Studi Komparasi Arsitektur GoogLeNet dan CNN untuk Klasifikasi Gambar Penyakit Daun Mangga

M. Alfan Tsalits[1], Aries Dwi Indriyanti[2]

[1,2] Sistem Informasi, Universitas Negeri Surabaya
[1]mtsalits.21054@mhs.unesa.ac.id
[2]ariesdwi@unesa.ac.id

***Abstrak*— Penyakit daun mangga secara signifikan mempengaruhi produktivitas pertanian, sehingga diperlukan sistem identifikasi otomatis dan efisien. Penelitian ini bertujuan untuk melakukan studi komparasi antara arsitektur *Convolutional Neural Network* (CNN) konvensional dengan arsitektur GoogLeNet dalam mengklasifikasikan delapan kategori kondisi daun mangga. Metodologi penelitian ini menggabungkan pendekatan SEMMA (*Sample, Explore, Modify, Model, Assess*) untuk tahapan pengembangan model, serta metode RAD (*Rapid Application Development*) untuk tahapan implementasi sistem *dashboard* berbasis *website*. Hasil eksperimen menunjukkan bahwa GoogLeNet mengungguli CNN konvensional, mencapai akurasi validasi yang lebih tinggi sebesar 90,68% dan *loss* validasi yang lebih rendah sebesar 0,2580, dibandingkan dengan akurasi CNN sebesar 89,55% dan *loss* 0,3309. Selain aspek akurasi, GoogLeNet jauh lebih efisien secara komputasi karena hanya menggunakan 2,87 juta parameter sementara CNN konvensional memerluka hingga 25,8 juta parameter. Melalui implementasi sistem yang telah dibangun, model terbaik mampu menyajikan hasil diagnosis penyakit dan nilai *confidence score* secara *real-time*.**

***Kata Kunci*— Penyakit Daun Mangga, *Deep Learning*, *Convolutional Neural Network*, *GoogLeNet*, Klasifikasi Citra.**

I. PENDAHULUAN

Pesatnya Kemajuan teknologi di bidang kecerdasan buatan, khususnya dalam cabang *deep learning*, telah membawa dampak signifikan dalam penyelesaian berbagai permasalahan berbasis data citra, salah satunya adalah klasifikasi gambar [1]. Salah satu teknik yang paling banyak digunakan Adalah CNN (*Convolutional Neural Network*), yang menjadi fondasi utama dalam pengolahan dan pengenalan pola visual [2]. CNN memiliki kemampuan dalam mengekstraksi fitur dari citra secara otomatis, sehingga banyak dimanfaatkan dalam bidang seperti medis, otomotif, pertanian, dan keamanan. Seiring meningkatnya kebutuhan terhadap pengolahan data visual yang kompleks, berbagai arsitektur CNN telah dikembangkan untuk meningkatkan akurasi dan efisiensi dalam klasifikasi citra [2].

Salah satu arsitektur paling awal adalah LeNet-5, yang dikembangkan oleh Yann LeCun pada tahun 1998 untuk system pengenalan dokumen [2]. *LeNet* menjadi dasar dari perkembangan CNN modern dengan memperkenalkan konsep *layer* konvolusi dan *pooling*. Pada tahun 2012, AlexNet merevolusi bidang visi komputer melalui penggunaan jaringan yang lebih dalam dan penggunaan GPU dalam pelatihan. Setelah itu, VGGNet hadir dengan arsitektur yang lebih dalam namun memiliki jumlah parameter yang sangat besar [3]. Perkembangan ini akhirnya mengarah pada desain arsitektur yang lebih efisien namun tetap dalam, yaitu GoogLeNet (*Inception V1*), yang menjadi titik penting dalam evolusi CNN modern [4].

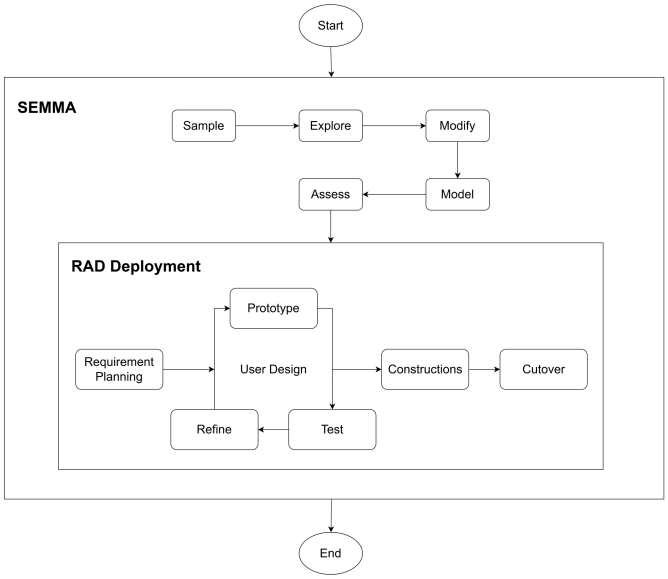
GoogLeNet diperkenalkan pada tahun 2014 dalam kompetisi *ImageNet Large Scale Visual Recognition Challenge* (ILSVRC) [4]. Arsitektur ini merupakan salah satu model terbaik dalam hal akurasi dan efisiensi melalui pendekatan modular yang kompleks. Keunggulan GoogLeNet terletak pada *Inception Module*, yang menjalankan beberapa operasi konvolusional secara paralel dengan ukuran filter yang berbeda, seperti 1×1, 3×3, dan 5×5, serta *max-pooling* [4]. Operasi 1×1 konvolusi diterapkan untuk mengurangi dimensi dan mengoptimalkan efisiensi komputasi. Desain ini menjadikan GoogLeNet lebih ringan dari segi parameter namun tetap mampu menangkap kompleksitas visual secara mendalam, berbeda dengan arsitektur lain seperti VGGNet, AlexNet maupun CNN konvensional yang cenderung padat parameter [1].

