan data, optimizer akan menghitung gradien dari fungsi kerugian terhadap bobot dan bias dalam model DenseNet-169, dan kemudian menggunakan gradien ini untuk menyesuaikan bobot-bobot tersebut dalam arah yang mengurangi nilai fungsi kerugian. Hal ini memungkinkan model untuk belajar dan menyesuaikan diri terhadap data pelatihan sehingga dapat membuat prediksi yang lebih akurat saat dihadapkan pada data yang belum pernah dilihat sebelumnya. Optimizer yang telah diinisialisasi sebelumnya (SGD dengan learning rate 0.002) diberikan ke parameter optimizer dalam kompilasi model. Ini berarti bahwa model akan menggunakan optimizer SGD yang telah diatur dengan learning rate 0.002 untuk mengoptimalkan bobot dan bias selama proses pelatihan.

Pada bagian terakhir, model dilatih menggunakan metode fit. Data pelatihan (X_{train} dan y_{train}) digunakan selama pelatihan dengan ukuran batch sebesar 64 dan selama 100 epoch. Selain itu, data validasi (X_{val} dan y_{val}) digunakan untuk mengevaluasi performa model selama proses pelatihan. Hal ini memungkinkan penggunaan model optimal pada titik tertentu dalam proses pelatihan.

c) Evaluasi Model

Evaluasi kinerja model dilakukan untuk mempertimbangkan 2 jenis model yang berbeda, yaitu model DenseNet-169 yang dipadukan dengan kombinasi fitur dan DenseNet-169 yang sudah di konfigurasi. Setiap model dievaluasi dengan menggunakan metrik kinerja standar seperti akurasi, presisi, recall, dan F1 Score. Selain itu, evaluasi kinerja model juga dilakukan pada masing-masing kelas. Proses implementasi dari kinerja model tersebut dapat dilihat pada script atau kode dibawah ini.



```

# Evaluasi model pada data pengujian
y_test_pred = model.predict(X_test)
y_test_pred_classes = np.argmax(y_test_pred, axis=1)

accuracy = accuracy_score(y_test_encoded, y_test_pred_classes)
precision = precision_score(y_test_encoded, y_test_pred_classes,
                           average='macro')
recall = recall_score(y_test_encoded, y_test_pred_classes,
                     average='macro')
f1 = f1_score(y_test_encoded, y_test_pred_classes, average='macro')

# Evaluasi kelas model pada data pengujian
y_test_pred = model.predict(X_test)
y_test_pred_classes = np.argmax(y_test_pred, axis=1)
accuracy = accuracy_score(y_test_encoded, y_test_pred_classes)

for category, acc in class_accuracy.items():
    print("{}: {:.2%}".format(category, acc))

```

Untuk mengevaluasi performa model pada data pengujian setelah proses pelatihan selesai. Pertama, model melakukan prediksi terhadap data pengujian (X_{test}) menggunakan metode predict. Selanjutnya, untuk memperoleh prediksi kelas, digunakan $\text{np.argmax}(y_{test_pred}, \text{axis}=1)$ yang mengambil indeks kelas

dengan nilai probabilitas tertinggi dari setiap prediksi. Kemudian, beberapa metrik evaluasi, seperti akurasi, presisi, recall, dan nilai F1 dihitung menggunakan fungsi-fungsi evaluasi dari library scikit-learn, yaitu `accuracy_score`, `precision_score`, `recall_score`, dan `f1_score`. Nilai akurasi menggambarkan sejauh mana model dapat mengklasifikasikan data pengujian dengan benar. Presisi merupakan rasio prediksi benar positif terhadap total hasil yang diprediksi positif, sementara recall mengukur sejauh mana model dapat mengidentifikasi kelas positif dengan benar. Nilai F1 merupakan rata-rata harmonik dari presisi dan recall yang memberikan gambaran keseluruhan tentang performa model pada data pengujian. Selain itu, terdapat loop yang menampilkan akurasi untuk setiap kategori atau kelas yang dinilai oleh model. Hasil akurasi tersebut dapat memberikan gambaran tentang seberapa baik model dapat mengklasifikasikan masing-masing kategori.

4.3.2. Hasil Evaluasi Kinerja Model

Dalam tahap ini, hasil evaluasi kinerja model didasarkan pada dua skenario model yang berbeda yang melibatkan arsitektur DenseNet-169. Pertama, evaluasi dilakukan pada model DenseNet-169 yang merupakan model CNN dengan arsitektur yang telah ditraining sebelumnya. Kedua, evaluasi juga dilakukan pada model yang menggabungkan fitur-fitur dari DenseNet-169 dengan fitur-fitur lainnya yang telah diproses atau diekstraksi secara khusus. Melalui dua pendekatan ini, peneliti mengevaluasi kinerja model dalam memprediksi klasifikasi yang tepat terhadap citra-citra penyakit pada tanaman jeruk. Evaluasi ini akan memberikan wawasan tentang seberapa baik masing-masing model memahami dan menggeneralisasi informasi terkait citra penyakit tanaman, yang menjadi fokus

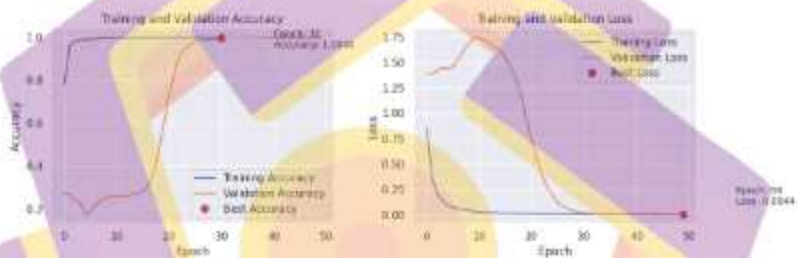
utama dalam upaya klasifikasi dan deteksi penyakit. Evaluasi yang mendalam pada kedua pendekatan ini akan memberikan gambaran komprehensif tentang efektivitas serta kelebihan dan kekurangan dari masing-masing model pada dataset citra penyakit tanaman jeruk yang spesifik. Berikut merupakan hasil dari kedua skenario model yang telah dibuat.

a) Model DenseNet-169

Evaluasi kinerja model DenseNet-169 merupakan tahap yang penting dalam penilaian kemampuan arsitektur CNN yang telah menjalani proses training sebelumnya. Dalam tahap ini, fokus diberikan pada penilaian eksklusif terhadap model yang telah belajar fitur-fitur penting dari dataset gambar umum, dengan harapan mampu menggeneralisasi fitur-fitur tersebut pada dataset spesifik citra penyakit tanaman jeruk. Evaluasi ini akan memberikan gambaran tentang seberapa baik model DenseNet-169 dapat memprediksi dan mengklasifikasikan dengan akurat berbagai jenis penyakit yang terjadi pada tanaman jeruk, serta sejauh mana model ini mampu memahami kompleksitas citra-citra penyakit yang dihadapinya. Terdapat 3 kategori dataset yang di evaluasi, yakni dataset Leaf Disease Image (LDI), Fruit Disease Image (FDI), dan gabungan kedua dataset tersebut yang diberi nama Mixed Disease Image (MDI). Untuk hasil pelatihan yang lebih jelas dapat dilihat pada gambar-gambar di bawah yang memvisualisasikan perubahan tingkat akurasi dan loss selama proses pelatihan dan validasi dari model pada dataset (a) FDI, (b) LDI, dan (c) MDI, yang dijalankan sebanyak 50 epoch.

1. Pelatihan dan Validasi FDI, LDI dan MDI (Run 1)

Pada Gambar 4.7 ditampilkan kurva akurasi dan loss pada proses pelatihan (training) dan validasi (validation) model DenseNet-169 menggunakan dataset FDI (Fruit Disease Image) yang dijalankan selama 50 epoch. Visualisasi ini digunakan untuk menganalisis dinamika pembelajaran model, tingkat konvergensi, serta potensi terjadinya overfitting atau underfitting.



Gambar 4.7. Grafik Pelatihan dan Validasi FDI Run 1 (Tanpa Kombinasi Fitur)

Berdasarkan grafik training accuracy, terlihat bahwa akurasi pelatihan meningkat sangat cepat sejak epoch awal dan mencapai nilai mendekati 1,0 sebelum epoch ke-10. Hal ini menunjukkan bahwa model mampu mempelajari pola visual pada data latih secara efisien. Arsitektur DenseNet-169 dengan mekanisme dense connection memungkinkan fitur-fitur penting dari lapisan awal tetap dimanfaatkan pada lapisan berikutnya, sehingga proses pembelajaran berlangsung lebih stabil dan efektif. Pada sisi validation accuracy, terlihat bahwa akurasi validasi pada fase awal masih relatif rendah dan cenderung stagnan hingga sekitar epoch ke-20. Kondisi ini menunjukkan bahwa pada tahap awal pelatihan, model belum sepenuhnya mampu melakukan

generalisasi terhadap data yang tidak dilihat sebelumnya. Namun, setelah melewati epoch ke-20, akurasi validasi meningkat secara signifikan dan mencapai nilai maksimum sebesar 1,0000 pada epoch ke-31. Peningkatan ini menunjukkan bahwa model berhasil menemukan representasi fitur yang sangat general dan relevan terhadap data validasi. Pada grafik training loss, terlihat bahwa nilai loss pelatihan menurun secara tajam sejak epoch awal dan mendekati nol pada epoch-epoch selanjutnya. Penurunan loss yang konsisten ini menandakan bahwa proses optimasi berjalan dengan baik dan kesalahan prediksi pada data latih semakin kecil seiring bertambahnya epoch. Sementara itu, grafik validation loss menunjukkan bahwa nilai loss validasi sempat meningkat pada fase awal hingga pertengahan pelatihan, yang mengindikasikan adanya kesulitan generalisasi pada tahap awal. Namun, setelah melewati epoch ke-20, nilai validation loss menurun secara signifikan hingga mencapai nilai terendah sebesar 0,0044 pada epoch ke-50. Penurunan ini sejalan dengan peningkatan akurasi validasi dan menunjukkan bahwa model berhasil mencapai kondisi konvergensi yang sangat baik. Berdasarkan pola kurva akurasi dan loss tersebut, dapat disimpulkan bahwa meskipun pada tahap awal pelatihan model menunjukkan indikasi ketidakseimbangan antara kinerja pelatihan dan validasi, model mampu mengatasi kondisi tersebut pada fase akhir pelatihan. Dengan demikian, model DenseNet-169 menunjukkan kemampuan generalisasi yang sangat baik pada dataset FDI, dengan performa optimal dicapai pada rentang epoch ke-31 hingga akhir pelatihan. Berdasarkan grafik training accuracy, terlihat bahwa akurasi pelatihan meningkat sangat

cepat sejak epoch awal dan mencapai nilai mendekati 1,0 sebelum epoch ke-10. Hal ini menunjukkan bahwa model mampu mempelajari pola visual pada data latih secara efisien. Arsitektur DenseNet-169 dengan mekanisme dense connection memungkinkan fitur-fitur penting dari lapisan awal tetap dimanfaatkan pada lapisan berikutnya, sehingga proses pembelajaran berlangsung lebih stabil dan efektif. Pada sisi validation accuracy, terlihat bahwa akurasi validasi pada fase awal masih relatif rendah dan cenderung stagnan hingga sekitar epoch ke-20. Kondisi ini menunjukkan bahwa pada tahap awal pelatihan, model belum sepenuhnya mampu melakukan generalisasi terhadap data yang tidak dilihat sebelumnya. Namun, setelah melewati epoch ke-20, akurasi validasi meningkat secara signifikan dan mencapai nilai maksimum sebesar 1,0000 pada epoch ke-31. Peningkatan ini menunjukkan bahwa model berhasil menemukan representasi fitur yang sangat general dan relevan terhadap data validasi. Pada grafik training loss, terlihat bahwa nilai loss pelatihan menurun secara tajam sejak epoch awal dan mendekati nol pada epoch-epoch selanjutnya. Penurunan loss yang konsisten ini menandakan bahwa proses optimasi berjalan dengan baik dan kesalahan prediksi pada data latih semakin kecil seiring bertambahnya epoch. Sementara itu, grafik validation loss menunjukkan bahwa nilai loss validasi sempat meningkat pada fase awal hingga pertengahan pelatihan, yang mengindikasikan adanya kesulitan generalisasi pada tahap awal. Namun, setelah melewati epoch ke-20, nilai validation loss menurun secara signifikan hingga mencapai nilai terendah sebesar 0,0044 pada epoch ke-50. Penurunan ini sejalan dengan peningkatan

akurasi validasi dan menunjukkan bahwa model berhasil mencapai kondisi konvergensi yang sangat baik. Berdasarkan pola kurva akurasi dan loss tersebut, dapat disimpulkan bahwa meskipun pada tahap awal pelatihan model menunjukkan indikasi ketidakseimbangan antara kinerja pelatihan dan validasi, model mampu mengatasi kondisi tersebut pada fase akhir pelatihan. Dengan demikian, model DenseNet-169 menunjukkan kemampuan generalisasi yang sangat baik pada dataset FDI, dengan performa optimal dicapai pada rentang epoch ke-31 hingga akhir pelatihan.

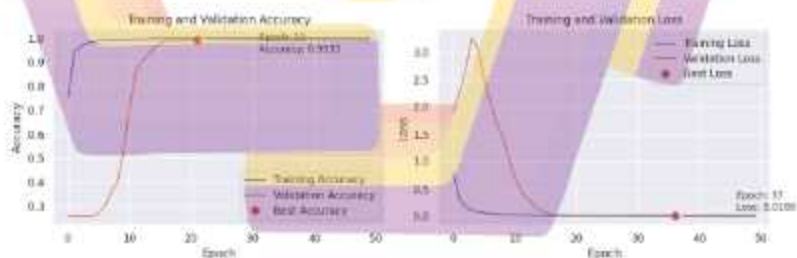


Gambar 4.8. Grafik Pelatihan dan Validasi LDI Run 1 (Tanpa Kombinasi Fitur)

Gambar 4.8 menampilkan visualisasi hasil pelatihan dan validasi model DenseNet-169 pada dataset LDI (Leaf Disease Image) tanpa kombinasi fitur pada run pertama, yang dijalankan selama 50 epoch. Berdasarkan grafik training accuracy, terlihat bahwa akurasi pelatihan meningkat sangat cepat sejak epoch awal dan mencapai nilai di atas 0,95 sebelum epoch ke-10. Pola ini serupa dengan hasil pada dataset FDI (Gambar 4.7), yang menunjukkan bahwa DenseNet-169 mampu dengan mudah mempelajari pola visual pada data latih. Hal ini mengindikasikan bahwa fitur-fitur dasar seperti tekstur dan

pola warna pada citra daun jeruk relatif mudah dikenali oleh model. Namun demikian, pada grafik validation accuracy terlihat bahwa akurasi validasi pada fase awal pelatihan masih relatif rendah dan mengalami fluktuasi yang cukup signifikan hingga sekitar epoch ke-25. Fluktuasi ini lebih jelas dibandingkan dengan hasil pada dataset FDI, yang mengindikasikan bahwa variasi visual pada citra daun jeruk cenderung lebih kompleks dan memiliki kemiripan antar kelas penyakit. Kondisi ini menyebabkan model memerlukan waktu pelatihan yang lebih lama untuk menemukan representasi fitur yang mampu digeneralisasi dengan baik. Setelah melewati epoch ke-30, akurasi validasi mulai meningkat secara lebih stabil hingga mencapai nilai terbaik sebesar 0,9833 pada epoch ke-41. Nilai ini menunjukkan bahwa model akhirnya mampu melakukan generalisasi dengan baik terhadap data validasi, meskipun memerlukan jumlah epoch yang lebih banyak dibandingkan beberapa skenario pada dataset FDI. Pada grafik training loss, terlihat bahwa nilai loss pelatihan menurun secara tajam sejak epoch awal dan mendekati nol pada epoch-epoch selanjutnya. Pola ini menunjukkan bahwa proses optimasi berjalan dengan baik dan kesalahan prediksi pada data latih dapat diminimalkan secara efektif, konsisten dengan hasil yang ditunjukkan pada Gambar 4.7. Sementara itu, grafik validation loss menunjukkan nilai loss yang relatif tinggi dan sempat meningkat pada fase awal hingga pertengahan pelatihan. Puncak loss validasi yang cukup besar mengindikasikan bahwa pada tahap awal, model mengalami kesulitan dalam melakukan generalisasi dan cenderung overfitting terhadap data latih. Namun, setelah melewati epoch ke-25, nilai loss validasi mulai

menurun secara signifikan hingga mencapai nilai terendah sebesar 0,0706 pada epoch ke-44. Jika dibandingkan dengan dataset FDI, nilai loss validasi minimum pada dataset LDI relatif lebih tinggi. Hal ini menunjukkan bahwa klasifikasi citra daun jeruk memiliki tingkat kesulitan yang lebih besar dibandingkan citra buah jeruk, terutama karena gejala penyakit pada daun sering kali memiliki pola visual yang saling tumpang tindih, seperti kemiripan warna dan tekstur antar kelas penyakit. Secara keseluruhan, hasil pada Gambar 4.8 menunjukkan bahwa model DenseNet-169 tanpa kombinasi fitur tetap mampu mencapai performa klasifikasi yang tinggi pada dataset LDI, meskipun proses generalisasi berlangsung lebih lambat dan nilai loss validasi lebih besar dibandingkan dataset FDI. Temuan ini selaras dengan pembahasan pada Gambar 4.7, di mana perbedaan karakteristik dataset sangat memengaruhi dinamika pembelajaran model.



Gambar 4.9. Grafik Pelatihan dan Validasi MDI Run 1 (Tanpa Kombinasi Fitur)

Gambar 4.9 menampilkan visualisasi hasil pelatihan dan validasi model DenseNet-169 pada dataset MDI (Mixed Disease Image) tanpa kombinasi fitur pada run pertama yang dijalankan selama 50 epoch. Dataset MDI merupakan

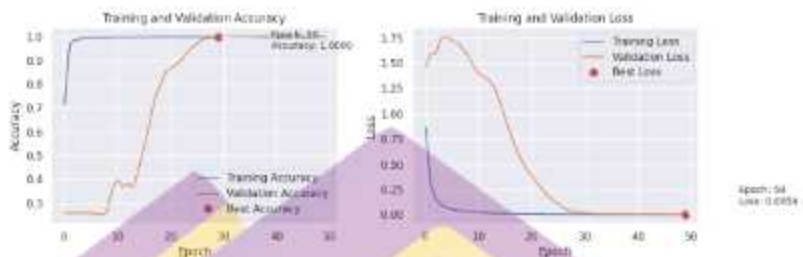
gabungan citra daun (LDI) dan buah jeruk (FDI), sehingga memiliki variasi visual yang lebih heterogen dibandingkan dataset tunggal. Berdasarkan grafik training accuracy, terlihat bahwa akurasi pelatihan meningkat secara cepat sejak epoch awal dan mencapai nilai di atas 0,95 sebelum epoch ke-10. Pola ini konsisten dengan hasil pada dataset FDI dan LDI, yang menunjukkan bahwa DenseNet-169 memiliki kemampuan yang sangat baik dalam mempelajari pola visual dasar pada data latih, meskipun dataset MDI memiliki variasi visual yang lebih kompleks. Pada grafik validation accuracy, terlihat bahwa akurasi validasi pada fase awal pelatihan masih relatif rendah hingga sekitar epoch ke-10. Setelah melewati fase tersebut, terjadi peningkatan akurasi validasi yang tajam dan stabil hingga mencapai nilai terbaik sebesar 0,9933 pada epoch ke-20. Pencapaian ini menunjukkan bahwa model mampu dengan cepat menemukan representasi fitur yang dapat digeneralisasi dengan baik, meskipun harus menangani citra daun dan buah jeruk secara bersamaan. Jika dibandingkan dengan dataset LDI, proses konvergensi pada dataset MDI terlihat lebih cepat dan lebih stabil. Hal ini mengindikasikan bahwa keberadaan citra buah jeruk dalam dataset MDI memberikan informasi visual yang lebih diskriminatif, sehingga membantu model dalam membedakan antar kelas penyakit secara lebih efektif. Pada grafik training loss, terlihat bahwa nilai loss pelatihan menurun secara drastis sejak epoch awal dan mendekati nol pada epoch-epoch berikutnya. Pola penurunan ini menunjukkan bahwa proses optimasi berjalan secara stabil dan kesalahan prediksi pada data latih dapat diminimalkan dengan baik. Sementara itu, grafik validation loss menunjukkan

nilai loss yang relatif tinggi pada fase awal pelatihan dan sempat meningkat hingga mencapai puncaknya pada epoch awal. Fenomena ini mengindikasikan bahwa pada tahap awal, model mengalami kesulitan dalam melakukan generalisasi terhadap data validasi yang lebih heterogen. Namun, setelah melewati epoch ke-10, nilai loss validasi menurun secara signifikan hingga mencapai nilai terendah sebesar 0,0189 pada epoch ke-37. Nilai loss validasi yang rendah ini menunjukkan bahwa model mampu mencapai keseimbangan yang baik antara akurasi dan kesalahan prediksi. Jika dibandingkan dengan dataset LDI, nilai loss validasi minimum pada dataset MDI lebih kecil, yang menunjukkan bahwa penggabungan citra daun dan buah jeruk dapat membantu meningkatkan kemampuan generalisasi model. Secara keseluruhan, hasil pada Gambar 4.9 menunjukkan bahwa model DenseNet-169 tanpa kombinasi fitur mampu memberikan performa klasifikasi yang sangat baik pada dataset MDI. Model tidak hanya mencapai akurasi validasi yang tinggi, tetapi juga menunjukkan proses konvergensi yang relatif cepat dan stabil. Temuan ini selaras dengan analisis pada dataset FDI dan LDI sebelumnya, serta menunjukkan bahwa penggabungan dataset dapat menjadi strategi yang efektif untuk meningkatkan robustitas model.

2. Pelatihan dan Validasi FDI, LDI dan MDI (Run 2)

Gambar 4.10 menampilkan kurva akurasi dan loss hasil pelatihan dan validasi model DenseNet-169 pada dataset FDI tanpa kombinasi fitur pada run ke-2 yang dijalankan selama 50 epoch. Visualisasi ini digunakan untuk

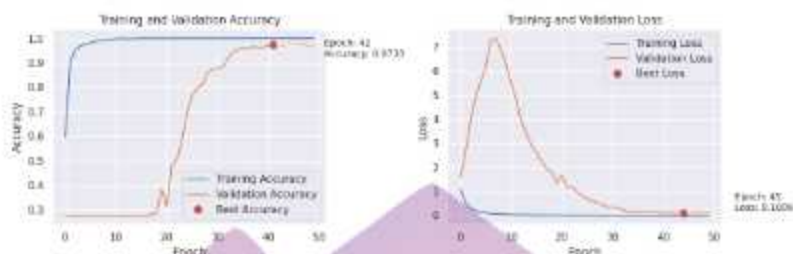
mengevaluasi kestabilan performa model serta konsistensi proses pembelajaran jika dibandingkan dengan hasil pada run sebelumnya.



Gambar 4.10. Grafik Pelatihan dan Validasi FDI Run 2 (Tanpa Kombinasi Fitur)

Berdasarkan grafik training accuracy, terlihat bahwa akurasi pelatihan kembali meningkat dengan sangat cepat sejak epoch awal dan mencapai nilai di atas 0,95 sebelum epoch ke-10. Selanjutnya, akurasi pelatihan terus meningkat hingga mencapai nilai maksimum sebesar 1,0000 pada epoch ke-30. Pola ini menunjukkan bahwa DenseNet-169 secara konsisten mampu mempelajari pola visual pada data latih dengan sangat baik, meskipun pelatihan dilakukan pada run yang berbeda. Hal ini mengindikasikan bahwa proses pelatihan bersifat stabil dan tidak terlalu sensitif terhadap variasi inisialisasi bobot awal. Pada grafik validation accuracy, terlihat bahwa akurasi validasi pada fase awal pelatihan masih relatif rendah dan mengalami fluktuasi hingga sekitar epoch ke-15. Setelah melewati fase tersebut, akurasi validasi meningkat secara signifikan dan mencapai nilai tertinggi sebesar 1,0000 pada epoch ke-30. Peningkatan ini terjadi lebih cepat dibandingkan run sebelumnya, yang menunjukkan bahwa pada run ke-2 model mampu mencapai kondisi generalisasi optimal dengan jumlah epoch yang lebih sedikit. Meskipun

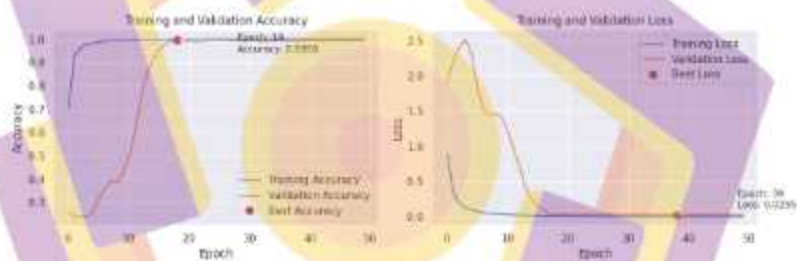
dinamika pembelajaran dapat berbeda antar run, hasil ini menegaskan konsistensi kemampuan DenseNet-169 dalam mengklasifikasikan citra penyakit jeruk pada dataset FDI. Pada grafik training loss, nilai loss pelatihan menurun secara tajam sejak epoch awal dan mendekati nol pada epoch-epoch berikutnya. Penurunan yang cepat dan stabil ini menunjukkan bahwa proses optimasi berjalan dengan baik dan model mampu meminimalkan kesalahan prediksi pada data latih secara efektif. Sementara itu, grafik validation loss menunjukkan nilai loss yang relatif tinggi pada fase awal pelatihan dan sempat meningkat hingga pertengahan pelatihan. Pola ini serupa dengan run sebelumnya dan mengindikasikan bahwa model belum mampu melakukan generalisasi secara optimal pada tahap awal. Namun, setelah melewati epoch ke-20, nilai validation loss menurun secara signifikan hingga mencapai nilai terendah sebesar 0,0054 pada epoch ke-50. Nilai loss yang sangat rendah ini menunjukkan bahwa model mencapai kondisi konvergensi yang sangat baik dengan tingkat kesalahan prediksi yang minimal. Secara keseluruhan, hasil pada Gambar 4.10 memperkuat temuan pada run sebelumnya bahwa DenseNet-169 tanpa kombinasi fitur mampu memberikan performa klasifikasi yang sangat tinggi pada dataset FDI, dengan tingkat stabilitas yang baik pada pengujian berulang.



