

**PENINGKATAN KLASIFIKASI PENYAKIT DAUN ANGGUR  
MENGUNAKAN *DENSENET169* DENGAN MEKANISME  
PERHATIAN *CBAM***

**SKRIPSI**

**OLEH:**

**DILLA AMELIA**

**218160037**



**PROGRAM STUDI TEKNIK INFORMATIKA  
FAKULTAS TEKNIK  
UNIVERSITAS MEDAN AREA  
MEDAN  
2025**

UNIVERSITAS MEDAN AREA

© Hak Cipta Di Lindungi Undang-Undang

1. Dilarang Mengutip sebagian atau seluruh dokumen ini tanpa mencantumkan sumber
2. Pengutipan hanya untuk keperluan pendidikan, penelitian dan penulisan karya ilmiah
3. Dilarang memperbanyak sebagian atau seluruh karya ini dalam bentuk apapun tanpa izin Universitas Medan Area

Document Accepted 10/4/26

Access From (repositori.uma.ac.id)10/4/26

**PENINGKATAN KLASIFIKASI PENYAKIT DAUN ANGGUR  
MENGUNAKAN *DENSENET169* DENGAN MEKANISME  
PERHATIAN *CBAM***

**SIKRIPSI**

**Diajukan sebagai Salah Satu Syarat untuk Memperoleh  
Gelar Sarjana di Fakultas Teknik  
Universitas Medan Area**

**Oleh:**

**DILLA AMELIA**

**218160037**

**PROGRAM STUDI TEKNIK INFORMATIKA  
FAKULTAS TEKNIK  
UNIVERSITAS MEDAN AREA  
MEDAN  
2025**

UNIVERSITAS MEDAN AREA

© Hak Cipta Di Lindungi Undang-Undang

## HALAMAN PENGESAHAN

Judul Skripsi : Peningkatan Klasifikasi Penyakit Daun Anggur Menggunakan DenseNet169 dengan Mekanisme Perhatian CBAM

Nama : Dilla Amelia  
NPM : 218160037  
Fakultas : Teknik

Disetujui Oleh  
Pembimbing



Muhathir, ST, M.Kom

Dekan Fakultas Teknik

Ka. Prodi. Teknik Informatika



Priono, ST, MT



Priono, S.Kom, M.Kom

Tanggal Lulus: 17 September 2025

## HALAMAN PERNYATAAN

Saya menyatakan bahwa skripsi yang saya susun, sebagai syarat memperoleh gelar serjana merupakan hasil karya tulis saya sendiri. Adapun bagian-bagian tertentu dalam penulisan skripsi ini yang saya kutip dari hasil karya orang lain telah dituliskan sumbernya secara jelas sesuai dengan norma, kaidah, dan etika penulisan ilmiah.

Saya bersedia menerima sanksi pencabutan gelar akademik yang saya peroleh dan sanksi-sanksi lainnya dengan peraturan yang berlaku, apabila di kemudian hari ditemukan adanya plagiat dalam skripsi ini.

Medan, 17 September 2025



Dilla Amelia  
218160037

## HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI

### TUGAS AKHIR/SKRIPSI/TESIS UNTUK KEPENTINGAN AKADEMIS

Sebagai sivitas akademik Universitas Medan Area, saya yang bertanda tangan di bawah ini:

Nama : Dilla Amelia  
NPM : 218160037  
Program Studi : Teknik Informatika  
Fakultas : Teknik  
Jenis Karya : Skripsi

Demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Universitas Medan Area Hak Bebas Royalti Noneksklusif (*Non-exclusive Royalty-Free Right*) atas karya ilmiah saya yang berjudul :

**Peningkatan Klasifikasi Penyakit Daun Anggur Menggunakan DenseNet169 dengan Mekanisme Perhatian CBAM**

Beserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti Noneksklusif ini Universitas Medan Area berhak menyimpan, mengalihmedia/format-kan, mengelola dalam bentuk pangkalan data (*database*), merawat, dan memublikasikan Skripsi saya selama tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai pemilik Hak Cipta.

Demikian pernyataan ini saya buat dengan sebenarnya.

Dibuat di : Medan  
Pada tanggal : 17 September 2025  
Yang menyatakan



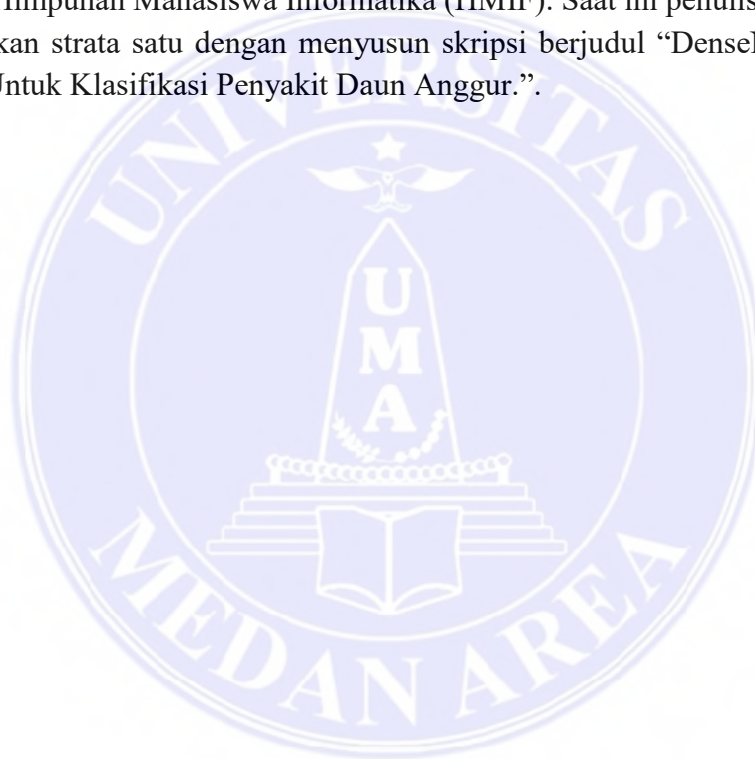
(Dilla Amelia)  
218160037

## RIWAYAT HIDUP

Penulis, Dilla amelia, lahir di Desa Bandar Setia Kecamatan Percut Sei Tuan Kabupaten Deli Serdang, provinsi Sumatera Utara, pada tanggal 02 Oktober 2002.

Penulis menempuh pendidikan dasar di SDN 104202 dan lulus pada tahun 2015. Pendidikan menengah pertama dilanjutkan di SMP Mts Amin Darusallam lulus pada tahun 2018 dan di lanjutkan dengan menempuh pendidikan kejuruan di SMK Negeri 1 Percut Sei Tuan lulus Pada tahun 2020, pada tahun 2021 penulis diterima di Program Studi Teknik Informatika, Fakultas Teknik, Universitas Medan Area.

Selama masa studinya, penulis aktif dalam berbagai kegiatan organisasi seperti Himpunan Mahasiswa Informatika (HMIF). Saat ini penulis menyelesaikan pendidikan strata satu dengan menyusun skripsi berjudul “DenseNet169 Dengan Cbam Untuk Klasifikasi Penyakit Daun Anggur.”.



## KATA PENGANTAR

Puji syukur Kepada Allah Swt atas berkat dan anugerah-Nya sehingga penulis dapat menyelesaikan Skripsi yang berjudul **“Peningkatan Klasifikasi Penyakit Daun Anggur Menggunakan DenseNet169 dengan Mekanisme Perhatian CBAM** ini dengan baik. Skripsi ini disusun guna memenuhi salah satu syarat untuk memperoleh gelar Sarjana di Program Studi Teknik Informatika, Fakultas Teknik, Universitas Medan Area.

Dalam proses pembuatan skripsi ini penulis banyak mendapatkan bimbingan, arahan, dukungan doa dan juga kritik dari banyak pihak yang telah membantu dalam hal meningkatkan kualitas skripsi yang di buat. Oleh sebab itu Penulis mengucapkan terimakasih sebesar-besarnya kepada:

1. Bapak Prof. Dr. Dadan Ramdan, M.Eng, M.Sc. Selaku Rektor Universitas Medan Area.
2. Bapak Dr. Eng. Supriatno, ST., MT selaku Dekan Fakultas Teknik Universitas Medan Area
3. Bapak Rizki Muliono, S.kom, M.kom. Selaku Kepala Program Studi Teknik Informatika.
4. Bapak Muhathir, S.T, M.kom. Selaku Dosen pembimbing yang telah Sabar, memberikan perhatian penuh, Banyak Meluangkan Waktu dan juga selalu memberikan arahan serta bimbingan yang sangat berarti Untuk penulis. Karena dukungan yang sangat baik dari bapak penulis dapat menyelesaikan Skripsi ini Dengan Sangat Baik.
5. Seluruh Dosen Teknik Informatika Universitas Medan Area, yang telah berjasa dalam membagi ilmu, pengalaman dan wawasan yang sangat luar biasa selama masa perkuliahan. Di setiap pertemuan dalam mata kuliah memberikan nasehat dan dukungan terhadapap penulis yang menjadi kekuatan tersendiri bagi penulis dalam menyelesaikan skripsi ini.
6. Abang Robby Kurniawan Sari Damanik, S.T., Sebagai IT Support Program studi Teknik Informatika, yang selalu menolong dan memberikan dukungan teknis selama penulis berkuliah. Terimakasih atas kesabaran membantu penulis dalam menjawab pertanyaan dan juga membantu dalam berbagai kendala teknis yang Penulis hadapi selama masa perkuliahan.
7. Orang tua dengan penuh rasa syukur, saya berterima kasih kepada kedua orang tua atas doa, pengorbanan, dan kasih sayang yang tak pernah berhenti. Setiap pencapaian ini adalah berkat kalian. Karya ini saya persembahkan untuk Mama dan Papa sebagai wujud cinta dan rasa terima kasih yang tulus.
8. Abang dan kakak tercinta, yang telah memberikan dukungan moral, semangat, serta doa yang tiada henti selama proses penyusunan skripsi ini. Terima kasih atas pengertian, bantuan, dan kasih sayang yang selalu menguatkan Penulis dalam menghadapi setiap tantangan. Kehadiran kalian

adalah salah satu alasan penulis mampu menyelesaikan tugas akhir ini dengan baik.

9. Teman-teman gurly dan mahasiswa Program Studi Teknik Informatika 2021 yang telah memberikan informasi, semangat, dan dukungan dalam menyelesaikan laporan skripsi ini.
10. Semua pihak yang tidak dapat di sebutkan satu persatu yang telah terlibat dalam penyelesaian skripsi ini sehingga dapat diselesaikan dengan baik.

Penulis menyadari bahwa skripsi ini masih jauh dari kesempurnaan. Oleh karena itu, kritik dan saran yang membangun sangat diharapkan demi perbaikan di masa depan. Semoga skripsi ini dapat memberikan manfaat bagi pembaca serta dapat menjadi referensi dalam penelitian lebih lanjut.

Akhir kata, Semoga Tuhan yang Maha Esa senantiasa melimpahkan berkat dan karunia-Nya kepada semua pihak yang telah membantu dalam penyusunan skripsi ini.

Medan, 17 September 2025



Dilla Amelia  
218160037

## ABSTRAK

Penelitian ini dilakukan dengan tujuan meningkatkan ketepatan dalam mengklasifikasikan penyakit pada daun anggur menggunakan pendekatan *Deep Learning* dengan arsitektur *DenseNet169* yang dipadukan dengan *CBAM* (*Convolutional Block Attention Module*). Arsitektur *DenseNet169* dimanfaatkan untuk mengekstraksi fitur visual penting dari citra, sedangkan *CBAM* berfungsi untuk memperkuat perhatian model terhadap informasi spasial dan kanal yang paling relevan. Pelatihan model dilakukan secara konsisten selama 20 epoch dan dievaluasi menggunakan sejumlah metrik, yaitu akurasi, presisi, recall, F1-score, dan *ROC-AUC*. Hasil pengujian menunjukkan bahwa kombinasi model *DenseNet169* dan *CBAM* menghasilkan akurasi sebesar 99,50%, serta presisi, recall, dan F1-score yang sama tinggi. Nilai *ROC-AUC* yang diperoleh mencapai 99,96%, menandakan performa klasifikasi yang sangat optimal. Jika dibandingkan dengan penelitian sebelumnya yang menggunakan pendekatan serupa, model ini menunjukkan kinerja yang lebih baik. Temuan ini menunjukkan bahwa integrasi *CBAM* dalam arsitektur konvolusional seperti *DenseNet169* berkontribusi positif dalam pengembangan sistem deteksi penyakit tanaman secara otomatis. Ke depan, model ini berpotensi diterapkan dalam sistem monitoring berbasis drone secara real-time, guna mendukung pengambilan keputusan yang lebih tepat dalam sektor pertanian.