Berkat efisiensinya, GoogLeNet banyak digunakan dalam berbagai studi klasifikasi citra, termasuk untuk mendeteksi penyakit dari tanaman mangga. Penelitian terdahulu terkait klasifikasi penyakit daun mangga berbasis *deep learning* masih terbatas, terutama untuk *dataset* multi-kelas. Sebagian besar penelitian terdahulu hanya mengklasifikasikan daun ke dalam dua kategori (sehat dan berpenyakit) atau hanya menggunakan arsitektur CNN konvensional.

Penelitian ini mengisi celah tersebut dengan membandingkan performa GoogLeNet terhadap CNN konvensional dalam mengklasifikasikan delapan kelas citra daun mangga (tujuh kelas daun berpenyakit dan satu kelas daun sehat). Pemilihan GoogLeNet didasarkan pada keunggulan *Inception Module* dalam efisiensi parameter dan kemampuannya menangkap fitur visual secara mendalam [1]. Penelitian ini diharapkan dapat memberikan kontribusi ilmiah terhadap pengembangan metode klasifikasi citra multi-kelas di bidang pertanian, serta mengimplementasikan model terbaik ke dalam *dashboard* website untuk mendukung sistem klasifikasi penyakit daun mangga.

II. METODE PENELITIAN

Metode penelitian ini disusun secara sistematis untuk menggambarkan alur pengembangan model klasifikasi penyakit daun mangga, mulai dari pengumpulan data hingga implementasi ke dalam sistem dashboard, seperti yang tertera pada Gambar 1.



Gbr. 1 Kerangka Penelitian

Berdasarkan Gambar 1, penelitian ini menggabungkan metode pengembangan data mining SEMMA (*Sample*, *Explore*, *Modify*, *Model*, *Assess*) untuk fokus pada pemodelan dan metode RAD (*Rapid Application Development*) untuk pengembangan aplikasi dashboard.

A. *Sample*

Proses pengambilan data dilakukan sebagai dasar dalam membangun model klasifikasi. Data yang digunakan dalam penelitian ini diperoleh dari platform publik Kaggle, sebuah situs penyedia data dan kompetisi dalam bidang data science dan *machine learning*. *Dataset* ini terdiri dari 4.400 gambar daun mangga yang terbagi ke dalam 8 kelas yang berbeda.

B. *Explore*

Dataset yang digunakan dalam penelitian ini mencakup total 4.400 gambar yang terdiri dari 8 kelas berbeda dengan proporsi yang seimbang seperti ditunjukkan pada Tabel I.

TABEL I
JUMLAH DATASET PER KELAS

No	Nama Kelas	Jumlah Gambar
1	*Anthracnose*	550
2	*Bacterial Canker*	550
3	*Cutting Weevil*	550
4	*Die Back*	550
5	*Gall Midge*	550
6	*Healthy*	550
7	*Powdery Mildew*	550
8	*Sooty Mould*	550
Total		**4.400**

C. *Modify*

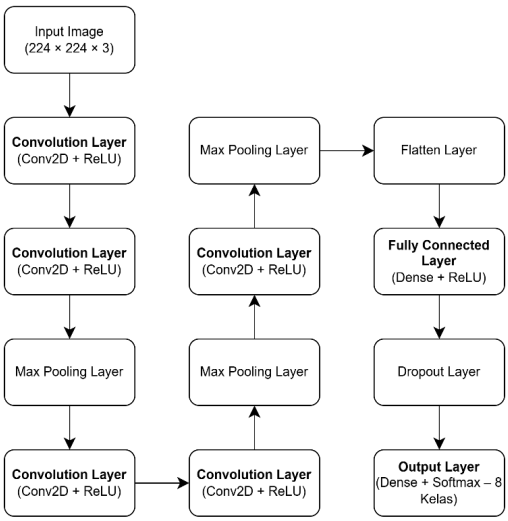
Sebelum melakukan pemodelan, data gambar perlu dimodifikasi melalui tahap pre-proses untuk memastikan data dalam kondisi optimal dan seragam. Tahap *preprocessing* merupakan proses penting untuk meningkatkan kualitas data sehingga model dapat melakukan proses pembelajaran secara lebih optimal sebelum dilakukan pemodelan [5]. Berdasarkan alur pra-pemrosesan pada penelitian ini, transformasi citra dilakukan melalui beberapa tahapan teknis sebagai berikut:

1) *Resizing*: Mengingat data asli memiliki resolusi yang sangat bervariasi (berkisar antara 18 KB hingga 3,83 MB), seluruh citra dikonversi menjadi ukuran tetap 224×224 piksel. Hal ini dilakukan agar sesuai dengan standar dimensi input arsitektur GoogLeNet dan mencegah terjadinya *shape mismatch* pada jaringan.
2) *Color Formatting*: Untuk menjaga konsistensi informasi visual, dilakukan konversi seluruh citra ke format warna RGB. Langkah ini memastikan seluruh kanal warna seragam, terutama jika terdapat data asli yang memiliki format *grayscale* atau BGR
3) *Labeling*: Proses pelabelan dilakukan secara otomatis berdasarkan struktur direktori dataset. Nama sub-folder seperti *Anthracnose*, *Healthy* dan lainnya diekstraksi menjadi label kategori tiap kelas yang dapat dikenali oleh model selama proses pelatihan dan pengujian.
4) *Data Splitting*: Pembagian data menjadi training dan testing dilakukan menggunakan rasio 80:20, yang merupakan praktik umum dalam pengembangan model machine learning untuk menjaga kemampuan generalisasi model [6]. Sebanyak 80% data (3.520 citra) dialokasikan sebagai data latih (*training*) untuk membantu model mempelajari pola penyakit, sementara 20% sisanya (880 citra) digunakan sebagai data uji (*testing*) untuk validasi independen guna memantau performa dan mencegah *overfitting*.

D. *Model*

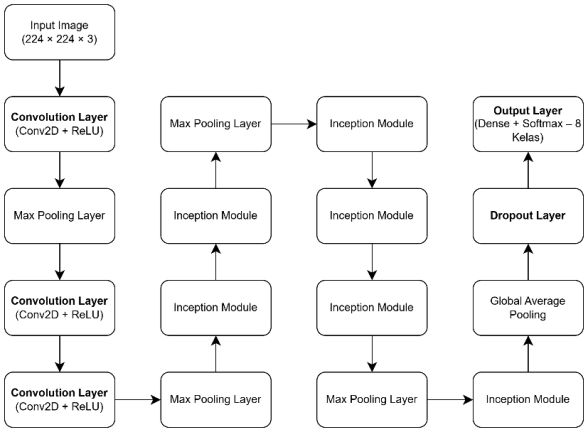
Tahap *Model* melibatkan perancangan dan pelatihan arsitektur *deep learning* untuk mengklasifikasikan citra daun mangga ke dalam delapan kategori. Pada penelitian ini, dilakukan pengembangan serta perbandingan antara dua model utama, yaitu CNN (*Convolutional Neural Network*) konvensional dan GoogLeNet (*Inception V1*). Kedua arsitektur ini dibangun secara manual menggunakan *framework* TensorFlow untuk memberikan kendali penuh terhadap struktur jaringan dan menjamin perbandingan performa yang adil.

1) *Arsitektur CNN Konvensional:* Model ini dibangun sebagai *baseline* dengan susunan sekuensial yang terdiri dari lapisan konvolusi untuk ekstraksi fitur visual, lapisan *pooling* untuk reduksi dimensi, dan lapisan *fully connected* untuk klasifikasi akhir [7], [8]. Secara teknis, model ini menggunakan filter konvolusi berukuran 3×3 dengan fungsi aktivasi ReLU, diikuti oleh *Global Average Pooling* untuk menekan jumlah parameter sebelum masuk ke lapisan *output*. Visualisasi susunan lapisan pada model CNN konvensional ditunjukkan pada Gambar 2.



Gbr. 2 Alur Proses Arsitektur CNN

2) *Arsitektur GoogLeNet (Inception V1):* Arsitektur ini dikembangkan dengan memodifikasi struktur asli GoogLeNet menjadi lebih efisien dengan mengimplementasikan enam *Inception Module* yang disusun secara bertumpuk. Inti dari pengembangan ini adalah penggunaan konvolusi paralel berukuran 1×1, 3×3, dan 5×5 dalam satu modul. Penggunaan konvolusi 1×1 berperan sebagai *dimensionality reduction* untuk menjaga beban komputasi tetap ringan [4]. Arsitektur ini diakhiri dengan lapisan *dropout* untuk mencegah *overfitting* sebelum lapisan proyeksi akhir. Diagram blok arsitektur GoogLeNet secara keseluruhan ditampilkan pada Gambar 3.



Gbr. 3 Alur Proses Arsitektur GoogLeNet

Selama pelatihan, performa model dievaluasi secara berkala pada *validation set* di setiap *epoch* guna memantau stabilitas dan mencegah terjadinya *overfitting*. Model terbaik dipilih berdasarkan nilai akurasi validasi tertinggi dan tren kurva *loss* yang stabil.