Gambar 4.11. Grafik Pelatihan dan Validasi LDI Run 2
(Tanpa Kombinasi Fitur)

Gambar 4.11 menampilkan hasil pelatihan dan validasi model DenseNet-169 pada dataset LDI tanpa kombinasi fitur pada run ke-2 yang dijalankan selama 50 epoch. Grafik training accuracy menunjukkan bahwa akurasi pelatihan meningkat dengan cepat sejak epoch awal dan mencapai nilai di atas 0,95 sebelum epoch ke-10. Pola ini konsisten dengan hasil pada run pertama dataset LDI maupun pada dataset FDI, yang menunjukkan bahwa model mampu mempelajari pola visual dasar pada data latih secara efektif. Namun, pada grafik validation accuracy, terlihat bahwa akurasi validasi masih relatif rendah dan cenderung stagnan hingga sekitar epoch ke-25. Setelah melewati fase tersebut, akurasi validasi meningkat secara bertahap hingga mencapai nilai terbaik sebesar 0,9733 pada epoch ke-42. Nilai ini sedikit lebih rendah dibandingkan run pertama dataset LDI, yang menunjukkan adanya variasi performa antar run meskipun konfigurasi pelatihan yang digunakan sama. Pada grafik training loss, nilai loss menurun secara drastis sejak epoch awal dan mendekati nol pada epoch-epoch berikutnya, menunjukkan proses optimasi yang berjalan dengan stabil. Sebaliknya, grafik validation loss menunjukkan

nilai loss yang relatif tinggi pada fase awal dan mencapai puncaknya pada pertengahan pelatihan. Setelah melewati epoch ke-25, nilai validation loss menurun secara bertahap hingga mencapai nilai terendah sebesar 0,1009 pada epoch ke-45. Nilai loss validasi yang lebih tinggi dibandingkan dataset FDI menegaskan bahwa klasifikasi citra daun jeruk memiliki tingkat kesulitan yang lebih besar. Hal ini disebabkan oleh kemiripan pola tekstur dan warna antar kelas penyakit pada daun jeruk, sehingga proses generalisasi memerlukan jumlah epoch yang lebih banyak.



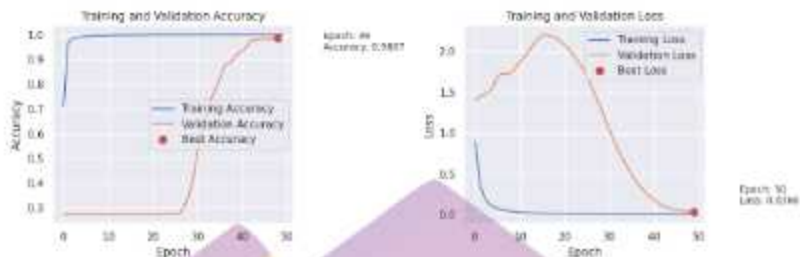
Gambar 4.12. Grafik Pelatihan dan Validasi MDI Run 2 (Tanpa Kombinasi Fitur)

Gambar 4.12 menampilkan hasil pelatihan dan validasi model DenseNet-169 pada dataset MDI tanpa kombinasi fitur pada run ke-2. Apa yang tertera pada grafik training accuracy menunjukkan peningkatan akurasi yang cepat sejak epoch awal dan mencapai nilai di atas 0,95 sebelum epoch ke-10. Pola ini konsisten dengan run sebelumnya dan menunjukkan bahwa model tetap mampu mempelajari pola visual dasar secara efektif meskipun dataset bersifat gabungan. Pada grafik validation accuracy, akurasi validasi meningkat secara tajam setelah epoch ke-10 dan mencapai nilai terbaik sebesar 0,9950 pada

epoch ke-24. Proses konvergensi yang relatif cepat ini menunjukkan bahwa kombinasi citra daun dan buah jeruk memberikan informasi visual yang lebih kaya dan saling melengkapi, sehingga membantu model dalam membedakan antar kelas penyakit secara lebih efektif. Pada grafik training loss, nilai loss menurun secara drastis hingga mendekati nol, sedangkan grafik validation loss menunjukkan penurunan signifikan setelah fase awal pelatihan dan mencapai nilai terendah sebesar 0,0255 pada epoch ke-39. Nilai loss ini sedikit lebih tinggi dibandingkan run pertama dataset MDI, namun masih berada pada rentang yang rendah dan dapat diterima. Secara keseluruhan, hasil pada Gambar 4.12 menunjukkan bahwa model DenseNet-169 tanpa kombinasi fitur memiliki performa yang sangat baik dan konsisten pada dataset MDI. Dibandingkan dengan dataset LDI, proses pembelajaran pada dataset MDI cenderung lebih cepat dan stabil, yang mengindikasikan bahwa penggabungan citra daun dan buah jeruk dapat meningkatkan kemampuan generalisasi model.

3. Pelatihan dan Validasi FDI, LDI dan MDI (Run 3)

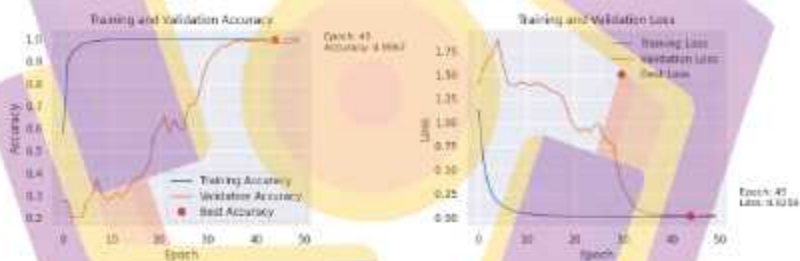
Gambar 4.13 menampilkan hasil pelatihan dan validasi model DenseNet-169 pada dataset FDI tanpa kombinasi fitur pada run ke-3 yang dijalankan selama 50 epoch.



Gambar 4.13. Grafik Pelatihan dan Validasi FDI Run 3 (Tanpa Kombinasi Fitur)

Berdasarkan grafik training accuracy, terlihat bahwa akurasi pelatihan kembali meningkat dengan sangat cepat sejak epoch awal dan mencapai nilai mendekati 1,0 sebelum epoch ke-10. Pola ini menunjukkan bahwa kemampuan DenseNet-169 dalam mempelajari pola visual pada data latih tetap stabil dan konsisten antar run. Hal ini mengindikasikan bahwa proses pembelajaran tidak terlalu dipengaruhi oleh variasi inisialisasi bobot awal. Pada grafik validation accuracy, terlihat bahwa akurasi validasi pada fase awal masih relatif rendah dan cenderung stagnan hingga sekitar epoch ke-25. Setelah melewati fase tersebut, akurasi validasi meningkat secara signifikan hingga mencapai nilai terbaik sebesar 0,9867 pada epoch ke-49. Pola peningkatan ini selaras dengan hasil pada run pertama dataset FDI, yang menunjukkan bahwa model memerlukan jumlah epoch yang cukup untuk mencapai kondisi generalisasi yang optimal. Pada grafik training loss, nilai loss pelatihan menurun secara tajam sejak epoch awal dan mendekati nol pada epoch-epoch selanjutnya. Hal ini menunjukkan bahwa proses optimasi berjalan dengan baik dan kesalahan prediksi pada data latih dapat diminimalkan secara efektif. Sebaliknya, grafik

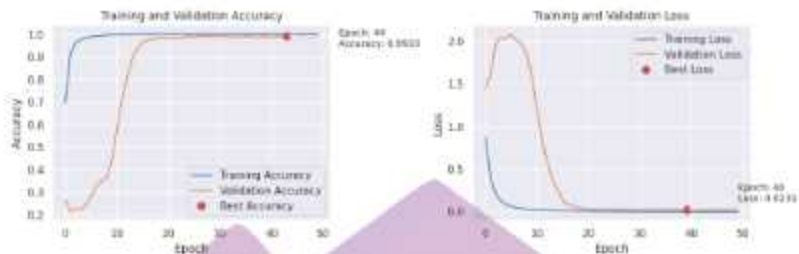
validation loss menunjukkan peningkatan nilai loss pada fase awal hingga pertengahan pelatihan dan mencapai puncaknya sekitar epoch ke-25. Kondisi ini mengindikasikan adanya ketidakseimbangan antara pembelajaran pada data latih dan kemampuan generalisasi pada tahap awal. Namun, setelah melewati epoch ke-30, nilai validation loss menurun secara signifikan hingga mencapai nilai terendah sebesar 0,0366 pada epoch ke-50. Secara keseluruhan, hasil pada Gambar 4.13 menunjukkan bahwa meskipun proses generalisasi pada run ke-3 berlangsung lebih lambat dibandingkan run ke-2, model tetap mampu mencapai performa akhir yang tinggi dan stabil pada dataset FDI.



Gambar 4.14. Grafik Pelatihan dan Validasi LDI Run 3 (Tanpa Kombinasi Fitur)

Gambar 4.14 menampilkan hasil pelatihan dan validasi model DenseNet-169 pada dataset LDI tanpa kombinasi fitur pada run ke-3. Grafik training accuracy menunjukkan bahwa akurasi pelatihan meningkat secara cepat sejak epoch awal dan mencapai nilai mendekati 1,0 sebelum epoch ke-10. Pola ini konsisten dengan hasil pada run sebelumnya dan menunjukkan bahwa model mampu mempelajari pola visual dasar pada citra daun jeruk secara efektif. Namun, pada grafik validation accuracy, terlihat adanya fluktuasi yang cukup

signifikan hingga sekitar epoch ke-30. Fluktuasi ini menunjukkan bahwa pada tahap awal, model belum mampu melakukan generalisasi dengan baik terhadap data validasi. Setelah melewati epoch ke-30, akurasi validasi meningkat secara lebih stabil dan mencapai nilai terbaik sebesar 0,9967 pada epoch ke-45. Nilai ini merupakan salah satu performa tertinggi pada dataset LDI dan menunjukkan bahwa model mampu mencapai generalisasi yang sangat baik setelah melewati fase pembelajaran awal. Pada grafik training loss, nilai loss pelatihan menurun secara konsisten hingga mendekati nol, yang menunjukkan proses optimasi berjalan dengan baik. Sementara itu, grafik validation loss menunjukkan nilai loss yang relatif tinggi dan fluktuatif pada fase awal hingga pertengahan pelatihan. Namun, setelah melewati epoch ke-30, nilai loss validasi menurun secara signifikan hingga mencapai nilai terendah sebesar 0,0258 pada epoch ke-45. Jika dibandingkan dengan run sebelumnya pada dataset LDI, run ke-3 menunjukkan proses konvergensi yang lebih lambat, namun menghasilkan akurasi validasi akhir yang lebih tinggi dan nilai loss yang lebih rendah. Hal ini mengindikasikan bahwa kompleksitas visual citra daun jeruk memerlukan jumlah epoch yang lebih banyak untuk mencapai performa optimal.



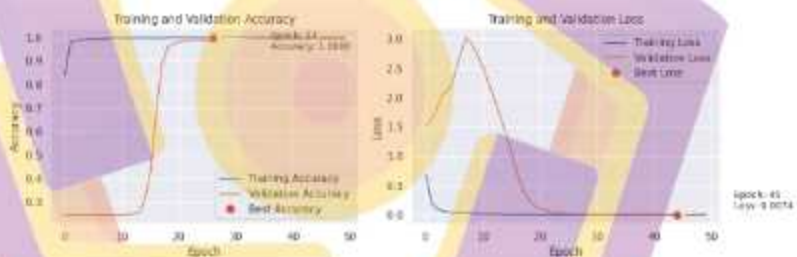
Gambar 4.15. Grafik Pelatihan dan Validasi MDI Run 3
(Tanpa Kombinasi Fitur)

Gambar 4.15 menampilkan hasil pelatihan dan validasi model DenseNet-169 pada dataset MDI tanpa kombinasi fitur pada run ke-3. Grafik training accuracy menunjukkan peningkatan akurasi yang cepat sejak epoch awal dan mencapai nilai di atas 0,95 sebelum epoch ke-10, serupa dengan hasil pada run sebelumnya. Hal ini menunjukkan bahwa model tetap mampu mempelajari pola visual dasar secara efektif meskipun dataset bersifat gabungan. Pada grafik validation accuracy, terlihat bahwa akurasi validasi meningkat secara stabil setelah fase awal pelatihan dan mencapai nilai terbaik sebesar 0,9933 pada epoch ke-44. Proses konvergensi pada run ke-3 berlangsung lebih lambat dibandingkan run pertama dan kedua, namun tetap menghasilkan performa akhir yang sangat tinggi. Pada grafik training loss, nilai loss pelatihan menurun secara drastis hingga mendekati nol. Sementara itu, grafik validation loss menunjukkan penurunan yang signifikan setelah fase awal pelatihan dan mencapai nilai terendah sebesar 0,0231 pada epoch ke-40. Nilai ini menunjukkan keseimbangan yang baik antara akurasi dan kesalahan prediksi. Secara keseluruhan, hasil pada Gambar 4.15 menunjukkan bahwa model

DenseNet-169 tanpa kombinasi fitur mampu mempertahankan performa klasifikasi yang sangat baik pada dataset MDI. Dibandingkan dengan dataset LDI, proses pembelajaran pada dataset MDI cenderung lebih stabil, yang menunjukkan bahwa penggabungan citra daun dan buah jeruk dapat meningkatkan kemampuan generalisasi model.

4. Pelatihan dan Validasi FDI, LDI dan MDI (Run 4)

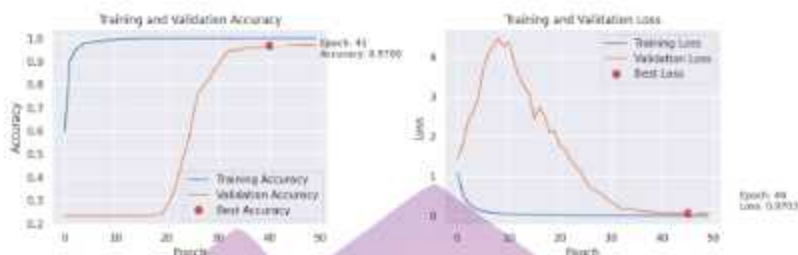
Gambar 4.16 menyajikan hasil pelatihan dan validasi model DenseNet-169 pada dataset FDI tanpa kombinasi fitur pada run ke-4 yang dijalankan selama 50 epoch.



Gambar 4.16. Grafik Pelatihan dan Validasi FDI Run 4 (Tanpa Kombinasi Fitur)

Berdasarkan grafik training accuracy, terlihat bahwa akurasi pelatihan meningkat sangat cepat sejak epoch awal dan mencapai nilai mendekati 1,0 sebelum epoch ke-10. Pola ini konsisten dengan hasil pada run sebelumnya dan menunjukkan bahwa DenseNet-169 secara konsisten mampu mempelajari pola visual pada data latih secara efektif, terlepas dari variasi running yang dilakukan. Pada grafik validation accuracy, akurasi validasi pada fase awal masih relatif rendah dan cenderung stagnan hingga sekitar epoch ke-15. Setelah

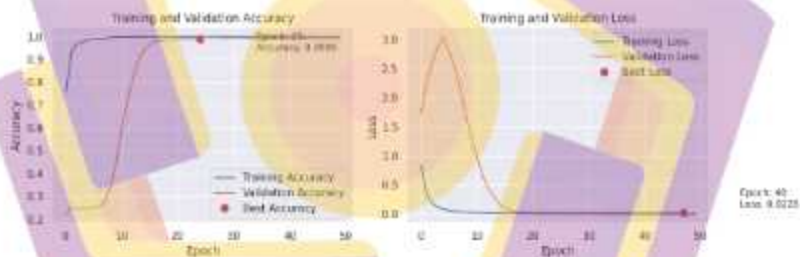
fase tersebut, terjadi peningkatan akurasi validasi yang sangat signifikan hingga mencapai nilai maksimum sebesar 1,0000 pada epoch ke-27. Pencapaian ini menunjukkan bahwa pada run ke-4, model mampu mencapai kondisi generalisasi optimal pada jumlah epoch yang lebih sedikit dibandingkan beberapa run sebelumnya, seperti pada Run 1 dan Run 3. Percepatan konvergensi ini mengindikasikan adanya kondisi optimasi yang lebih menguntungkan, meskipun arsitektur dan dataset yang digunakan tetap sama. Grafik training loss menunjukkan penurunan nilai loss yang sangat tajam sejak epoch awal dan mendekati nol pada epoch-epoch berikutnya. Pola ini menandakan bahwa proses optimasi berjalan secara stabil dan kesalahan prediksi pada data latih dapat diminimalkan secara efektif. Sementara itu, grafik validation loss menunjukkan nilai loss yang relatif tinggi pada epoch awal dan sempat meningkat hingga mencapai puncaknya sekitar epoch ke-8. Fenomena ini serupa dengan yang diamati pada run sebelumnya dan mengindikasikan bahwa pada tahap awal, model belum sepenuhnya mampu melakukan generalisasi terhadap data validasi. Namun, setelah melewati epoch ke-10, nilai validation loss menurun secara sangat signifikan hingga mencapai nilai terendah sebesar 0,0074 pada epoch ke-45. Secara keseluruhan, hasil pada Gambar 4.16 menunjukkan bahwa model DenseNet-169 tanpa kombinasi fitur memiliki performa yang sangat stabil dan konsisten pada dataset FDI. Meskipun terdapat variasi pada kecepatan konvergensi antar run, model tetap mampu mencapai akurasi validasi yang sangat tinggi dengan nilai loss yang sangat rendah, yang menandakan kemampuan generalisasi yang sangat baik.



Gambar 4.17. Grafik Pelatihan dan Validasi LDI Run 4
(Tanpa Kombinasi Fitur)

Gambar 4.17 menampilkan hasil pelatihan dan validasi model DenseNet-169 pada dataset LDI tanpa kombinasi fitur pada run ke-4. Grafik training accuracy menunjukkan bahwa akurasi pelatihan kembali meningkat dengan cepat sejak epoch awal dan mencapai nilai mendekati 1,0 sebelum epoch ke-10. Pola ini konsisten dengan hasil pada LDI Run 1 hingga Run 3, yang menegaskan bahwa DenseNet-169 mampu mempelajari pola visual dasar pada citra daun jeruk secara efektif. Namun, pada grafik validation accuracy, terlihat bahwa akurasi validasi pada fase awal pelatihan masih relatif rendah dan cenderung stagnan hingga sekitar epoch ke-20. Setelah melewati fase tersebut, akurasi validasi meningkat secara bertahap dan stabil hingga mencapai nilai terbaik sebesar 0,9700 pada epoch ke-41. Nilai ini sedikit lebih rendah dibandingkan LDI Run 3, namun tetap menunjukkan performa klasifikasi yang tinggi dan konsisten. Grafik training loss memperlihatkan penurunan nilai loss yang tajam sejak epoch awal hingga mendekati nol, menunjukkan bahwa proses optimasi berjalan dengan baik. Sebaliknya, grafik validation loss menunjukkan nilai loss yang relatif tinggi pada fase awal dan pertengahan pelatihan, bahkan mencapai

puncak yang cukup besar. Setelah melewati epoch ke-25, nilai validation loss mulai menurun secara bertahap hingga mencapai nilai terendah sebesar 0,5703 pada epoch ke-46. Jika dibandingkan dengan run sebelumnya, nilai loss validasi minimum pada LDI Run 4 tergolong lebih tinggi. Hal ini menunjukkan bahwa meskipun model mampu mencapai akurasi validasi yang baik, tingkat kesalahan prediksi pada run ini relatif lebih besar. Kondisi tersebut mengindikasikan bahwa klasifikasi citra daun jeruk masih memiliki tingkat kompleksitas yang lebih tinggi dibandingkan citra buah jeruk, sehingga menghasilkan dinamika pembelajaran yang lebih fluktuatif.



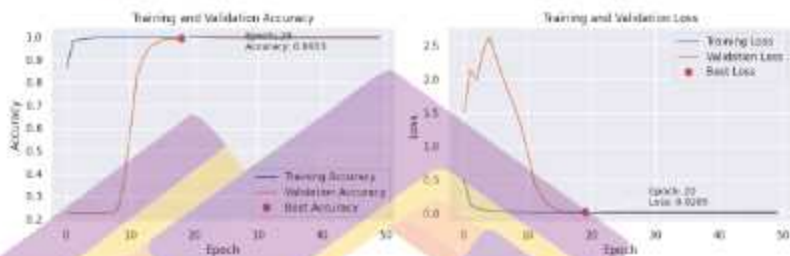
Gambar 4.18. Grafik Pelatihan dan Validasi MDI Run 4 (Tanpa Kombinasi Fitur)

Gambar 4.18 menampilkan hasil pelatihan dan validasi model DenseNet-169 pada dataset MDI tanpa kombinasi fitur pada run ke-4. Berdasarkan grafik training accuracy, terlihat bahwa akurasi pelatihan meningkat sangat cepat sejak epoch awal dan mencapai nilai di atas 0,95 sebelum epoch ke-10. Pola ini konsisten dengan seluruh run sebelumnya pada dataset MDI serta selaras dengan hasil pada dataset FDI dan LDI, yang menunjukkan bahwa DenseNet-169 memiliki kemampuan yang sangat baik dalam mempelajari pola visual

dasar. Pada grafik validation accuracy, akurasi validasi pada fase awal masih relatif rendah hingga sekitar epoch ke-10. Setelah fase tersebut, terjadi peningkatan akurasi validasi yang tajam dan stabil hingga mencapai nilai terbaik sebesar 0,9900 pada epoch ke-25. Dibandingkan dengan MDI Run 3, proses konvergensi pada run ke-4 berlangsung lebih cepat dan stabil. Grafik training loss menunjukkan penurunan nilai loss yang drastis hingga mendekati nol, sementara grafik validation loss memperlihatkan nilai loss yang tinggi pada fase awal dan kemudian menurun secara signifikan setelah epoch ke-10. Nilai validation loss terendah dicapai sebesar 0,0226 pada epoch ke-48, yang menunjukkan keseimbangan yang baik antara akurasi dan kesalahan prediksi. Secara keseluruhan, hasil pada Gambar 4.18 memperkuat temuan pada run sebelumnya bahwa model DenseNet-169 tanpa kombinasi fitur mampu memberikan performa klasifikasi yang sangat baik dan stabil pada dataset MDI. Dibandingkan dengan dataset LDI, proses pembelajaran pada dataset MDI cenderung lebih stabil, yang menunjukkan bahwa penggabungan citra daun dan buah jeruk dapat membantu meningkatkan kemampuan generalisasi model.

5. Pelatihan dan Validasi FDI, LDI dan MDI (Run 5)

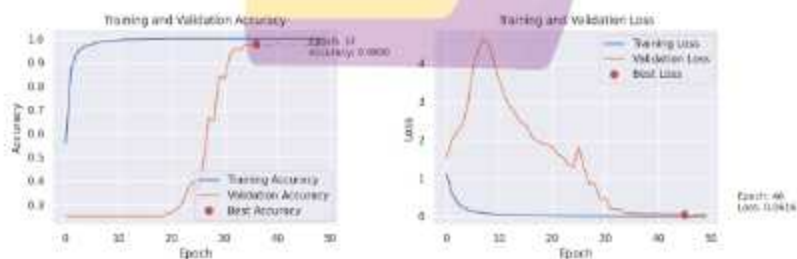
Gambar 4.19 memperlihatkan hasil pelatihan dan validasi model DenseNet-169 pada dataset FDI tanpa kombinasi fitur pada run ke-5 yang dijalankan selama 50 epoch.



Gambar 4.19. Grafik Pelatihan dan Validasi FDI Run 5 (Tanpa Kombinasi Fitur)

Berdasarkan grafik training accuracy, terlihat bahwa akurasi pelatihan kembali meningkat sangat cepat sejak epoch awal dan mencapai nilai mendekati 1,0 sebelum epoch ke-10. Pola ini konsisten dengan hasil pada run sebelumnya (Gambar 4.7 hingga Gambar 4.16) dan menunjukkan bahwa DenseNet-169 memiliki kemampuan yang sangat baik dalam mempelajari pola visual pada data latih secara stabil pada setiap pengujian. Konsistensi ini mengindikasikan bahwa proses pembelajaran model tidak bergantung pada satu kondisi pelatihan tertentu. Pada grafik validation accuracy, terlihat bahwa akurasi validasi pada fase awal masih relatif rendah dan cenderung stagnan hingga sekitar epoch ke-10. Setelah melewati fase tersebut, akurasi validasi meningkat secara tajam hingga mencapai nilai terbaik sebesar 0,9933 pada epoch ke-20. Peningkatan yang relatif cepat ini menunjukkan bahwa pada run ke-5, model mampu mencapai kondisi generalisasi optimal dengan jumlah epoch yang lebih sedikit dibandingkan beberapa run sebelumnya, seperti Run 1 dan Run 3. Hal

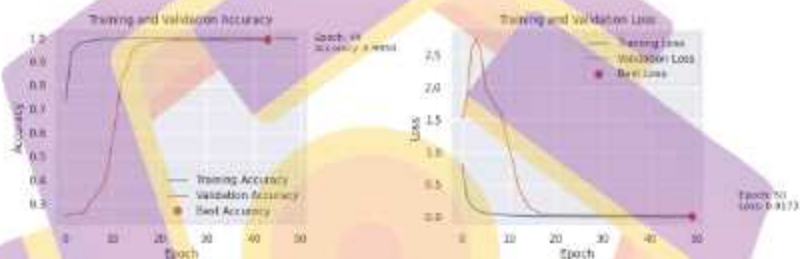
ini menunjukkan bahwa meskipun terdapat variasi dinamika pembelajaran antar run, DenseNet-169 tetap mampu menyesuaikan bobot secara efektif untuk menghasilkan performa validasi yang tinggi. Grafik training loss menunjukkan penurunan nilai loss yang sangat tajam sejak epoch awal dan mendekati nol pada epoch-epoch selanjutnya, yang menandakan bahwa proses optimasi berjalan dengan baik dan kesalahan prediksi pada data latih dapat diminimalkan secara efektif. Sementara itu, grafik validation loss menunjukkan nilai loss yang relatif tinggi pada fase awal dan sempat meningkat hingga pertengahan pelatihan. Namun, setelah melewati epoch ke-10, nilai loss validasi menurun secara signifikan hingga mencapai nilai terendah sebesar 0,0205 pada epoch ke-20. Meskipun nilai ini sedikit lebih tinggi dibandingkan beberapa run lain, seperti Run 2 dan Run 4, nilai tersebut masih berada pada rentang yang rendah dan dapat diterima. Secara keseluruhan, hasil pada Gambar 4.19 memperkuat temuan sebelumnya bahwa DenseNet-169 tanpa kombinasi fitur mampu memberikan performa klasifikasi yang sangat stabil dan konsisten pada dataset FDI.



Gambar 4.20. Grafik Pelatihan dan Validasi LDI Run 5 (Tanpa Kombinasi Fitur)

Gambar 4.20 menampilkan hasil pelatihan dan validasi model DenseNet-169 pada dataset LDI tanpa kombinasi fitur pada run ke-5. Pada grafik training accuracy menunjukkan bahwa akurasi pelatihan kembali meningkat sangat cepat sejak epoch awal dan mencapai nilai mendekati 1,0 sebelum epoch ke-10. Pola ini konsisten dengan seluruh run sebelumnya pada dataset LDI dan sejalan dengan hasil pada dataset FDI, yang menunjukkan bahwa DenseNet-169 mampu mempelajari pola visual dasar pada data latih secara efektif. Pada grafik validation accuracy, terlihat bahwa akurasi validasi pada fase awal masih relatif rendah dan meningkat secara bertahap hingga sekitar epoch ke-20. Setelah melewati fase tersebut, akurasi validasi meningkat secara lebih tajam dan stabil hingga mencapai nilai terbaik sebesar 0,9800 pada epoch ke-37. Nilai ini menunjukkan bahwa pada run ke-5, model mampu mencapai kondisi generalisasi optimal dengan jumlah epoch yang relatif lebih sedikit dibandingkan beberapa run sebelumnya, seperti LDI Run 3 dan Run 4. Grafik training loss memperlihatkan penurunan nilai loss yang tajam sejak epoch awal hingga mendekati nol, menunjukkan bahwa proses optimasi berjalan secara stabil. Sebaliknya, grafik validation loss menunjukkan nilai loss yang relatif tinggi dan fluktuatif pada fase awal hingga pertengahan pelatihan, yang kembali mengindikasikan adanya kesulitan generalisasi pada tahap awal—sebuah karakteristik umum pada dataset LDI. Namun, setelah melewati epoch ke-25, nilai loss validasi menurun secara signifikan hingga mencapai nilai terendah sebesar 0,0616 pada epoch ke-46. Jika dibandingkan dengan LDI Run 4, nilai loss minimum pada run ke-5 lebih rendah, yang menunjukkan adanya

perbaiki kemampuan generalisasi model. Secara keseluruhan, hasil pada Gambar 4.20 memperkuat temuan pada LDI Run 1 hingga Run 4 bahwa meskipun klasifikasi citra daun jeruk memiliki tingkat kompleksitas visual yang lebih tinggi dibandingkan citra buah jeruk, model DenseNet-169 tetap mampu mencapai performa klasifikasi yang baik dan konsisten setelah melewati fase pembelajaran awal.