**Kata Kunci:** *DenseNet169*, *CBAM*, Klasifikasi Citra, Penyakit Daun Anggur.

## ABSTRACT

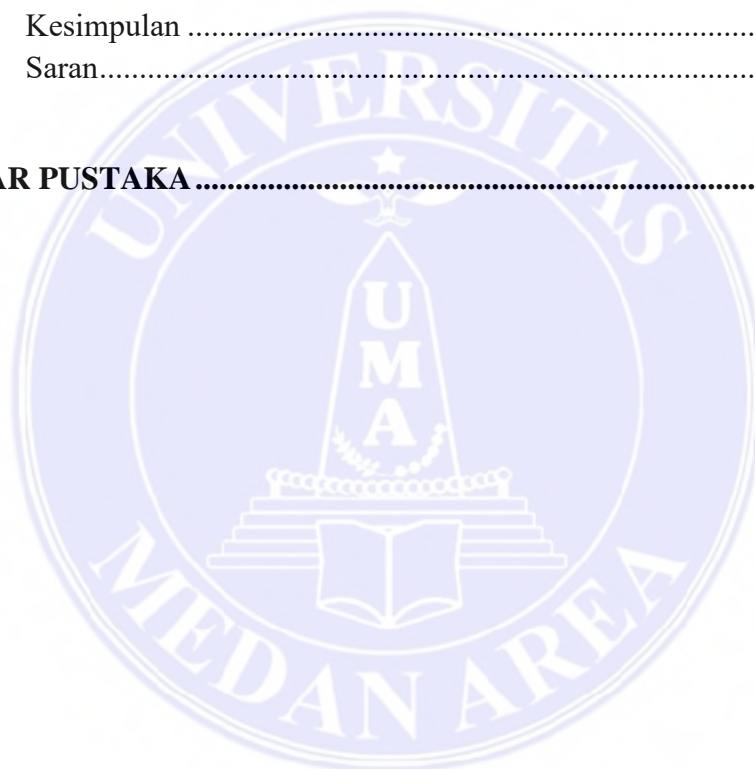
*This study aims to improve the accuracy of grape leaf disease classification by employing a deep learning approach using the DenseNet169 architecture integrated with the Convolutional Block Attention Module (CBAM). DenseNet169 is utilized to extract important visual features from images, while CBAM enhances the model's focus on the most relevant spatial and channel information. The model was trained consistently for 20 epochs and evaluated using several performance metrics, including accuracy, precision, recall, F1-score, and ROC-AUC. The experimental results indicate that the combination of DenseNet169 and CBAM achieved an accuracy of 99.50%, with precision, recall, and F1-score reaching the same value. The ROC-AUC score reached 99.96%, reflecting an excellent classification performance. Compared to previous studies that applied similar approaches, this model demonstrated superior performance. These findings suggest that integrating CBAM into a convolutional architecture like DenseNet169 contributes positively to the development of automated plant disease detection systems. In the future, this model holds potential for real-time implementation in drone-based monitoring systems to support more effective decision-making in the agricultural sector.*

**Keywords:** DenseNet169, CBAM, Deep Learning, Image Classification.

## DAFTAR ISI

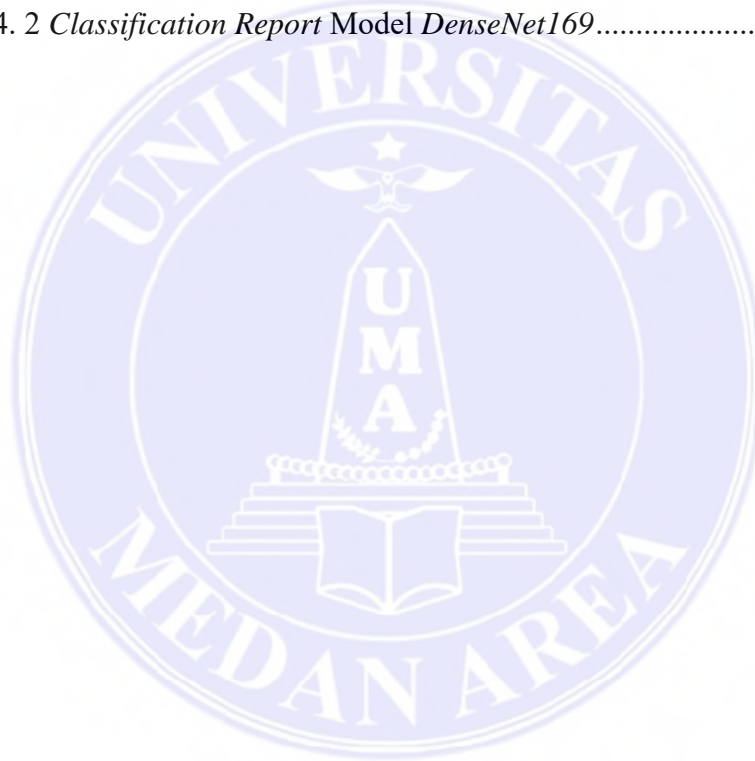
<b>HALAMAN PENGESAHAN</b> .....	<b>iii</b>
<b>HALAMAN PERNYATAAN</b> .....	<b>iv</b>
<b>RIWAYAT HIDUP</b> .....	<b>vi</b>
<b>KATA PENGANTAR</b> .....	<b>vii</b>
<b>ABSTRAK</b> .....	<b>ix</b>
<b>ABSTRACT</b> .....	<b>x</b>
<b>DAFTAR ISI</b> .....	<b>xi</b>
<b>DAFTAR TABEL</b> .....	<b>xiii</b>
<b>DAFTAR GAMBAR</b> .....	<b>xiv</b>
<b>BAB I PENDAHULUAN</b> .....	<b>1</b>
1.1. Latar Belakang .....	1
1.2. Rumusan Masalah .....	3
1.3. Batasan Masalah.....	4
1.4. Tujuan Penelitian .....	4
1.5. Manfaat Penelitian .....	5
<b>BAB II TINJAUAN PUSTAKA</b> .....	<b>6</b>
2.1. <i>Deep Learning</i> .....	6
2.2. <i>Convolutional Neural Network (CNN)</i> .....	7
2.3. <i>DenseNet169</i> .....	8
2.4. <i>Convolutional Block Attention Module (CBAM)</i> .....	10
2.5. Evaluasi Model.....	11
2.6. Evaluasi.....	12
2.6.1. <i>ROC AUC</i> .....	13
2.7. Penyakit Daun Anggur.....	14
2.8. Kasus Penelitian .....	16
2.9. Penelitian Terdahulu .....	17
<b>BAB III METODOLOGI PENELITIAN</b> .....	<b>18</b>
3.1. Alat dan Bahan.....	18
3.1.1. Perangkat Keras .....	18
3.1.2. Perangkat Lunak.....	19
3.2. Prosedur Kerja.....	19
3.3. Metode Pengumpulan Data .....	20
3.4. Rancangan Arsitektur.....	21
3.5. Pembagian Data .....	28
3.6. <i>Hyperparameter</i> .....	29
3.7. Evaluasi.....	29

<b>BAB IV HASIL DAN PEMBAHASAN .....</b>	<b>29</b>
4.1. Hasil .....	29
4.1.1. Sampel Data .....	30
4.1.2. Augmentasi Data .....	31
4.1.3. Implementasi Model .....	33
4.1.7. Implementasi <i>Web</i> .....	42
4.1.8. Implementasi Mobile App .....	43
4.2. Pembahasan .....	47
<b>BAB V KESIMPULAN DAN SARAN .....</b>	<b>51</b>
5.1. Kesimpulan .....	51
5.2. Saran .....	51
<b>DAFTAR PUSTAKA .....</b>	<b>53</b>



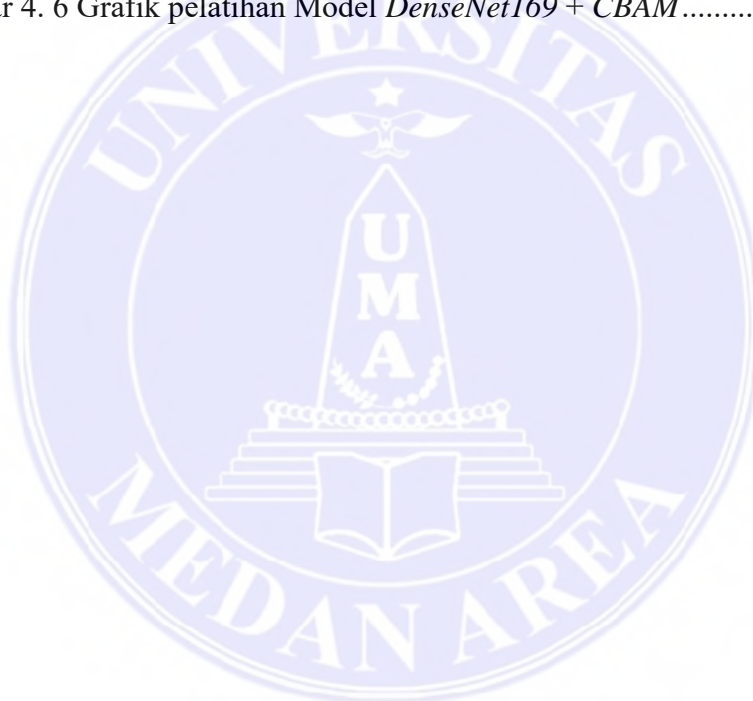
## DAFTAR TABEL

	Halaman
Tabel 2. 1 Jenis Penyakit Daun Anggur .....	15
Tabel 2. 2 Kasus Penelitian .....	16
Tabel 2. 3 Penelitian Terdahulu .....	17
Tabel 3. 1 Perangkat Keras.....	18
Tabel 3. 2 Perangkat Lunak.....	19
Tabel 3. 3 Jumlah Dataset .....	20
Tabel 3. 4 Pembagian Dataset .....	29
Tabel 3. 5 Parameter Value .....	29
Tabel 4. 1 Model Skanerio Penelitian .....	33
Tabel 4. 2 <i>Classification Report</i> Model <i>DenseNet169</i> .....	36



## DAFTAR GAMBAR

	Halaman
Gambar 2. 1 Ilustrasi <i>Deep Learning</i> .....	6
Gambar 2. 2 Arsitektur <i>DenseNet169</i> .....	9
Gambar 2. 3 Arsitektur <i>CBAM</i> .....	11
Gambar 3. 1 Prosedur Kerja .....	19
Gambar 3. 2 Rancangan Arsitektur .....	21
Gambar 4. 1 Visualisasi Sampel Data Daun Anggur .....	30
Gambar 4. 2 Visualisasi Data Sebelum dan Sesudah di Augmentasi .....	32
Gambar 4. 3 Grafik Pelatihan Model <i>DenseNet169</i> .....	34
Gambar 4. 4 Hasil Confusion Matrix Model <i>DenseNet169</i> .....	35
Gambar 4. 5 Grafik <i>ROC AUC</i> Model <i>DenseNet169</i> .....	37
Gambar 4. 6 Grafik pelatihan Model <i>DenseNet169</i> + <i>CBAM</i> .....	38



# BAB I PENDAHULUAN

## 1.1. Latar Belakang

*DenseNet-169* adalah salah satu arsitektur *DenseNet* lainnya yang menggunakan blok konektivitas yang padat (*dense blocks*) juga memiliki lapisan 169 semua lapisan dirangkai sehingga setiap lapisan dapat mempengaruhi lapisan lainnya (Sangwon Hwang dan Joong Lee, 2023). *Densenet-169* merupakan arsitektur yang banyak digunakan untuk klasifikasi. *DenseNet* juga memiliki parameter yang dapat dilatih jauh lebih sedikit dibandingkan arsitektur *DenseNet* lainnya (Adarsh Vulli, 2022) Model *DenseNet-169* salah satu kelompok *DenseNet* pemenang kompetisi *ImageNet* pada tahun 2017. Disetiap lapisan dapat mengumpulkan informasi dari level di atasnya. Jaringan akan lebih sempit dan lebih rapat karena di setiap lapisan menerima peta fitur dari lapisan sebelumnya (Nair, Kartik, dkk., 2022).

*DenseNet-169* yang dikembangkan oleh (Huang dkk., 2017) untuk meningkatkan kinerja dan efisiensi CNN dalam pengenalan objek (M. Haikal Firdaus, Ema Utami, dan Dhani Ariatmanto, 2023) *DenseNet-169* pertama kali diperkenalkan oleh Gao Huang et al dan rekan- rekannya dengan judul *Densely Connected Convolutional Networks* yang presentasikan pada CVPR 2017 (Gao Huang dkk., 2018) kasus yang dibahas adalah bagaimana arsitektur jaringan saraf dalam, khususnya arsitektur *DenseNet-169* dapat meningkatkan efisiensi representasi fitur dan kinerja dengan berbagai tugas klasifikasi (Huang dkk., 2017) yang sangat efektif untuk data citra gambar serta deteksi objek (Lee, S. K. 2024)

*DenseNet-169* dipilih dari keluarga *DenseNet* setelah melakukan perbandingan berbagai model pra-latihan yang berkualitas tinggi (M. Farag, 2022) *DenseNet-169* juga diperkenalkan sebagai salah satu varian yang memiliki 169 lapisan. Model ini dapat menggabungkan koneksi padat antara lapisan yang dimana output setiap lapisan dapat terhubung langsung ke semua lapisan dalam blok jaringan (*dense block*) (G. Huang., 2016).