E. *Assess*

Penilaian perbandingan hasil training antar model pada penelitian ini menggunakan metrik evaluasi yang mencakup akurasi, presisi, *recall*, dan *F1-score*, yang merupakan standar pengukuran pada permasalahan klasifikasi *multi-class*. Akurasi digunakan untuk mengukur tingkat ketepatan prediksi model secara keseluruhan. Presisi dan *recall* digunakan untuk mengevaluasi kemampuan model dalam mengenali masing-masing kelas penyakit secara lebih spesifik, sementara *F1-score* dimanfaatkan sebagai metrik gabungan yang menyeimbangkan nilai presisi dan *recall*.

Selain evaluasi numerik, alat bantu analitik berupa *confusion matrix* digunakan untuk mengidentifikasi pola kesalahan klasifikasi pada masing-masing arsitektur. Visualisasi ini memetakan hubungan antara label sebenarnya (*true label*) dan hasil prediksi model (*predicted label*), sehingga dapat menganalisis apakah terdapat kelas penyakit dengan kemiripan visual yang sering mengalami kesalahan prediksi.

F. *Deployment*

Tahap *deployment* dilakukan untuk mengimplementasikan model *deep learning* terbaik ke dalam sebuah sistem *dashboard* berbasis *website*. Pengembangan sistem ini menggunakan metode *Rapid Application Development* (RAD) yang menekankan pada kecepatan dan efisiensi melalui pembuatan prototipe [9], [10]. Tahapan RAD dalam penelitian ini mencakup beberapa fase sebagai berikut:

- *Requirement Planning:* Tahap ini bertujuan untuk mengidentifikasi kebutuhan sistem yang diperlukan agar *dashboard* berfungsi sesuai tujuan penelitian [9]. Kebutuhan fungsional mencakup fitur unggah citra daun, proses klasifikasi otomatis, serta penyajian hasil prediksi label dan nilai *confidence*. Kebutuhan non-fungsional memastikan sistem dapat diakses secara optimal melalui peramban *web*.
- *User Design:* Fokus pada perancangan alur interaksi pengguna secara iteratif melalui proses *prototype*, *test*, dan *refine*. Rancangan sistem divisualisasikan menggunakan *activity diagram* untuk memodelkan alur kerja sistem untuk memberikan kenyamanan bagi pengguna [10].
- *Construction:* Merupakan fase implementasi di mana rancangan konseptual diubah menjadi sistem yang siap dijalankan [9]. Proses ini melibatkan pengkodean (*coding*) serta integrasi antara model klasifikasi yang telah dilatih dengan antarmuka *website* menggunakan bahasa pemrograman Python dan *framework* Flask.
- *Cutover:* Tahap ini berperan sebagai pengecekan akhir untuk memastikan stabilitas sistem. Pengujian dilakukan dengan menjalankan sistem secara menyeluruh guna memverifikasi bahwa proses *input* citra hingga tampilan *output* prediksi berjalan tanpa kendala teknis.

III. HASIL DAN PEMBAHASAN

A. *Performa Pelatihan Model*

Proses pelatihan untuk arsitektur CNN (*Convolutional Neural Network*) konvensional dan GoogLeNet dilakukan selama 20 *epoch* menggunakan *optimizer* Adam dan fungsi *loss*

categorical cross-entropy. Kedua model dievaluasi menggunakan pembagian data 80:20. Peningkatan akurasi pelatihan (*training accuracy*) dan validasi (*validation accuracy*), serta nilai kerugian (*loss*) untuk kedua model, menunjukkan kemampuan pembelajarannya secara jelas.

Tabel II berikut menyajikan log pelatihan untuk model CNN. Model tersebut menunjukkan peningkatan akurasi yang stabil, hingga akhirnya mencapai akurasi validasi tertinggi sebesar 89,55% pada *epoch* 19 dengan nilai *validation loss* sebesar 0,3309.

TABEL III
HASIL PELATIHAN MODEL CNN

Epoch	Train_acc	Train_loss	Val_acc	Val_loss
1	26.90%	1.8309	63.30%	1.1422
2	61.52%	1.0959	74.32%	0.7657
3	72.34%	0.7664	80.57%	0.6234
4	78.51%	0.6422	82.61%	0.5405
5	82.46%	0.5293	83.18%	0.5066
6	84.07%	0.4789	86.70%	0.4127
7	86.02%	0.4112	86.93%	0.4015
8	86.07%	0.4110	85.91%	0.4010
9	87.07%	0.3622	85.23%	0.4068
10	88.98%	0.3041	86.82%	0.4007
11	90.49%	0.2731	88.64%	0.3443
12	91.06%	0.2635	88.64%	0.3472
13	90.93%	0.2400	86.70%	0.3798
14	91.54%	0.2261	87.61%	0.3774
15	92.11%	0.2116	88.41%	0.3176
16	94.40%	0.1663	87.84%	0.3663
17	93.52%	0.1804	88.64%	0.3445
18	93.75%	0.1775	89.09%	0.3508
19	95.45%	0.1289	89.55%	0.3309
20	95.85%	0.1267	89.09%	0.3336

Seperti yang ditunjukkan pada Tabel II, model CNN menunjukkan konvergensi yang stabil, di mana akurasi pelatihan dan validasi mengikuti tren yang serupa. Akurasi validasi secara bertahap meningkat dan stabil di sekitar angka 89%, menunjukkan bahwa model dapat melakukan generalisasi dengan baik tanpa terjadi *overfitting* yang signifikan. Hasil pelatihan model GoogLeNet ditunjukkan pada Tabel III.