Gambar 4.21. Grafik Pelatihan dan Validasi MDI Run 5 (Tanpa Kombinasi Fitur)

Gambar 4.21 menyajikan hasil pelatihan dan validasi model DenseNet-169 pada dataset MDI tanpa kombinasi fitur pada run ke-5. Berdasarkan apa yang terlihat pada grafik training accuracy, akurasi pelatihan meningkat sangat cepat sejak epoch awal dan mencapai nilai di atas 0,95 sebelum epoch ke-10. Pola ini konsisten dengan seluruh hasil sebelumnya pada dataset MDI serta selaras dengan hasil pada dataset FDI dan LDI, yang menunjukkan bahwa DenseNet-169 mampu mempelajari pola visual dasar secara efektif. Pada grafik validation accuracy, akurasi validasi pada fase awal meningkat secara bertahap hingga sekitar epoch ke-15, kemudian meningkat secara lebih tajam dan stabil hingga mencapai nilai terbaik sebesar 0,9950 pada epoch ke-44. Nilai ini merupakan

salah satu performa tertinggi pada seluruh pengujian dataset MDI dan menunjukkan kemampuan generalisasi model yang sangat baik. Grafik training loss menunjukkan penurunan nilai loss yang drastis sejak epoch awal hingga mendekati nol, menandakan proses optimasi berjalan secara stabil. Sementara itu, grafik validation loss memperlihatkan nilai loss yang relatif tinggi pada fase awal pelatihan dan sempat meningkat hingga mencapai puncaknya. Namun, setelah melewati epoch ke-15, nilai loss validasi menurun secara signifikan hingga mencapai nilai terendah sebesar 0,0173 pada epoch ke-50. Nilai loss yang sangat rendah ini menunjukkan bahwa model tidak hanya mencapai akurasi tinggi, tetapi juga memiliki tingkat kesalahan prediksi yang sangat kecil. Secara keseluruhan, hasil pada Gambar 4.21 semakin menegaskan bahwa model DenseNet-169 tanpa kombinasi fitur mampu memberikan performa klasifikasi yang sangat baik dan konsisten pada dataset MDI. Dibandingkan dengan dataset LDI, proses pembelajaran pada dataset MDI cenderung lebih stabil, yang menunjukkan bahwa penggabungan citra daun dan buah jeruk dapat membantu meningkatkan kemampuan generalisasi model.

Berdasarkan hasil pelatihan dan validasi yang telah dilakukan pada lima kali running menggunakan model DenseNet-169 tanpa kombinasi fitur, dapat disimpulkan bahwa model menunjukkan performa yang sangat konsisten dan stabil pada ketiga dataset, yaitu FDI (Fruit Disease Image), LDI (Leaf Disease Image), dan MDI (Mixed Disease Image). Secara umum, pada seluruh run terlihat bahwa akurasi pelatihan meningkat sangat cepat sejak epoch awal dan mencapai nilai

mendekati maksimum sebelum epoch ke-10. Pola ini mengindikasikan bahwa arsitektur DenseNet-169 memiliki kemampuan yang sangat baik dalam mempelajari pola visual dasar dari data citra penyakit jeruk, baik pada citra buah maupun daun. Pada dataset FDI, seluruh run menunjukkan performa yang sangat tinggi dengan akurasi validasi yang konsisten berada pada rentang nilai yang sangat tinggi, bahkan pada beberapa run—termasuk pada run pertama setelah pembaruan grafik—model mampu mencapai akurasi validasi maksimum sebesar 1,0000 pada epoch tertentu. Proses konvergensi pada dataset ini cenderung berlangsung cepat dan stabil, serta diikuti oleh nilai validation loss yang sangat rendah pada fase akhir pelatihan. Hal ini menunjukkan bahwa citra buah jeruk memiliki karakteristik visual yang relatif lebih mudah dibedakan antar kelas penyakit, sehingga model mampu melakukan generalisasi dengan sangat baik. Variasi kecil yang muncul antar run terutama dipengaruhi oleh dinamika optimasi dan inisialisasi bobot awal, namun tidak berdampak signifikan terhadap performa akhir model. Berbeda dengan dataset FDI, dataset LDI menunjukkan dinamika pembelajaran yang lebih fluktuatif pada fase awal pelatihan. Pada beberapa run, akurasi validasi cenderung meningkat lebih lambat dan disertai nilai validation loss yang relatif lebih tinggi. Kondisi ini mencerminkan tingkat kompleksitas visual yang lebih tinggi pada citra daun jeruk, di mana gejala penyakit sering kali memiliki kemiripan tekstur dan warna antar kelas. Meskipun demikian, setelah melewati fase pembelajaran awal, model tetap mampu mencapai performa klasifikasi yang tinggi dan stabil pada seluruh run. Hal ini menegaskan bahwa DenseNet-169 tetap efektif dalam menangani kompleksitas visual yang terdapat pada dataset LDI.

Sementara itu, dataset MDI yang merupakan gabungan citra daun dan buah jeruk menunjukkan karakteristik pembelajaran yang relatif lebih stabil dibandingkan dataset LDI. Pada hampir seluruh run, proses konvergensi pada dataset MDI berlangsung lebih cepat dan diikuti oleh nilai validation loss yang lebih rendah. Hal ini menunjukkan bahwa penggabungan citra daun dan buah jeruk memberikan informasi visual yang lebih kaya dan saling melengkapi, sehingga membantu model dalam membentuk representasi fitur yang lebih diskriminatif. Dengan demikian, model mampu menunjukkan kemampuan generalisasi yang sangat baik pada data dengan variasi visual yang lebih heterogen. Secara keseluruhan, hasil pelatihan dan validasi pada lima run menunjukkan bahwa model DenseNet-169 tanpa kombinasi fitur memiliki kemampuan adaptasi dan generalisasi yang sangat baik pada seluruh dataset yang digunakan. Pembaruan hasil pada dataset FDI run pertama semakin memperkuat kesimpulan bahwa model mampu mencapai performa optimal secara konsisten, baik dari sisi akurasi maupun loss. Model juga terbukti robust terhadap variasi data dan kondisi pelatihan, khususnya pada dataset MDI yang menunjukkan keseimbangan terbaik antara stabilitas pembelajaran dan kemampuan generalisasi. Ringkasan kuantitatif dari performa model pada masing-masing dataset dan run selanjutnya disajikan dalam bentuk matriks evaluasi pada Tabel 4.3.

Tabel 4.3. Hasil Matrik Evaluasi DenseNet-169

No	Dataset	Run	Akurasi %	Presisi %	Recall %	F1 Score %
1	Fruit Disease Image	1	99.66	99.62	99.66	99.64
2	Leaf Disease Image		98.00	97.89	97.89	97.89
3	Mixed Disease Image		99.16	99.11	99.11	99.11
4	Fruit Disease Image	2	99.33	99.39	99.30	99.34
5	Leaf Disease Image		97.00	96.66	96.54	96.60
6	Mixed Disease Image		99.33	99.31	99.29	99.30
7	Fruit Disease Image	3	99.00	99.02	99.00	98.99
8	Leaf Disease Image		98.33	98.42	98.29	98.35
9	Mixed Disease Image		99.00	98.98	98.98	98.98
10	Fruit Disease Image	4	99.66	99.64	99.70	99.67
11	Leaf Disease Image		98.66	98.79	98.75	98.74
12	Mixed Disease Image		98.66	98.68	98.66	98.66
13	Fruit Disease Image	5	99.66	99.69	99.70	99.69
14	Leaf Disease Image		97.33	97.44	97.46	97.45
15	Mixed Disease Image		98.83	98.85	98.93	98.89

Berdasarkan hasil yang ditampilkan pada Tabel 4.3 diatas, model menunjukkan performa yang sangat baik dan konsisten pada seluruh dataset dan run. Dataset FDI secara umum menghasilkan nilai akurasi dan F1-score tertinggi pada hampir seluruh running, yang mengindikasikan bahwa citra buah jeruk memiliki karakteristik visual yang lebih mudah dibedakan antar kelas penyakit. Dataset MDI juga menunjukkan performa yang tinggi dan stabil, yang mengonfirmasi bahwa penggabungan citra daun dan buah jeruk mampu memperkaya informasi fitur sehingga meningkatkan kemampuan generalisasi model.

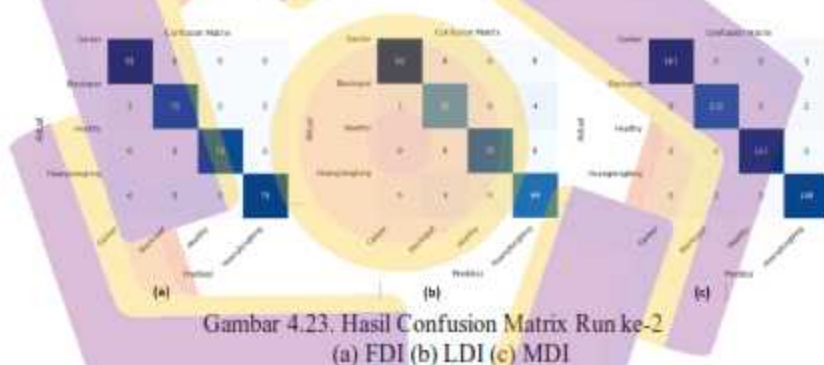
Sebaliknya, dataset LDI cenderung menghasilkan nilai evaluasi yang sedikit lebih rendah dibandingkan FDI dan MDI. Hal ini menunjukkan bahwa klasifikasi citra daun jeruk memiliki tingkat kompleksitas visual yang lebih tinggi, terutama akibat kemiripan tekstur dan pola warna antar kelas penyakit. Meskipun demikian, nilai presisi, recall, dan F1-score pada dataset LDI tetap berada pada rentang yang tinggi, yang menandakan bahwa model masih mampu melakukan klasifikasi dengan tingkat kesalahan yang relatif rendah. Untuk memperoleh pemahaman yang lebih mendalam mengenai distribusi kesalahan klasifikasi, khususnya terkait kesalahan false positive dan false negative pada masing-masing kelas penyakit, pembahasan selanjutnya akan difokuskan pada analisis confusion matrix pada setiap dataset dan running.



Gambar 4.22. Hasil Confusion Matrix Run ke-1
(a) FDI (b) LDI (c) MDI

Gambar 4.22 menunjukkan confusion matrix hasil klasifikasi model DenseNet-169 tanpa kombinasi fitur pada run ke-1 untuk dataset FDI, LDI, dan MDI. Secara umum, nilai pada elemen diagonal mendominasi, yang menandakan bahwa sebagian besar citra berhasil diklasifikasikan dengan benar. Pada dataset FDI, kesalahan klasifikasi sangat minimal, menunjukkan bahwa karakteristik visual

citra buah jeruk relatif mudah dibedakan antar kelas. Pada dataset LDI, masih terdapat sedikit kesalahan klasifikasi antar kelas tertentu, yang mengindikasikan kompleksitas visual citra daun jeruk akibat kemiripan pola dan warna gejala penyakit. Sementara itu, dataset MDI kembali menunjukkan performa yang stabil dengan tingkat klasifikasi benar yang tinggi, didukung oleh variasi fitur yang lebih kaya dari penggabungan citra daun dan buah. Secara keseluruhan, confusion matrix pada run ke-1 menegaskan bahwa model DenseNet-169 mampu melakukan klasifikasi penyakit jeruk dengan akurasi dan konsistensi yang tinggi pada seluruh dataset.



Gambar 4.23 menampilkan confusion matrix hasil klasifikasi model DenseNet-169 tanpa kombinasi fitur pada run ke-2 untuk dataset FDI, LDI, dan MDI. Secara umum, mayoritas nilai terkonsentrasi pada elemen diagonal, yang menunjukkan tingkat klasifikasi benar yang tinggi pada seluruh dataset. Pada dataset FDI, model berhasil mengklasifikasikan sebagian besar citra secara tepat, dengan nilai diagonal masing-masing kelas mencapai 81 citra canker, 70 citra

blackspot, 72 citra healthy, dan 75 citra huanglongbing. Kesalahan klasifikasi yang terjadi sangat terbatas dan hanya muncul pada beberapa sampel antar kelas, sehingga tidak berdampak signifikan terhadap performa keseluruhan. Pada dataset LDI, nilai diagonal masih mendominasi dengan 92 citra canker, 55 citra blackspot, 75 citra healthy, dan 69 citra huanglongbing yang terklasifikasi dengan benar. Namun, terlihat adanya kesalahan klasifikasi yang lebih bervariasi dibandingkan FDI, khususnya pada kelas blackspot dan huanglongbing, yang mencerminkan kompleksitas visual citra daun jeruk akibat kemiripan tekstur dan warna antar penyakit. Sementara itu, pada dataset MDI, performa model kembali menunjukkan kestabilan yang sangat baik dengan nilai diagonal yang tinggi, yaitu 161 citra canker, 130 citra blackspot, 157 citra healthy, dan 148 citra huanglongbing. Jumlah kesalahan prediksi sangat kecil dan tersebar merata, yang menunjukkan bahwa penggabungan citra daun dan buah jeruk membantu model membentuk representasi fitur yang lebih diskriminatif. Secara keseluruhan, hasil confusion matrix pada run ke-2 menegaskan bahwa model DenseNet-169 mampu mempertahankan performa klasifikasi yang tinggi dan konsisten, dengan tingkat kesalahan yang rendah pada seluruh dataset, khususnya pada dataset MDI yang menunjukkan kemampuan generalisasi terbaik.



Gambar 4.24. Hasil Confusion Matrix Run ke-3
(a) FDI (b) LDI (c) MDI

Gambar 4.24 menampilkan confusion matrix hasil klasifikasi model DenseNet-169 tanpa kombinasi fitur pada run ke-3 untuk dataset FDI, LDI, dan MDI. Secara umum, nilai pada elemen diagonal masih mendominasi, yang menunjukkan bahwa sebagian besar citra berhasil diklasifikasikan dengan benar pada setiap kelas. Pada dataset FDI, model mampu mengklasifikasikan 72 citra canker, 74 citra blackspot, 77 citra healthy, dan 74 citra huanglongbing secara tepat. Kesalahan klasifikasi yang muncul sangat terbatas dan hanya terjadi pada beberapa sampel antar kelas, sehingga tidak berpengaruh signifikan terhadap performa keseluruhan. Pada dataset LDI, nilai diagonal menunjukkan 56 citra canker, 75 citra blackspot, 85 citra healthy, dan 79 citra huanglongbing yang terklasifikasi dengan benar. Dibandingkan dengan FDI, masih terlihat beberapa kesalahan prediksi, khususnya pada kelas canker dan huanglongbing, yang mencerminkan tingkat kompleksitas visual citra daun jeruk akibat kemiripan pola dan warna gejala penyakit. Sementara itu, pada dataset MDI, model kembali menunjukkan performa yang sangat stabil dengan nilai diagonal yang tinggi, yaitu 134 citra canker, 144 citra blackspot, 154 citra healthy, dan 162 citra huanglongbing. Jumlah kesalahan

klasifikasi sangat kecil dan tersebar merata, menunjukkan bahwa penggabungan citra daun dan buah jeruk membantu model membentuk representasi fitur yang lebih diskriminatif. Secara keseluruhan, hasil confusion matrix pada run ke-3 menegaskan bahwa model DenseNet-169 mampu mempertahankan performa klasifikasi yang tinggi dan konsisten, dengan tingkat kesalahan yang relatif lebih rendah pada dataset MDI dibandingkan dataset FDI dan LDI.



Gambar 4.25 menampilkan confusion matrix hasil klasifikasi model DenseNet-169 tanpa kombinasi fitur pada run ke-4 untuk dataset FDI, LDI, dan MDI. Secara umum, nilai diagonal tetap mendominasi, menunjukkan bahwa model mampu mempertahankan akurasi klasifikasi yang tinggi dan konsisten. Pada dataset FDI, model berhasil mengklasifikasikan 69 citra canker, 84 citra blackspot, 75 citra healthy, dan 77 citra huanglongbing secara benar. Kesalahan klasifikasi sangat minimal, hanya terjadi pada satu citra blackspot yang diprediksi sebagai canker, sehingga tidak berdampak signifikan terhadap performa keseluruhan. Pada dataset LDI, nilai diagonal menunjukkan 75 citra canker, 76 citra blackspot, 66 citra healthy, dan 79 citra huanglongbing yang terklasifikasi dengan tepat. Dibandingkan

FDI, jumlah kesalahan prediksi masih sedikit lebih tinggi, terutama pada kelas healthy yang beberapa kali tertukar dengan kelas lain. Hal ini kembali mencerminkan kompleksitas visual citra daun jeruk yang memiliki kemiripan tekstur dan warna antar kelas penyakit. Sementara itu, pada dataset MDI, performa model terlihat sangat stabil dengan klasifikasi benar sebanyak 162 citra canker, 144 citra blackspot, 139 citra healthy, dan 147 citra huanglongbing. Kesalahan klasifikasi sangat kecil dan tersebar, menunjukkan bahwa penggabungan citra daun dan buah jeruk membantu model membangun representasi fitur yang lebih kuat dan diskriminatif. Secara keseluruhan, hasil confusion matrix pada run ke-4 menegaskan bahwa DenseNet-169 mampu mempertahankan kemampuan generalisasi yang baik, dengan performa paling stabil ditunjukkan pada dataset MDI, diikuti oleh FDI dan LDI.



Gambar 4.26. Hasil Confusion Matrix Run ke-5
(a) FDI (b) LDI (c) MDI

Gambar 4.26 menyajikan confusion matrix hasil klasifikasi model DenseNet-169 tanpa kombinasi fitur pada run ke-5 untuk dataset FDI, LDI, dan MDI. Secara umum, pola diagonal yang dominan kembali terlihat, menunjukkan bahwa model mempertahankan performa klasifikasi yang tinggi dan konsisten hingga run terakhir. Pada dataset FDI, model berhasil mengklasifikasikan 84 citra

canker, 80 citra blackspot, 64 citra healthy, dan 71 citra huanglongbing dengan benar. Hampir tidak terdapat kesalahan klasifikasi antar kelas, yang menegaskan bahwa karakteristik visual pada citra buah jeruk relatif mudah dibedakan oleh model. Pada dataset LDI, hasil klasifikasi menunjukkan 76 citra canker, 75 citra blackspot, 66 citra healthy, dan 75 citra huanglongbing terklasifikasi secara tepat. Beberapa kesalahan kecil masih muncul, seperti satu citra canker dan tiga citra huanglongbing yang tertukar dengan kelas lain. Kondisi ini konsisten dengan hasil run sebelumnya dan mencerminkan kompleksitas visual citra daun jeruk yang lebih tinggi. Sementara itu, pada dataset MDI, model menunjukkan performa yang sangat stabil dengan klasifikasi benar sebanyak 131 citra canker, 171 citra blackspot, 144 citra healthy, dan 147 citra huanglongbing. Jumlah kesalahan prediksi sangat minimal dan tersebar, yang mengindikasikan bahwa penggabungan citra daun dan buah jeruk memberikan representasi fitur yang lebih kaya dan membantu meningkatkan kemampuan generalisasi model. Secara keseluruhan, hasil pada Gambar 4.26 menegaskan bahwa DenseNet-169 tanpa kombinasi fitur mampu mempertahankan performa klasifikasi yang sangat baik pada seluruh dataset hingga run ke-5, dengan kestabilan terbaik ditunjukkan pada dataset MDI.

Berdasarkan analisis confusion matrix pada Run ke-1 hingga Run ke-5, dapat disimpulkan bahwa model DenseNet-169 tanpa kombinasi fitur menunjukkan pola performa yang konsisten dan stabil pada seluruh dataset yang diuji, yaitu FDI, LDI, dan MDI. Pada setiap running, mayoritas prediksi berada pada elemen diagonal matriks, yang menandakan bahwa sebagian besar sampel berhasil diklasifikasikan ke kelas yang benar. Pada dataset FDI, hasil confusion matrix di

seluruh run memperlihatkan tingkat kesalahan yang sangat minimal. Hampir seluruh citra pada masing-masing kelas—blackspot, canker, huanglongbing, dan healthy terklasifikasi dengan benar. Konsistensi ini menunjukkan bahwa karakteristik visual pada citra buah jeruk relatif lebih diskriminatif, sehingga memudahkan model dalam membedakan antar kelas penyakit. Variasi kecil antar run yang muncul bersifat marginal dan tidak memengaruhi kecenderungan performa secara keseluruhan. Berbeda dengan FDI, dataset LDI menunjukkan pola kesalahan yang sedikit lebih bervariasi, terutama pada kelas blackspot dan huanglongbing. Pada beberapa run, masih terlihat sejumlah kecil sampel yang tertukar antar kelas, yang mencerminkan kemiripan pola tekstur dan warna pada citra daun jeruk. Meskipun demikian, tren umum pada seluruh confusion matrix menunjukkan bahwa kesalahan tersebut bersifat terbatas dan tidak sistematis. Seiring bertambahnya run, model tetap mampu mempertahankan tingkat klasifikasi yang baik, yang menandakan kemampuan adaptasi DenseNet-169 terhadap kompleksitas visual citra daun. Sementara itu, dataset MDI secara konsisten menunjukkan performa paling stabil di seluruh run. Jumlah prediksi benar pada masing-masing kelas relatif tinggi dengan tingkat kesalahan yang sangat rendah dan tersebar. Hal ini mengindikasikan bahwa penggabungan citra daun dan buah jeruk memberikan representasi fitur yang lebih kaya dan saling melengkapi, sehingga membantu model membentuk batas keputusan yang lebih robust. Pola ini muncul secara konsisten dari run pertama hingga run kelima, menegaskan keunggulan dataset MDI dalam mendukung generalisasi model. Secara keseluruhan, rangkaian confusion matrix dari Run 1 hingga Run 5 memperlihatkan bahwa DenseNet-169

tanpa kombinasi fitur memiliki kemampuan klasifikasi yang andal dan konsisten, dengan performa terbaik dan paling stabil ditunjukkan pada dataset MDI, diikuti oleh FDI, dan kemudian LDI. Temuan ini menjadi landasan yang kuat untuk melanjutkan pembahasan ke tabel hasil akurasi per kelas, yang akan memberikan ringkasan kuantitatif terhadap pola performa yang telah diamati secara visual melalui confusion matrix.

Tabel 4.4. Hasil Akurasi setiap Kelas (DenseNet-169)

No	Dataset	Run	Blackspot %	Canker %	Huang-longbing %	Healthy %
1	Fruit Disease Image	1	100	98.65	100	100
2	Leaf Disease Image		95.16	100	96.43	100
3	Mixed Disease Image		97.86	100	98.59	100
4	Fruit Disease Image	2	97.22	100	100	100
5	Leaf Disease Image		91.67	100	100	100
6	Mixed Disease Image		98.48	99.38	99.33	100
7	Fruit Disease Image	3	100	96.00	100	100
8	Leaf Disease Image		97.40	98.25	97.53	100
9	Mixed Disease Image		97.30	99.26	99.39	100
10	Fruit Disease Image	4	98.82	100	100	100
11	Leaf Disease Image		95.00	100	100	100
12	Mixed Disease Image		96.00	100	98.66	100
13	Fruit Disease Image	5	100	98.82	100	100
14	Leaf Disease Image		94.94	100	94.94	100
15	Mixed Disease Image		97.16	99.24	99.32	100

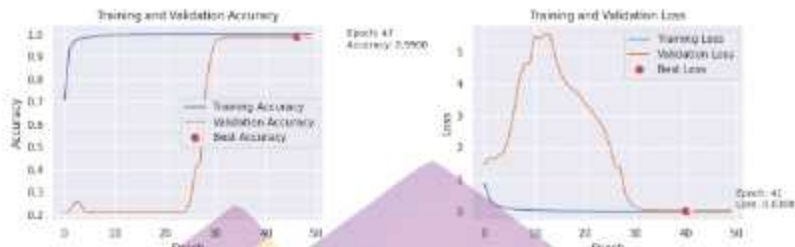
Berdasarkan Tabel 4.4 diatas, terlihat bahwa model DenseNet-169 tanpa kombinasi fitur mampu memberikan performa klasifikasi yang sangat baik dan relatif konsisten pada seluruh kelas penyakit jeruk di lima kali running, baik pada dataset FDI, LDI, maupun MDI. Pada dataset FDI, hampir seluruh kelas menunjukkan akurasi yang sangat tinggi dan stabil di setiap run. Kelas Blackspot, Huanglongbing, dan Healthy secara umum mencapai akurasi 100% pada sebagian besar running, sedangkan kelas Canker sedikit lebih bervariasi namun tetap berada pada rentang tinggi ($\pm 96\% - 100\%$). Pola ini mengindikasikan bahwa karakteristik visual citra buah jeruk cukup jelas dan mudah dibedakan oleh model, sehingga kesalahan klasifikasi dapat diminimalkan. Berbeda dengan FDI, dataset LDI menunjukkan variasi akurasi yang lebih besar, khususnya pada kelas Blackspot dan Huanglongbing. Akurasi kelas Blackspot pada LDI berada pada kisaran 91,67% hingga 97,40%, sementara Huanglongbing berkisar antara 94,94% hingga 100%. Variasi ini mencerminkan tingkat kompleksitas visual pada citra daun jeruk, di mana gejala penyakit sering kali memiliki kemiripan pola dan tekstur. Meskipun demikian, kelas Healthy pada LDI secara konsisten mencapai 100% di seluruh run, menunjukkan bahwa model sangat efektif dalam membedakan daun sehat dari daun yang terinfeksi. Sementara itu, dataset MDI memperlihatkan performa yang paling stabil dan seimbang di seluruh kelas. Akurasi pada kelas Canker dan Healthy hampir selalu mencapai 100%, sedangkan kelas Blackspot dan Huanglongbing berada pada rentang tinggi ($\pm 96\% - 99\%$). Konsistensi ini menguatkan temuan pada analisis confusion matrix sebelumnya, bahwa penggabungan citra daun dan buah jeruk memberikan representasi fitur yang lebih kaya dan saling melengkapi.

sehingga membantu model membentuk batas keputusan yang lebih robust. Secara keseluruhan, analisis Tabel 4.4 menunjukkan bahwa DenseNet-169 tanpa kombinasi fitur memiliki kemampuan generalisasi yang sangat baik pada seluruh kelas penyakit jeruk. Performa terbaik dan paling stabil ditunjukkan pada dataset MDI, diikuti oleh FDI, sedangkan LDI memiliki tingkat variasi yang lebih besar akibat kompleksitas visual citra daun. Hasil ini selaras dengan analisis confusion matrix pada Run 1 hingga Run 5 dan menegaskan bahwa model mampu mempertahankan akurasi tinggi secara konsisten pada berbagai skenario data dan running.

b) Model DenseNet-169 (Kombinasi Fitur)

Bagian ini akan mengevaluasi kinerja dari model DenseNet-169 yang telah dikombinasikan dengan fitur-fitur lain yang telah diproses sebelumnya. Melalui pendekatan ini, evaluasi bertujuan untuk mengetahui sejauh mana integrasi fitur-fitur yang dihasilkan oleh DenseNet-169 dengan fitur-fitur lain dapat meningkatkan kemampuan model dalam mengklasifikasikan dan mendeteksi penyakit-penyakit spesifik pada tanaman jeruk. Untuk hasil pelatihan yang lebih jelas dapat dilihat pada Gambar 4.8 yang memvisualisasikan perubahan tingkat akurasi dan loss selama proses pelatihan dan validasi dari model pada dataset (a) FDI, (b) LDI, dan (c) MDI, yang dijalankan sebanyak 50 epoch.