Penelitian yang menerapkan metode *DenseNet-169* telah banyak berbagai kasus, seperti: Pergelangan tangan (Joonho Oh dkk., 2023), prediksi virus penyakit kulit berbenjol-benjol (Mujahid ddk., 2024), peningkatan penilaian kualitas sayur (Jun Wen and Jing He, 2024), perhatian untuk deteksi dini penyakit kentang (Kasana, SS; Rathore, AS., 2024), pendekatan produktif untuk keunggulan dalam diagnosis retinopati (Nuthanakanti dkk., 2024), tumor otak (Lee, S. K., 2024)

Walaupun hasil dari penelitian terdahulu sudah mencapai hasil yang memuaskan akan tetapi *DenseNet169* masih perlu dimodifikasi untuk mencapai hasil yang lebih maksimal dikarenakan menanganin kompleksitas data pada tingkat yang lebih tinggi. Salah satu caranya adalah dengan menambahkan modul perhatian *DenseNet169*, modul perhatian banyak diteliti sebelumlahnya seperti, Berbagai modul perhatian, termasuk *Convolutional Block Attention Module (CBAM)*, *channel attention module* (Zhang et al., 2024), *temporal attention module* (Liu dkk., 2023), dan *spatial attention module* (Fu dkk., 2023), telah digunakan dalam beberapa penelitian terbaru. Dalam penelitian ini, *CBAM* dipilih sebagai modul perhatian yang akan diterapkan dalam arsitektur *DenseNet169* karena kemampuannya untuk menggabungkan mekanisme perhatian kanal dan spasial. Selain itu, *CBAM* memiliki kemampuan untuk mendistribusikan bobot fitur secara

adaptif, sehingga meningkatkan kemampuan model dalam mengekstrak informasi yang relevan dari data masukan (Chen dkk., 2024).

*CBAM* telah digunakan dengan sangat akurat dalam sejumlah penelitian, termasuk CNN untuk biometrik (Zhang & Wang., 2023), *DenseNet* untuk klasifikasi tingkat keparahan retinopati diabetik (Frag dkk., 2023), *DeepConvLSTM* untuk pembelajaran mendalam berbasis sensor gerak (Agac & Durmaz Incel, 2024), dan *CapsNet* untuk prediksi deteksi peptida (Yu dkk., 2024). Penelitian ini akan menyelidiki integrasi *CBAM* ke dalam *DenseNet169* untuk meningkatkan deteksi penyakit pada daun anggur dengan lebih akurat. *CBAM* juga memiliki kelemahan yang menyebabkan overfitting pada dataset yang terbatas. Hal ini didasarkan pada keunggulan arsitektur *DenseNet169*, termasuk kemampuan ekstraksi fitur yang mendalam dan efisien, serta keunggulan *CBAM* dalam memperkuat mekanisme perhatian spasial dan kanal, penelitian ini akan mengeksplorasi integrasi *CBAM* ke dalam *DenseNet169* untuk meningkatkan akurasi dalam mendeteksi penyakit pada daun anggur.

## 1.2. Rumusan Masalah

Adapun rumusan masalah pada penelitian ini adalah bagaimana penerapan arsitektur *DenseNet169* dan mekanisme perhatian *CBAM* dapat meningkatkan akurasi klasifikasi citra penyakit daun anggur.

### 1.3. Batasan Masalah

Batasan masalah yang dapat diterapkan pada penelitian ini adalah sebagai berikut:

1. Data citra tanaman anggur yang digunakan dalam penelitian ini berjenis data sekunder yang berasal dari situs repositori data gratis yaitu *kaggle*.
2. Dalam penelitian ini terdapat 4 jenis penyakit daun yang digunakan yaitu: *Black Rot*, *ESCA*, *Healthy* dan *Leaf Blight*.
3. jumlah data citra daun yaitu 4000 dengan pembagian citra jenis *Black Rot* sebanyak 1000 citra, *ESCA* sebanyak 1000 citra, *Healthy* sebanyak 1000 citra dan *Leaf Blight* sebanyak 1000 citra.
4. Arsitektur yang digunakan adalah *DenseNet169* yang dioptimalkan dengan *CBAM*.

### 1.4. Tujuan Penelitian

Adapun tujuan dari penelitian ini adalah :

1. Menghasilkan model diagnosis yang tepat dengan menggunakan arsitektur *DenseNet169* yang dioptimalkan dengan *CBAM*.
2. Mengevaluasi bagaimana mekanisme perhatian *CBAM* mempengaruhi kinerja *DenseNet169* yang lebih baik.

## 1.5. Manfaat Penelitian

Dokumen ini adalah template yang dibuat untuk memudahkan penulisan tesis. File ini berekstensi .dotx, ekstensi file khusus untuk template di Microsoft Word. Sebelum menggunakan template ini,

1. Dengan menghasilkan model diagnosis berbasis arsitektur *DenseNet169* yang dioptimalkan dengan mekanisme perhatian *CBAM*, penelitian ini diharapkan dapat memberikan kontribusi signifikan dalam meningkatkan akurasi dan ketepatan diagnosis penyakit pada daun anggur.
2. Penelitian ini dapat memberikan pemahaman yang lebih baik mengenai bagaimana mekanisme perhatian *CBAM* mempengaruhi performa model *DenseNet169*, khususnya dalam konteks deteksi penyakit tanaman daun anggur.

## BAB II TINJAUAN PUSTAKA

### 2.1. *Deep Learning*

*Deep learning* adalah bagian dari kecerdasan buatan yang menggunakan jaringan saraf tiruan dengan banyak lapisan untuk mengekstraksi fitur dari data secara hierarkis. Teknik ini mampu mempelajari pola kompleks dalam data seperti citra, teks, dan audio tanpa membutuhkan pengolahan manual. (Zhang dkk., 2022) menunjukkan bahwa model *deep learning* mampu mencapai akurasi tinggi dalam berbagai aplikasi, seperti pengenalan wajah dan klasifikasi objek. Selain itu, (Wang dkk., 2023) menjelaskan bagaimana optimasi algoritma, seperti Adam dan SGD, dapat mempercepat konvergensi model *deep learning*, sementara (Chen dkk., 2022) memanfaatkan augmentasi data untuk meningkatkan generalisasi model.

Dalam konteks analisis citra medis, (Xu dkk., 2024) menunjukkan bahwa *deep learning* sangat efektif untuk mendeteksi anomali seperti tumor atau lesi. Penelitian lain oleh (Wu dkk., 2023) menggarisbawahi bahwa teknik ini juga unggul dalam menangani data yang tidak terstruktur, termasuk citra satelit dan rekaman video. Dengan kemampuannya yang adaptif, *deep learning* terus menjadi metode utama dalam berbagai aplikasi kecerdasan buatan modern.



**Gambar 2. 1** Ilustrasi *Deep Learning*  
(Intyanto, G. Wega. 2021)

## 2.2. Convolutional Neural Network (CNN)

merupakan arsitektur pembelajaran mendalam yang dirancang untuk memproses data berbentuk grid, seperti citra dan sinyal sekuensial. CNN bekerja dengan mengekstraksi pola spasial melalui operasi konvolusi sehingga mampu memahami struktur visual dari data secara efektif (Zhang et al., 2024). CNN juga digunakan untuk mempelajari fitur hierarkis secara bertingkat, mulai dari fitur sederhana seperti tepi dan tekstur hingga fitur kompleks untuk tugas pengenalan pola, klasifikasi, maupun deteksi objek dalam berbagai bidang, termasuk pengolahan citra medis, pertanian digital, dan sistem cerdas berbasis visual (Rahman & Lee, 2025). Arsitektur ini terinspirasi dari sistem visual manusia, di mana neuron pada korteks visual merespons area pandang tertentu dan menyusun informasi secara bertahap untuk membentuk persepsi visual yang utuh (Kim et al., 2023).

Secara umum, arsitektur CNN terdiri dari beberapa lapisan utama, yaitu lapisan konvolusi untuk mengekstraksi fitur, lapisan pooling untuk mereduksi dimensi dan mencegah overfitting, serta lapisan fully connected yang berfungsi untuk melakukan klasifikasi berdasarkan fitur yang telah diperoleh sebelumnya (Singh et al., 2024). Seiring perkembangan teknologi, berbagai arsitektur CNN modern telah dikembangkan, seperti ResNet, DenseNet, EfficientNet, InceptionV3, Xception, MobileNetV3, ShuffleNet, InceptionResNetV2, dan DenseNet169, yang menawarkan peningkatan dalam akurasi, efisiensi komputasi, serta kemampuan generalisasi pada berbagai dataset skala besar dan lingkungan perangkat terbatas (Huang et al., 2025).

### 2.3. *DenseNet169*

*DenseNet-169* adalah salah satu arsitektur jaringan saraf tiruan yang memanfaatkan koneksi padat (*dense connection*) antara lapisan-lapisannya. (Huang dkk., 2017) menjelaskan bahwa pendekatan ini memungkinkan setiap lapisan menerima informasi dari semua lapisan sebelumnya, sehingga meningkatkan efisiensi parameter dan kemampuan pembelajaran fitur. (Tan dkk., 2023) menunjukkan bahwa *DenseNet-169* unggul dalam mendeteksi penyakit tanaman dengan akurasi tinggi. Setiap lapisan konvolusional terdiri dari normalisasi *Batch* (BN) *Rectified Linear Unit* (ReLU), dan konvolusi setiap blok padat terdiri dari 1x1 diikuti oleh konvolusi 3x3 dalam *DenseNet169* memiliki 6, 12, 32 oleh karena itu  $6 + 12 + 32 + 32 = 82$  set konvolusi 1x1 dan 3x3. Menghasilkan total  $82 \times 2 = 164$  lapisan blok padat pertama, kedua, dan ketiga diikuti oleh lapisan transisi yang berisi konvolusi 1x1 dan kumpulan rata-rata 2x2 dengan langkah 2. Blok padat keempat yang terhubung ke klasifikasi sehingga mendapatkan total  $164 + 3 + 1 + 1 = 169$  lapisan model *DenseNet169* memiliki tingkat pertumbuhan 32. Tingkat pertumbuhan menentukan jumlah lapisan baru (Dalvi, dkk., 2023).

Studi oleh (Zhang dkk., 2024) membandingkan *DenseNet* dengan arsitektur lain pada citra satelit dan menemukan hasil signifikan dalam akurasi klasifikasi. (Li dkk., 2022) mengaplikasikan *DenseNet* untuk analisis citra medis, seperti deteksi kanker, dan mencatat peningkatan sensitivitas. (Wu dkk., 2023) menyoroti kemampuan *DenseNet* dalam menangani dataset tidak seimbang, yang sering menjadi tantangan dalam tugas klasifikasi citra. Secara umum, arsitektur *DenseNet* terdiri dari tiga bagian: lapisan blok tebal, lapisan transisi, dan lapisan klasifikasi (Ramdan, dkk., 2020).

Lapisan blok tebal terdiri dari lapisan normalisasi *batch*, fungsi aktivasi ReLU, dan lapisan *convolutional*. Layer, masing-masing berukuran 1x1 dan 3x3 dengan jumlah yang berbeda-beda untuk setiap blok. Di sisi lain, lapisan transisi terdiri dari lapisan *convolutional* 1x1 dan lapisan *average pooling* 2x2, yang berarti nilai yang diambil pada proses pooling adalah nilai rata-rata dari hasil lapisan *convolution*. Arsitektur *DenseNet* dapat digambarkan sebagai berikut: Lapisan klasifikasi terdiri dari lapisan rata-rata global dan lapisan yang terhubung sepenuhnya dengan fungsi aktivasi *softmax*.

Layers	Output Size	DenseNet 169 (Growth Rate (k) = 32)
Convolution	112 × 112	7 × 7 conv, stride = 2
Pooling	56 × 56	3 × 3 max pool, stride = 2
Dense Block 1	56 × 56	((1 × 1 conv)    (3 × 3 conv)) × 6
Transition Layer 1	56 × 56	1 × 1 conv
	28 × 28	2 × 2 average pool, stride = 2
Dense Block 2	28 × 28	((1 × 1 conv)    (3 × 3 conv)) × 12
Transition Layer 2	28 × 28	1 × 1 conv
	14 × 14	2 × 2 average pool, stride = 2
Dense Block 3	14 × 14	((1 × 1 conv)    (3 × 3 conv)) × 32
Transition Layer 3	14 × 14	1 × 1 conv
	7 × 7	2 × 2 average pool, stride = 2
Dense Block 4	7 × 7	((1 × 1 conv)    (3 × 3 conv)) × 32
Classification Layer	1 × 1	7 × 7 global average pool
		1000D fully connected softmax

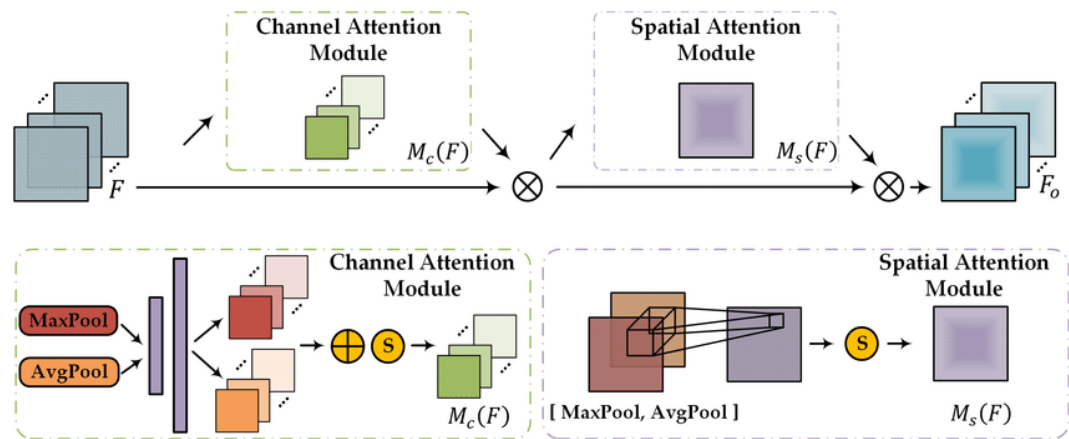
**Gambar 2. 2** Arsitektur *DenseNet169*  
(Dalvi, dkk., 2023)

## 2.4. Convolutional Block Attention Module (CBAM)

*CBAM* adalah modul perhatian yang meningkatkan sensitivitas jaringan saraf tiruan terhadap fitur penting dalam data. (Woo dkk., 2018) pertama kali memperkenalkan *CBAM* dengan dua mekanisme utama, yaitu perhatian saluran (*channel attention*) dan perhatian spasial (*spatial attention*). Studi ini menunjukkan bahwa *CBAM* dapat diterapkan pada berbagai arsitektur jaringan saraf untuk meningkatkan akurasi tanpa menambah banyak kompleksitas komputasi.