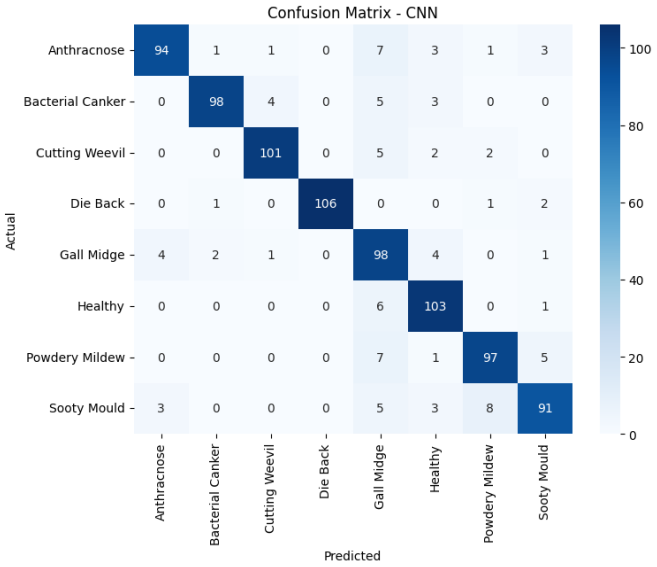
TABEL IIIII
HASIL PELATIHAN MODEL GOOLENET

Epoch	Train_acc	Train_loss	Val_acc	Val_loss
1	14.39%	2.0545	26.82%	1.7370
2	33.36%	1.6423	52.50%	1.3848
3	49.26%	1.3934	57.61%	1.2827
4	60.85%	1.0902	61.02%	0.9620
5	72.72%	0.8062	66.02%	0.8489
6	69.81%	0.7837	74.43%	0.6655
7	78.77%	0.5774	76.93%	0.6165
8	81.91%	0.5104	84.89%	0.4025
9	85.28%	0.4217	83.30%	0.4496
10	86.66%	0.3842	88.98%	0.3294
11	86.76%	0.3888	90.68%	0.2818
12	88.99%	0.3176	74.43%	0.7326
13	84.93%	0.4176	88.86%	0.3335
14	90.25%	0.2800	90.68%	0.2680
15	89.41%	0.2918	89.55%	0.3048
16	88.75%	0.2903	87.73%	0.3576
17	90.07%	0.2950	90.68%	0.2580
18	91.24%	0.2544	84.32%	0.4399
19	89.79%	0.2703	78.41%	0.5942
20	87.12%	0.3330	89.09%	0.2878

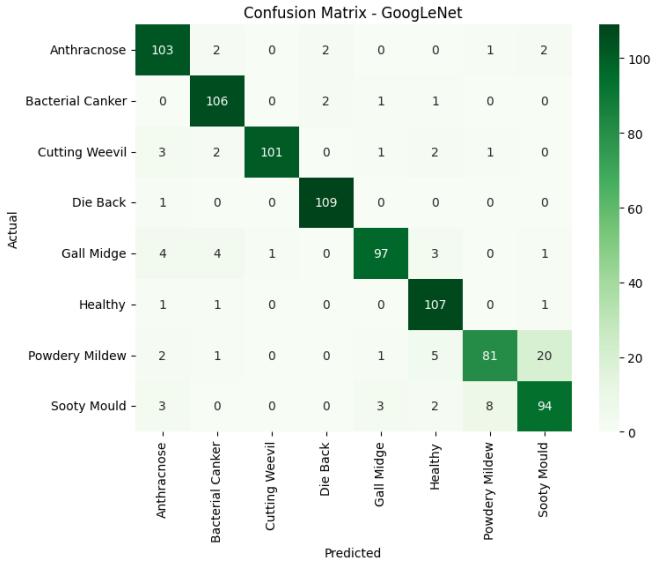
Tabel III menampilkan log pelatihan untuk GoogLeNet. Arsitektur ini menunjukkan perilaku pembelajaran yang lebih dinamis dan mencapai puncak akurasi validasinya sebesar 90,68% lebih awal, secara spesifik pada *epoch* 11, 14, dan 17. Nilai *validation loss* terendah yang diamati adalah 0,2580 pada *epoch* 17. Teknik penghentian dini (*early stopping*) diterapkan pada *epoch* ini untuk mencegah model mengalami *overfitting* sekaligus mempertahankan bobot (*weights*) yang paling optimal.

B. *Evaluasi Klasifikasi*

Untuk lebih memahami performa klasifikasi, *confusion matrix* dihasilkan untuk kedua model. Matriks ini memberikan distribusi rinci dari kelas-kelas yang diprediksi secara benar maupun salah. Hasil *confusion matrix* dari model CNN dan GoogLeNet ditunjukkan pada Gambar 4 dan Gambar 5.