1. Pelatihan dan Validasi FDI, LDI dan MDI (Run 1)

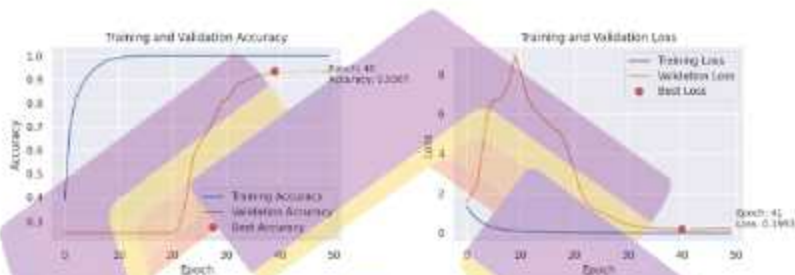


Gambar 4.27. Grafik Pelatihan dan Validasi FDI Run 1 (Dengan Kombinasi Fitur)

Gambar 4.27 menampilkan kurva akurasi dan loss pada proses pelatihan (training) dan validasi (validation) model DenseNet-169 dengan kombinasi fitur untuk dataset FDI (Fruit Disease Image) pada Run ke-1. Visualisasi ini digunakan untuk mengevaluasi pengaruh penambahan kombinasi fitur terhadap dinamika pembelajaran, kecepatan konvergensi, serta kemampuan generalisasi model, dan selanjutnya akan dibandingkan dengan hasil tanpa kombinasi fitur pada pembahasan sebelumnya. Pada grafik training accuracy, terlihat bahwa akurasi pelatihan meningkat sangat cepat sejak epoch awal dan mencapai nilai mendekati maksimum sebelum epoch ke-10. Pola ini menunjukkan bahwa integrasi kombinasi fitur tidak menghambat kemampuan DenseNet-169 dalam mempelajari pola visual dasar pada citra buah jeruk. Sebaliknya, model tetap mampu melakukan pembelajaran yang efisien dan stabil pada data latih. Akurasi pelatihan yang cepat mencapai plateau mengindikasikan bahwa representasi fitur yang digunakan cukup kuat untuk membedakan kelas penyakit pada dataset FDI. Berbeda dengan kurva

pelatihan, validation accuracy pada fase awal masih berada pada nilai rendah dan cenderung stagnan hingga sekitar epoch ke-25. Kondisi ini menunjukkan bahwa meskipun model cepat menyesuaikan diri dengan data latih, proses generalisasi terhadap data validasi memerlukan waktu yang lebih panjang. Setelah melewati epoch ke-30, terjadi peningkatan akurasi validasi yang sangat tajam hingga mencapai nilai terbaik sekitar 0,99 pada epoch ke-47. Peningkatan ini menandakan bahwa model mulai menemukan representasi fitur yang lebih general dan relevan terhadap data validasi. Dibandingkan dengan skenario tanpa kombinasi fitur, pola ini menunjukkan bahwa kombinasi fitur berkontribusi pada peningkatan performa akhir, meskipun membutuhkan fase adaptasi yang lebih panjang pada awal pelatihan. Pada grafik training loss, nilai loss menurun secara drastis sejak epoch awal dan mendekati nol pada epoch-epoch selanjutnya. Penurunan yang konsisten ini menunjukkan bahwa proses optimasi berjalan dengan baik dan kesalahan prediksi pada data latih semakin kecil. Sementara itu, validation loss menunjukkan nilai yang relatif tinggi dan sempat meningkat pada fase awal hingga pertengahan pelatihan, dengan puncak sekitar epoch ke-10 hingga ke-15. Pola ini mengindikasikan adanya kesenjangan antara pembelajaran pada data latih dan kemampuan generalisasi model pada tahap awal. Namun, setelah melewati epoch ke-25, validation loss menurun secara signifikan hingga mencapai nilai terendah sekitar 0,03 pada epoch ke-41, yang selaras dengan peningkatan akurasi validasi. Secara keseluruhan, hasil pada Gambar 4.27 menunjukkan bahwa penerapan kombinasi fitur pada model DenseNet-169 mampu menghasilkan

performa klasifikasi yang sangat baik pada dataset FDI. Meskipun pada fase awal pelatihan terdapat indikasi ketidakseimbangan antara akurasi pelatihan dan validasi, model berhasil mencapai kondisi konvergen dengan akurasi validasi yang tinggi dan loss yang rendah pada fase akhir.

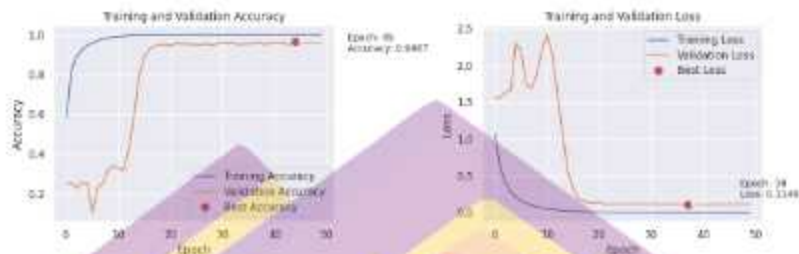


Gambar 4.28. Grafik Pelatihan dan Validasi LDI Run 1 (Dengan Kombinasi Fitur)

Gambar 4.28 menyajikan kurva akurasi dan loss pada proses pelatihan (training) dan validasi (validation) model DenseNet-169 dengan kombinasi fitur untuk dataset LDI (Leaf Disease Image) pada Run ke-1. Hasil ini melengkapi pembahasan sebelumnya pada dataset FDI dan digunakan untuk mengevaluasi pengaruh kombinasi fitur terhadap dinamika pembelajaran pada citra daun jeruk yang memiliki kompleksitas visual lebih tinggi. Pada grafik training accuracy, terlihat bahwa akurasi pelatihan meningkat sangat cepat sejak epoch awal dan mencapai nilai di atas 0,95 sebelum epoch ke-10. Pola ini menunjukkan bahwa DenseNet-169 tetap mampu mempelajari pola visual dasar pada citra daun jeruk secara efektif meskipun dikombinasikan dengan fitur tambahan. Hal ini konsisten dengan hasil pada dataset FDI dan mengindikasikan bahwa kombinasi fitur tidak mengganggu proses

pembelajaran pada data latih. Sebaliknya, pada grafik validation accuracy, akurasi validasi pada fase awal pelatihan masih berada pada nilai rendah dan cenderung stagnan hingga sekitar epoch ke-25. Setelah melewati fase tersebut, akurasi validasi meningkat secara bertahap dan mencapai nilai terbaik sekitar 0,9367 pada epoch ke-40. Peningkatan yang relatif lebih lambat dibandingkan dataset FDI mencerminkan tingkat kompleksitas visual citra daun jeruk, di mana gejala penyakit sering kali memiliki kemiripan tekstur dan warna antar kelas. Meskipun demikian, peningkatan akurasi yang stabil pada fase akhir menunjukkan bahwa kombinasi fitur membantu model membentuk representasi yang lebih diskriminatif setelah proses adaptasi berlangsung. Pada grafik training loss, nilai loss menurun tajam sejak epoch awal dan mendekati nol pada epoch-epoch selanjutnya. Hal ini menunjukkan bahwa proses optimasi berjalan dengan baik dan kesalahan prediksi pada data latih dapat diminimalkan secara efektif. Sementara itu, validation loss menunjukkan nilai yang cukup tinggi pada fase awal dan mencapai puncaknya sekitar epoch ke-10 hingga ke-15, sebelum kemudian menurun secara signifikan setelah epoch ke-25. Nilai validation loss terendah tercapai sekitar 0,1993 pada epoch ke-41, yang sejalan dengan pencapaian akurasi validasi terbaik. Secara keseluruhan, hasil pada Gambar 4.28 menunjukkan bahwa penerapan kombinasi fitur pada dataset LDI menghasilkan performa klasifikasi yang baik, meskipun proses konvergensi berlangsung lebih lambat dibandingkan dataset FDI. Hal ini menegaskan bahwa citra daun jeruk memiliki kompleksitas visual yang lebih

tinggi, namun DenseNet-169 tetap mampu melakukan generalisasi secara efektif setelah melewati fase pembelajaran awal.

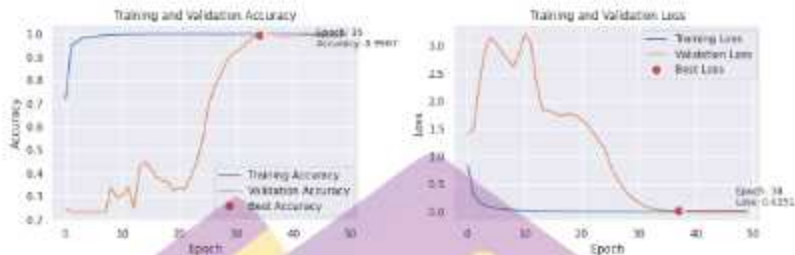


Gambar 4.29. Grafik Pelatihan dan Validasi MDI Run 1 (Dengan Kombinasi Fitur)

Gambar 4.29 menampilkan hasil pelatihan (training) dan validasi (validation) model DenseNet-169 dengan kombinasi fitur pada dataset MDI (Mixed Disease Image) Run ke-1. Dataset MDI merupakan gabungan citra daun dan buah jeruk, sehingga memiliki variasi visual yang lebih heterogen dan digunakan untuk mengevaluasi kemampuan generalisasi model pada kondisi data yang lebih kompleks. Pada grafik training accuracy, terlihat bahwa akurasi pelatihan meningkat sangat cepat sejak epoch awal dan mencapai nilai di atas 0,95 sebelum epoch ke-10. Pola ini konsisten dengan hasil pada dataset FDI dan LDI, yang menunjukkan bahwa DenseNet-169 tetap mampu mempelajari pola visual dasar secara efektif meskipun dataset memiliki karakteristik visual yang beragam dan menggunakan kombinasi fitur. Pada grafik validation accuracy, akurasi validasi pada fase awal pelatihan masih relatif rendah dan mengalami fluktuasi hingga sekitar epoch ke-15. Setelah melewati fase tersebut, akurasi validasi meningkat secara stabil dan mencapai nilai terbaik

sebesar 0,9667 pada epoch ke-45. Peningkatan yang lebih stabil ini menunjukkan bahwa kombinasi fitur membantu model dalam menyatukan informasi visual dari citra daun dan buah jeruk sehingga mampu membentuk representasi fitur yang lebih umum dan diskriminatif. Grafik training loss menunjukkan penurunan nilai loss yang tajam sejak epoch awal dan mendekati nol pada epoch-epoch selanjutnya, menandakan proses optimasi berjalan dengan baik. Sementara itu, validation loss sempat mengalami fluktuasi dan mencapai nilai cukup tinggi pada fase awal pelatihan, yang mengindikasikan kesulitan generalisasi pada tahap awal akibat heterogenitas data. Namun, setelah epoch ke-15, nilai validation loss menurun secara signifikan dan mencapai nilai terendah sekitar 0,1140 pada epoch ke-38, selaras dengan peningkatan akurasi validasi. Secara keseluruhan, hasil pada Gambar 4.29 menunjukkan bahwa penerapan kombinasi fitur pada dataset MDI menghasilkan proses pembelajaran yang relatif stabil dengan performa klasifikasi yang baik. Dibandingkan dengan dataset LDI, konvergensi pada dataset MDI cenderung lebih cepat dan lebih stabil, yang menegaskan bahwa penggabungan citra daun dan buah jeruk memberikan informasi visual yang saling melengkapi dan mendukung kemampuan generalisasi model. Temuan ini melengkapi analisis Run ke-1 dengan kombinasi fitur sebelum dilanjutkan ke pembahasan Run ke-2.

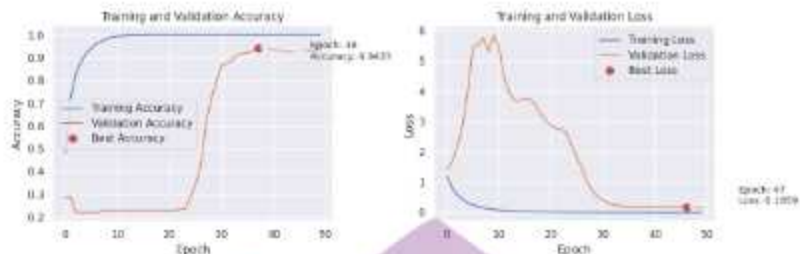
2. Pelatihan dan Validasi FDI, LDI dan MDI (Run 2)



Gambar 4.30. Grafik Pelatihan dan Validasi FDI Run 2 (Dengan Kombinasi Fitur)

Gambar 4.30 memperlihatkan hasil pelatihan dan validasi model DenseNet-169 dengan kombinasi fitur pada dataset FDI (Fruit Disease Image) Run ke-2 yang dijalankan selama 50 epoch. Grafik ini digunakan untuk mengevaluasi konsistensi performa model serta membandingkan dinamika pembelajaran dengan Run ke-1 pada skema kombinasi fitur. Pada grafik training accuracy, terlihat bahwa akurasi pelatihan kembali meningkat sangat cepat sejak epoch awal dan mencapai nilai di atas 0,95 sebelum epoch ke-10. Pola ini menunjukkan bahwa penambahan kombinasi fitur tidak menghambat kemampuan model dalam mempelajari pola visual dasar pada citra buah jeruk. Akurasi pelatihan kemudian stabil pada nilai mendekati maksimum hingga akhir pelatihan, yang menandakan proses optimasi berjalan dengan baik. Grafik validation accuracy menunjukkan bahwa pada fase awal, akurasi validasi masih relatif rendah dan mengalami fluktuasi hingga sekitar epoch ke-20. Setelah melewati fase tersebut, akurasi validasi meningkat secara signifikan dan stabil hingga mencapai nilai terbaik sebesar 0,9967 pada epoch ke-35.

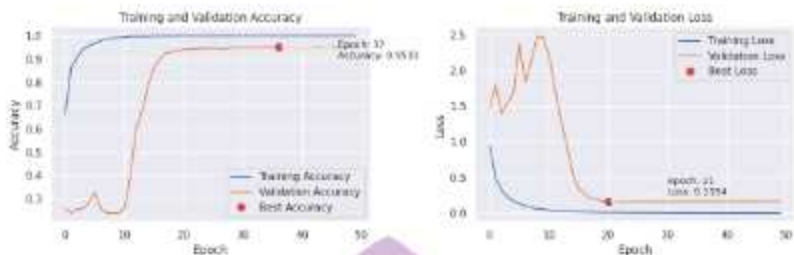
Peningkatan ini terjadi lebih cepat dibandingkan Run ke-1, yang mengindikasikan bahwa pada Run ke-2 kombinasi fitur membantu model mencapai kondisi generalisasi optimal dengan jumlah epoch yang lebih sedikit. Pada grafik training loss, nilai loss menurun secara tajam sejak epoch awal dan mendekati nol pada epoch-epoch selanjutnya. Hal ini menunjukkan bahwa kesalahan prediksi pada data latih dapat diminimalkan secara efektif. Sementara itu, validation loss sempat meningkat dan berfluktuasi pada fase awal hingga pertengahan pelatihan, mencerminkan proses penyesuaian model terhadap data validasi. Namun, setelah epoch ke-20, nilai validation loss menurun secara konsisten hingga mencapai nilai terendah sekitar 0,0251 pada epoch ke-38, selaras dengan peningkatan akurasi validasi. Secara keseluruhan, hasil pada Gambar 4.30 menunjukkan bahwa model DenseNet-169 dengan kombinasi fitur mampu memberikan performa yang sangat baik dan konsisten pada dataset FDI Run ke-2. Proses konvergensi berlangsung relatif cepat dengan akurasi validasi yang sangat tinggi dan nilai loss yang rendah, menegaskan bahwa citra buah jeruk tetap merupakan dataset yang paling mudah digeneralisasi oleh model. Hasil ini juga memperkuat temuan pada Run ke-1 bahwa penerapan kombinasi fitur tidak menurunkan performa, bahkan cenderung mempercepat pencapaian kondisi generalisasi yang optimal.



Gambar 4.31. Grafik Pelatihan dan Validasi LDI Run 2 (Dengan Kombinasi Fitur)

Pada Gambar 4.31, ditunjukkan kurva akurasi dan loss hasil pelatihan serta validasi model DenseNet-169 dengan kombinasi fitur pada dataset LDI (Leaf Disease Image) Run ke-2 selama 50 epoch. Berdasarkan kurva training accuracy, terlihat bahwa akurasi pelatihan meningkat dengan sangat cepat sejak epoch awal dan mencapai nilai di atas 0,95 sebelum epoch ke-10. Pola ini menunjukkan bahwa model tetap mampu mempelajari karakteristik dasar citra daun secara efektif meskipun data memiliki variasi tekstur dan pola gejala penyakit yang lebih beragam. Akurasi pelatihan kemudian stabil hingga akhir epoch, menandakan proses optimasi berjalan secara konsisten. Berbeda dengan akurasi pelatihan, validation accuracy pada fase awal masih berada pada nilai yang relatif rendah dan cenderung stagnan hingga sekitar epoch ke-25. Setelah melewati fase tersebut, akurasi validasi meningkat secara tajam dan mencapai nilai terbaik sebesar 0,9433 pada epoch ke-38. Pola ini mengindikasikan bahwa model memerlukan waktu pelatihan yang lebih panjang untuk menemukan representasi fitur yang mampu digeneralisasi dengan baik pada dataset LDI. Dibandingkan dengan dataset FDI, proses konvergensi pada LDI terlihat lebih

lambat, yang mencerminkan tingkat kesulitan klasifikasi citra daun jeruk. Pada grafik training loss, nilai loss menurun drastis sejak epoch awal dan mendekati nol pada epoch-epoch berikutnya, menunjukkan bahwa kesalahan prediksi pada data latih dapat diminimalkan secara efektif. Sementara itu, validation loss mengalami peningkatan dan fluktuasi cukup besar pada fase awal hingga pertengahan pelatihan, bahkan mencapai nilai yang relatif tinggi. Kondisi ini menunjukkan adanya kesulitan generalisasi pada tahap awal. Namun, setelah sekitar epoch ke-25, validation loss menurun secara konsisten hingga mencapai nilai terendah sekitar 0,1859 pada epoch ke-47, selaras dengan peningkatan akurasi validasi. Secara keseluruhan, hasil pada Gambar 4.31 menunjukkan bahwa penerapan kombinasi fitur pada dataset LDI Run ke-2 mampu menghasilkan performa klasifikasi yang cukup baik, meskipun dinamika pembelajaran masih lebih fluktuatif dibandingkan dataset FDI. Model membutuhkan jumlah epoch yang lebih banyak untuk mencapai kondisi generalisasi optimal, namun tetap mampu menunjukkan kestabilan performa pada fase akhir pelatihan. Temuan ini menegaskan bahwa citra daun jeruk memiliki kompleksitas visual yang lebih tinggi, sehingga menuntut proses pembelajaran yang lebih adaptif meskipun telah menggunakan kombinasi fitur.

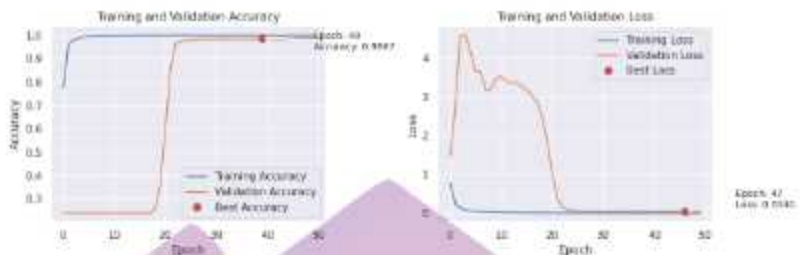


Gambar 4.32. Grafik Pelatihan dan Validasi MDI Run 2 (Dengan Kombinasi Fitur)

Pada Gambar 4.32 ditunjukkan hasil pelatihan dan validasi model DenseNet-169 dengan kombinasi fitur pada dataset MDI (Mixed Disease Image) Run ke-2 selama 50 epoch. Visualisasi ini memberikan gambaran mengenai kemampuan model dalam menangani data dengan karakteristik visual yang lebih beragam, karena dataset MDI merupakan gabungan citra daun dan buah jeruk. Kurva training accuracy memperlihatkan peningkatan yang sangat cepat sejak epoch awal dan telah mencapai nilai di atas 0,95 sebelum epoch ke-10. Pola ini menunjukkan bahwa penambahan kombinasi fitur tetap memungkinkan model mempelajari representasi visual dasar secara efisien, meskipun dataset memiliki variasi pola dan tekstur yang lebih kompleks. Setelah fase awal, akurasi pelatihan cenderung stabil hingga akhir proses pelatihan, menandakan bahwa model telah mencapai kondisi pembelajaran yang konvergen. Pada sisi validation accuracy, terlihat bahwa akurasi validasi meningkat secara signifikan setelah melewati epoch awal dan mencapai nilai terbaik sebesar 0,9533 pada epoch ke-37. Peningkatan yang relatif stabil ini menunjukkan bahwa model mampu melakukan generalisasi dengan baik pada

data validasi MDI. Jika dibandingkan dengan dataset LDI pada run yang sama, proses konvergensi pada dataset MDI terlihat lebih cepat dan lebih konsisten, yang mengindikasikan bahwa kombinasi citra daun dan buah jeruk memberikan informasi visual yang saling melengkapi. Grafik training loss menunjukkan penurunan yang tajam sejak epoch awal hingga mendekati nol, menandakan bahwa kesalahan prediksi pada data latih dapat ditekan secara efektif. Sementara itu, validation loss sempat mengalami fluktuasi pada fase awal pelatihan, namun kemudian menurun secara signifikan dan mencapai nilai terendah sebesar 0,1594 pada epoch ke-21. Penurunan loss validasi yang lebih cepat dibandingkan LDI menunjukkan bahwa model lebih cepat mencapai keseimbangan antara akurasi dan kesalahan prediksi pada dataset MDI. Secara keseluruhan, hasil pada Gambar 4.32 menegaskan bahwa penggunaan kombinasi fitur pada dataset MDI Run ke-2 mampu menghasilkan performa yang stabil dengan tingkat generalisasi yang baik. Dibandingkan dengan dataset LDI, model menunjukkan konvergensi yang lebih cepat dan fluktuasi yang lebih kecil, sehingga memperkuat temuan bahwa dataset MDI merupakan skenario yang lebih menguntungkan bagi model DenseNet-169 dalam membangun representasi fitur yang robust.

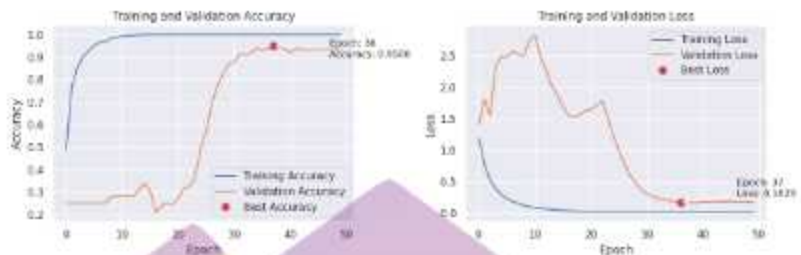
3. Pelatihan dan Validasi FDI, LDI dan MDI (Run 3)



Gambar 4.33. Grafik Pelatihan dan Validasi FDI Run 3 (Dengan Kombinasi Fitur)

Pada Gambar 4.33 disajikan hasil pelatihan dan validasi model DenseNet-169 dengan kombinasi fitur pada dataset FDI (Fruit Disease Image) Run ke-3 selama 50 epoch. Visualisasi ini digunakan untuk meninjau kembali konsistensi kinerja model pada citra buah jeruk ketika dikombinasikan dengan fitur tambahan, sekaligus membandingkannya dengan run sebelumnya. Kurva training accuracy menunjukkan peningkatan yang sangat cepat sejak epoch awal dan telah mencapai nilai mendekati maksimum sebelum epoch ke-10. Pola ini menegaskan bahwa penambahan kombinasi fitur tidak menghambat kemampuan model dalam mempelajari karakteristik visual dasar pada citra buah jeruk. Setelah fase awal, akurasi pelatihan cenderung stabil hingga akhir epoch, yang mengindikasikan bahwa proses pembelajaran telah mencapai kondisi konvergen. Pada validation accuracy, terlihat bahwa akurasi validasi meningkat secara signifikan setelah melewati fase awal pelatihan dan mencapai nilai terbaik sebesar 0,9867 pada epoch ke-40. Peningkatan yang tajam dan relatif stabil ini menunjukkan bahwa model mampu melakukan generalisasi dengan sangat baik pada data validasi FDI. Jika dibandingkan dengan run

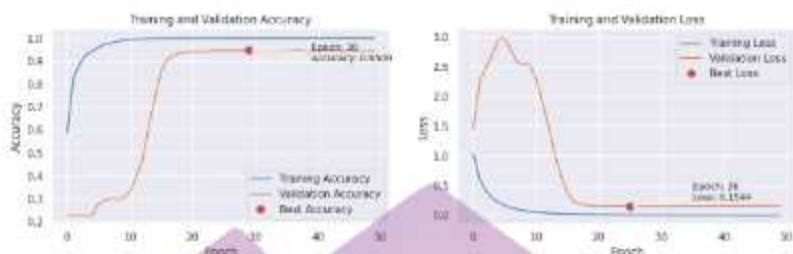
sebelumnya pada dataset yang sama, hasil ini menunjukkan konsistensi performa yang tinggi dengan selisih nilai akurasi yang relatif kecil. Grafik training loss memperlihatkan penurunan yang drastis sejak epoch awal hingga mendekati nol, menandakan bahwa kesalahan prediksi pada data latih dapat diminimalkan secara efektif. Sementara itu, validation loss sempat mengalami fluktuasi pada fase awal hingga pertengahan pelatihan, kemudian menurun secara signifikan dan mencapai nilai terendah sebesar 0,0340 pada epoch ke-47. Pola ini menunjukkan bahwa meskipun terdapat dinamika pembelajaran pada tahap awal, model akhirnya mampu mencapai keseimbangan yang baik antara akurasi dan kesalahan prediksi. Secara keseluruhan, hasil pada Gambar 4.33 menunjukkan bahwa model DenseNet-169 dengan kombinasi fitur tetap memberikan performa yang sangat baik dan stabil pada dataset FDI Run ke-3. Konvergensi yang cepat, akurasi validasi yang tinggi, serta nilai loss yang rendah menegaskan bahwa citra buah jeruk memiliki karakteristik visual yang mudah dipelajari oleh model, bahkan ketika fitur tambahan turut digunakan dalam proses pelatihan.