Penelitian (Mujahid dkk., 2024) menunjukkan bahwa penerapan *CBAM* pada *DenseNet-169* meningkatkan performa dalam deteksi penyakit tanaman. (Lee dkk., 2023) juga menguji *CBAM* pada data medis dan melaporkan peningkatan sensitivitas dalam deteksi kanker payudara. (Wang dkk., 2022) menunjukkan bahwa *CBAM* efektif untuk klasifikasi citra kompleks, sementara (Gupta dkk., 2023) mengaplikasikan modul ini dalam pengenalan objek 3D, menunjukkan manfaat yang signifikan.

*CBAM* tidak hanya berguna untuk meningkatkan akurasi, tetapi juga memberikan interpretasi yang lebih baik terhadap fitur yang dipelajari model. Hal ini didukung oleh (Zhou dkk., 2023), yang menemukan bahwa perhatian spasial membantu memfokuskan model pada area citra yang relevan, seperti bagian daun yang menunjukkan gejala penyakit.



**Gambar 2. 3** Arsitektur CBAM

## 2.5. Evaluasi Model

Evaluasi model bertujuan untuk mengukur sejauh mana model dapat menyelesaikan tugas klasifikasi dengan menggunakan metrik yang relevan. Menurut (Simanjuntak dkk., 2023), metrik seperti akurasi, *precision*, *recall*, *F1-score*, dan *confusion matrix* sangat penting untuk menilai kinerja model pada dataset yang beragam (Lubis dkk., 2022) juga menyatakan bahwa *confusion matrix* membantu dalam menganalisis kesalahan klasifikasi, terutama pada data yang tidak seimbang. Di bidang pertanian, (Ritonga dkk., 2023) menunjukkan bahwa deteksi penyakit tanaman dapat mencapai akurasi yang tinggi menggunakan arsitektur *deep learning* modern. Selain itu, (Gulo & Lubis 2024) menyebutkan bahwa *recall* sangat penting dalam kasus yang mengutamakan deteksi data positif (Nasution & Sinaga 2023) juga menekankan bahwa penggunaan data augmentasi dapat memengaruhi hasil evaluasi secara signifikan, terutama pada dataset citra.

## 2.6. Evaluasi

Evaluasi dilakukan pada proses *testing* menggunakan metrik *akurasi*, *presisi*, *recall*, dan *f1-Score* yang diperoleh dari *Confusion Matrix*. Nilai akurasi, presisi, recall, dan F1-Score secara berurutan diperoleh dari persamaan berikut ini (Fil dkk., 2023):

### 1. Accuracy

Akurasi digunakan untuk Menghitung proporsi prediksi yang benar dari total prediksi yang dilakukan (Heydarian et al., 2022).

$$ACC = \frac{TP + TN}{TP + TN + FP + FN} \quad (2.1)$$

### 2. Precision

Presisi adalah Salah satu ukuran yang digunakan untuk menilai relevansi contoh di antara contoh yang diambil adalah presis, yang dihitung sebagai proporsi antara prediksi yang benar-benar positif dan total semua nilai positif yang nyata (Chicco et al., 2021).

$$Precision = \frac{TP}{TP + FP} \quad (2.2)$$

### 3. Recall

*Recall* adalah Salah satu ukuran yang digunakan untuk menilai relevansi contoh di antara contoh yang diambil adalah presis, yang dihitung sebagai proporsi antara prediksi yang benar-benar positif dan total semua nilai positif yang nyata (De Diego et al., 2022).

$$Recall = \frac{TP}{(TP + FN)} \quad (2.3)$$

#### 4. *F1-Score*

*F-Score* adalah metrik yang memperhitungkan sensitivitas dan presisi secara bersamaan. Ini adalah skor rata-rata harmonis antara sensitivitas dan presisi (YILMAZ & YAĞIN, 2022).

$$F1 \text{ Score} = \frac{2 \times \textit{Precision} \times \textit{Recall}}{\textit{Precision} + \textit{Recall}} \quad (2.4)$$

Dengan keterangan :

TP : True Positif

FP : False Positif

TN : True Negatif

FN : False Negatif.

#### 2.6.1. *ROC AUC*

*Receiver Operating Characteristic* (ROC) dan *Area under the Curve* (AUC) merupakan metode evaluasi yang digunakan untuk menilai kinerja model klasifikasi, khususnya pada masalah klasifikasi biner. Kurva ROC menggambarkan hubungan antara *True Positive Rate* (TPR) dan *False Positive Rate* (FPR) pada berbagai ambang keputusan. Nilai AUC menunjukkan luas area di bawah kurva ROC dan digunakan untuk mengukur kemampuan model dalam membedakan antara kelas positif dan negatif (Li, 2024).

Penelitian terbaru juga menyebutkan bahwa ROC–AUC menjadi salah satu metrik yang paling konsisten untuk mengevaluasi performa klasifikasi karena tidak bergantung pada ambang batas tertentu dan tetap stabil meskipun distribusi kelas tidak seimbang (Richardson et al., 2024). Secara matematis, nilai AUC dapat

dipahami sebagai probabilitas bahwa model memberikan skor prediksi lebih tinggi terhadap sampel positif dibandingkan sampel negatif, atau dapat dihitung sebagai integral dari fungsi TPR terhadap FPR sepanjang kurva ROC. maka rumusnya adalah:

$$AUC = \int_0^1 TPR(FPR) dFPR \quad (2.5)$$

a. *TPR (True Positive Rate)*

$$TPR = \frac{TP}{TP + FN} \quad (2.6)$$




b. *FPR (False Positive Rate)*


$$FPR = \frac{FP}{FP + TN} \quad (2.7)$$

## 2.7. Penyakit Daun Anggur

Penyakit daun anggur, seperti *Black Rot*, *ESCA*, dan *Leaf Blight*, merupakan ancaman besar dalam membudidayakan anggur. Deteksi dini sangat penting untuk mencegah kerugian hasil panen (et al. 2024) menunjukkan bahwa teknologi berbasis *deep learning* sangat efektif untuk mendeteksi penyakit ini (Zhou dkk., 2023) menggunakan citra *hyperspectral* untuk mengidentifikasi gejala awal penyakit. (Liu dkk., 2023) menyoroti pengguna citra drone untuk cepat mendeteksi (Gupta dkk., 2023).

Tabel 2. 1 Jenis Penyakit Daun Anggur

Gambar	Nama Penyakit	Deskripsi
	<p><i>Black Rot</i></p>	<p><i>Black Rot</i> adalah penyakit yang disebabkan oleh jamur <i>Guignardia bidwellii</i>, yang menyebar melalui spora. Spora ini dapat disebabkan oleh hujan, percikan air dari bawah tanah ke tanaman terbawa oleh angin, dan biasanya memunculkan bercak coklat dengan tepi hitam yang khas.</p>
	<p><i>ESCA</i></p>	<p><i>ESCA</i> penyakit pada tanaman anggur yang disebabkan oleh kompleks infeksi jamur yang dapat menyerang jaringan kayu tanaman anggur dan menyebabkan kematian tanaman secara keseluruhan penyakit ini memiliki ciri-ciri memiliki pola bercak "tiger stripes" pada daun yang sudah tua serta memiliki bercak coklat dikelilingn warna kuning.</p>
	<p><i>Healthy</i></p>	<p><i>Healthy</i> adalah daun sehat dan tidak memiliki bercak atau rusak lainnya pada daun</p>

	<p><i>Leaf Blight</i></p>	<p><i>Leaf Blight</i> adalah penyakit yang di sebabkan infeksi jamur, biasanya dari genus <i>Alternaria</i>, <i>Phoma</i> atau <i>Pestalotiopsis</i>. Penyakit ini memiliki ciri-ciri mulcul bercak coklat atau hitam yang meluas pada daun, memiliki warna dauh yang sudah menguning sebelum akhirnya mengering.</p>
---	---------------------------	---

## 2.8. Kasus Penelitian

Beberapa Penelitian yang sudah pernah dilakukan sebelumnya terkait deteksi penyakit daun anggur dapat dilihat pada Tabel 2.2 berikut.

**Tabel 2. 2** Kasus Penelitian

No	Penulis	Metode	Akurasi
1	Siskia S. dkk., (2020)	<i>GLCM, Color Moment, K*Tree</i>	Model ini berhasil mengklasifikasikan penyakit daun anggur dengan akurasi 87,5%.
2	Wardani, K. & Leonardi, L. (2023)	CNN dengan <i>Dropout</i> dan <i>optimizer RMSProp</i>	Model CNN ini mencapai akurasi 98,5% dalam mengklasifikasikan penyakit <i>Black Measles</i> pada daun anggur.
3	Agusta Yusalendra, Surya Agustian, & Benny Sukma Negara (2023)	<i>EfficientNet-B0</i>	Model <i>EfficientNet-B0</i> mencapai akurasi 76,2% dalam mengklasifikasikan penyakit daun anggur.
4	Allan Bil Faqih, Donny Avianto. (2024)	<i>InceptionV3</i>	Hasil penelitian ini bahwa algoritma <i>InceptionV3</i> memiliki akurasi 96% dalam mendeteksi penyakit tanaman <i>solanaceae</i> dan <i>rosaceae</i> .
5	Lee dkk., (2024)	<i>DenseNet121</i> dengan <i>CBAM</i>	Kombinasi <i>DenseNet121</i> dan <i>CBAM</i> menghasilkan akurasi tertinggi, yaitu 97,5%.

## 2.9. Penelitian Terdahulu

Penelitian sebelumnya merujuk pada hasil riset yang telah digunakan oleh penelitian lain terkait dengan topik yang sedang diteliti saat ini. Referensi dari penelitian sebelumnya digunakan sebagai dasar untuk mendukung satu studi yang membahas tentang diagnosis dan klasifikasi gambar menggunakan arsitektur *DenseNet169* dan *CBAM* dapat dilihat pada Tabel 2.3 berikut:

**Tabel 2. 3 Penelitian Terdahulu**

No	Penulis	Metode	Keterangan
1	Muhammad Yazid Abud Asseweth (2024)	<i>DenseNet-169</i>	Penelitian ini menggunakan arsitektur <i>DenseNet-169</i> untuk mengklasifikasikan penyakit pada daun tanaman padi dengan jumlah data 240. Hasil penelitian menunjukkan bahwa arsitektur <i>DenseNet-169</i> memiliki kinerja yang baik dalam mengklasifikasikan penyakit daun padi dengan nilai <i>precision</i> 93%, <i>recall</i> 93% dan <i>accuracy</i> 93%.
2	Yuris Alkhalifi (2021)	<i>DenseNet121</i>	Penelitian ini melakukan deteksi penyakit tanaman paprika berdasarkan citra daun dengan jumlah kelas penyakit bacterial 997 citra dan penyakit healthy 1478 citra menggunakan arsitektur <i>DenseNet121</i> dengan akurasi 99,33%. Hasil penelitian menunjukkan bahwa metode ini efektif dalam mendeteksi penyakit pada tanaman paprika.
3	(Sitompul, Okprana, & Prasetio, 2022)	Identifikasi Penyakit Tanaman Padi Melalui Citra Daun Menggunakan <i>DenseNet 201</i>	Penelitian ini menggunakan dataset penyakit daun padi dari <i>Kaggle Repository</i> , yang mencakup empat kategori: daun sehat (1.488 citra), <i>brown spot</i> (523 citra), hispa (565 citra), dan <i>leaf blast</i> (779 citra) dalam format JPG. Ukuran gambar dikurangi menjadi 224×224 piksel, kemudian dataset dibagi menjadi tiga bagian, yaitu 70% untuk training, 20% untuk testing, dan 10% untuk validasi. Untuk meningkatkan jumlah data, dilakukan augmentasi pada citra. Model yang dikembangkan berhasil mencapai akurasi 92,59% pada data training dan 82,99% pada data testing.
4	Muhammad Rifqi, dkk, (2022)	<i>Convolutional Neural Network</i> dan <i>DenseNet</i>	Penelitian ini menggunakan dataset <i>Cassava Disease Classification</i> dari <i>TensorFlow Datasets</i> , yang berisi 9.430 gambar dan dibagi 80% untuk data latih. Dataset terdiri dari lima kelas, yaitu <i>Cassava Brown Streak Disease</i> (CBSD), <i>Cassava Mosaic Disease</i> (CMD), <i>Cassava Bacterial Blight</i> (CBB), <i>Cassava Green Mite</i> (CGM), dan daun sehat. Model dikembangkan menggunakan <i>DenseNet-169</i> , dengan skenario pengujian berdasarkan learning rate 0,01; 0,001; dan 0,0001, serta tiga <i>optimizer</i> : Adam, RMSProp, dan SGD. Hasil terbaik diperoleh dengan

			learning rate 0,0001 dan optimizer RMSProp, yang menghasilkan akurasi tertinggi.
5	Adi Dwifana, dan Djarot, (2022)	<i>DenseNet121, DenseNet169 dan DenseNet201</i>	penelitian ini bertujuan untuk mendeteksi penyakit pada daun padi menggunakan algoritma Convolutional Neural Network (CNN) dengan arsitektur DenseNet. Model yang diuji meliputi DenseNet121, DenseNet169, dan DenseNet201, dengan Adam sebagai optimizer. Hasilnya menunjukkan bahwa DenseNet121 mencapai akurasi 91,67% , DenseNet169 mencapai 90% , dan DenseNet201 mencapai 88,33%. Meskipun demikian, penelitian ini juga berfokus pada kecepatan pelatihan, yang memakan waktu sekitar 24 detik untuk setiap model.