Gbr. 4 *Confusion Matrix* CNN



Gbr. 5 *Confusion Matrix* GoogLeNet

Model GoogLeNet menunjukkan performa klasifikasi yang kuat pada hampir semua kelas penyakit, dengan nilai *true positive* yang tinggi pada kategori seperti *Healthy*, *Die Back*, dan *Bacterial Canker*. Namun, beberapa kesalahan klasifikasi (*misclassification*) masih terjadi, terutama antara kelas yang memiliki kemiripan visual yang tinggi seperti *Powdery Mildew* dan *Sooty Mould*. Kesalahan ini kemungkinan besar disebabkan oleh adanya kemiripan pola tekstur dan warna antara jenis penyakit tersebut.

Tahap evaluasi ini juga didukung secara kuantitatif oleh metrik evaluasi yang mencakup *Precision*, *Recall*, *F1-Score,* dan juga *Accuracy*, yang dihitung dari *dataset* pengujian sebagaimana ditunjukkan pada Tabel IV.

TABEL IV
HASIL PERBANDINGAN METRIKS EVALUASI

Model	***Accuracy***	***Precision***	***Recall***	***F1-score***
CNN	0.90	0.90	0.90	0.90
GoogLeNet	0.91	0.91	0.91	0.91

C. *Pembahasan*

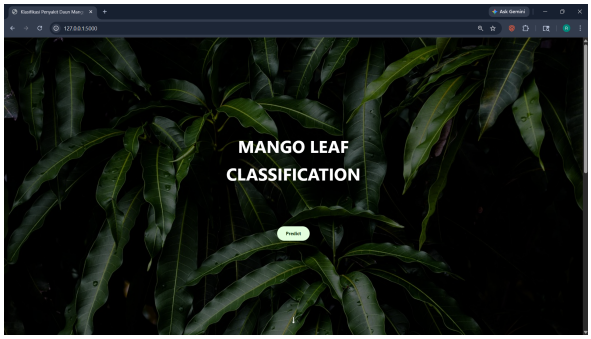
Hasil empiris mengonfirmasi bahwa GoogLeNet mengungguli CNN konvensional dalam hal akurasi validasi, dengan nilai akurasi tertinggi GoogLeNet mencapai 90,68% sedangkan CNN konvensional sebesar 89,55%, *validation loss* GoogLeNet 0,2580 berbanding dengan *validation loss* CNN sebesar 0,3309, dan perolehan *F1-Score* secara keseluruhan. Peningkatan performa ini utamanya dikaitkan dengan keunggulan struktural dari *Inception Module* di dalam GoogLeNet. Dengan memanfaatkan konvolusi paralel melalui ukuran filter yang bervariasi (1×1, 3×3, dan 5×5), GoogLeNet secara efektif mampu menangkap baik tekstur halus (*fine-grained textures*) seperti bercak kecil pada penyakit *Anthracnose*, maupun kelainan struktural yang lebih besar seperti kerusakan daun pada *Cutting Weevil*.

Selain itu, GoogLeNet mampu mencapai akurasi yang lebih tinggi ini dengan beban komputasi yang jauh lebih ringan. Model ini beroperasi dengan hanya sekitar 2,87 juta parameter, yang mana secara signifikan lebih sedikit dibandingkan dengan 25,8 juta parameter yang dibutuhkan oleh model CNN konvensional yang digunakan dalam studi ini. Penggunaan konvolusi 1×1 untuk mereduksi dimensi (*dimensionality reduction*) pada GoogLeNet berhasil membatasi biaya komputasi sistem sekaligus memaksimalkan kedalaman ekstraksi fitur, menjadikannya arsitektur yang sangat optimal untuk implementasi ke dalam alat bantu klasifikasi penyakit daun mangga berbasis web.

D. *Deployment*

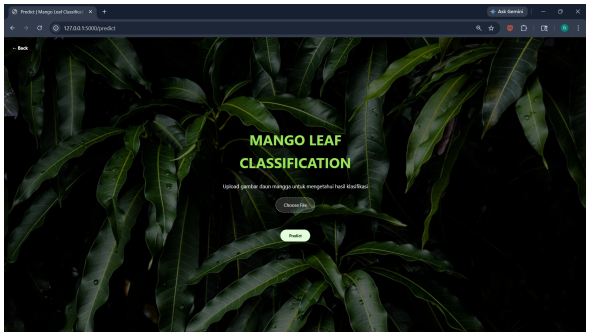
Tahap terakhir dari penelitian ini adalah implementasi model ke dalam sistem klasifikasi penyakit daun mangga berbasis *website*. Berdasarkan hasil evaluasi pada tahap sebelumnya, arsitektur GoogLeNet dipilih sebagai model utama yang diintegrasikan ke dalam sistem karena mencapai akurasi validasi tertinggi sebesar 90,68% dengan efisiensi jumlah parameter yang jauh lebih ringan dibandingkan CNN konvesional. Sistem *dashboard* ini dibangun menggunakan bahasa pemrograman python dengan *framework* flask untuk menghubungkan antarmuka pengguna dengan model klasifikasi yang telah disimpan dalam format *.h5*.