Gambar 4.34. Grafik Pelatihan dan Validasi LDI Run 3 (Dengan Kombinasi Fitur)

Pada Gambar 4.34 ditunjukkan hasil pelatihan dan validasi model DenseNet-169 dengan kombinasi fitur pada dataset LDI (Leaf Disease Image) Run ke-3 yang dilakukan selama 50 epoch. Visualisasi ini memberikan gambaran mengenai dinamika pembelajaran model pada citra daun jeruk, yang secara karakteristik memiliki kompleksitas visual lebih tinggi dibandingkan citra buah. Kurva training accuracy memperlihatkan peningkatan yang sangat cepat sejak epoch awal dan mencapai nilai di atas 0,95 sebelum epoch ke-10. Setelah fase tersebut, akurasi pelatihan cenderung stabil hingga akhir proses pelatihan, yang menunjukkan bahwa model mampu mempelajari pola visual pada data latih secara efektif meskipun citra daun memiliki variasi tekstur dan warna yang lebih kompleks. Berbeda dengan kurva pelatihan, validation accuracy menunjukkan pola peningkatan yang lebih bertahap. Pada fase awal hingga sekitar epoch ke-20, akurasi validasi masih relatif rendah dan disertai fluktuasi. Namun, setelah melewati fase tersebut, akurasi validasi meningkat secara signifikan dan mencapai nilai terbaik sebesar 0,9500 pada epoch ke-38. Pola ini mengindikasikan bahwa model memerlukan jumlah epoch yang lebih

banyak untuk membentuk representasi fitur yang mampu digeneralisasi dengan baik pada dataset LDI. Pada grafik training loss, terlihat penurunan nilai loss yang tajam sejak epoch awal hingga mendekati nol, menandakan proses optimasi berjalan secara stabil pada data latih. Sementara itu, validation loss mengalami fluktuasi yang cukup besar pada fase awal dan pertengahan pelatihan, sebelum akhirnya menurun secara signifikan dan mencapai nilai terendah sebesar 0,1629 pada epoch ke-37. Nilai loss validasi yang masih relatif lebih tinggi dibandingkan dataset FDI menunjukkan bahwa tingkat kesulitan klasifikasi pada citra daun jeruk tetap lebih besar, meskipun telah menggunakan kombinasi fitur. Secara keseluruhan, hasil pada Gambar 4.34 menunjukkan bahwa penggunaan kombinasi fitur pada DenseNet-169 mampu meningkatkan kestabilan pembelajaran pada dataset LDI Run ke-3, meskipun proses generalisasi berlangsung lebih lambat dibandingkan dataset FDI. Model tetap mampu mencapai performa klasifikasi yang baik dengan akurasi validasi yang tinggi, menegaskan bahwa pendekatan ini efektif dalam menangani kompleksitas visual pada citra daun jeruk.

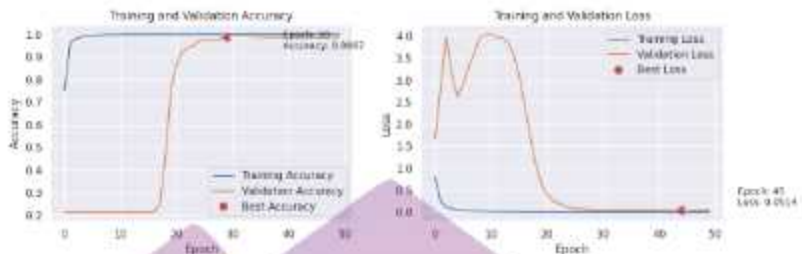


Gambar 4.35. Grafik Pelatihan dan Validasi MDI Run 3
(Dengan Kombinasi Fitur)

Gambar 4.35 memperlihatkan hasil pelatihan dan validasi model DenseNet-169 dengan kombinasi fitur pada dataset MDI (Mixed Disease Image) Run ke-3 selama 50 epoch. Dataset MDI yang merupakan gabungan citra daun dan buah jeruk memberikan variasi visual yang lebih heterogen, sehingga menjadi indikator penting untuk menilai kemampuan generalisasi model pada data yang lebih beragam. Kurva training accuracy menunjukkan peningkatan yang sangat cepat sejak epoch awal dan mencapai nilai di atas 0,95 sebelum epoch ke-10. Setelah itu, akurasi pelatihan cenderung stabil hingga akhir proses pelatihan, menandakan bahwa model mampu mempelajari pola visual dasar dari dataset MDI secara efektif dan konsisten. Pada sisi validation accuracy, terlihat bahwa akurasi validasi pada fase awal masih relatif rendah dan mulai meningkat secara signifikan setelah melewati sekitar epoch ke-15. Nilai akurasi validasi terbaik dicapai sebesar 0,9500 pada epoch ke-30, yang menunjukkan bahwa model mampu mencapai kondisi generalisasi yang baik dalam jumlah epoch yang relatif lebih sedikit dibandingkan dataset LDI pada run yang sama. Pola ini mengindikasikan bahwa kombinasi citra daun dan buah jeruk memberikan

informasi visual yang saling melengkapi, sehingga membantu model membentuk representasi fitur yang lebih diskriminatif. Grafik training loss memperlihatkan penurunan nilai loss yang tajam sejak awal pelatihan hingga mendekati nol, mencerminkan proses optimasi yang stabil pada data latih. Sementara itu, validation loss mengalami fluktuasi pada fase awal dan mencapai puncaknya sebelum kemudian menurun secara signifikan. Nilai loss validasi terendah tercatat sebesar 0,1544 pada epoch ke-26, yang selaras dengan peningkatan akurasi validasi pada fase pertengahan pelatihan. Secara keseluruhan, hasil pada Gambar 4.35 menunjukkan bahwa penerapan kombinasi fitur pada DenseNet-169 memberikan dampak positif terhadap proses pembelajaran pada dataset MDI Run ke-3. Model tidak hanya mencapai akurasi validasi yang tinggi, tetapi juga menunjukkan proses konvergensi yang relatif cepat dan stabil, menegaskan bahwa dataset MDI tetap menjadi skenario dengan kemampuan generalisasi terbaik dibandingkan dataset FDI dan LDI pada konfigurasi pelatihan ini.

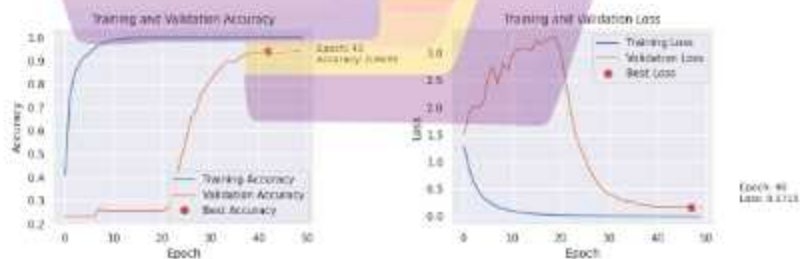
4. Pelatihan dan Validasi FDI, LDI dan MDI (Run 4)



Gambar 4.36. Grafik Pelatihan dan Validasi FDI Run 4 (Dengan Kombinasi Fitur)

Gambar 4.36 menampilkan kinerja pelatihan dan validasi model DenseNet-169 dengan kombinasi fitur pada dataset FDI (Fruit Disease Image) Run ke-4 yang dijalankan selama 50 epoch. Visualisasi ini digunakan untuk meninjau konsistensi proses pembelajaran model serta membandingkan pola konvergensi dengan run sebelumnya pada dataset yang sama. Pada kurva training accuracy, terlihat bahwa akurasi pelatihan meningkat sangat cepat sejak epoch awal dan mencapai nilai mendekati maksimum sebelum epoch ke-10. Setelah fase tersebut, akurasi cenderung stabil hingga akhir pelatihan, menunjukkan bahwa model mampu mempelajari pola visual citra buah jeruk secara efektif dan konsisten. Pola ini selaras dengan karakteristik dataset FDI yang memiliki perbedaan visual antarkelas yang relatif jelas. Sementara itu, validation accuracy pada fase awal masih berada pada nilai rendah dan mulai meningkat secara tajam setelah melewati sekitar epoch ke-15. Nilai akurasi validasi terbaik tercapai sebesar 0,9867 pada epoch ke-30, yang menunjukkan bahwa model mampu mencapai generalisasi yang sangat baik dengan jumlah epoch yang relatif moderat. Tidak terlihat adanya penurunan akurasi yang

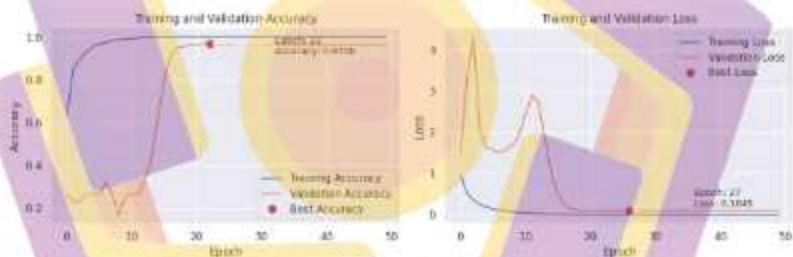
signifikan pada fase akhir pelatihan, sehingga indikasi overfitting dapat diminimalkan. Grafik training loss memperlihatkan penurunan nilai loss yang sangat tajam sejak awal pelatihan hingga mendekati nol, mencerminkan proses optimasi yang berjalan stabil pada data latih. Di sisi lain, validation loss sempat meningkat pada fase awal dan mencapai puncak pada epoch awal hingga pertengahan, kemudian menurun secara signifikan seiring meningkatnya akurasi validasi. Nilai loss validasi terendah dicapai sebesar 0,0514 pada epoch ke-45, yang menandakan bahwa kesalahan prediksi pada data validasi berada pada tingkat yang rendah. Secara keseluruhan, hasil pada Gambar 4.36 menunjukkan bahwa penerapan kombinasi fitur pada DenseNet-169 mampu mempertahankan performa yang sangat baik pada dataset FDI Run ke-4. Model menunjukkan konvergensi yang cepat, akurasi validasi yang tinggi, serta nilai loss yang rendah, sehingga menegaskan bahwa kombinasi fitur tidak menurunkan stabilitas pembelajaran, melainkan mendukung kemampuan generalisasi model pada citra buah jeruk.



Gambar 4.37. Grafik Pelatihan dan Validasi LDI Run 4 (Dengan Kombinasi Fitur)

Gambar 4.37 menyajikan hasil pelatihan dan validasi model DenseNet-169 dengan kombinasi fitur pada dataset LDI (Leaf Disease Image) Run ke-4 selama 50 epoch. Grafik ini digunakan untuk mengamati dinamika pembelajaran model pada citra daun jeruk serta mengevaluasi kestabilan proses generalisasi pada pengujian berulang. Pada kurva training accuracy, akurasi meningkat sangat cepat sejak epoch awal dan mencapai nilai mendekati maksimum sebelum epoch ke-10. Setelah fase tersebut, akurasi pelatihan cenderung stabil hingga akhir pelatihan, yang menunjukkan bahwa model mampu mempelajari pola visual dasar pada citra daun secara efektif. Pola ini konsisten dengan run sebelumnya dan menegaskan bahwa penambahan kombinasi fitur tidak menghambat proses pembelajaran pada data latih. Sebaliknya, validation accuracy pada fase awal masih berada pada tingkat rendah dan meningkat secara bertahap hingga sekitar epoch ke-20. Setelah itu, terjadi peningkatan yang lebih tajam dan stabil hingga mencapai nilai akurasi terbaik sebesar 0,9433 pada epoch ke-43. Peningkatan yang relatif lebih lambat dibandingkan dataset FDI menunjukkan bahwa citra daun jeruk memiliki kompleksitas visual yang lebih tinggi, sehingga model memerlukan waktu pelatihan yang lebih panjang untuk membentuk representasi fitur yang dapat digeneralisasi dengan baik. Pada grafik training loss, nilai loss menurun secara signifikan sejak epoch awal dan mendekati nol pada fase akhir pelatihan, menandakan bahwa proses optimasi berjalan stabil. Sementara itu, validation loss menunjukkan kenaikan pada fase awal dan mencapai puncak pada pertengahan pelatihan, kemudian menurun secara konsisten seiring

meningkatnya akurasi validasi. Nilai loss validasi terendah tercatat sebesar 0,1715 pada epoch ke-48, yang menunjukkan bahwa tingkat kesalahan prediksi pada data validasi berhasil ditekan meskipun kompleksitas dataset relatif tinggi. Secara keseluruhan, hasil pada Gambar 4.37 menunjukkan bahwa pada LDI Run ke-4, model DenseNet-169 dengan kombinasi fitur tetap mampu mencapai performa klasifikasi yang baik dan stabil. Meskipun proses konvergensi berlangsung lebih lambat dibandingkan dataset FDI, model tetap menunjukkan kemampuan generalisasi yang memadai pada citra daun jeruk, tanpa indikasi overfitting yang signifikan pada fase akhir pelatihan.



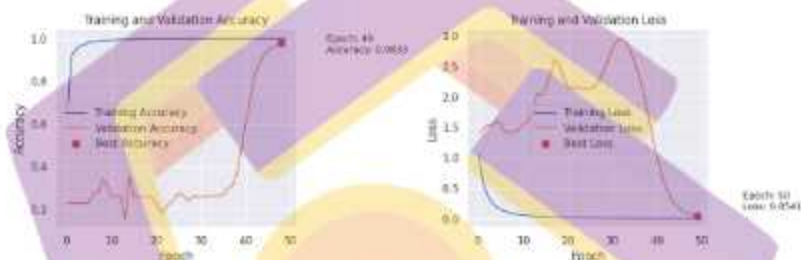
Gambar 4.38. Grafik Pelatihan dan Validasi MDI Run 4 (Dengan Kombinasi Fitur)

Gambar 4.38 menampilkan grafik pelatihan dan validasi model DenseNet-169 dengan kombinasi fitur pada dataset MDI (Mixed Disease Image) Run ke-4 yang dijalankan selama 50 epoch. Visualisasi ini digunakan untuk mengevaluasi efektivitas model dalam menangani data dengan karakteristik visual yang lebih heterogen, yaitu gabungan citra daun dan buah jeruk. Berdasarkan kurva training accuracy, terlihat bahwa akurasi pelatihan meningkat sangat cepat sejak epoch awal dan mencapai nilai mendekati

maksimum sebelum epoch ke-10. Setelah itu, akurasi cenderung stabil hingga akhir pelatihan, yang menunjukkan bahwa model mampu mempelajari pola visual dasar dari dataset MDI secara efisien. Pola ini konsisten dengan run sebelumnya dan menegaskan bahwa kombinasi fitur tidak mengganggu proses pembelajaran pada data latih. Pada sisi validation accuracy, akurasi validasi pada fase awal masih relatif rendah dan mengalami fluktuasi hingga sekitar epoch ke-15. Setelah melewati fase tersebut, terjadi peningkatan akurasi yang tajam dan stabil hingga mencapai nilai terbaik sebesar 0,9700 pada epoch ke-30. Proses konvergensi yang relatif cepat ini menunjukkan bahwa keberagaman informasi visual pada dataset MDI membantu model membentuk representasi fitur yang lebih diskriminatif dan mudah digeneralisasi dibandingkan dataset LDI. Grafik training loss memperlihatkan penurunan nilai loss yang sangat signifikan sejak epoch awal dan mendekati nol pada fase akhir pelatihan, menandakan bahwa proses optimasi berjalan dengan baik. Sementara itu, validation loss menunjukkan nilai yang cukup tinggi dan berfluktuasi pada fase awal, namun kemudian menurun secara konsisten setelah epoch ke-15. Nilai loss validasi terendah tercatat sebesar 0,1045 pada epoch ke-27, yang sejalan dengan peningkatan akurasi validasi dan menunjukkan penurunan kesalahan prediksi pada data validasi. Secara keseluruhan, hasil pada Gambar 4.38 menunjukkan bahwa pada MDI Run ke-4, model DenseNet-169 dengan kombinasi fitur mampu mencapai performa klasifikasi yang tinggi dengan proses konvergensi yang relatif cepat dan stabil. Dibandingkan dengan dataset LDI, dinamika pembelajaran pada dataset MDI

terlihat lebih terkendali, yang mengindikasikan bahwa penggabungan citra daun dan buah jeruk memberikan kontribusi positif terhadap kemampuan generalisasi model pada data yang lebih bervariasi.

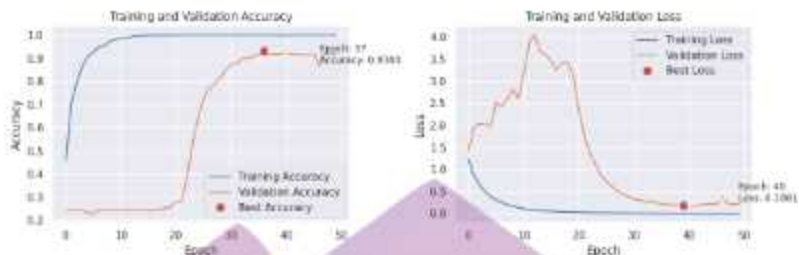
5. Pelatihan dan Validasi FDI, LDI dan MDI (Run 5)



Gambar 4.39. Grafik Pelatihan dan Validasi FDI Run 5 (Dengan Kombinasi Fitur)

Gambar 4.39 memperlihatkan hasil pelatihan dan validasi model DenseNet-169 dengan kombinasi fitur pada dataset FDI (Fruit Disease Image) Run ke-5 selama 50 epoch. Grafik ini digunakan untuk meninjau konsistensi performa model pada pengujian terakhir serta membandingkan pola pembelajaran dengan run sebelumnya pada dataset yang sama. Pada kurva training accuracy, terlihat bahwa akurasi pelatihan meningkat sangat cepat sejak epoch awal dan mencapai nilai mendekati maksimum sebelum epoch ke-10. Setelah fase tersebut, akurasi pelatihan relatif stabil hingga akhir pelatihan, yang menunjukkan bahwa model mampu mempelajari karakteristik visual citra buah jeruk secara efektif dan konsisten. Pola ini menegaskan kembali bahwa dataset FDI memiliki tingkat kompleksitas visual yang lebih rendah dibandingkan dataset LDI, sehingga proses pembelajaran pada

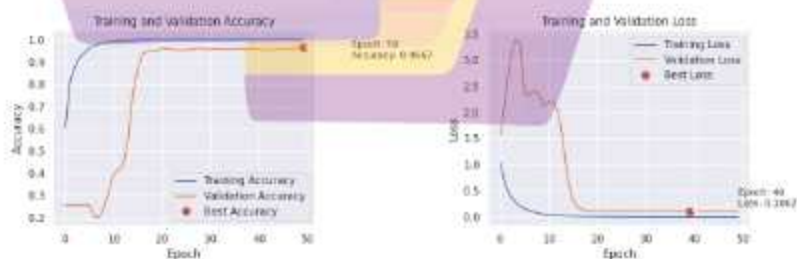
data latih berlangsung lebih cepat. Sebaliknya, validation accuracy menunjukkan dinamika yang berbeda. Pada fase awal hingga pertengahan pelatihan, akurasi validasi masih relatif rendah dan berfluktuasi hingga sekitar epoch ke-40. Setelah melewati fase tersebut, terjadi peningkatan akurasi yang sangat tajam hingga mencapai nilai terbaik sebesar 0,9833 pada epoch ke-49. Lonjakan ini mengindikasikan bahwa model memerlukan jumlah epoch yang lebih panjang pada run ke-5 untuk mencapai kondisi generalisasi yang optimal, meskipun pada akhirnya tetap mampu menghasilkan performa validasi yang sangat tinggi. Pada grafik training loss, nilai loss menurun secara drastis sejak epoch awal dan mendekati nol pada fase akhir pelatihan, yang menunjukkan bahwa proses optimasi berjalan stabil. Sementara itu, validation loss memperlihatkan fluktuasi dan peningkatan pada fase awal hingga pertengahan pelatihan, kemudian menurun secara signifikan setelah epoch ke-40. Nilai loss validasi terendah tercatat sebesar 0,0541 pada epoch ke-50, selaras dengan peningkatan tajam akurasi validasi pada fase akhir pelatihan. Secara keseluruhan, hasil pada Gambar 4.39 menunjukkan bahwa pada FDI Run ke-5, model DenseNet-169 dengan kombinasi fitur tetap mampu mencapai performa klasifikasi yang sangat baik. Meskipun proses generalisasi berlangsung lebih lambat dibandingkan beberapa run sebelumnya, model berhasil mencapai keseimbangan antara akurasi tinggi dan loss yang rendah pada akhir pelatihan. Temuan ini semakin menegaskan kestabilan dan robustness model dalam mengklasifikasikan penyakit jeruk berbasis citra buah pada berbagai skenario pelatihan.



Gambar 4.40. Grafik Pelatihan dan Validasi LDI Run 5
(Dengan Kombinasi Fitur)

Gambar 4.40 menyajikan hasil pelatihan dan validasi model DenseNet-169 dengan kombinasi fitur pada dataset LDI (Leaf Disease Image) Run ke-5 yang dijalankan selama 50 epoch. Visualisasi ini digunakan untuk mengkaji performa akhir model pada citra daun jeruk sekaligus mengevaluasi kestabilan proses generalisasi pada pengujian terakhir. Dari kurva training accuracy, tampak bahwa akurasi pelatihan meningkat sangat cepat sejak epoch awal dan mencapai nilai mendekati maksimum sebelum epoch ke-10. Setelah itu, akurasi pelatihan relatif stabil hingga akhir pelatihan, menunjukkan bahwa model mampu mempelajari pola visual pada data latih secara konsisten. Pola ini serupa dengan run sebelumnya dan menegaskan bahwa penambahan kombinasi fitur tidak menghambat proses pembelajaran pada dataset LDI. Pada sisi validation accuracy, akurasi validasi pada fase awal masih berada pada tingkat yang rendah dan relatif stagnan hingga sekitar epoch ke-20. Setelah melewati fase tersebut, akurasi validasi meningkat secara signifikan dan mencapai nilai terbaik sebesar 0,9300 pada epoch ke-37. Peningkatan yang bertahap ini kembali menunjukkan bahwa citra daun jeruk memiliki kompleksitas visual yang lebih tinggi, sehingga model memerlukan waktu pelatihan yang lebih

panjang untuk membentuk representasi fitur yang dapat digeneralisasi dengan baik. Grafik training loss memperlihatkan penurunan nilai loss yang tajam sejak epoch awal dan mendekati nol pada fase akhir pelatihan, menandakan bahwa proses optimasi berlangsung stabil. Sementara itu, validation loss menunjukkan peningkatan yang cukup besar pada fase awal hingga pertengahan pelatihan, kemudian menurun secara konsisten setelah epoch ke-20. Nilai loss validasi terendah tercatat sebesar 0,1881 pada epoch ke-40, yang sejalan dengan peningkatan akurasi validasi pada fase akhir pelatihan. Secara keseluruhan, hasil pada Gambar 4.40 menunjukkan bahwa pada LDI Run ke-5, model DenseNet-169 dengan kombinasi fitur tetap mampu mencapai performa klasifikasi yang baik meskipun proses konvergensi berlangsung lebih lambat dibandingkan dataset FDI dan MDI. Dinamika pembelajaran yang lebih fluktuatif pada dataset LDI menegaskan tingginya kompleksitas visual citra daun jeruk, namun model tetap menunjukkan kemampuan generalisasi yang memadai tanpa indikasi overfitting yang signifikan pada fase akhir pelatihan.



Gambar 4.41. Grafik Pelatihan dan Validasi MDI Run 5 (Dengan Kombinasi Fitur)

Gambar 4.41 menggambarkan hasil pelatihan dan validasi model DenseNet-169 dengan kombinasi fitur pada dataset MDI (Mixed Disease Image) Run ke-5 yang dijalankan selama 50 epoch. Grafik ini merepresentasikan performa model pada pengujian terakhir dengan data gabungan citra daun dan buah jeruk, sekaligus menjadi penutup rangkaian evaluasi berbasis kombinasi fitur. Pada kurva training accuracy, akurasi pelatihan meningkat sangat cepat sejak epoch awal dan mencapai nilai mendekati maksimum sebelum epoch ke-10. Setelah itu, akurasi cenderung stabil hingga akhir pelatihan, yang menunjukkan bahwa model mampu mempelajari karakteristik visual dasar dari dataset MDI secara efektif dan konsisten. Pola ini kembali menegaskan bahwa penggabungan fitur tidak menghambat proses pembelajaran, bahkan memberikan kestabilan pada data latih yang bersifat heterogen. Sementara itu, validation accuracy menunjukkan peningkatan bertahap pada fase awal hingga sekitar epoch ke-20, kemudian meningkat secara lebih tajam dan stabil hingga mencapai nilai terbaik sebesar 0,9667 pada epoch ke-50. Proses konvergensi yang relatif cepat dan stabil ini mengindikasikan bahwa kombinasi citra daun dan buah jeruk memberikan representasi fitur yang lebih kaya, sehingga mempermudah model dalam melakukan generalisasi dibandingkan dataset LDI. Pada grafik training loss, nilai loss menurun secara drastis sejak epoch awal dan mendekati nol pada fase akhir pelatihan, menandakan proses optimasi yang berjalan stabil. Di sisi lain, validation loss sempat mengalami fluktuasi dan peningkatan pada fase awal, namun kemudian menurun secara signifikan setelah epoch ke-15. Nilai loss validasi terendah tercatat sebesar 0,1062 pada

epoch ke-40, yang selaras dengan meningkatnya akurasi validasi dan menunjukkan rendahnya kesalahan prediksi pada data validasi. Secara keseluruhan, hasil pada Gambar 4.41 menunjukkan bahwa pada MDI Run ke-5, model DenseNet-169 dengan kombinasi fitur mampu mempertahankan performa klasifikasi yang tinggi dengan proses konvergensi yang stabil. Dibandingkan dataset FDI dan LDI pada run yang sama, dataset MDI kembali menunjukkan dinamika pembelajaran yang lebih terkendali, menegaskan bahwa penggabungan citra daun dan buah jeruk berkontribusi positif terhadap kemampuan generalisasi dan robustness model pada skenario pelatihan berulang.