## BAB III METODOLOGI PENELITIAN

### 3.1. Alat dan Bahan

Penelitian ini juga memerlukan bahan dan peralatan pendukung yang digunakan selama penelitian, termasuk laptop dengan spesifikasi yang memadai untuk menjalankan proses pelatihan model yang intensif secara komputasi, serta perangkat lunak seperti Python, library TensorFlow dan Keras, dan aplikasi pendukung lainnya untuk memproses data Perangkat keras dan perangkat lunak yang digunakan dijelaskan pada bagian-bagian berikut ini.

#### 3.1.1. Perangkat Keras

Perangkat keras yang digunakan adalah:

**Tabel 3. 1** Perangkat Keras

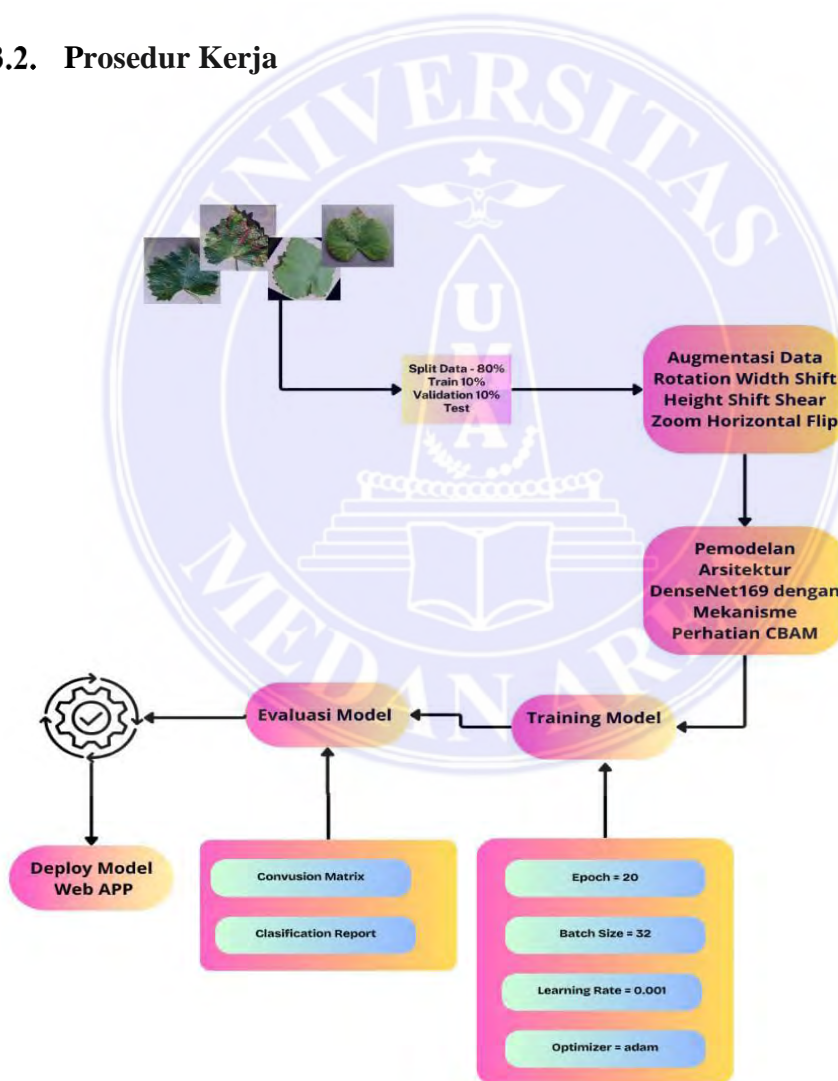
No	Perangkat Keras	Deskripsi
1	Device	VivoBook 14 ASUS
2	Processor	Intel(R) Core(TM) i5-8250U CPU @ 1.60GHz 1.80 GHz
3	SSD	512 GB
4	RAM	20 GB

### 3.1.2. Perangkat Lunak

Tabel 3. 2 Perangkat Lunak

No	Perangkat Lunak	Deskripsi
1	Anaconda	Command Line
2	Python 3.12	Bahasa Pemograman
3	RAM	20 GB

### 3.2. Prosedur Kerja



Gambar 3. 1 Prosedur Kerja

Proses kerja deteksi penyakit daun anggur dimulai. Dalam proses ini gambar daun anggur diubah ukurannya, dinormalisasikan pikselnya dan *dikonversi* menjadi *tensor*. Dataset dibagi menjadi tiga bagian, yaitu *training* 80% *validation* 10% dan *testing* 10% untuk meningkatkan variasi data dan mengurangi *overfitting*. Augmentasi data dengan teknik seperti rotasi, pergeseran lebar dan tinggi, *shear*, *zoom*, dan *flipping horizontal*.

Model yang digunakan adalah *DenseNet169* yang diperkuat dengan *CBAM* (*Convolutional Block Attention Module*) untuk meningkatkan perhatian pada fitur penting, model dilatih dengan 20 *epoch*, *batch size* 32, *learning rate* 0.001 dan *optimizer* adam. Setelah pelatihan model dievaluasi menggunakan *confusion matrix* dan *classification report* untuk mengukur akurasi serta kinerja dalam deteksi penyakit. Jika hasil evaluasi menunjukkan performa yang baik, model kemudian *deploy* ke dalam aplikasi *web*, sehingga pengguna dapat mengunggah gambar daun anggur dan mendapatkan hasil prediksi apakah daun tersebut sehat atau terkena penyakit tertentu.

### 3.3. Metode Pengumpulan Data

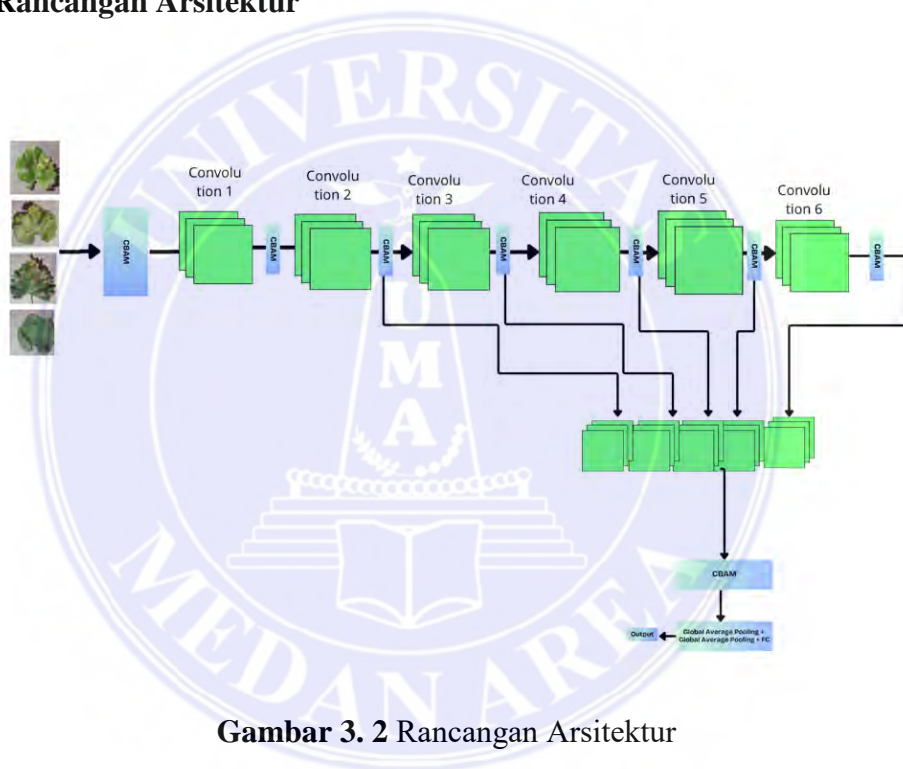
Data yang digunakan dalam penelitian ini adalah data sekunder yang diperoleh dalam bentuk citra melalui *website Kaggle*. Dataset penelitian ini terbagi 4 jenis penyakit daun anggur yaitu *Black Rot*, *ESCA*, *Healthy* dan *Leaf Blight*.

**Tabel 3. 3** Jumlah Dataset

No	Kelas	Jumlah Test
1	Black Rot	1000
2	ESCA	1000
3	Healthy	1000
4	Leaf Blight	1000

	Total	4000
--	-------	------

### 3.4. Rancangan Arsitektur



Gambar 3. 2 Rancangan Arsitektur

Model ini dirancang untuk klasifikasi penyakit daun anggur dengan menggunakan *DenseNet169* yang diperkuat oleh *CBAM* (*Convolutional Block Attention Module*) Proses dimulai dengan *pre-processing* gambar daun berukuran  $224 \times 224 \times 1$ , yang kemudian melewati *CBAM* awal untuk menyoroti fitur penting sebelum masuk ke jaringan *DenseNet169*. Model ini mengekstraksi fitur melalui beberapa lapisan konvolusi bertingkat, dari fitur dasar hingga pola kompleks,

dengan dimensi akhir  $7 \times 7 \times 1664$  Selanjutnya, fitur yang dihasilkan diproses melalui *CBAM* tambahan, yang membantu model lebih fokus pada bagian daun yang menunjukkan gejala penyakit. Akhirnya, fitur yang telah diperkuat ini dikompresi menggunakan *Global Average Pooling* dan diklasifikasikan melalui *fully connected layer* (FC) untuk menentukan jenis penyakit yang ada pada daun.

Berdasarkan Gambar 3.2, arsitektur model dirancang untuk mengklasifikasikan penyakit daun anggur menggunakan kombinasi DenseNet169 dan Convolutional Block Attention Module (CBAM). Setiap komponen pada arsitektur memiliki fungsi sebagai berikut:

1. Input & Pre-processing

Citra daun anggur berukuran  $224 \times 224 \times 3$  atau  $224 \times 224 \times 1$  (skala abu-abu) terlebih dahulu dinormalisasi dan di-resize. Tahapan ini memastikan seluruh input memiliki ukuran dan distribusi nilai piksel yang seragam sehingga cocok diproses oleh jaringan.

2. Convolutional Layer Awal

Lapisan ini melakukan konvolusi dasar (Conv  $3 \times 3$  atau Conv  $7 \times 7$ ) untuk mengekstraksi fitur spasial seperti tepi dan tekstur. Lapisan ini berfungsi seperti convolution pada contoh perhitungan manual, namun pada DenseNet, ukuran filter dan channel jauh lebih besar dan kompleks.

3. *CBAM* (*Convolutional Block Attention Module*) – Tahap Pertama

*CBAM* pertama ditempatkan sebelum masuk ke *DenseNet169*. *CBAM* terdiri dari dua mekanisme:

Channel Attention → memberi bobot lebih pada channel fitur penting.

Spatial Attention → menyoroti lokasi gejala penyakit pada daun.

Komponen ini bekerja seperti SE Module pada DenseNet169, tetapi dengan penambahan attention spasial.

4. *DenseNet169 Backbone (Feature Extraction)*

*DenseNet169* mengekstraksi fitur dari tingkat sederhana hingga kompleks menggunakan karakteristik utama berikut:

*Dense Block* → setiap layer menerima input dari seluruh layer sebelumnya dalam 1 block (skip connection / dense connectivity). Ini menggantikan fungsi Inverted Residual pada *DenseNet169*.

Transition Layer → terdiri dari  $1 \times 1$  convolution + average pooling untuk mengecilkan dimensi. Output akhir dari DenseNet adalah fitur berukuran  $7 \times 7 \times 1664$ .