1) *Tampilan Dashboard Website:* Antarmuka sistem dirancang secara intuitif untuk memudahkan pengguna melakukan diagnosis mandiri. Alur interaksi sistem diawali dari halaman beranda (*home*), yang bertindak sebagai titik akses utama di mana pengguna dapat menekan tombol *predict* untuk diarahkan ke halaman klasifikasi, seperti ditunjukkan pada Gambar 6.



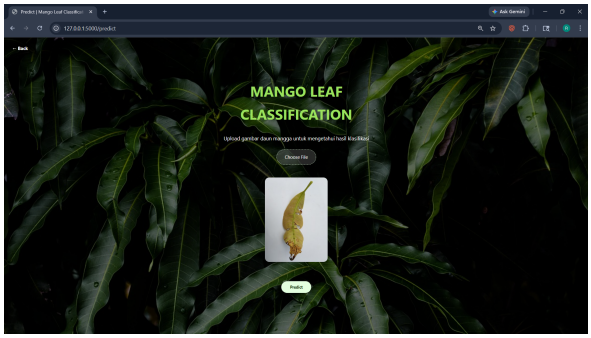
Gbr. 6 Tampilan halaman beranda

Setelah memasuki halaman prediksi (*predict page*), pengguna dihadapkan pada antarmuka yang memuat tombol *choose file* untuk memilih citra daun mangga dari penyimpanan perangkat pengguna. Pada halaman ini juga terdapat tombol *back* di pojok kiri atas untuk memberikan fleksibilitas navigasi kembali ke halaman utama, seperti ditunjukkan pada Gambar 7.



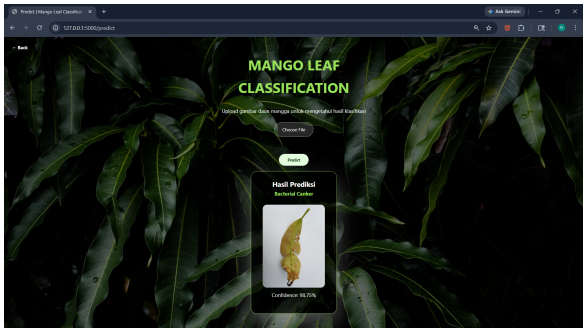
Gbr. 7 Tampilan halaman prediksi

Ketika citra telah dipilih, antarmuka sistem secara otomatis akan menampilkan *preview* gambar sebagai bentuk verifikasi visual bagi pengguna. Setelah memastikan citra sudah benar, pengguna cukup menekan tombol *predict* untuk memicu proses ekstraksi fitur dan klasifikasi oleh model pada sisi *backend*, seperti ditunjukkan pada Gambar 8.



Gbr. 8 Tampilan *preview* gambar

Sebagai hasil akhir, sistem akan langsung memunculkan diagnosis yang memuat label kelas kategori penyakit beserta *confidence score* yang merepresentasikan persentase tingkat keyakinan model terhadap prediksi tersebut, seperti ditunjukkan pada gambar 9.



Gbr. 9 Tampilan hasil klasifikasi beserta *confidence score*

2) *Testing Website:* Pengujian dilakukan menggunakan metode *blackbox testing* sebagai representasi tahap *cutover* pada metode RAD untuk memastikan stabilitas dan fungsionalitas sistem secara menyeluruh. Tabel V berikut menunjukkan scenario dan hasil uji coba fungsionalitas menggunakan *blackbox testing* untuk memastikan bahwa sistem mampu berjalan tanpa kendala teknis.

TABEL V
HASIL *BLACKBOX TESTING*

Halaman	Fitur	Hasil yang Diharapkan	Status
Beranda	Navigasi halaman klasifikasi	Sistem mengarahkan pengguna ke halaman prediksi setelah menekan tombol *predict*	✓
Prediksi	Fitur unggah citra (*upload file*)	Sistem menerima *file* masukan dari pengguna dan menampilkan *preview* gambar	✓
Prediksi	Navigasi kembali	Sistem mengembalikan pengguna ke halaman beranda saat menekan tombol *back*	✓
Prediksi	Tombol *predict* setelah memilih gambar	Sistem memproses gambar dan menampilkan hasil klasifikasi berupa keals penyakit beserta angka *confidence score*	✓

Berdasarkan hasil pengujian pada Tabel V, keseluruhan proses mulai dari navigasi halaman hingga pemunculan *output* diagnosis berjalan dengan lancar. Sistem terbukti mampu memproses data citra dan menampilkan hasil prediksi tanpa mengalami penundaan komputasi maupun *error* pada antarmuka.

IV. KESIMPULAN

Berdasarkan hasil penelitian dan eksperimen yang telah dilakukan, dapat ditarik beberapa kesimpulan sebagai berikut.