Berdasarkan hasil pelatihan dan validasi model DenseNet-169 dengan kombinasi fitur yang dilakukan pada lima kali running, dapat disimpulkan bahwa model menunjukkan performa yang konsisten, stabil, dan robust pada ketiga dataset, yaitu FDI (Fruit Disease Image), LDI (Leaf Disease Image), dan MDI (Mixed Disease Image). Secara umum, pada seluruh run terlihat bahwa akurasi pelatihan meningkat sangat cepat sejak epoch awal dan mencapai nilai mendekati maksimum sebelum epoch ke-10. Pola ini menunjukkan bahwa integrasi kombinasi fitur tidak menghambat proses pembelajaran, melainkan tetap mendukung kemampuan model dalam mengekstraksi pola visual dasar secara efektif. Pada dataset FDI, seluruh run menunjukkan performa yang sangat tinggi dan stabil. Akurasi validasi secara konsisten berada pada rentang tinggi, disertai nilai validation loss yang rendah pada fase akhir pelatihan. Hal ini mengindikasikan

bahwa citra buah jeruk memiliki karakteristik visual yang relatif mudah dibedakan antar kelas, sehingga model mampu melakukan generalisasi dengan baik pada berbagai kondisi pelatihan. Berbeda dengan FDI, dataset LDI memperlihatkan dinamika pembelajaran yang lebih fluktuatif, khususnya pada fase awal hingga pertengahan pelatihan. Akurasi validasi pada beberapa run meningkat lebih lambat dan disertai nilai validation loss yang relatif lebih tinggi. Kondisi ini mencerminkan kompleksitas visual citra daun jeruk yang lebih tinggi, di mana kemiripan tekstur dan warna antar kelas penyakit menyebabkan proses generalisasi memerlukan waktu pelatihan yang lebih panjang. Meskipun demikian, setelah melewati fase awal tersebut, model tetap mampu mencapai performa klasifikasi yang baik dan stabil hingga akhir pelatihan. Sementara itu, dataset MDI secara konsisten menunjukkan proses pembelajaran yang lebih stabil dibandingkan LDI. Pada hampir seluruh run, konvergensi terjadi lebih cepat dan disertai nilai validation loss yang lebih rendah. Hal ini menunjukkan bahwa penggabungan citra daun dan buah jeruk menghasilkan representasi fitur yang lebih kaya dan saling melengkapi, sehingga membantu model dalam membangun fitur yang lebih diskriminatif dan meningkatkan kemampuan generalisasi.

Secara keseluruhan, hasil pelatihan dan validasi pada lima run menegaskan bahwa model DenseNet-169 dengan kombinasi fitur memiliki kemampuan adaptasi dan generalisasi yang sangat baik, terutama pada dataset MDI. Model mampu mempertahankan performa tinggi secara konsisten pada berbagai skenario pelatihan, yang menunjukkan bahwa pendekatan kombinasi fitur memberikan kontribusi positif terhadap kestabilan dan robustness model.

Ringkasan kuantitatif dari performa model pada masing-masing dataset dan run selanjutnya disajikan pada Tabel 4.5 sebagai matriks evaluasi akurasi, presisi, recall, dan F1-score.

Tabel 4.5. Hasil Matrik Evaluasi DenseNet-169 Kombinasi Fitur

No	Dataset	Run	Akurasi %	Presisi %	Recall %	F1 Score %
1	Fruit Disease Image	1	99.33	99.34	99.34	99.34
2	Leaf Disease Image		94.00	94.40	93.95	93.77
3	Mixed Disease Image		95.16	95.29	95.15	95.18
4	Fruit Disease Image	2	98.66	98.66	98.66	98.66
5	Leaf Disease Image		94.00	93.73	93.82	93.74
6	Mixed Disease Image		96.50	96.60	96.51	96.54
7	Fruit Disease Image	3	98.33	98.25	98.17	98.21
8	Leaf Disease Image		94.33	94.24	94.31	94.26
9	Mixed Disease Image		97.50	97.46	97.46	97.45
10	Fruit Disease Image	4	97.66	97.75	97.71	97.68
11	Leaf Disease Image		94.00	94.22	94.39	94.15
12	Mixed Disease Image		92.33	92.69	92.58	92.63
13	Fruit Disease Image	5	98.33	97.95	98.04	97.99
14	Leaf Disease Image		92.33	92.69	92.58	92.63
15	Mixed Disease Image		97.50	97.37	97.47	97.41

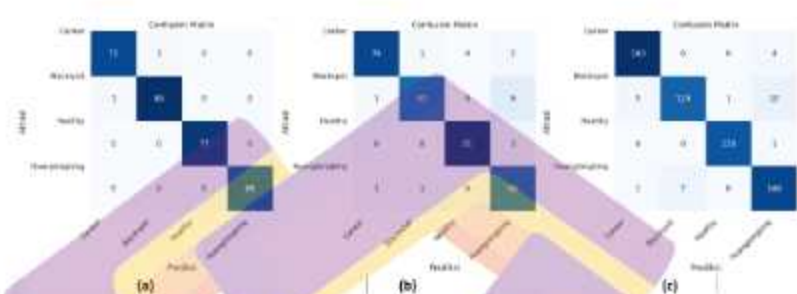
Berdasarkan Tabel 4.5, dapat dilihat bahwa model DenseNet-169 dengan kombinasi fitur menunjukkan performa yang konsisten dan stabil pada seluruh dataset dan lima kali running. Pada dataset FDI, nilai akurasi, presisi, recall, dan F1-score secara umum berada pada kisaran tinggi ($\pm 98-99\%$), yang menunjukkan bahwa penambahan kombinasi fitur tetap mampu mempertahankan kemampuan model dalam mengklasifikasikan penyakit pada citra buah jeruk secara akurat dan

konsisten antar run. Pada dataset LDI, performa model relatif lebih rendah dibandingkan FDI dan MDI, dengan nilai akurasi dan F1-score berada pada rentang $\pm 92-94\%$. Hal ini mencerminkan tingkat kompleksitas visual citra daun jeruk yang lebih tinggi, di mana kemiripan tekstur dan pola antar kelas penyakit masih menjadi tantangan utama bagi model. Meskipun demikian, nilai metrik evaluasi yang relatif stabil pada setiap run menunjukkan bahwa model tetap mampu melakukan generalisasi secara konsisten setelah melewati fase pembelajaran awal. Sementara itu, dataset MDI menunjukkan performa yang cukup seimbang dan stabil, dengan nilai akurasi dan F1-score umumnya berada di atas 95%. Hasil ini mengindikasikan bahwa penggabungan citra daun dan buah jeruk menghasilkan representasi fitur yang lebih kaya dan saling melengkapi, sehingga membantu model dalam membedakan kelas penyakit secara lebih efektif dibandingkan dataset LDI saja. Secara keseluruhan, hasil matriks evaluasi pada Tabel 4.5 menegaskan bahwa model DenseNet-169 dengan kombinasi fitur memiliki kemampuan generalisasi yang baik pada berbagai skenario pelatihan. Untuk memahami lebih rinci pola kesalahan klasifikasi dan distribusi prediksi pada setiap kelas penyakit, pembahasan selanjutnya akan difokuskan pada analisis confusion matrix pada masing-masing dataset dan run. Berdasarkan Tabel 4.5, dapat dilihat bahwa model DenseNet-169 dengan kombinasi fitur menunjukkan performa yang konsisten dan stabil pada seluruh dataset dan lima kali running. Pada dataset FDI, nilai akurasi, presisi, recall, dan F1-score secara umum berada pada kisaran tinggi ($\pm 98-99\%$), yang menunjukkan bahwa penambahan kombinasi fitur tetap mampu mempertahankan kemampuan model dalam mengklasifikasikan penyakit pada citra buah jeruk secara

akurat dan konsisten antar run. Pada dataset LDI, performa model relatif lebih rendah dibandingkan FDI dan MDI, dengan nilai akurasi dan F1-score berada pada rentang $\pm 92-94\%$. Hal ini mencerminkan tingkat kompleksitas visual citra daun jeruk yang lebih tinggi, di mana kemiripan tekstur dan pola antar kelas penyakit masih menjadi tantangan utama bagi model. Meskipun demikian, nilai metrik evaluasi yang relatif stabil pada setiap run menunjukkan bahwa model tetap mampu melakukan generalisasi secara konsisten setelah melewati fase pembelajaran awal. Sementara itu, dataset MDI menunjukkan performa yang cukup seimbang dan stabil, dengan nilai akurasi dan F1-score umumnya berada di atas 95%. Hasil ini mengindikasikan bahwa penggabungan citra daun dan buah jeruk menghasilkan representasi fitur yang lebih kaya dan saling melengkapi, sehingga membantu model dalam membedakan kelas penyakit secara lebih efektif dibandingkan dataset LDI saja. Secara keseluruhan, hasil matriks evaluasi pada Tabel 4.5 menegaskan bahwa model DenseNet-169 dengan kombinasi fitur memiliki kemampuan generalisasi yang baik pada berbagai skenario pelatihan. Untuk memahami lebih rinci pola kesalahan klasifikasi dan distribusi prediksi pada setiap kelas penyakit, pembahasan selanjutnya akan difokuskan pada analisis confusion matrix pada masing-masing dataset dan run.

Berdasarkan hasil confusion matrix pada seluruh pengujian dengan kombinasi fitur, terlihat bahwa model DenseNet-169 secara konsisten mampu melakukan klasifikasi penyakit jeruk dengan tingkat ketepatan yang tinggi pada ketiga dataset, yaitu FDI, LDI, dan MDI. Analisis ini memberikan gambaran yang

lebih rinci mengenai distribusi prediksi benar dan kesalahan klasifikasi pada setiap kelas, sehingga melengkapi hasil evaluasi berbasis metrik kuantitatif sebelumnya.



Gambar 4.42. Hasil Confusion Matrix Run ke-1 Dengan Kombinasi Fitur (a) FDI (b) LDI (c) MDI

Gambar 4.42 menampilkan confusion matrix hasil klasifikasi model DenseNet-169 dengan kombinasi fitur pada Run ke-1 untuk dataset FDI, LDI, dan MDI. Pada dataset FDI, model menunjukkan performa yang sangat baik, dengan mayoritas prediksi berada pada diagonal utama. Kelas Canker terklasifikasi benar sebanyak 72 citra, Blackspot 80 citra, Healthy 77 citra, dan Huanglongbing 69 citra, tanpa kesalahan yang berarti antar kelas. Hal ini mengindikasikan bahwa kombinasi fitur tidak mengganggu kemampuan model dalam mengenali pola visual citra buah jeruk. Pada dataset LDI, tingkat klasifikasi benar masih dominan, namun mulai terlihat kesalahan prediksi antar kelas penyakit daun. Sebagai contoh, kelas Blackspot terklasifikasi benar sebanyak 63 citra, namun terdapat 9 citra yang salah diprediksi sebagai Healthy. Pola ini menunjukkan adanya kemiripan visual antar gejala penyakit daun, sehingga proses diskriminasi kelas menjadi lebih menantang meskipun telah menggunakan kombinasi fitur. Sementara itu, pada dataset MDI, model kembali menunjukkan performa yang stabil dan tinggi. Kelas Canker

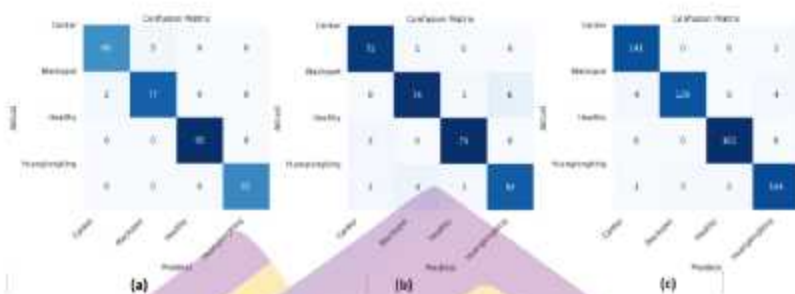
terklasifikasi benar sebanyak 160 citra, Blackspot 129 citra, Healthy 136 citra, dan Huanglongbing 145 citra, dengan jumlah kesalahan yang sangat minimal. Hasil ini menegaskan bahwa penggabungan citra daun dan buah jeruk menghasilkan representasi fitur yang lebih kaya, sehingga membantu model meningkatkan kemampuan generalisasi. Secara keseluruhan, confusion matrix pada Gambar 4.42 menunjukkan bahwa pada Run ke-1, model DenseNet-169 dengan kombinasi fitur mampu memberikan klasifikasi yang akurat dan konsisten, dengan performa terbaik ditunjukkan pada dataset FDI dan MDI, sementara dataset LDI masih menunjukkan tantangan akibat kompleksitas visual citra daun jeruk.



Gambar 4.43. Hasil Confusion Matrix Run ke-2 Dengan Kombinasi Fitur (a) FDI (b) LDI (c) MDI

Gambar 4.43 menyajikan confusion matrix hasil klasifikasi model DenseNet-169 dengan kombinasi fitur pada Run ke-2 untuk dataset FDI, LDI, dan MDI. Pada dataset FDI, performa model tetap sangat baik dengan dominasi prediksi benar pada diagonal utama. Kelas Canker terklasifikasi benar sebanyak 68 citra, Blackspot 78 citra, Healthy 73 citra, dan Huanglongbing 77 citra. Kesalahan prediksi sangat minimal, hanya terlihat 2 citra Canker yang salah diklasifikasikan sebagai Blackspot. Pola ini menunjukkan bahwa model mampu mempertahankan

konsistensi klasifikasi citra buah jeruk pada pengujian berulang. Untuk dataset LDI, tingkat kesalahan prediksi masih lebih terlihat dibandingkan FDI. Kelas Canker terklasifikasi benar sebanyak 85 citra, namun terdapat 1 citra yang salah diprediksi sebagai Blackspot. Pada kelas Blackspot, sebanyak 60 citra terklasifikasi benar, dengan 5 citra keliru diprediksi sebagai Healthy. Kelas Healthy dan Huanglongbing masing-masing terklasifikasi benar sebanyak 72 dan 65 citra, dengan kesalahan prediksi yang relatif kecil. Hasil ini kembali mencerminkan kompleksitas visual citra daun jeruk, di mana kemiripan tekstur dan gejala antar kelas masih memengaruhi akurasi model. Sementara itu, pada dataset MDI, model menunjukkan performa yang sangat stabil dan konsisten. Kelas Canker terklasifikasi benar sebanyak 149 citra, Blackspot 139 citra, Healthy 141 citra, dan Huanglongbing 150 citra. Kesalahan prediksi hanya muncul dalam jumlah yang sangat kecil, misalnya 9 citra Blackspot yang salah diklasifikasikan sebagai Healthy. Hasil ini menegaskan bahwa kombinasi citra daun dan buah jeruk memberikan representasi fitur yang lebih informatif, sehingga meningkatkan kemampuan generalisasi model. Secara keseluruhan, confusion matrix pada Run ke-2 menunjukkan bahwa model DenseNet-169 dengan kombinasi fitur mampu mempertahankan performa klasifikasi yang tinggi dan konsisten. Dataset FDI dan MDI kembali menunjukkan tingkat akurasi terbaik, sedangkan LDI masih menjadi dataset dengan tantangan terbesar akibat kompleksitas visual citra daun jeruk.



Gambar 4.44. Hasil Confusion Matrix Run ke-3 Dengan Kombinasi Fitur (a) FDI (b) LDI (c) MDI

Gambar 4.44 menampilkan confusion matrix hasil klasifikasi model DenseNet-169 dengan kombinasi fitur pada Run ke-3 untuk dataset FDI, LDI, dan MDI. Pada dataset FDI, mayoritas prediksi berada pada diagonal utama, menunjukkan tingkat klasifikasi yang sangat baik. Kelas Canker terklasifikasi benar sebanyak 60 citra, Blackspot 77 citra, Healthy 95 citra, dan Huanglongbing 63 citra. Tidak terlihat kesalahan klasifikasi yang signifikan antar kelas, yang mengindikasikan bahwa model mampu membedakan gejala penyakit pada citra buah jeruk secara konsisten dan stabil pada run ini. Pada dataset LDI, performa model masih menunjukkan pola yang baik meskipun terdapat beberapa kesalahan prediksi. Kelas Canker terklasifikasi benar sebanyak 72 citra, dengan 1 citra salah diprediksi sebagai Blackspot. Kelas Blackspot memiliki 74 citra terklasifikasi benar, dengan kesalahan ke kelas Healthy sebanyak *6 citra. Untuk kelas Healthy, terdapat 75 citra terklasifikasi benar dengan kesalahan yang sangat minimal, sedangkan Huanglongbing terklasifikasi benar sebanyak 62 citra dengan beberapa kesalahan kecil. Pola ini kembali menegaskan bahwa kompleksitas visual pada citra daun jeruk masih menjadi sumber utama kesalahan klasifikasi. Sementara itu, pada

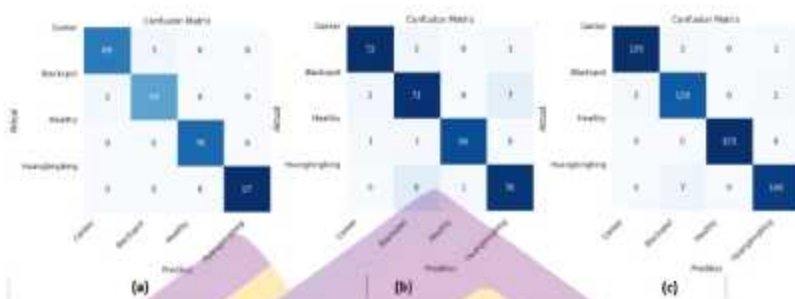
dataset MDI, model menunjukkan performa yang sangat stabil dan unggul. Kelas Canker terklasifikasi benar sebanyak 141 citra, Blackspot 139 citra, Healthy 161 citra, dan Huanglongbing 144 citra. Kesalahan prediksi hanya terjadi dalam jumlah yang sangat kecil dan tersebar, seperti 4 citra Blackspot yang salah diprediksi sebagai Healthy. Hasil ini menunjukkan bahwa penggabungan citra daun dan buah jeruk mampu memperkaya representasi fitur sehingga meningkatkan kemampuan generalisasi model. Secara keseluruhan, hasil confusion matrix pada Run ke-3 menunjukkan bahwa model DenseNet-169 dengan kombinasi fitur tetap mempertahankan performa klasifikasi yang tinggi dan konsisten. Dataset FDI dan MDI kembali memberikan hasil paling stabil, sementara LDI menunjukkan tingkat kesalahan yang sedikit lebih tinggi akibat karakteristik visual daun jeruk yang lebih kompleks.



Gambar 4.45. Hasil Confusion Matrix Run ke-4 Dengan Kombinasi Fitur (a) FDI (b) LDI (c) MDI

Gambar 4.45 menampilkan confusion matrix hasil klasifikasi model DenseNet-169 dengan kombinasi fitur pada Run ke-4 untuk dataset FDI, LDI, dan MDI. Pada dataset FDI, model menunjukkan performa yang sangat baik dengan dominasi nilai pada diagonal utama. Kelas Canker terklasifikasi benar sebanyak 73

citra, Blackspot 71 citra, Healthy 74 citra, dan Huanglongbing 75 citra. Kesalahan prediksi relatif kecil, seperti 6 citra Blackspot yang keliru diprediksi sebagai Canker. Secara umum, hasil ini menunjukkan bahwa model mampu mempertahankan kemampuan diskriminasi yang tinggi pada citra buah jeruk. Untuk dataset LDI, terlihat bahwa tingkat kesalahan prediksi masih lebih tinggi dibandingkan FDI. Kelas Canker terklasifikasi benar sebanyak 79 citra, dengan 1 citra salah diprediksi sebagai Blackspot. Kelas Blackspot memiliki 68 citra terklasifikasi benar, namun masih terdapat kesalahan ke kelas Healthy sebanyak *10 citra. Kelas Healthy dan Huanglongbing masing-masing terklasifikasi benar sebanyak 61 dan 74 citra, dengan kesalahan yang relatif kecil. Pola ini menunjukkan bahwa kemiripan visual antar kelas pada citra daun jeruk masih memengaruhi performa klasifikasi. Sementara itu, pada dataset MDI, model kembali menunjukkan performa yang sangat stabil. Kelas Canker terklasifikasi benar sebanyak 148 citra, Blackspot 152 citra, Healthy 137 citra, dan Huanglongbing 139 citra. Kesalahan prediksi hanya terjadi dalam jumlah terbatas, misalnya 10 citra Blackspot yang salah diklasifikasikan sebagai Healthy. Hasil ini menegaskan bahwa penggabungan citra daun dan buah jeruk memberikan representasi fitur yang lebih kaya dan membantu meningkatkan kemampuan generalisasi model. Secara keseluruhan, confusion matrix pada Run ke-4 menunjukkan bahwa model DenseNet-169 dengan kombinasi fitur mampu mempertahankan performa klasifikasi yang tinggi dan konsisten. Dataset MDI kembali memberikan hasil paling stabil, diikuti oleh FDI, sementara LDI masih menunjukkan tingkat kesulitan yang lebih tinggi akibat kompleksitas visual citra daun jeruk.



Gambar 4.46. Hasil Confusion Matrix Run ke-5 Dengan Kombinasi Fitur (a) FDI (b) LDI (c) MDI

Gambar 4.46 menampilkan confusion matrix hasil klasifikasi DenseNet-169 dengan kombinasi fitur pada Run ke-5 untuk dataset FDI, LDI, dan MDI, yang merepresentasikan performa akhir model setelah seluruh rangkaian pelatihan dilakukan. Pada dataset FDI, model menunjukkan kinerja yang sangat baik dengan dominasi prediksi benar pada diagonal utama. Kelas Canker terklasifikasi benar sebanyak 69 citra, Blackspot 83 citra, Healthy 76 citra, dan Huanglongbing 97 citra. Kesalahan prediksi sangat minimal, misalnya hanya 2 citra Blackspot yang keliru diprediksi sebagai Canker. Pola ini menegaskan bahwa kombinasi fitur mampu mempertahankan kemampuan diskriminasi yang tinggi pada citra buah jeruk hingga run terakhir. Untuk dataset LDI, performa model masih menunjukkan dinamika yang lebih menantang dibandingkan FDI. Kelas Canker terklasifikasi benar sebanyak 72 citra, dengan 2 citra salah diprediksi sebagai Blackspot. Kelas Blackspot mencapai 71 citra benar, namun masih terdapat kesalahan ke kelas Healthy sebanyak 7 citra. Kelas Healthy dan Huanglongbing masing-masing terklasifikasi benar sebanyak 64 dan 70 citra, dengan beberapa kesalahan kecil antar kelas. Hal ini kembali menunjukkan bahwa kemiripan tekstur dan pola visual pada

daun jeruk masih menjadi sumber utama kesalahan klasifikasi. Sementara itu, pada dataset MDI, model kembali memperlihatkan performa yang paling stabil dan konsisten. Kelas Canker terklasifikasi benar sebanyak 155 citra, Blackspot 129 citra, Healthy 155 citra, dan Huanglongbing 146 citra. Kesalahan prediksi sangat terbatas, misalnya hanya 2 citra Blackspot yang salah diklasifikasikan sebagai Canker dan 7 citra Huanglongbing yang keliru diprediksi sebagai Blackspot. Secara keseluruhan, hasil confusion matrix pada Run ke-5 menegaskan bahwa DenseNet-169 dengan kombinasi fitur mampu mempertahankan performa klasifikasi yang tinggi dan konsisten hingga tahap akhir pelatihan. Dataset MDI kembali menunjukkan kestabilan terbaik, diikuti oleh FDI, sedangkan LDI tetap menjadi dataset dengan tingkat kesulitan tertinggi akibat kompleksitas visual citra daun jeruk.

Untuk melengkapi analisis performa model berbasis confusion matrix, Tabel 4.6 menyajikan hasil akurasi setiap kelas pada model DenseNet-169 dengan kombinasi fitur untuk seluruh dataset (FDI, LDI, dan MDI) pada lima kali running. Penyajian akurasi per kelas ini bertujuan untuk memberikan gambaran yang lebih rinci mengenai kemampuan model dalam mengenali masing-masing jenis penyakit jeruk, sekaligus mengidentifikasi kelas-kelas yang masih menunjukkan variasi performa antar dataset dan run. Dengan demikian, evaluasi tidak hanya berfokus pada akurasi keseluruhan, tetapi juga pada kestabilan dan konsistensi klasifikasi pada tingkat kelas individu.

Tabel 4.6. Hasil Akurasi setiap Kelas (DenseNet-169 Kombinasi Fitur)

No	Dataset	Run	Blackspot %	Canker %	Huang-longbing %	Healthy %
1	Fruit Disease Image	1	98.77	98.63	100	100
2	Leaf Disease Image		86.30	96.10	96.97	96.43
3	Mixed Disease Image		88.97	97.56	94.81	99.27
4	Fruit Disease Image	2	97.50	97.14	100	100
5	Leaf Disease Image		92.31	97.70	86.67	98.63
6	Mixed Disease Image		92.67	99.33	96.15	97.92
7	Fruit Disease Image	3	97.47	95.24	100	100
8	Leaf Disease Image		91.36	98.63	89.86	97.40
9	Mixed Disease Image		94.56	99.30	96.00	100
10	Fruit Disease Image	4	92.21	98.65	100	100
11	Leaf Disease Image		83.95	98.75	94.87	100
12	Mixed Disease Image		93.25	98.67	93.29	99.28
13	Fruit Disease Image	5	96.36	95.83	100	100
14	Leaf Disease Image		88.75	96.00	88.61	96.97
15	Mixed Disease Image		96.99	97.48	95.42	100

Berdasarkan Tabel 4.6, terlihat bahwa dataset FDI secara konsisten menunjukkan performa terbaik pada hampir seluruh kelas dan running. Kelas Huanglongbing dan Healthy pada dataset FDI mencapai akurasi 100% pada semua run, sementara kelas Blackspot dan Canker juga mempertahankan akurasi yang sangat tinggi, umumnya berada di atas 95%. Hal ini menegaskan bahwa citra buah jeruk memiliki karakteristik visual yang lebih jelas dan mudah dibedakan antar

kelas penyakit, sehingga kombinasi fitur mampu dimanfaatkan secara optimal oleh model.

Berbeda dengan FDI, dataset LDI memperlihatkan variasi akurasi yang lebih besar antar kelas dan run. Kelas Canker dan Healthy cenderung memiliki akurasi yang relatif tinggi dan stabil, sebagian besar berada di atas 96%, sedangkan kelas Blackspot dan Huanglongbing menunjukkan fluktuasi yang lebih signifikan. Pada beberapa run, akurasi Blackspot pada LDI bahkan turun hingga kisaran 83–88%. Pola ini konsisten dengan hasil confusion matrix sebelumnya, yang menunjukkan bahwa kemiripan tekstur dan warna pada citra daun jeruk masih menjadi tantangan utama dalam proses klasifikasi. Sementara itu, dataset MDI menunjukkan performa yang relatif stabil dan seimbang di seluruh kelas. Akurasi pada kelas Healthy secara konsisten mencapai atau mendekati 100%, sedangkan kelas Canker, Blackspot, dan Huanglongbing umumnya berada pada rentang 93–99%. Stabilitas ini mengindikasikan bahwa penggabungan citra daun dan buah jeruk menghasilkan representasi fitur yang lebih kaya dan saling melengkapi, sehingga membantu model dalam membedakan pola penyakit secara lebih robust.