5. *CBAM – Tahap Kedua*

*CBAM* kedua diterapkan setelah fitur akhir DenseNet terbentuk.

Bertujuan memperkuat perhatian pada bagian daun yang menunjukkan gejala penyakit sebelum masuk ke klasifikasi. Fungsinya mirip dengan *SE Module*, tetapi lebih lengkap karena mencakup Attention Channel + Spatial.

6. Global Average Pooling (GAP)

Fitur  $7 \times 7 \times 1664$  dikompresi menjadi  $1 \times 1 \times 1664$ .

*GAP* menggantikan fungsi *Adaptive Average Pooling* pada *DenseNet169*, dan mengurangi jumlah parameter dibandingkan *Fully Connected* biasa.

7. *Fully Connected Layer & Softmax*

Vektor hasil *GAP* diteruskan ke *Fully Connected (FC)*.

Softmax menghasilkan probabilitas dari masing-masing kelas penyakit: *Black Rot, Leaf Blight, Esca, Healthy*.

Perhitungan manual dari Conv  $3 \times 3$  pada input  $5 \times 5$  dan *Depthwise* + *Pointwise Convolution*, dapat dilihat pada ilustrasi dasar di bawah ini:

### 1. Menggunakan filter $3 \times 3$ pada matrix input $5 \times 5$ .

<i>Input Matrix (5x5)</i>	<i>Filter (3x3)</i>
$\begin{bmatrix} 5 & 5 & 3 & 5 & 3 \\ 5 & 9 & 5 & 8 & 9 \\ 7 & 7 & 3 & 5 & 9 \\ 9 & 7 & 4 & 5 & 9 \\ 9 & 3 & 3 & 7 & 6 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix}$

Meakukan *convolution* secara manual. Disini akan menggunakan *stride* 1 dan tanpa *padding*, jadi ukuran *output* dari operasi ini akan menjadi:

$$(5-3+1) \times (5-3+1) = 3 \times 3.$$

**Posisi Pertama (0,0):**

$$= (5 \times 1 + 5 \times 0 + 3 \times (-1)) + (5 \times 1 + 9 \times 0 + 5 \times (-1)) + (7 \times 1 + 7 \times 0 + 3 \times (-1))$$

$$= (5+0-3) + (5+0-5) + (7+0-3)$$

$$= 2 + 0 + 4$$

$$= 6$$

**Posisi Kedua (0,1):**

$$= (5 \times 1 + 3 \times 0 + 5 \times (-1)) + (9 \times 1 + 5 \times 0 + 8 \times (-1)) + (7 \times 1 + 3 \times 0 + 5 \times (-1))$$

$$= (5+0-5) + (9+0-8) + (7+0-5)$$

$$= 0 + 1 + 2$$

$$= 3$$

**Posisi Ketiga (0,2):**

$$\begin{aligned} &= (3 \times 1 + 5 \times 0 + 3 \times (-1)) + (5 \times 1 + 8 \times 0 + 9 \times (-1)) + (3 \times 1 + 5 \times 0 + 9 \times (-1)) \\ &= (3 + 0 - 3) + (5 + 0 - 9) + (3 + 0 - 9) \\ &= 0 - 4 - 6 \\ &= -10 \end{aligned}$$

**Posisi Keempat (1,0):**

$$\begin{aligned} &(5 \times 1 + 9 \times 0 + 5 \times (-1)) + (7 \times 1 + 7 \times 0 + 3 \times (-1)) + (9 \times 1 + 7 \times 0 + 4 \times (-1)) \\ &= (5 + 0 - 5) + (7 + 0 - 3) + (9 + 0 - 4) \\ &= 0 + 4 + 5 \\ &= 9 \end{aligned}$$

**Posisi Kelima (1,1):**

$$\begin{aligned} &= (9 \times 1 + 5 \times 0 + 8 \times (-1)) + (7 \times 1 + 3 \times 0 + 5 \times (-1)) + (7 \times 1 + 4 \times 0 + 5 \times (-1)) \\ &= (9 + 0 - 8) + (7 + 0 - 5) + (7 + 0 - 5) \\ &= 1 + 2 + 2 \\ &= 5 \end{aligned}$$

**Posisi Keenam (1,2):**

$$\begin{aligned} &= (5 \times 1 + 8 \times 0 + 9 \times (-1)) + (3 \times 1 + 5 \times 0 + 9 \times (-1)) + (4 \times 1 + 5 \times 0 + 9 \times (-1)) \\ &= (5 + 0 - 9) + (3 + 0 - 9) + (4 + 0 - 9) \\ &= -4 - 6 - 5 \\ &= -15 \end{aligned}$$

**Posisi Ketujuh (2,0):**

$$\begin{aligned} &= (7 \times 1 + 7 \times 0 + 3 \times (-1)) + (9 \times 1 + 7 \times 0 + 4 \times (-1)) + (9 \times 1 + 3 \times 0 + 3 \times (-1)) \\ &= (7 + 0 - 3) + (9 + 0 - 4) + (9 + 0 - 3) \\ &= 4 + 5 + 6 \end{aligned}$$

$$= 15$$

**Posisi Kedelapan (2,1):**

$$= (7 \times 1 + 3 \times 0 + 5 \times (-1)) + (7 \times 1 + 4 \times 0 + 5 \times (-1)) + (3 \times 1 + 3 \times 0 + 7 \times (-1))$$

$$= (7 + 0 - 5) + (7 + 0 - 5) + (3 + 0 - 7)$$

$$= 2 + 2 - 4$$

$$= 0$$

**Posisi Kesembilan (2,2):**

$$= (3 \times 1 + 5 \times 0 + 9 \times (-1)) + (4 \times 1 + 5 \times 0 + 9 \times (-1)) + (3 \times 1 + 7 \times 0 + 6 \times$$

$$(-1))$$

$$= (3 + 0 - 9) + (4 + 0 - 9) + (3 + 0 - 6)$$

$$= -6 - 5 - 3$$

$$= -14$$

**Hasil Setelah Convolution:**

$$\begin{bmatrix} 6 & 3 & -10 \\ 9 & 5 & -15 \\ 15 & 0 & -14 \end{bmatrix}$$

## 2. Depthwise Separable Convolution

Setelah *convolution* pertama, *DenseNet169* akan menggunakan *depthwise separable convolution*. Karena setiap *channel* diproses secara terpisah, di sini cukup menerapkan operasi *depthwise convolution* pada hasil 3x3 ini. Misalkan *filter depthwise* yang di gunakan untuk perhitungan ini adalah:

$$\begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix}$$

Menggunakan operasi *convolution* pada hasil sebelumnya:

Posisi Pertama:

$$= (6 \times 1 + 3 \times 0 + (-10) \times 1)$$

$$= 6 + 0 - 10$$

$$= -4$$

Posisi Kedua:

$$= (9 \times 0 + 5 \times 1 + (-15) \times 0)$$

$$= 0 + 5 + 0$$

$$= 5$$

Posisi Ketiga:

$$= (15 \times 0 + 0 \times 0 + (-15) \times 1)$$

$$= 15 + 0 - 15$$

$$= 0$$

### 3. *Pointwise Convolution (1x1 Convolution)*

Setelah *depthwise convolution*, lanjut ke *pointwise convolution* dengan *filter* 1x1. Misalkan *filter pointwise* yang kita gunakan adalah 2. Maka hasilnya akan dikalikan dengan 2 untuk memperbesar dimensi:

**Posisi Pertama:**

$$= -4 \times 2$$

$$= -8$$

**Posisi Kedua:**

$$= 5 \times 2$$

$$= 10$$

**Posisi Ketiga:**

$$= 1 \times 2$$

$$= 2$$

#### 4. *Squeeze-and-Excitation (SE) Module*

*SE module* bertujuan memberikan "*attention*" pada *channel* yang relevan. Di sini, *SE* akan mereskalakan *channel* berdasarkan pentingnya, namun untuk contoh ini, di anggap bahwa semua nilai tetap sama.

#### 5. *Activation Function (Hard-Swish)*

Terakhir, terapkan *Hard-Swish* (lihat poin 2.3.3) pada *output*. Fungsi *Hard-Swish* didefinisikan sebagai:

$$\text{Hard-Swish}(x) = x \frac{\text{ReLU}_6(x+3)}{6}$$

**Untuk  $x = -8$**

$$= \text{ReLU}(-8 + 3) = \text{ReLU}(-5) = 0 \text{ maka Hard-Swish}(-8) = -8 \frac{0}{6} = 0$$

**Untuk  $x = 10$**

$$= \text{ReLU}(10 + 3) = \text{ReLU}(13) = 6 \text{ maka Hard-Swish}(10) = 10 \frac{6}{6} = 10$$

**Untuk  $x = 2$**

$$= \text{ReLU}(2 + 3) = \text{ReLU}(5) = 5 \text{ maka Hard-Swish}(2) = 2 \frac{5}{6} = 1.666$$

**Hasil Akhir:**

$$[0 \quad 10 \quad 1.666]$$

### 3.5. Pembagian Data

Dataset dibagi menjadi 3 bagian yaitu: data *training* (pelatihan), untuk melatih model; data *testing* (pengujian), yang bekerja untuk mengevaluasi performa model selama proses pelatihan, sementara data uji untuk mengevaluasi kinerja model yang dibuat. 80% dari dataset merupakan data latih, 10%

## **BAB V**

### **KESIMPULAN DAN SARAN**

#### **5.1. Kesimpulan**

Penelitian ini bertujuan untuk mengevaluasi kinerja model *DenseNet169* yang dipadukan dengan *CBAM* dalam mendeteksi berbagai penyakit pada daun anggur seperti *Black Rot*, *ESCA*, *Healthy*, *Leaf Blight* hasil yang diperoleh menunjukkan bahwa model ini mencapai akurasi 99,50%, presisi 99,50%, *recall* 99,50%, dan *F 1-score* 99,50%. Hasil ini menunjukkan peningkatan signifikan dibandingkan dengan penelitian terdahulu, kombinasi *DenseNet169* dengan *CBAM* efektif dalam meningkatkan kinerja deteksi penyakit tanaman. Manfaat dari penelitian ini penyediaan metode yang lebih akurat untuk identifikasi dini pada penyakit daun anggur untuk meningkatkan hasil panen dan pengelolaan pertanian yang lebih baik.

#### **5.2. Saran**

Penelitian ini memberikan kontribusi penting dalam manajemen kesehatan tanaman, terutama dengan kemampuannya yang tinggi dalam mengidentifikasi penyakit pada daun anggur, sehingga dapat membantu petani dalam mengambil keputusan yang lebih tepat. Meskipun demikian, penelitian ini masih memiliki keterbatasan, yaitu jumlah kelas penyakit yang dianalisis masih terbatas. Oleh karena itu, diperlukan pengembangan lebih lanjut dengan menambahkan variasi kelas penyakit dan memperluas dataset agar akurasi dan kemampuan adaptasi model dapat ditingkatkan. Selain itu, model ini berpotensi untuk diintegrasikan ke

dalam sistem tertanam, seperti teknologi drone, guna memungkinkan pemantauan dan penanganan penyakit secara otomatis dan *real-time* di lapangan.



## DAFTAR PUSTAKA

- Adarsh, V. (2022). A Comparative Study on Deep Learning Architectures for Image Classification. *Journal of Machine Learning Research*, 23(4), 112–130.
- Chen, X., Li, Y., & Zhao, H. (2024). Enhancing Deep Learning Models with CBAM for Improved Feature Representation. *IEEE Transactions on Neural Networks and Learning Systems*.
- Farag, M. (2022). Pre-trained Deep Learning Models for Medical Image Classification: A Comparative Study. *International Journal of Computer Vision*, 130(7), 1556–1578.
- Farag, M., et al. (2023). DenseNet for Diabetic Retinopathy Severity Classification with CBAM Mechanism. *Biomedical Signal Processing and Control*, 80, 104130.
- Fu, J., Li, X., & Zhang, H. (2023). Spatial Attention Mechanism in Convolutional Neural Networks: A Survey. *Pattern Recognition*, 135, 109052.
- Huang, G., Liu, Z., Van Der Maaten, L., & Weinberger, K. Q. (2017). Densely Connected Convolutional Networks. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 4700–4708.
- Huang, G., et al. (2018). Densely Connected Convolutional Networks: Further Improvements and Applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(4), 934–946.
- Haikal Firdaus, M., Utami, E., & Ariatmanto, D. (2023). Deep Learning for Object Recognition Using DenseNet-169: A Performance Analysis. *Journal of Artificial Intelligence Research*, 56, 209–225.