1. Pengembangan arsitektur GoogLeNet dan CNN konvensional untuk klasifikasi penyakit daun mangga berhasil dilakukan secara manual menggunakan *framework* TensorFlow dengan mengikuti tahapan metodologi SEMMA. Arsitektur CNN dibangun dengan struktur sekuensial lapisan konvolusi standar, sedangkan GoogLeNet dikembangkan secara khusus dengan mengintegrasikan enam *Inception Module* yang menerapkan mekanisme konvolusi paralel dan reduksi dimensi 1×1. Proses ini didukung oleh tahap *preprocessing* citra berupa *resizing* menjadi

224×224 piksel dan normalisasi data untuk mengoptimalkan ekstraksi fitur visual.

2. Hasil perbandingan menunjukkan bahwa arsitektur GoogLeNet memiliki performa yang lebih unggul dibandingkan CNN konvensional. GoogLeNet mencapai akurasi validasi tertinggi sebesar 90,68% sementara CNN konvensional mencapai akurasi tertinggi sebesar 89,55%. Selain itu, GoogLeNet terbukti jauh lebih efisien secara komputasi karena hanya memerlukan 2,87 juta parameter, dibandingkan dengan CNN konvensional yang memerlukan 25,8 juta parameter. Hal ini membuktikan bahwa penggunaan *Inception Module* sangat efektif dalam menangkap fitur penyakit yang kompleks dengan beban parameter yang lebih rendah.
3. Model GoogLeNet sebagai arsitektur yang lebih unggul telah berhasil diimplementasikan ke dalam *dashboard website* dengan metode RAD. Sistem ini mampu melakukan klasifikasi untuk menampilkan label kategori penyakit dan nilai *confidence score* setelah pengguna mengunggah citra daun.

UCAPAN TERIMA KASIH

Penulis mengucapkan puji syukur kepada Allah SWT atas limpahan rahmat dan karunia-Nya sehingga penelitian ini dapat diselesaikan dengan baik. Penulis juga menyampaikan terima kasih kepada dosen pembimbing atas bimbingan, arahan, serta masukan yang diberikan selama proses penelitian berlangsung. Selain itu, apresiasi turut diberikan kepada keluarga dan rekan-rekan yang telah memberikan dukungan, motivasi, dan bantuan selama penyusunan penelitian ini.

REFERENSI

[1] D. B. D. Hanggoro, "Analisis Komparatif Arsitektur Deep Learning Untuk Aplikasi Computer Vision: Studi Literature Review," *JUKTISI (Jurnal Komputer Teknologi Informasi Sistem Komputer)*, vol. 4, no. 2, hal. 1001-1008, Sep. 2025.

[2] J. Briliantio, N. Santosa, G. Ardian, dan L. Hakim, "Penerapan Convolutional Neural Network untuk Handwriting Recognition pada Aplikasi Belajar Aritmatika Dasar Berbasis Web," *Jurnal Teknik Informatika Unika St. Thomas (JTIUST)*, vol. 05, no. 02, hal. 137-146, Des. 2020.

[3] R. A. Rizvee *et al.*, "LeafNet: A proficient convolutional neural network for detecting seven prominent mango leaf diseases," *Journal of Agriculture and Food Research*, vol. 14, hal. 100787, 2023.

[4] S. Sa'idah, I. P. Y. N. Suparta, dan E. Suhartono, "Modifikasi Convolutional Neural Network Arsitektur GoogLeNet dengan Dull Razor Filtering untuk Klasifikasi Kanker Kulit," *Jurnal Nasional Teknik Elektro dan Teknologi Informasi*, vol. 11, no. 2, Mei 2022.

[5] A. D. Indriyanti, R. Gernowo, E. Sediyono, dan A. E. Pratiwi, "Smart and Sustainable Road Development: A Case Study of Sentiment Analysis and GIS for East Java Infrastructure," *E3S Web of Conferences*, vol. 645, hal. 02013, 2025.

[6] A. D. Indriyanti, R. Gernowo, dan E. Sediyono, "Revealing East Java Community Sentiments Towards Poverty: A Comparative Study Using LDA and BERT," Cuestiones de Fisioterapia, vol. 54, no. 4, hal. 7790–7801, 2025.

[7] A. B. Handoko, I. K. Timotius, dan D. Utomo, "Klasifikasi Citra X-Ray COVID-19 Menggunakan Three-layered CNN Model," *Techné: Jurnal Ilmiah Elektroteknika*, vol. 21, no. 2, hal. 155-168, Okt. 2022.

[8] T. Ayu, V. Dwi, dan A. E. Minarno, "Pendiagnosa Daun Mangga Dengan Model Convolutional Neural Network," *CESS (Journal of Computer Engineering System and Science)*, vol. 6, no. 2, hal. 230-235, Jul. 2021.

[9] Y. D. Atma, P. P. I. Prayitno, dan N. B. Idris, "Penerapan Model Rapid Application Development dalam Perancangan Aplikasi Monitoring Bahan Baku Produksi UMKM," *Jurnal Sains Informatika Terapan (JSIT)*, vol. 04, no. 03, hal. 465-472, Okt. 2025.

[10] S. N. Bakri dan M. I. P. Nasution, "Penerapan Metodologi Rekayasa Perangkat Lunak untuk Efisiensi Pengembangan Sistem," *JSITIK*, vol. 3, no. 1, Des. 2024.