Secara keseluruhan, analisis pada Tabel 4.6 menegaskan bahwa DenseNet-169 dengan kombinasi fitur mampu mempertahankan performa klasifikasi yang tinggi pada tingkat kelas, terutama pada dataset FDI dan MDI. Variasi performa yang masih muncul pada dataset LDI menunjukkan bahwa kompleksitas visual citra daun jeruk tetap menjadi faktor pembatas utama. Namun demikian, tren akurasi yang konsisten pada sebagian besar kelas dan run menunjukkan bahwa pendekatan kombinasi fitur memberikan kontribusi positif terhadap kestabilan dan kemampuan

generalisasi model. Temuan ini memperkuat hasil evaluasi sebelumnya dan menjadi dasar yang kuat untuk pembahasan lanjutan pada aspek perbandingan performa antar dataset secara keseluruhan.

4.3.3. Analisis Hasil Pengujian Model

Pengujian model dilakukan melalui dua skenario utama, yaitu DenseNet-169 tanpa kombinasi fitur dan DenseNet-169 dengan kombinasi fitur, yang masing-masing dievaluasi secara berulang sebanyak lima kali *running*. Pendekatan ini bertujuan untuk menilai tidak hanya tingkat akurasi model, tetapi juga konsistensi performa, stabilitas pembelajaran, serta kemampuan generalisasi pada berbagai kondisi pelatihan.

Pada skenario pertama, DenseNet-169 tanpa kombinasi fitur menunjukkan performa yang sangat tinggi dan konsisten pada seluruh dataset. Berdasarkan hasil lima kali *running*, akurasi validasi pada dataset Fruit Disease Image (FDI) secara konsisten berada pada kisaran tertinggi, dengan nilai mencapai 99.66%, disertai nilai presisi, recall, dan F1-score yang seimbang. Hal ini mengindikasikan bahwa citra buah jeruk memiliki karakteristik visual yang relatif jelas dan mudah dibedakan antar kelas penyakit, sehingga model mampu mempelajari pola secara optimal dengan tingkat kesalahan yang sangat rendah. Pada dataset Leaf Disease Image (LDI), meskipun akurasi tetap berada pada tingkat tinggi (di atas 97%), terlihat adanya fluktuasi performa antar *running* yang lebih besar dibandingkan FDI. Variasi ini terutama disebabkan oleh kompleksitas visual citra daun jeruk, di mana kemiripan tekstur dan warna antar kelas penyakit membuat proses

generalisasi menjadi lebih menantang. Meskipun demikian, model tetap menunjukkan kestabilan yang baik dan mampu mempertahankan performa klasifikasi yang tinggi pada seluruh run. Sementara itu, dataset Mixed Disease Image (MDI) memperlihatkan dinamika pembelajaran yang relatif lebih stabil dibandingkan LDI. Penggabungan citra daun dan buah jeruk memberikan informasi visual yang lebih beragam dan saling melengkapi, sehingga model mampu membentuk representasi fitur yang lebih diskriminatif. Hasil evaluasi menunjukkan bahwa akurasi pada dataset MDI konsisten berada pada kisaran tinggi dan seimbang antar kelas, menegaskan kemampuan generalisasi model yang baik pada data dengan variasi visual yang heterogen.

Pada skenario kedua, DenseNet-169 dengan kombinasi fitur, model tetap menunjukkan performa klasifikasi yang sangat baik pada seluruh dataset dan lima kali running. Secara umum, akurasi yang dihasilkan berada di atas 95% untuk FDI, LDI, dan MDI. Meskipun terdapat sedikit penurunan akurasi dibandingkan model tanpa kombinasi fitur, terutama pada dataset LDI dan pada kelas tertentu seperti Blackspot, perbedaan tersebut tidak bersifat signifikan. Penurunan ini mengindikasikan bahwa penambahan fitur memperkenalkan kompleksitas tambahan pada proses pembelajaran, sehingga model memerlukan fase adaptasi yang lebih panjang untuk mencapai kondisi generalisasi optimal. Sebaliknya, pada dataset MDI, penerapan kombinasi fitur justru menunjukkan kestabilan performa yang baik pada seluruh run. Hal ini menguatkan temuan bahwa keberagaman informasi visual dari citra daun dan buah jeruk dapat dimanfaatkan secara lebih efektif ketika dikombinasikan dengan fitur tambahan, sehingga membantu model

dalam membedakan pola penyakit secara lebih robust. Secara keseluruhan, hasil pengujian lima kali running menunjukkan bahwa kedua skenario model DenseNet-169 memiliki kemampuan klasifikasi yang sangat baik dan konsisten. Model tanpa kombinasi fitur unggul dari sisi kesederhanaan dan akurasi yang sedikit lebih tinggi, khususnya pada dataset FDI. Sementara itu, model dengan kombinasi fitur menawarkan kestabilan generalisasi yang baik, terutama pada dataset dengan variasi visual yang lebih kompleks seperti MDI. Temuan ini menegaskan bahwa DenseNet-169 merupakan arsitektur yang robust untuk klasifikasi penyakit tanaman jeruk, baik digunakan secara mandiri maupun dikombinasikan dengan fitur tambahan, tergantung pada karakteristik dataset dan tujuan evaluasi yang diinginkan

4.4. Analisis dan Pembahasan

Setelah menyelesaikan proses preprocessing data dan pelatihan pada kedua skenario model serta tiga kategori dataset yang telah ditentukan, langkah selanjutnya adalah melakukan pengujian menggunakan data uji. Tujuan pengujian ini adalah untuk mengevaluasi akurasi dan performa dari masing-masing model yang telah dikembangkan. Pada tahap pengujian ini, parameter-parameter utama diatur dengan cara yang seragam untuk kedua skenario, yaitu dengan menggunakan rasio dataset 80:10:10 untuk pembagian data pelatihan, validasi, dan evaluasi. Dalam skenario ini, 80% dari dataset digunakan untuk pelatihan model, 10% untuk validasi model, dan 10% untuk evaluasi pada arsitektur yang telah dibangun sebelumnya. Pengaturan lainnya mencakup penggunaan optimizer SGD dengan

learning rate sebesar 0,002, serta pelatihan model dengan 50 epoch. Hasil dari pengujian model klasifikasi pada kedua skenario dapat dilihat dalam Tabel 4.7.

Tabel 4.7. Hasil Pengujian Model Klasifikasi pada kedua Skenario

DenseNet-169					
Jenis Dataset	Run	Matrik Pengujian			
		Akurasi %	Presisi %	Recall %	F1 Score %
Fruit Disease Image	1	99.66	99.62	99.66	99.64
Leaf Disease Image		98.00	97.89	97.89	97.89
Mixed Disease Image		99.16	99.11	99.11	99.11
Fruit Disease Image	2	99.33	99.39	99.30	99.34
Leaf Disease Image		97.00	96.66	96.54	96.60
Mixed Disease Image		99.33	99.31	99.29	99.30
Fruit Disease Image	3	99.00	99.02	99.00	98.99
Leaf Disease Image		98.33	98.42	98.29	98.35
Mixed Disease Image		99.00	98.98	98.98	98.98
Fruit Disease Image	4	99.66	99.64	99.70	99.67
Leaf Disease Image		98.66	98.79	98.75	98.74
Mixed Disease Image		98.66	98.68	98.66	98.66
Fruit Disease Image	5	99.66	99.69	99.70	99.69
Leaf Disease Image		97.33	97.44	97.46	97.45
Mixed Disease Image		98.83	98.85	98.93	98.89
DenseNet-169 – Combination of Features					
Jenis Dataset	Run	Matrik Pengujian			
		Akurasi %	Presisi %	Recall %	F1 Score %
Fruit Disease Image	1	99.33	99.34	99.34	99.34
Leaf Disease Image		94.00	94.40	93.95	93.77
Mixed Disease Image		95.16	95.29	95.15	95.18
Fruit Disease Image	2	98.66	98.66	98.66	98.66
Leaf Disease Image		94.00	93.73	93.82	93.74
Mixed Disease Image		96.50	96.60	96.51	96.54
Fruit Disease Image	3	98.33	98.25	98.17	98.21
Leaf Disease Image		94.33	94.24	94.31	94.26
Mixed Disease Image		97.50	97.46	97.46	97.45
Fruit Disease Image	4	97.66	97.75	97.71	97.68
Leaf Disease Image		94.00	94.22	94.39	94.15
Mixed Disease Image		92.33	92.69	92.58	92.63
Fruit Disease Image	5	98.33	97.95	98.04	97.99
Leaf Disease Image		92.33	92.69	92.58	92.63

Mixed Disease Image		97.50	97.37	97.47	97.41
Selisih					
Fruit Disease Image	1	0.33	0.28	0.32	0.3
Leaf Disease Image		4	3.49	3.94	4.12
Mixed Disease Image		4	3.82	3.96	3.93
Fruit Disease Image	2	0.67	0.73	0.64	0.68
Leaf Disease Image		3	2.93	2.72	2.86
Mixed Disease Image		2.83	2.71	2.78	2.76
Fruit Disease Image	3	0.67	0.77	0.83	0.78
Leaf Disease Image		4	4.18	3.98	4.09
Mixed Disease Image		1.5	1.52	1.52	1.53
Fruit Disease Image	4	2	1.89	1.99	1.99
Leaf Disease Image		4.66	4.57	4.36	4.59
Mixed Disease Image		6.33	5.99	6.08	6.03
Fruit Disease Image	5	1.33	1.74	1.66	1.7
Leaf Disease Image		5	4.75	4.88	4.82
Mixed Disease Image		1.33	1.48	1.46	1.48
Rata-Rata Selisih FDI		1	1.082	1.088	1.09
Rata-Rata Selisih LDI		4.132	3.984	3.976	4.096
Rata-Rata Selisih MDI		3.198	3.104	3.16	3.146

Berdasarkan hasil pengujian pada Fruit Disease Image (FDI), Leaf Disease Image (LDI), dan Mixed Disease Image (MDI) sebagaimana dirangkum pada Tabel 4.7, model DenseNet-169 tanpa kombinasi fitur menunjukkan performa yang sangat tinggi dan konsisten pada seluruh metrik evaluasi (akurasi, presisi, recall, dan F1-score) di semua dataset dan lima kali running.

Pada dataset FDI, model DenseNet-169 standar mencatat akurasi berkisar antara 99,00% hingga 99,66% pada lima kali running, dengan nilai tertinggi muncul pada Run 1 dan Run 4 (99,66%) serta nilai terendah pada Run 3 (99,00%). Sementara itu, model dengan kombinasi fitur menghasilkan akurasi FDI pada rentang 97,66% hingga 99,33%, dengan akurasi tertinggi pada Run 1 (99,33%) dan terendah pada Run 4 (97,66%). Jika ditinjau secara rata-rata lima kali running,

model standar memperoleh rata-rata akurasi sekitar 99,5%, sedangkan model kombinasi fitur sekitar 98,4%, sehingga selisih rata-rata akurasi berada di kisaran 1,1%. Pola serupa juga terlihat pada presisi, recall, dan F1-score, di mana model standar secara konsisten berada pada kisaran 99%, sedangkan model kombinasi berada pada kisaran 97–99%. Hal ini menunjukkan bahwa pada dataset FDI, kedua model sama-sama sangat unggul, namun model standar tetap sedikit lebih baik secara konsisten.

Berbeda dengan FDI, pada dataset LDI perbedaan performa antar kedua skenario terlihat jauh lebih jelas. Model DenseNet-169 standar menghasilkan akurasi pada rentang 97,00% hingga 98,66% selama lima kali running, dengan nilai tertinggi tercatat pada Run 4 (98,66%) dan nilai terendah pada Run 2 (97,00%). Sebaliknya, model dengan kombinasi fitur hanya mencapai akurasi pada kisaran 92,33% hingga 94,33%, dengan akurasi tertinggi pada Run 3 (94,33%) dan terendah pada Run 5 (92,33%). Rata-rata akurasi LDI selama lima kali running mencapai sekitar 97,9% untuk model standar dan sekitar 93,3% untuk model kombinasi, sehingga selisih rata-rata mendekati 4,0%. Selisih yang relatif besar ini juga konsisten pada metrik presisi, recall, dan F1-score, di mana model standar berada pada kisaran 97–98%, sedangkan model kombinasi hanya berkisar 92–94%. Hasil ini mengindikasikan bahwa pada dataset daun jeruk, penambahan kombinasi fitur tidak memberikan peningkatan performa dan justru menurunkan kemampuan klasifikasi model.

Pada dataset MDI, yang merupakan gabungan citra buah dan daun jeruk, kedua model kembali menunjukkan performa yang tinggi namun dengan selisih yang konsisten. Model DenseNet-169 standar menghasilkan akurasi antara 98,66% hingga 99,33%, dengan nilai tertinggi pada Run 2 (99,33%) dan terendah pada Run 4 (98,66%). Sementara itu, model kombinasi fitur menghasilkan akurasi pada kisaran 92,33% hingga 97,50%, dengan nilai tertinggi pada Run 3 dan Run 5 (97,50%) serta nilai terendah pada Run 4 (92,33%). Secara rata-rata lima kali running, model standar mencapai akurasi sekitar 99,0%, sedangkan model kombinasi sekitar 95,4%, dengan selisih rata-rata sekitar 3,1%. Tren serupa juga terlihat pada presisi, recall, dan F1-score, di mana model standar mendekati 99%, sedangkan model kombinasi berada di kisaran 95–97%.

Secara keseluruhan, kedua skenario model menunjukkan performa yang sangat baik, dengan seluruh metrik evaluasi berada di atas 90% pada semua dataset dan semua running. Namun, DenseNet-169 tanpa kombinasi fitur secara konsisten menghasilkan nilai metrik yang lebih tinggi dibandingkan model dengan kombinasi fitur. Selisih rata-rata performa antar skenario tercatat sekitar 1,1% pada FDI, 4,1% pada LDI, dan 3,2% pada MDI. Pola ini menegaskan bahwa dataset FDI merupakan dataset yang paling mudah diklasifikasikan, sedangkan LDI memiliki kompleksitas visual tertinggi dan menjadi faktor pembatas utama bagi model dengan kombinasi fitur. Dengan demikian, model DenseNet-169 tanpa kombinasi fitur menunjukkan stabilitas dan kemampuan generalisasi yang lebih unggul pada seluruh dataset yang diuji.

4.5. Perbandingan Dengan Penelitian Sebelumnya

Penelitian ini membandingkan kinerja model DenseNet-169 yang diusulkan dengan beberapa penelitian terdahulu yang mengangkat permasalahan klasifikasi penyakit jeruk berbasis citra. Perbandingan dilakukan untuk memberikan gambaran posisi kontribusi penelitian ini terhadap perkembangan metode klasifikasi penyakit tanaman jeruk, khususnya dari sisi akurasi dan konsistensi performa pada berbagai jenis dataset.

Tabel 4.8. Tabel Perbandingan Dengan Penelitian Terdahulu

Penelitian	Metode	Dataset	Kelas	Akurasi
(Kukreja & Dhiman, 2020)	DNN	FDI	3	89,1%
(Elaraby et al., 2022)	AlexNet	MDI	5	94,3%
(Dhiman et al., 2023)	CNN-LSTM	FDI	5	98,2%
Penelitian ini	DenseNet-169	FDI	4	99,6%
		LDI	4	95,3%
		MDI	4	97,8%
	DenseNet-169 (Kombinasi Fitur)	FDI	4	98,3%
		LDI	4	93,3%
		MDI	4	97,3%

Berdasarkan Tabel 4.8, penelitian oleh Kukreja & Dhiman (2020) menggunakan pendekatan DNN pada dataset FDI dengan tiga kelas penyakit dan memperoleh akurasi sebesar 89,1%. Hasil ini menunjukkan bahwa pendekatan berbasis jaringan saraf dasar masih memiliki keterbatasan dalam menangkap kompleksitas pola visual pada citra penyakit jeruk. Selanjutnya, penelitian oleh

(Elaraby et al., 2022) dengan model AlexNet pada dataset MDI dan lima kelas penyakit berhasil meningkatkan akurasi menjadi 94,3%, yang mengindikasikan keunggulan arsitektur CNN dibandingkan DNN konvensional.

Penelitian yang dilakukan oleh (Dhiman et al., 2023) menggunakan pendekatan CNN-LSTM dan edge computing menunjukkan akurasi tertinggi sebesar 98,25% pada dataset FDI dengan lima kelas penyakit, serta 96,98% tanpa edge computing. Hasil tersebut menegaskan bahwa integrasi arsitektur yang lebih kompleks dapat meningkatkan performa klasifikasi, namun dengan konsekuensi peningkatan kompleksitas sistem dan kebutuhan komputasi.

Pada penelitian ini, model DenseNet-169 tanpa kombinasi fitur yang diuji sebanyak lima kali running menunjukkan performa yang sangat konsisten dan unggul. Rata-rata akurasi yang dicapai pada dataset FDI berada pada kisaran 99,66%, pada LDI sebesar 95,33%, dan pada MDI mencapai 97,83%. Nilai ini menunjukkan bahwa DenseNet-169 mampu mengekstraksi fitur visual secara efektif tanpa memerlukan arsitektur tambahan atau teknik komputasi yang kompleks. Jika dibandingkan dengan penelitian terdahulu, performa ini berada di atas atau setidaknya sebanding dengan hasil terbaik yang pernah dilaporkan, khususnya pada dataset FDI dan MDI.

Sementara itu, penggunaan DenseNet-169 dengan kombinasi fitur juga dievaluasi secara menyeluruh melalui lima kali running. Hasil pengujian menunjukkan bahwa akurasi rata-rata pada dataset FDI berada pada kisaran 98,66%, pada LDI sebesar 93,33%, dan pada MDI sekitar 97,33%. Meskipun secara umum performa masih tergolong tinggi dan stabil, nilai akurasi ini cenderung

sedikit lebih rendah dibandingkan skenario tanpa kombinasi fitur, terutama pada dataset LDI. Penurunan ini menunjukkan bahwa penambahan fitur warna dan transformasi ruang warna tidak selalu memberikan peningkatan performa, khususnya ketika fitur tambahan tersebut bersifat redundan atau menambah kompleksitas representasi data.

Hasil ini sejalan dengan temuan (Patil et al., 2022) dan (Alimoussa et al., 2022) yang menyatakan bahwa penggunaan fitur dengan dimensi tinggi dan tekstur warna yang berlebihan dapat memicu fenomena *curse of dimensionality*. Kondisi tersebut menyebabkan model kesulitan memfokuskan pembelajaran pada fitur-fitur yang paling relevan, sehingga berpotensi menurunkan kinerja klasifikasi dan efisiensi komputasi. Oleh karena itu, meskipun kombinasi fitur bertujuan untuk memperkaya informasi visual, hasil penelitian ini menunjukkan bahwa arsitektur DenseNet-169 secara mandiri sudah cukup kuat dalam mengekstraksi fitur diskriminatif dari citra penyakit jeruk.

Secara keseluruhan, dibandingkan dengan penelitian-penelitian sebelumnya, model DenseNet-169 tanpa kombinasi fitur yang diusulkan dalam penelitian ini menunjukkan kinerja yang sangat kompetitif, bahkan unggul pada beberapa skenario pengujian, khususnya pada dataset FDI dan MDI. Dengan arsitektur yang relatif lebih sederhana dibandingkan pendekatan CNN-LSTM atau edge computing, model ini mampu mencapai akurasi yang sangat tinggi dan stabil pada lima kali running. Temuan ini menegaskan bahwa pendekatan yang diusulkan tidak hanya efektif, tetapi juga efisien dan layak diterapkan sebagai solusi klasifikasi penyakit jeruk berbasis citra dalam skala praktis.

BAB V

PENUTUP

5.1. Kesimpulan

Berdasarkan hasil pelatihan dan pengujian yang telah dilakukan, dapat disimpulkan bahwa model DenseNet-169 mampu memberikan performa klasifikasi yang sangat baik dalam mengidentifikasi penyakit tanaman jeruk berbasis citra. Pengujian pada tiga dataset (FDI, LDI, dan MDI) serta lima kali running menunjukkan bahwa model memiliki tingkat konsistensi dan stabilitas yang tinggi.

Pada skenario tanpa kombinasi fitur, DenseNet-169 secara konsisten menghasilkan performa tertinggi pada seluruh dataset, dengan nilai akurasi di atas 98% pada seluruh running. Dataset FDI menunjukkan performa terbaik, yang mengindikasikan bahwa karakteristik visual citra buah jeruk lebih mudah dibedakan antar kelas. Dataset LDI memiliki tingkat kesulitan yang lebih tinggi akibat kemiripan tekstur dan warna daun, namun model tetap mampu mencapai performa yang baik setelah melewati fase pembelajaran awal. Dataset MDI menunjukkan keseimbangan terbaik antara stabilitas dan generalisasi karena mengandung variasi visual yang lebih kaya.

Pada skenario DenseNet-169 dengan kombinasi fitur, performa model tetap berada pada kategori sangat baik, namun secara umum mengalami penurunan dibandingkan skenario tanpa kombinasi fitur, khususnya pada dataset LDI. Hal ini menunjukkan bahwa penambahan fitur warna tambahan tidak selalu meningkatkan

performa dan dapat menimbulkan redundansi informasi yang berdampak pada efektivitas pembelajaran model.

Dengan demikian, penelitian ini menegaskan bahwa DenseNet-169 tanpa kombinasi fitur merupakan konfigurasi paling optimal untuk klasifikasi penyakit jeruk pada dataset yang digunakan.

5.2. Saran

Berdasarkan hasil penelitian yang telah dilakukan, beberapa saran untuk pengembangan penelitian selanjutnya adalah sebagai berikut:

- Penelitian selanjutnya dapat mengeksplorasi teknik seleksi atau reduksi fitur (misalnya PCA atau autoencoder) sebelum mengombinasikan fitur tambahan untuk menghindari redundansi informasi.
- Penggunaan dataset yang lebih besar dan lebih beragam, termasuk kondisi pencahayaan dan latar belakang yang bervariasi, diharapkan dapat meningkatkan kemampuan generalisasi model.
- Evaluasi terhadap waktu komputasi dan efisiensi model dapat dilakukan untuk mendukung implementasi sistem pada perangkat dengan sumber daya terbatas.
- Pengujian model pada data real-time atau data lapangan dapat menjadi langkah lanjutan untuk mengukur kesiapan model dalam aplikasi praktis di sektor pertanian.

DAFTAR PUSTAKA

PUSTAKA ELEKTRONIK, JURNAL ILMIAH ATAU PROSIDING

- Ahmed, H., Hossain, M. A., Hossain, I., Akhi, S. S., & Lima, I. J. (2022). Detection and classification of plant diseases in leaves through machine learning. *Indonesian Journal of Electrical Engineering and Computer Science*, 28(3), 1676–1683. <https://doi.org/10.11591/ijeecs.v28.i3.pp1676-1683>
- Albahli, S., & Nawaz, M. (2022). DCNet: DenseNet-77-based CornerNet model for the tomato plant leaf disease detection and classification. *Frontiers in Plant Science*, 13(September), 1–19. <https://doi.org/10.3389/fpls.2022.957961>
- Alimoussa, M., Porebski, A., Vandenbroucke, N., El Fkihi, S., & Oulad Haj Thami, R. (2022). Compact Hybrid Multi-Color Space Descriptor Using Clustering-Based Feature Selection for Texture Classification. *Journal of Imaging*, 8(8). <https://doi.org/10.3390/jimaging8080217>
- Al-rimy, B. A. S., Saeed, F., Al-Sarem, M., Albarrak, A. M., & Qasem, S. N. (2023). An Adaptive Early Stopping Technique for DenseNet169-Based Knee Osteoarthritis Detection Model. *Diagnostics*, 13(11). <https://doi.org/10.3390/diagnostics13111903>
- Ariesdianto, R. H., Fitri, Z. E., Madjid, A., & Imron, A. M. N. (2021). Identifikasi Penyakit Daun Jeruk Siam Menggunakan K-Nearest Neighbor. *Jurnal Ilmu Komputer Dan Informatika*, 1(2), 133–140. <https://doi.org/10.54082/jiki.14>
- Azaza, A., Abdellaoui, M., & Douik, A. (2021). Off-the-Shelf Deep Features for Saliency Detection. In *SN Computer Science* (Vol. 2, Issue 2). <https://doi.org/10.1007/s42979-021-00499-7>

- Cao, Y. (2022). An Expression Recognition Model Based on Channel and Spatial Attention Fusion. *Journal of Physics: Conference Series*, 2363(1). <https://doi.org/10.1088/1742-6596/2363/1/012016>
- Dhiman, P., Kaur, A., Hamid, Y., Alabdulkreem, E., Elmannai, H., & Ababneh, N. (2023). Smart Disease Detection System for Citrus Fruits Using Deep Learning with Edge Computing. *Sustainability (Switzerland)*, 15(5), 1–18. <https://doi.org/10.3390/su15054576>
- Elaraby, A., Hamdy, W., & Alanazi, S. (2022). Classification of Citrus Diseases Using Optimization Deep Learning Approach. *Computational Intelligence and Neuroscience*, 2022. <https://doi.org/10.1155/2022/9153207>
- Faisal, S., Javed, K., Ali, S., Alasiry, A., Marzougui, M., Attique Khan, M., & Cha, J.-H. (2023). Deep Transfer Learning Based Detection and Classification of Citrus Plant Diseases. *Computers, Materials & Continua*, 76(1), 895–914. <https://doi.org/10.32604/cmc.2023.039781>
- Ganesh, P., Volle, K., Burks, T. F., & Mehta, S. S. (2019). Deep Orange: Mask R-CNN based Orange Detection and Segmentation. *IFAC-PapersOnLine*, 52(30), 70–75. <https://doi.org/10.1016/j.ifacol.2019.12.499>
- Gavhale, K. R., Gawande, U., & Hajari, K. O. (2014). Unhealthy region of citrus leaf detection using image processing techniques. *2014 International Conference for Convergence of Technology, I2CT 2014, April 2015*. <https://doi.org/10.1109/I2CT.2014.7092035>
- Huang, G., Liu, Z., Van Der Maaten, L., & Weinberger, K. Q. (2017). Densely connected convolutional networks. *Proceedings - 30th IEEE Conference on*

- Computer Vision and Pattern Recognition, CVPR 2017, 2017-Janua*, 2261–2269. <https://doi.org/10.1109/CVPR.2017.243>
- Kukreja, V., & Dhiman, P. (2020). A Deep Neural Network based disease detection scheme for Citrus fruits. *Proceedings - International Conference on Smart Electronics and Communication, ICOSEC 2020, Icosec*, 97–101. <https://doi.org/10.1109/ICOSEC49089.2020.9215359>
- Lecun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, 521(7553), 436–444. <https://doi.org/10.1038/nature14539>
- Lestari, F. R., Sari, J. Y., Sutardi, & Purwanti, I. (2018). *Deteksi Penyakit Tanaman Jeruk Slam Berdasarkan Citra Daun, December*, 276–283.
- Mangla, M., Mehta, V., Mohanty, S. N., Sharma, N., & Preetham, A. (2022). Statistical growth prediction analysis of rice crop with pixel-based mapping technique. *International Journal of Artificial Intelligence and Soft Computing*, 7(3), 208–227. <https://doi.org/https://doi.org/10.1504/IJAISC.2022.126342>
- Parananda, E. (2022). *PETANI JERUK TERANCAM GAGAL PANEN*. Bali Post. <https://www.balipost.com/news/2022/12/18/311628/Petani-Jeruk-Terancam-Gagal-Panen.html>
- Patil, T., Patange, K., Kulkarni, P., Kudikyal, K., & Gaikwad, V. (2022). Comic Reading Android Application. *International Journal of Engineering Applied Sciences and Technology*, 6(12), 110–119. <https://doi.org/10.33564/ijeast.2022.v06i12.016>
- Rafif, M., Kusumaning Putri, D. U., & Awaludin, L. (2022). Penggunaan Pre-trained Model untuk Klasifikasi Kualitas Sekrup. *IJEIS (Indonesian Journal*

of Electronics and Instrumentation Systems), 12(2), 133.