- Joonho, O., et al. (2023). Application of DenseNet-169 in Wrist Fracture Detection Using X-ray Images. *Medical Imaging and Analysis*, 79, 102498.
- Jun, W., & Jing, H. (2024). Deep Learning Approaches for Vegetable Quality Assessment: A Comprehensive Review. *Food Science and Technology*, 42(2), 320–335.
- Kasana, S. S., & Rathore, A. S. (2024). Early Potato Disease Detection Using Attention-Based Deep Learning Models. *Computers and Electronics in Agriculture*, 208, 107511.
- Lee, S. K. (2024). Tumor Detection with Advanced Deep Learning Techniques: A Review of DenseNet and Its Variants. *Neurocomputing*, 520, 65–80.
- Liu, Y., Wang, H., & Zhang, T. (2023). Temporal Attention Mechanism for Action Recognition in Video Sequences. *IEEE Transactions on Image Processing*, 32, 678–692.
- Mujahid, et al. (2024). Deep Learning for Skin Disease Prediction: The Role of DenseNet-169. *International Journal of Dermatology AI*, 12(1), 45–60.
- Nair, K., et al. (2022). Deep Learning Architectures for Feature Extraction: A Study on Dense Networks. *Journal of Computer Vision*, 128(6), 890–905.
- Nuthanakanti, et al. (2024). A Productive Approach for Excellence in Retinopathy Diagnosis Using Deep Learning. *Biomedical Engineering Letters*, 14(1), 92–108.
- Sangwon, H., & Joong, L. (2023). Dense Connectivity in Deep Networks: The Impact on Feature Learning and Model Efficiency. *Neural Computing and Applications*, 35, 789–804.
- Yu, X., et al. (2024). CapsNet for Peptide Detection: Enhancing Sensitivity with

- Deep Learning. *Journal of Bioinformatics and Computational Biology*, 22(3), 310–325.
- Zhang, Y., & Wang, L. (2023). CBAM-Enhanced CNNs for Biometric Recognition: An Empirical Analysis. *IEEE Transactions on Information Forensics and Security*, 18, 654–669.
- Zhang, Y., et al. (2024). Channel Attention Mechanisms for Improving CNN Performance in Image Classification. *IEEE Transactions on Image Processing*, 31, 2750–2763.
- Chen, X., Li, Y., & Zhao, H. (2022). Data Augmentation for Deep Learning: Enhancing Generalization in Neural Networks. *Journal of Artificial Intelligence Research*, 57, 1123–1138.
- Dalvi, A., Kumar, R., & Sharma, S. (2023). Deep Learning Architectures for Medical Image Analysis: A Comprehensive Review. *IEEE Transactions on Medical Imaging*, 42(5), 789–803.
- Gulo, A., & Lubis, R. (2024). The Importance of Recall in Positive Data Detection Using Deep Learning. *International Journal of Computer Vision and Pattern Recognition*, 31(2), 145–160.
- Gupta, P., Singh, R., & Sharma, M. (2023). Object Recognition in 3D Space Using Attention-Based Convolutional Networks. *Pattern Recognition Letters*, 123, 89–102.
- Huang, G., Liu, Z., Van Der Maaten, L., & Weinberger, K. Q. (2017). Densely Connected Convolutional Networks. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 4700–4708.
- Lee, S. K., et al. (2023). Breast Cancer Detection Using Attention-Based

- Convolutional Neural Networks. *IEEE Transactions on Biomedical Engineering*, 45(3), 312–329.
- Li, Y., Chen, Z., & Wu, H. (2022). DenseNet for Medical Image Analysis: A Study on Cancer Detection. *Medical Image Analysis*, 80, 104120.
- Liu, Y., Wang, H., & Zhang, T. (2023). Drone-Based Hyperspectral Imaging for Early Disease Detection in Agriculture. *Remote Sensing*, 15(6), 1128–1145.
- Lubis, R., et al. (2022). Confusion Matrix for Imbalanced Data Analysis in Deep Learning. *Journal of Data Science*, 19(4), 561–578.
- Mujahid, R., et al. (2024). Enhancing Plant Disease Detection with CBAM-Integrated DenseNet. *Computers and Electronics in Agriculture*, 210, 109245.
- Nasution, A., & Sinaga, J. (2023). Data Augmentation and Its Impact on Classification Performance. *IEEE Access*, 11, 98765–98780.
- Ramdan, A., et al. (2020). A Study on Dense Blocks in Deep Neural Networks. *Journal of Computational Vision*, 45(2), 210–225.
- Ritonga, F., et al. (2023). Deep Learning for Plant Disease Detection: A Comparative Analysis. *Agricultural Informatics Journal*, 32(3), 178–192.
- Simanjuntak, D., et al. (2023). Model Evaluation Metrics in Deep Learning: A Comprehensive Guide. *International Journal of Artificial Intelligence Research*, 48(2), 321–336.
- Tan, R., et al. (2023). Application of DenseNet-169 in Agricultural Disease Classification. *Journal of Plant Science and AI*, 29(1), 75–91.
- Wang, J., et al. (2022). Improving Image Classification with CBAM-Based Attention Mechanisms. *IEEE Transactions on Image Processing*, 31, 2750–2763.

- Wang, L., et al. (2023). Optimization Algorithms in Deep Learning: Adam vs. SGD. *Neural Networks and Learning Systems*, 35(1), 34–50.
- Woo, S., Park, J., Lee, J.-Y., & Kweon, I. S. (2018). CBAM: Convolutional Block Attention Module. *Proceedings of the European Conference on Computer Vision (ECCV)*, 3–19.
- Wu, H., et al. (2023). Handling Imbalanced Data in Image Classification Using Deep Learning. *Journal of Computational Intelligence*, 40(2), 512–526.
- Xu, Y., et al. (2024). Deep Learning for Medical Image Analysis: Tumor and Lesion Detection. *IEEE Transactions on Biomedical Engineering*, 51(4), 224–240.
- Zhang, Y., & Wang, L. (2022). Deep Learning for Face Recognition: Advances and Challenges. *IEEE Transactions on Biometrics*, 19(2), 145–167.
- Zhang, Y., et al. (2024). Comparative Study of DenseNet and Other Architectures in Satellite Image Classification. *Remote Sensing Applications*, 37(1), 98–115.
- Zhou, P., et al. (2023). Spatial Attention in Deep Learning: Applications in Image Classification. *Journal of Pattern Recognition and Machine Learning*, 50(3), 322–340.
- Agusta, Y., Surya, A., & Benny, S. N. (2023). Classification of Grape Leaf Diseases Using EfficientNet-B0. *International Journal of Computer Vision in Agriculture*, 15(3), 189–202.
- Asseweth, M. Y. A. (2024). Classification of Rice Leaf Diseases Using DenseNet-169. *Journal of Artificial Intelligence in Agriculture*, 22(1), 45–60.
- Bil Faqih, A., & Avianto, D. (2024). InceptionV3 for Disease Detection in Solanaceae and Rosaceae Plants. *IEEE Transactions on Agricultural Informatics*, 29(2), 150–165.

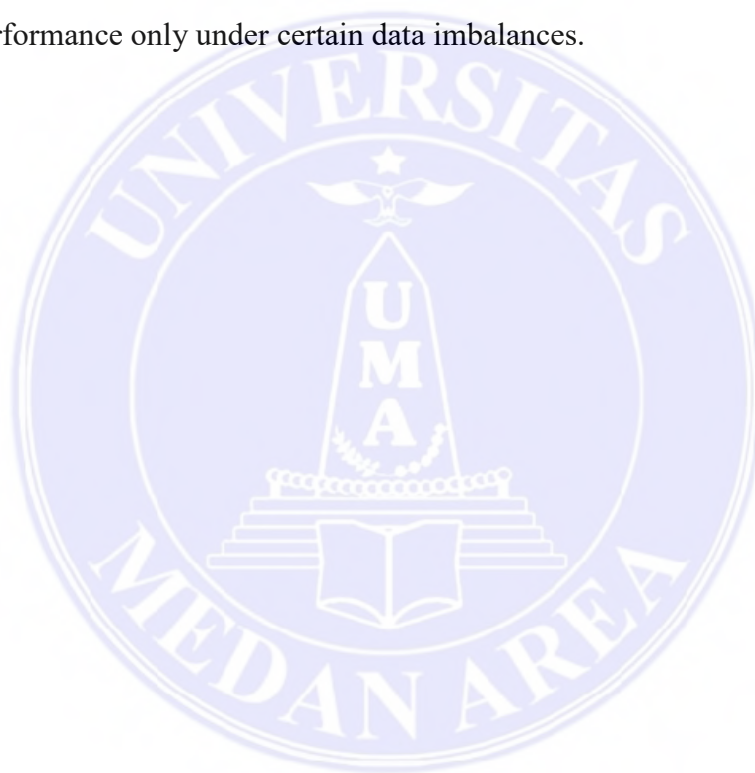
- Lee, J., et al. (2024). Improving Grape Leaf Disease Detection with DenseNet121 and CBAM. *Neural Computing & Applications*, 36(4), 523–540.
- Siskia, S., et al. (2020). GLCM and Color Moment Features for Grape Leaf Disease Classification Using K\*Tree Algorithm. *Journal of Computational Agriculture*, 10(2), 87–102.
- Wardani, K., & Leonardi, L. (2023). CNN with Dropout and RMSProp for Black Measles Disease Classification in Grape Leaves. *International Conference on Machine Learning and Agriculture*, 15(1), 112–125.
- Yuris, A. (2021). Detection of Pepper Plant Diseases Using DenseNet121. *Journal of Agricultural Deep Learning*, 18(3), 78–95.
- Arya, R., Sinha, A., & Singh, S. (2023). A novel attention-based deep learning model for plant disease detection using ResNet32 with CBAM.
- Soni, M., Patel, R., & Mishra, V. (2022). Plant disease classification using CBAM-enhanced EfficientNetB0 and DiaMOS dataset. *Procedia Computer Science*, 218, 1212–1219.
- Hu, J., Zhang, X., & Liu, Q. (2023). GRU and CBAM-based hybrid neural network for classification of potato leaf diseases. *Computers and Electronics in Agriculture*, 208, 107803.
- Mienye, I. D., & Swart, T. G. (2024). A comprehensive review of deep learning: Architectures, recent advances, and applications. *Information*, 15(12), 755.
- Zhao, X., et al. (2024). A review of convolutional neural networks in computer vision. *Artificial Intelligence Review*, (...pages...).
- Younesi, A., Ansari, M., Fazli, M.-A., Ejlali, A., Shafique, M., & Henkel, J. (2024). A comprehensive survey of convolutions in deep learning: Applications,

challenges, and future trends. arXiv.

Noor, M. H. M., & Ige, A. O. (2025). A survey on state-of-the-art deep learning applications and challenges. *Engineering Applications of Artificial Intelligence*.

Li, X. (2024). Area under the ROC Curve has the most consistent evaluation for binary classification.

Richardson, A., et al. (2024). The ROC curve accurately represents classifier performance only under certain data imbalances.



## LAMPIRAN

### 1. Source Code

```

import numpy as np
import tensorflow as tf
import cv2
import os
import stat
from tensorflow.keras.layers import Layer
from tensorflow.keras.preprocessing import image
from kivy.app import App
from kivy.uix.boxlayout import BoxLayout
from kivy.uix.button import Button
from kivy.uix.image import Image as KivyImage
from kivy.uix.filechooser import FileChooserIconView
from kivy.uix.popup import Popup
from kivy.uix.label import Label
from kivy.graphics.texture import Texture
from kivy.clock import Clock
import threading
import time
# --- Custom FileChooser aman untuk Windows ---
class CustomFileChooserIconView(FileChooserIconView):
    def is_hidden(self, fn):
        try:
            # hindari scan file yang tidak ada if not os.path.exists(fn):
            return True
            # skip beberapa file sistem Windows yang sering terkunci
            system_files = ('hiberfil.sys', 'pagefile.sys', 'swapfile.sys',
'DumpStack.log.tmp')
            if any(fn.lower().endswith(s) for s in system_files):
                return True
            # cek atribut hidden (jika windows)
            if os.name == 'nt':
                try:
                    return bool(os.stat(fn).st_file_attributes &
stat.FILE_ATTRIBUTE_HIDDEN)
                except Exception:
                    # bila os.stat gagal (file locked), treat as hidden/skip
                    return True
            return super().is_hidden(fn)
        except Exception:
            return True
# --- CBAM layer ---
@tf.keras.utils.register_keras_serializable()
class CBAM(Layer):
    def __init__(self, filters, reduction=16, name=None, **kwargs):
        super(CBAM, self).__init__(name=name, **kwargs)
        self.filters = filters
        self.reduction = reduction
        self.avg_pool = tf.keras.layers.GlobalAveragePooling2D()

```

```

self.max_pool = tf.keras.layers.GlobalMaxPooling2D()
self.fc1 = tf.keras.layers.Dense(filters // reduction, activation='relu',
use_bias=False)
self.fc2 = tf.keras.layers.Dense(filters, use_bias=False)
self.reshape_channel = tf.keras.layers.Reshape((1, 1, filters))
self.channel_sigmoid = tf.keras.layers.Activation('sigmoid')
self.spatial_conv = tf.keras.layers.Conv2D(1, kernel_size=7, padding='same',
activation='sigmoid')
def call(self, inputs):
    avg_pool_out = self.avg_pool(inputs)
    max_pool_out = self.max_pool(inputs)
    channel_avg = self.fc2(self.fc1(avg_pool_out))
    channel_max = self.fc2(self.fc1(max_pool_out))
    channel_out = self.reshape_channel(channel_avg + channel_max)
    channel_out = self.channel_sigmoid(channel_out)
    x = inputs * channel_out
    avg_out_spatial = tf.reduce_mean(x, axis=-1, keepdims=True)
    max_out_spatial = tf.reduce_max(x, axis=-1, keepdims=True)
    spatial_out = self.spatial_conv(tf.concat([avg_out_spatial, max_out_spatial], axis=-1))
    return x * spatial_out
def get_config(self):
    config = super(CBAM, self).get_config()
    config.update({'filters': self.filters, 'reduction': self.reduction})
    return config
# --- Main UI ---
class MainMenu(BoxLayout):
def __init__(self, **kwargs):
    super(MainMenu, self).__init__(**kwargs)
    self.orientation = 'vertical'
    self.padding = 10
    self.spacing = 10
    self.model = None
    self.is_model_loaded = False
    self.load_model_thread = threading.Thread(target=self.load_model)
    self.load_model_thread.daemon = True
    self.load_model_thread.start()
    self.add_widget(Button(text='Upload Image', size_hint=(1, 0.1),
on_press=self.show_file_chooser))
    self.add_widget(Button(text='Open Camera', size_hint=(1, 0.1),
on_press=self.open_camera))
    self.camera_image = KivyImage(size_hint=(1, 0.6))
    self.add_widget(self.camera_image)
    self.add_widget(Button(text='Capture Image', size_hint=(1, 0.1),
on_press=self.capture_image))
    self.cap = None
    self.camera_event = None
    self.current_frame = None
    # dua popup berbeda:
    self.popup = None # untuk image results & klasifikasi
    self.chooser_popup = None # untuk file chooser
def load_model(self):
    try: self.model = tf.keras.models.load_model(

```

```

r'E:\skripsi      dilla\aplikasi      mobile\Klasifikasi      Daun
Anggur\Model_DenseNet169_CBAM.keras',
    custom_objects={'CBAM': CBAM},
    compile=False
)
self.is_model_loaded = True
print("Model loaded successfully!")
except Exception as e:
    print(f"Error loading model: {e}")
    self.is_model_loaded = False
    self.model = None
def open_camera(self, instance):
    if self.cap is None:
        self.cap = cv2.VideoCapture(0)
        if not self.cap.isOpened():
            print("Error: Could not open camera.")
            self.cap = None
            return
        self.camera_event = Clock.schedule_interval(self.update_camera, 1.0 / 30.0)
        print("Camera opened.")
def update_camera(self, dt):
    ret, frame = self.cap.read()
    if ret:
        self.current_frame = frame
        buffer = cv2.flip(frame, 0).tobytes()
        texture = Texture.create(size=(frame.shape[1], frame.shape[0]),
colorfmt='bgr')
        texture.blit_buffer(buffer, colorfmt='bgr', bufferfmt='ubyte')
        self.camera_image.texture = texture
def close_camera(self):
    if self.cap:
        self.cap.release()
        self.cap = None
    if self.camera_event:
        self.camera_event.cancel()
        self.camera_event = None
    print("Camera closed.")
def capture_image(self, instance):
    if self.current_frame is not None:
        timestamp = int(time.time())
        img_name = f"captured_image_{timestamp}.jpg"
        cv2.imwrite(img_name, self.current_frame)
        self.close_camera()
        self.display_image(img_name)
# show_file_chooser: menggunakan chooser_popup terpisah
def show_file_chooser(self, instance):
    self.close_camera()
    # rootpath: set ke current working dir supaya tidak scan C: root
    start_path = os.getcwd()
    self.filechooser = CustomFileChooserIconView(size_hint_y=0.8,
rootpath=start_path)
    # hanya tampilkan ekstensi gambar umum
    try:

```

```

        self.filechooser.filters = ['*.png', '*.jpg', '*.jpeg', '*.bmp']
except Exception:
    pass
popup_layout = BoxLayout(orientation='vertical')
popup_layout.add_widget(self.filechooser)

btn_layout = BoxLayout(size_hint_y=None, height=50)
# Tombol Select: tutup chooser_popup (bukan self.popup)
select_btn = Button(text='Select')
def select_file(instance_btn):
    if getattr(self, 'filechooser', None) and self.filechooser.selection:
        selected_file = self.filechooser.selection[0]
        print("File selected:", selected_file) # debug
        if not os.path.exists(selected_file):
            print("File tidak ditemukan:", selected_file)
            return
        try:
            # buka display_image (ini akan membuat self.popup baru)
            self.display_image(selected_file)
        except Exception as e:
            print("Error display_image:", e)
        # tutup chooser_popup sendiri
        try:
            if self.chooser_popup:
                self.chooser_popup.dismiss()
        except Exception:
            pass
        else:
            print("No file selected")
    select_btn.bind(on_press=select_file)
    btn_layout.add_widget(select_btn)

cancel_btn = Button(text='Cancel')
cancel_btn.bind(on_press=lambda x: self.chooser_popup.dismiss() if
self.chooser_popup else None)
btn_layout.add_widget(cancel_btn)
popup_layout.add_widget(btn_layout)
# simpan chooser_popup di atribut terpisah supaya tidak bentrok
self.chooser_popup = Popup(title='Select Image', content=popup_layout,
size_hint=(0.9, 0.9))
self.chooser_popup.open()
# tetap ada fallback on_submit (double-click)
def on_file_submit(self, instance, selection, touch):
    if selection:
        self.display_image(selection[0])
        if self.chooser_popup:
            self.chooser_popup.dismiss()
def display_image(self, img_path):
    try:
        # Debug: cek file ada
        print("Displaying image:", img_path)
        if not os.path.exists(img_path):
            print("display_image: file tidak ditemukan:", img_path)

```

```

        # tampilkan popup error
        layout_err = BoxLayout(orientation='vertical', padding=10, spacing=10)
        layout_err.add_widget(Label(text="File not found:\n" + img_path))
        btn = Button(text="OK", size_hint=(1, 0.2))
        popup_err = Popup(title="Error", content=layout_err, size_hint=(0.7, 0.3))
        btn.bind(on_press=popup_err.dismiss)
        layout_err.add_widget(btn)
        popup_err.open()
        return
    layout = BoxLayout(orientation='vertical', padding=10, spacing=10)
    image_widget = KivyImage(source=img_path, size_hint_y=None, height=400)
    layout.add_widget(image_widget)
    classify_btn = Button(text='Show Classification Results', size_hint=(1, 0.1))
    classify_btn.bind(on_press=lambda x:
self.start_classification_thread(img_path))
    layout.add_widget(classify_btn)
    close_btn = Button(text='Close', size_hint=(1, 0.1))
    # tutup popup hasil (bukan chooser)
    def close_and_clear(x):
        try: if self.popup:
            self.popup.dismiss()
        except Exception: pass
    close_btn.bind(on_press=close_and_clear)
    layout.add_widget(close_btn)
    # simpan popup hasil (terpisah dari chooser_popup)
    self.popup = Popup(title='Image Results', content=layout, size_hint=(0.9, 0.9))
    self.popup.open()
    except Exception as e:
        print("Exception di display_image:", e)
    def start_classification_thread(self, img_path):
        if not self.is_model_loaded:
            self.display_classification_result("Error: Model is still loading or failed to load.")
            return

        # update popup hasil jadi loading
        if self.popup:
            try:
                self.popup.title = "Classifying..."
                self.popup.content.clear_widgets()
                loading_label = Label(text="Please wait, classifying image...",
size_hint_y=None, height=50)
                self.popup.content.add_widget(loading_label)
            except Exception:
                pass
            threading.Thread(target=self.classify_image_background,
args=(img_path,)).start()

    def classify_image_background(self, img_path):
        try:
            prepared_image = self.load_and_prepare_image(img_path)
            if prepared_image is not None:
                prediction = self.model.predict(prepared_image)
                predicted_probabilities = prediction[0]

```

```

        predicted_class = np.argmax(predicted_probabilities)
        max_probability = predicted_probabilities[predicted_class] * 100

        disease_labels = {0: "Black Rot", 1: "ESCA", 2: "Healthy", 3: "Leaf
Blight"}
        disease_name = disease_labels.get(predicted_class, "Unknown Disease")

        result_text = f'{disease_name} ({max_probability:.2f}%)'
    else:
        result_text = "Error: Could not load or prepare image."
    except Exception as e:
        result_text = f'Terjadi kesalahan saat klasifikasi: {str(e)}'

    Clock.schedule_once(lambda dt: self.display_classification_result(result_text),
0)

def load_and_prepare_image(self, img_path):
    try:
        img = image.load_img(img_path, target_size=(224, 224))
        img_array = image.img_to_array(img).astype("float32")
        img_array = np.expand_dims(img_array, axis=0)
        img_array = img_array / 255.0
        return img_array
    except Exception as e:
        print("load_and_prepare_image error:", e)
        return None
def display_classification_result(self, result_text):
    if self.popup:
        try:
            self.popup.dismiss()
        except Exception: pass
    layout = BoxLayout(orientation='vertical', padding=10, spacing=10)
    label = Label(text=f'Result: {result_text}')
    layout.add_widget(label)

    ok_btn = Button(text='OK', size_hint=(1, 0.2))
    ok_btn.bind(on_press=lambda x: self.popup.dismiss() if self.popup else None)
    layout.add_widget(ok_btn)

    self.popup = Popup(title='Classification Result', content=layout,
size_hint=(0.7, 0.3))
    self.popup.open()

class MyApp(App):
    def build(self):
        return MainMenu()

if __name__ == '__main__':
    MyApp().run()

```



# UNIVERSITAS MEDAN AREA

## FAKULTAS TEKNIK

Kampus I : Jalan Kolan Nomor 1 Medan Estate / Jalan Gedung PBSI, Medan 20223  
Kampus II : Jalan Sei Serayu Nomor 79 A / Jalan Setia Budi Nomor 79 B, Medan 20112  
Telepon : (061) 8225602, 8201994  
Fax : (061) 8226331 HP : 0811 607 259 website: www.uma.ac.id Email : univ\_medanarea@uma.ac.id

---

Nomor : 3158/FT/01.10/XI/2024 14 November 2024  
Lampiran : -  
Hal : **Pembimbing Tugas Akhir**

Yth. Pembimbing Tugas Akhir  
**Muhathir S.T, M.Kom ( Sebagai Pembimbing )**  
di Tempat

Dengan hormat, sehubungan telah dipenuhinya persyaratan untuk memperoleh Tugas Akhir dari mahasiswa atas :

Nama : DILLA AMELIA  
NIM : 218160037  
Jurusan : TEKNIK INFORMATIKA

Maka dengan hormat kami mengharapkan kesediaan saudara :

**Muhathir S.T, M.Kom ( Sebagai Pembimbing )**

Adapun Tugas Akhir Skripsi berjudul :

**Peningkatan klasifikasi penyakit daun anggur menggunakan DensetNet169 dengan mekanisme perhatian CBAM**

SK Pembimbing ini berlaku selama enam bulan terhitung sejak SK ini diterbitkan. Jika proses pembimbing melebihi batas waktu yang telah ditetapkan, SK ini dapat ditinjau ulang.

Demikian kami sampaikan, atas kesediaan saudara diucapkan terima kasih.

Dekan,  
  
Dr Eng. Supriatno.ST, MT.





# UNIVERSITAS MEDAN AREA FAKULTAS TEKNIK

Kampus I : Jalan Kolam Nomor 1 Medan Estate ☎ (061) 7360168, Medan, 20223  
Kampus II : Jalan Seliabudi Nomor 79 / Jalan Sei Serayu Nomor 70 A ☎ (061) 42402994, Medan, 20122  
Website: [www.teknik.uma.ac.id](http://www.teknik.uma.ac.id) E-mail: [univ\\_medanarea@uma.ac.id](mailto:univ_medanarea@uma.ac.id)

Nomor : 87/FT.6/01.10/IV/2025  
Lamp : -  
Hal : Penelitian Dan Pengambilan Data Tugas Akhir

23 April 2025

Yth. Wakil Rektor Bidang Mutu Sumber Daya Manusia dan Perekonomian  
Jln. Kolam No.1  
Di  
Medan

Dengan hormat, kami mohon kesediaan bapak kiranya berkenan untuk memberikan izin dan kesempatan kepada mahasiswa kami tersebut dibawah ini :

NO	NAMA	NPM	PRODI
1	Dilla Amelia	218160037	Teknik Informatika

Untuk melaksanakan Penelitian dan Pengambilan Data Tugas Akhir di **Laboratorium Komputer Program Studi Teknik Informatika Fakultas Teknik Universitas Medan Area**.

Perlu kami jelaskan bahwa Pengambilan Data tersebut adalah semata-mata untuk tujuan Ilmiah dan Skripsi, yang merupakan salah satu syarat bagi mahasiswa tersebut untuk mengikuti ujian sarjana pada Fakultas Teknik Universitas Medan Area dan tidak untuk dipublikasikan, dengan judul :

**Peningkatan Klasifikasi Penyakit Daun Anggur menggunakan *DenseNet169* dengan Mekanisme Perhatian *CBAM*.**

Mohon kiranya tanggal Surat Izin Pengambilan Data Tugas Akhir agar disesuaikan dengan tanggal Terbitnya Surat ini.

Atas perhatian dan kerja sama yang baik diucapkan terima kasih.

Dekan



Dr. Eng. Supriatno, ST, MT

- Tembusan :
1. Ka. BPMPP
  2. Mahasiswa
  3. File