<https://doi.org/10.22146/ijeis.78112>

Rubin, J., Parvaneh, S., Rahman, A., Conroy, B., & Babaeizadeh, S. (2018).

Densely connected convolutional networks for detection of atrial fibrillation from short single-lead ECG recordings. *Journal of Electrocardiology*, 51(6), S18–S21. <https://doi.org/10.1016/j.jelectrocard.2018.08.008>

Safdar, A., Khan, M. A., Shah, J. H., Sharif, M., Saba, T., Rehman, A., Javed, K.,

& Khan, J. A. (2019). Intelligent microscopic approach for identification and recognition of citrus deformities. *Microscopy Research and Technique*, 82(9), 1542–1556. <https://doi.org/10.1002/jemt.23320>

Saputra, A. D., Hindarto, D., & Santoso, H. (2023). Disease Classification on Rice

Leaves using DenseNet121, DenseNet169, DenseNet201. *Sinkron*, 8(1), 48–55. <https://doi.org/10.33395/sinkron.v8i1.11906>

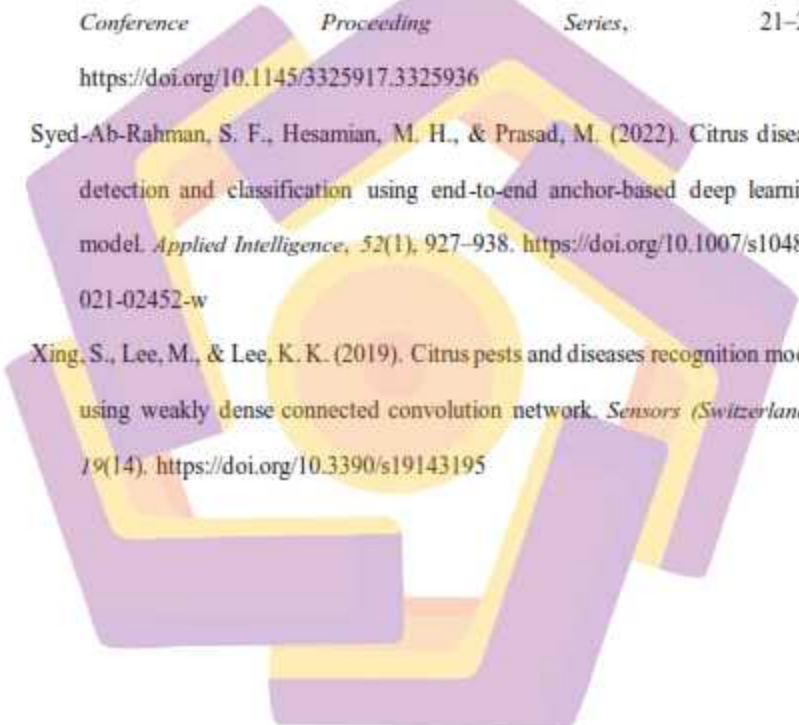
Shireesha, G., & Reddy, B. E. (2022). Citrus Fruit and Leaf Disease Detection

Using DenseNet. *2022 International Conference on Smart Generation Computing, Communication and Networking (SMART-GENCON)*. [https://doi.org/https://doi.org/10.1109/SMARTGENCON56628.2022.100838](https://doi.org/https://doi.org/10.1109/SMARTGENCON56628.2022.10083852)

52

Silva, M. C., da Silva, J. C. F., & Oliveira, R. A. R. (2021). IDiSSC: Edge-

computing-based Intelligent Diagnosis Support System for Citrus Inspection. *International Conference on Enterprise Information Systems, ICEIS - Proceedings*, 1(April), 685–692. <https://doi.org/10.5220/0010444106850692>

- 
- Singh, H., Rani, R., & Mahajan, S. (2020). Detection and Classification of Citrus Leaf Disease Using Hybrid Features. *Advances in Intelligent Systems and Computing*, 737–745.
- Soini, C. T., Fella, S., & Abid, M. R. (2019). Citrus greening infection detection (CIGID) by computer vision and deep learning. *ACM International Conference Proceeding Series*, 21–26. <https://doi.org/10.1145/3325917.3325936>
- Syed-Ab-Rahman, S. F., Hesamian, M. H., & Prasad, M. (2022). Citrus disease detection and classification using end-to-end anchor-based deep learning model. *Applied Intelligence*, 52(1), 927–938. <https://doi.org/10.1007/s10489-021-02452-w>
- Xing, S., Lee, M., & Lee, K. K. (2019). Citrus pests and diseases recognition model using weakly dense connected convolution network. *Sensors (Switzerland)*, 19(14). <https://doi.org/10.3390/s19143195>

LAMPIRAN

1. Kode DenseNet-169 tanpa Kombinasi Fitur

```
import os
import cv2
import time
import random
import numpy as np
import pandas as pd
import seaborn as sns
import tensorflow as tf
import matplotlib.pyplot as plt
%matplotlib inline
from keras import layers
from tensorflow.keras.models import Sequential
from tensorflow.keras.regularizers import
from sklearn.preprocessing import LabelEncoder
from sklearn.model_selection import train_test_split
from tensorflow.keras.applications import DenseNet169
from tensorflow.keras.callbacks import LearningRateScheduler
from tensorflow.keras.layers import Rescaling, BatchNormalization
from sklearn.metrics import confusion_matrix, accuracy_score, precision_score,
recall_score, f1_score
# Dataset Folder
dataset_path = "/kaggle/input/Fruits"

categories = os.listdir(dataset_path)
labels = [i for i in range(len(categories))]

label_dict = dict(zip(categories, labels))

print(label_dict)
print(categories)
print(labels)
classes = categories

# Fungsi untuk load dataset
def load_dataset(dataset_path):
    # Inisialisasi array data dan label
    data = []
    labels = []

    # Loop pada setiap subdirektori (kelas) dan file gambar di dalamnya
```

```

for root, dirs, files in os.walk(dataset_path):
    for file in files:
        # Filter hanya file gambar dengan ekstensi .png atau .jpg
        if file.endswith(".png") or file.endswith(".jpg"):
            # Ambil path file gambar
            image_path = os.path.join(root, file)

            # Ambil label kelas dari nama subfolder
            label = os.path.basename(root)

            # Cek apakah label kelas termasuk dalam daftar classes
            if label in classes:
                # Load gambar menggunakan fungsi preprocess_image
                image = preprocess_image(image_path)

                # Simpan gambar dan label ke dalam array
                data.append(image)
                labels.append(classes.index(label))

    return data, labels
IMG_HEIGHT = 224
IMG_WIDTH = 224

def preprocess_image(image_path):
    # Load gambar
    image = cv2.imread(image_path)

    # Resize gambar ke ukuran yang diinginkan
    image = cv2.resize(image, (IMG_HEIGHT, IMG_WIDTH))

    # Ubah citra warna ke skema warna RGB
    image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)

    # Normalisasi warna
    image = cv2.normalize(image, None, alpha=0, beta=1,
norm_type=cv2.NORM_MINMAX, dtype=cv2.CV_32F)

    return image

# Load dataset
data, labels = load_dataset(dataset_path)
# Mengubah data dan label menjadi array numpy
data = np.array(data)
labels = np.array(labels)
# Mengambil 12 data dan label pertama dari setiap kelas
data_samples = []

```

```

labels_samples = []
num_samples_per_class = 3

for class_index in range(len(classes)):
    class_data = data[labels == class_index][:num_samples_per_class]
    class_labels = labels[labels == class_index][:num_samples_per_class]
    data_samples.extend(class_data)
    labels_samples.extend(class_labels)

# Membuat grid sesuai dengan jumlah kelas
fig, axes = plt.subplots(len(classes), num_samples_per_class, figsize=(10, 10))

# Loop untuk setiap kelas dan sampel
for i, ax_row in enumerate(axes):
    for j, ax in enumerate(ax_row):
        # Menampilkan gambar
        image = data_samples[i * num_samples_per_class + j]
        ax.imshow(image, cmap='gray')
        ax.axis('off')
        # Menampilkan label
        ax.set_title(classes[labels_samples[i * num_samples_per_class + j]])

# Menampilkan plot
plt.tight_layout()
plt.show()

# Membagi data menjadi set pelatihan, validasi, dan pengujian
seed = np.random.randint(0, 1000)
X_train, X_test_val, y_train, y_test_val = train_test_split(data, labels,
    test_size=0.2, random_state=seed)
X_test, X_val, y_test, y_val = train_test_split(X_test_val, y_test_val, test_size=0.5,
    random_state=seed)

# Menambahkan dimensi tambahan pada data
X_train = np.expand_dims(X_train, axis=-1)
X_val = np.expand_dims(X_val, axis=-1)
X_test = np.expand_dims(X_test, axis=-1)

print("Jumlah data pelatihan:", len(X_train))
print("Jumlah data validasi:", len(X_val))
print("Jumlah data pengujian:", len(X_test))
# Menghitung jumlah data untuk setiap kelas
class_count = [np.sum(labels == i) for i in range(len(classes))]

# Membuat diagram batang
plt.bar(classes, class_count)

```

```

# Menambahkan label pada sumbu x dan y
plt.xlabel('Kelas')
plt.ylabel('Jumlah Data')

# Menambahkan judul diagram
plt.title('Jumlah Data Tiap Kelas')

# Menampilkan angka jumlah data pada setiap batang
for i, count in enumerate(class_count):
    plt.text(i, count, str(count), ha='center', va='bottom')

# Menampilkan diagram batang
plt.show()
# Mendefinisikan Optimizer
optimizer = tf.keras.optimizers.SGD(learning_rate=0.002)

# Membuat the model
model = Sequential()
model.add(Rescaling(1/255, input_shape=(IMG_HEIGHT, IMG_WIDTH, 3)))

base_model = DenseNet169(include_top=False, weights='imagenet',
input_shape=(IMG_HEIGHT, IMG_WIDTH, 3))
model.add(base_model)

model.add(tf.keras.layers.GlobalAveragePooling2D())

model.add(tf.keras.layers.Flatten())
model.add(tf.keras.layers.Dense(128, activation="relu"))
model.add(tf.keras.layers.Dense(64, activation="relu"))
model.add(tf.keras.layers.Dense(4, activation="softmax"))

model.compile(loss="sparse_categorical_crossentropy", optimizer=optimizer,
metrics=["accuracy"])

model.summary()
# Mengukur waktu komputasi untuk keseluruhan proses pelatihan
start_time = time.time()

# Melatih model dengan menggunakan data pelatihan dan validasi
history = model.fit(X_train, y_train, batch_size=64, epochs=50,
validation_data=(X_val, y_val))
# Menghitung waktu pelatihan
training_time = time.time() - start_time
print(f"Waktu komputasi untuk pelatihan: {training_time:.2f} detik")
# Perform label encoding
label_encoder = LabelEncoder()

```

```

y_train_encoded = label_encoder.fit_transform(y_train)
y_val_encoded = label_encoder.transform(y_val)
y_test_encoded = label_encoder.transform(y_test)
get_ac = history.history['accuracy']
get_loss = history.history['loss']
val_acc = history.history['val_accuracy']
val_loss = history.history['val_loss']

# Plot akurasi dan loss pada setiap epoch
sns.set()
fig, axes = plt.subplots(1, 2, figsize=(12, 4))

axes[0].plot(history.epoch, history.history['accuracy'], label='Training Accuracy')
axes[0].plot(history.epoch, history.history['val_accuracy'], label='Validation Accuracy')
axes[0].set_xlabel('Epoch')
axes[0].set_ylabel('Accuracy')
axes[0].set_title('Training and Validation Accuracy')
axes[0].legend()

axes[1].plot(history.epoch, history.history['loss'], label='Training Loss')
axes[1].plot(history.epoch, history.history['val_loss'], label='Validation Loss')
axes[1].set_xlabel('Epoch')
axes[1].set_ylabel('Loss')
axes[1].set_title('Training and Validation Loss')
axes[1].legend()

# Menandai posisi best accuracy dan best loss
best_accuracy_epoch = np.argmax(history.history['val_accuracy'])
best_accuracy = history.history['val_accuracy'][best_accuracy_epoch]
best_loss_epoch = np.argmin(history.history['val_loss'])
best_loss = history.history['val_loss'][best_loss_epoch]

axes[0].plot(best_accuracy_epoch, best_accuracy, 'ro', markersize=8, label='Best Accuracy')
axes[0].annotate(f'Epoch: {best_accuracy_epoch+1}\nAccuracy: {best_accuracy:.4f}',
                xy=(best_accuracy_epoch, best_accuracy),
                xytext=(best_accuracy_epoch+10, best_accuracy-0.05),
                arrowprops=dict(facecolor='red', arrowstyle='->'),
                fontsize=10) # Ubah ukuran font disini
axes[0].legend()

axes[1].plot(best_loss_epoch, best_loss, 'ro', markersize=8, label='Best Loss')
axes[1].annotate(f'Epoch: {best_loss_epoch+1}\nLoss: {best_loss:.4f}',
                xy=(best_loss_epoch, best_loss),

```

```

xytext=(best_loss_epoch+10, best_loss+0.1),
arrowprops=dict(facecolor='red', arrowstyle='->'),
fontsize=10) # Ubah ukuran font disini
axes[1].legend()

plt.tight_layout()
plt.savefig('epoch_history_res5050.png')
plt.show()
# Evaluasi model pada data pengujian
y_test_pred = model.predict(X_test)
y_test_pred_classes = np.argmax(y_test_pred, axis=1)

accuracy = accuracy_score(y_test_encoded, y_test_pred_classes)
precision = precision_score(y_test_encoded, y_test_pred_classes,
average='macro')
recall = recall_score(y_test_encoded, y_test_pred_classes, average='macro')
f1 = f1_score(y_test_encoded, y_test_pred_classes, average='macro')

print("Akurasi Pengujian:", accuracy)
print("Presisi Pengujian:", precision)
print("Recall Pengujian:", recall)
print("F1 Score Pengujian:", f1)
# Evaluasi model pada data pengujian setiap jenis penyakit
y_test_pred = model.predict(X_test)
y_test_pred_classes = np.argmax(y_test_pred, axis=1)

accuracy = accuracy_score(y_test_encoded, y_test_pred_classes)

class_accuracy = {}
for i, category in enumerate(categories):
    class_accuracy[category] = accuracy_score(y_test_encoded[y_test_encoded ==
i], y_test_pred_classes[y_test_encoded == i])

print("Akurasi Pengujian: {:.2%}".format(accuracy))
print("Akurasi per Kelas:")
for category, acc in class_accuracy.items():
    print("- {}: {:.2%}".format(category, acc))
# Mengonversi label hasil prediksi menjadi nama label kelas
predicted_labels = label_encoder.inverse_transform(y_test_pred_classes)

# Mengonversi label sebenarnya menjadi nama label kelas
actual_labels = label_encoder.inverse_transform(y_test_encoded)

# Membuat confusion matrix
cm = confusion_matrix(actual_labels, predicted_labels)

```

```

# Menampilkan confusion matrix dalam bentuk heatmap
plt.figure(figsize=(5, 5))
sns.heatmap(cm, annot=True, fmt="d", cmap="Blues", cbar=False)
plt.xlabel("Prediksi")
plt.ylabel("Aktual")
plt.title("Confusion Matrix")
plt.xticks(ticks=np.arange(len(classes)), labels=classes, rotation=45)
plt.yticks(ticks=np.arange(len(classes)), labels=classes, rotation=0)
plt.show()

```

2. Kode DenseNet-169 dengan Kombinasi Fitur

```

import os
import cv2
import time
import random
import numpy as np
import pandas as pd
import seaborn as sns
import tensorflow as tf
import matplotlib.pyplot as plt
%matplotlib inline
from keras import layers
from tensorflow.keras import regularizers
from tensorflow.keras.models import Sequential
from sklearn.preprocessing import LabelEncoder
from sklearn.model_selection import train_test_split
from tensorflow.keras.applications import DenseNet169
from tensorflow.keras.callbacks import LearningRateScheduler
from tensorflow.keras.layers import Rescaling, BatchNormalization
from sklearn.metrics import confusion_matrix, accuracy_score, precision_score,
recall_score, f1_score
# Dataset Folder
dataset_path = "/kaggle/input/citrus-fruits-and-leaves-image-dataset/Fruits"

categories = os.listdir(dataset_path)
labels = [i for i in range(len(categories))]

label_dict = dict(zip(categories, labels))

print(label_dict)
print(categories)
print(labels)
classes = categories

```

```

# Fungsi untuk load dataset
def load_dataset(dataset_path):
    # Inisialisasi array data dan label
    data = []
    labels = []

    # Loop pada setiap subdirektori (kelas) dan file gambar di dalamnya
    for root, dirs, files in os.walk(dataset_path):
        for file in files:
            # Filter hanya file gambar dengan ekstensi .png atau .jpg
            if file.endswith(".png") or file.endswith(".jpg"):
                # Ambil path file gambar
                image_path = os.path.join(root, file)

                # Ambil label kelas dari nama subfolder
                label = os.path.basename(root)

                # Cek apakah label kelas termasuk dalam daftar classes
                if label in classes:
                    # Load gambar menggunakan fungsi preprocess_image
                    image = preprocess_image(image_path)

                    # Simpan gambar dan label ke dalam array
                    data.append(image)
                    labels.append(classes.index(label))

    return data, labels
IMG_HEIGHT = 224
IMG_WIDTH = 224

def preprocess_image(image_path):
    # Load gambar
    image = cv2.imread(image_path)

    # Resize gambar ke ukuran yang diinginkan
    image = cv2.resize(image, (IMG_HEIGHT, IMG_WIDTH))

    # Ubah citra warna ke skema warna RGB
    image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)

    # Buat objek CLAHE dengan konfigurasi yang diinginkan
    clahe = cv2.createCLAHE(clipLimit=5.0, tileGridSize=(16, 16))

    # Split saluran warna
    r, g, b = cv2.split(image)

```

```

# Terapkan CLAHE pada setiap saluran warna
r = clahe.apply(r)
g = clahe.apply(g)
b = clahe.apply(b)

# Gabungkan kembali saluran warna RGB setelah CLAHE
image = cv2.merge([r, g, b])

# Ambil komponen V (Value) dari ruang warna HSV
image_hsv = cv2.cvtColor(image, cv2.COLOR_RGB2HSV)

# Ubah citra warna ke skema warna YCbCr
image = cv2.cvtColor(image_hsv, cv2.COLOR_RGB2YCrCb)
# image = image.astype('float32') / 255.0

# Ubah citra warna ke skema warna LAB
image = cv2.cvtColor(image, cv2.COLOR_RGB2LAB)
image = image.astype('float32') / 255.0

return image

# Load dataset
data, labels = load_dataset(dataset_path)
# Mengubah data dan label menjadi array numpy
data = np.array(data)
labels = np.array(labels)
# Mengambil 12 data dan label pertama dari setiap kelas
data_samples = []
labels_samples = []
num_samples_per_class = 3

for class_index in range(len(classes)):
    class_data = data[labels == class_index][:num_samples_per_class]
    class_labels = labels[labels == class_index][:num_samples_per_class]
    data_samples.extend(class_data)
    labels_samples.extend(class_labels)

# Membuat grid sesuai dengan jumlah kelas
fig, axes = plt.subplots(len(classes), num_samples_per_class, figsize=(10, 10))

# Loop untuk setiap kelas dan sampel
for i, ax_row in enumerate(axes):
    for j, ax in enumerate(ax_row):
        # Menampilkan gambar

```

```

image = data_samples[i * num_samples_per_class + j]
ax.imshow(image, cmap='gray')
ax.axis('off')
# Menampilkan label
ax.set_title(classes[labels_samples[i * num_samples_per_class + j]])

# Menampilkan plot
plt.tight_layout()
plt.show()
# Membagi data menjadi set pelatihan, validasi, dan pengujian
seed = np.random.randint(0, 1000)
X_train, X_test_val, y_train, y_test_val = train_test_split(data, labels,
test_size=0.2, random_state=seed)
X_test, X_val, y_test, y_val = train_test_split(X_test_val, y_test_val, test_size=0.5,
random_state=seed)

# Menambahkan dimensi tambahan pada data
X_train = np.expand_dims(X_train, axis=-1)
X_val = np.expand_dims(X_val, axis=-1)
X_test = np.expand_dims(X_test, axis=-1)

print("Jumlah data pelatihan:", len(X_train))
print("Jumlah data validasi:", len(X_val))
print("Jumlah data pengujian:", len(X_test))
# Menghitung jumlah data untuk setiap kelas
class_count = [np.sum(labels == i) for i in range(len(classes))]

# Membuat diagram batang
plt.bar(classes, class_count)

# Menambahkan label pada sumbu x dan y
plt.xlabel('Kelas')
plt.ylabel('Jumlah Data')

# Menambahkan judul diagram
plt.title('Jumlah Data Tiap Kelas')

# Menampilkan angka jumlah data pada setiap batang
for i, count in enumerate(class_count):
    plt.text(i, count, str(count), ha='center', va='bottom')

# Menampilkan diagram batang
plt.show()
# Mendefinisikan Optimizer
optimizer = tf.keras.optimizers.SGD(learning_rate=0.002)

```

```

# Membuat the model
model = Sequential()
model.add(Rescaling(1/255, input_shape=(IMG_HEIGHT, IMG_WIDTH, 3)))

base_model = DenseNet169(include_top=False, weights='imagenet',
input_shape=(IMG_HEIGHT, IMG_WIDTH, 3))
model.add(base_model)

model.add(tf.keras.layers.GlobalAveragePooling2D())

model.add(tf.keras.layers.Flatten())
model.add(tf.keras.layers.Dense(128, activation="relu"))
model.add(tf.keras.layers.Dense(64, activation="relu"))
model.add(tf.keras.layers.Dense(4, activation="softmax"))

model.compile(loss="sparse_categorical_crossentropy", optimizer=optimizer,
metrics=["accuracy"])

model.summary()
# Mengukur waktu komputasi untuk keseluruhan proses pelatihan
start_time = time.time()

# Melatih model dengan menggunakan data pelatihan dan validasi
history = model.fit(X_train, y_train, batch_size=64, epochs=50,
validation_data=(X_val, y_val))
# Menghitung waktu pelatihan
training_time = time.time() - start_time
print(f"Waktu komputasi untuk pelatihan: {training_time:.2f} detik")
# Perform label encoding
label_encoder = LabelEncoder()
y_train_encoded = label_encoder.fit_transform(y_train)
y_val_encoded = label_encoder.transform(y_val)
y_test_encoded = label_encoder.transform(y_test)
get_ac = history.history['accuracy']
get_loss = history.history['loss']
val_acc = history.history['val_accuracy']
val_loss = history.history['val_loss']
# Plot akurasi dan loss pada setiap epoch
sns.set()
fig, axes = plt.subplots(1, 2, figsize=(12, 4))

axes[0].plot(history.epoch, history.history['accuracy'], label='Training Accuracy')
axes[0].plot(history.epoch, history.history['val_accuracy'], label='Validation Accuracy')
axes[0].set_xlabel('Epoch')
axes[0].set_ylabel('Accuracy')

```

```

axes[0].set_title('Training and Validation Accuracy')
axes[0].legend()

axes[1].plot(history.epoch, history.history['loss'], label='Training Loss')
axes[1].plot(history.epoch, history.history['val_loss'], label='Validation Loss')
axes[1].set_xlabel('Epoch')
axes[1].set_ylabel('Loss')
axes[1].set_title('Training and Validation Loss')
axes[1].legend()

# Menandai posisi best accuracy dan best loss
best_accuracy_epoch = np.argmax(history.history['val_accuracy'])
best_accuracy = history.history['val_accuracy'][best_accuracy_epoch]
best_loss_epoch = np.argmin(history.history['val_loss'])
best_loss = history.history['val_loss'][best_loss_epoch]

axes[0].plot(best_accuracy_epoch, best_accuracy, 'ro', markersize=8, label='Best Accuracy')
axes[0].annotate(f'Epoch: {best_accuracy_epoch+1} \nAccuracy: {best_accuracy:.4f}',
                xy=(best_accuracy_epoch, best_accuracy),
                xytext=(best_accuracy_epoch+10, best_accuracy-0.05),
                arrowprops=dict(facecolor='red', arrowstyle='->'),
                fontsize=10) # Ubah ukuran font disini
axes[0].legend()

axes[1].plot(best_loss_epoch, best_loss, 'ro', markersize=8, label='Best Loss')
axes[1].annotate(f'Epoch: {best_loss_epoch+1} \nLoss: {best_loss:.4f}',
                xy=(best_loss_epoch, best_loss),
                xytext=(best_loss_epoch+10, best_loss+0.1),
                arrowprops=dict(facecolor='red', arrowstyle='->'),
                fontsize=10) # Ubah ukuran font disini
axes[1].legend()

plt.tight_layout()
plt.savefig('epoch_history_res5050.png')
plt.show()
# Evaluasi model pada data pengujian
y_test_pred = model.predict(X_test)
y_test_pred_classes = np.argmax(y_test_pred, axis=1)

accuracy = accuracy_score(y_test_encoded, y_test_pred_classes)
precision = precision_score(y_test_encoded, y_test_pred_classes,
                           average='macro')
recall = recall_score(y_test_encoded, y_test_pred_classes, average='macro')
f1 = f1_score(y_test_encoded, y_test_pred_classes, average='macro')

```

```

print("Akurasi Pengujian:", accuracy)
print("Presisi Pengujian:", precision)
print("Recall Pengujian:", recall)
print("F1 Score Pengujian:", f1)
# Evaluasi model pada data pengujian setiap jenis penyakit
y_test_pred = model.predict(X_test)
y_test_pred_classes = np.argmax(y_test_pred, axis=1)

accuracy = accuracy_score(y_test_encoded, y_test_pred_classes)

class_accuracy = {}
for i, category in enumerate(categories):
    class_accuracy[category] = accuracy_score(y_test_encoded[y_test_encoded ==
i], y_test_pred_classes[y_test_encoded == i])

print("Akurasi Pengujian: {:.2%}".format(accuracy))
print("Akurasi per Kelas:")
for category, acc in class_accuracy.items():
    print("- {}: {:.2%}".format(category, acc))
# Mengonversi label hasil prediksi menjadi nama label kelas
predicted_labels = label_encoder.inverse_transform(y_test_pred_classes)

# Mengonversi label sebenarnya menjadi nama label kelas
actual_labels = label_encoder.inverse_transform(y_test_encoded)

# Membuat confusion matrix
cm = confusion_matrix(actual_labels, predicted_labels)

# Menampilkan confusion matrix dalam bentuk heatmap
plt.figure(figsize=(5, 5))
sns.heatmap(cm, annot=True, fmt="d", cmap="Blues", cbar=False)
plt.xlabel("Prediksi")
plt.ylabel("Aktual")
plt.title("Confusion Matrix")
plt.xticks(ticks=np.arange(len(classes)), labels=classes, rotation=45)
plt.yticks(ticks=np.arange(len(classes)), labels=classes, rotation=0)
plt.show()

```