

**KLASIFIKASI CITRA PENYAKIT DAUN PADA TANAMAN
CABAI MERAH MENGGUNAKAN MODEL
CONVOLUTIONAL NEURAL NETWORK DENGAN
ARSITEKTUR VGG-16 DAN ALEXNET**

SKRIPSI

DICKY CANDID SARAGIH

198160061



FAKULTAS TEKNIK

PROGRAM STUDI TEKNIK INFORMATIKA

UNIVERSITAS MEDAN AREA

2024

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 9/1/25

Access From (repository.uma.ac.id)9/1/25

**KLASIFIKASI CITRA PENYAKIT DAUN PADA TANAMAN
CABAI MERAH MENGGUNAKAN MODEL
CONVOLUTIONAL NEURAL NETWORK DENGAN
ARSITEKTUR VGG-16 DAN ALEXNET**

SKRIPSI

*Diajukan Untuk Memenuhi Sebagai Persyaratan Dalam Memperoleh
Gelara Sarjana Universitas Medan Area*



Oleh :
DICKY CANDID SARAGIH
198160061

**FAKULTAS TEKNIK
PROGRAM STUDI TEKNIK INFORMATIKA
UNIVERSITAS MEDAN AREA
2024**

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 9/1/25

Access From (repository.uma.ac.id)9/1/25

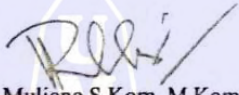
LEMBAR PENGESAHAN

Judul Skripsi : Klasifikasi penyakit daun tanaman cabai merah menggunakan model convolutional neural network dengan arsitektur VGG16 dan AlexNet

Nama : Dicky Candid Saragih
NPM : 198160061
Fakultas : Teknik
Prodi : Teknik Informatika

Disetujui Oleh

Komisi Pembimbing


Rizki Muliono S.Kom, M.Kom.

Mengetahui

Dekan Fakultas Teknik

Ketua prodi Teknik Informatika



Dr. Djenar Priatno, S.T., M.T.
NIDN : 0102027402



Rizki Muliono S.Kom M.Kom
NIDN : 0109038902

Tanggal Lulus : 19 Juni 2024

iii

HALAMAN PERNYATAAN

Saya menyatakan bahwa skripsi ini adalah hasil penelitian dan ide saya sendiri, sebagai syarat memperoleh gelar sarjana merupakan hasil karya tulis saya sendiri. Adapun bagian-bagian tertentu dalam penulisan skripsi ini yang saya kutip dari hasil karya orang lain yang telah dituliskan sumbernya secara jelas sesuai dengan norma, kaidah, dan etika penulisan ilmiah. Saya bersedia menerima sanksi akademik yang saya peroleh dan sanksi-sanksi lainnya dengan peraturan yang berlaku, apabila di kemudian hari ditemukan adanya plagiat dalam skripsi ini.

Medan, 19 Juni 2024


METERAN
TELEPON
E82BALX406973353

Dicky Candid Saragih
198160061

**HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI TUGAS
AKHIR/SKRIPSI/TESIS UNTUK KEPENTINGAN AKADEMIS**

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

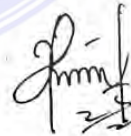
Nama : Dicky Candid Saragih
NPM : 198160061
Program Studi : Teknik Informatika
Fakultas : Fakultas Teknik
Jenis karya : Tugas Akhir/Skripsi

Demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Universitas Medan Area **Hak Bebas Royalti Non-eksklusif (Non-Exclusive Royalty-Free Right)** atas karya ilmiah saya yang berjudul : Klasifikasi penyakit daun tanaman cabai merah menggunakan model convolutional neural network dengan arsitektur VGG16 dan AlexNet.

Dengan Hak Bebas Royalti Non-ekskusif ini Universitas Medan Area berhak menyimpan, mengalih media/format-kan, mengelola dalam bentuk pangkalan data (database), merawat, dan memublikasikan tugas akhir/skripsi/tesis 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: 19 Juni 2024
Yang menyatakan:



Dicky Candid Saragih
198160061

v

ABSTRAK

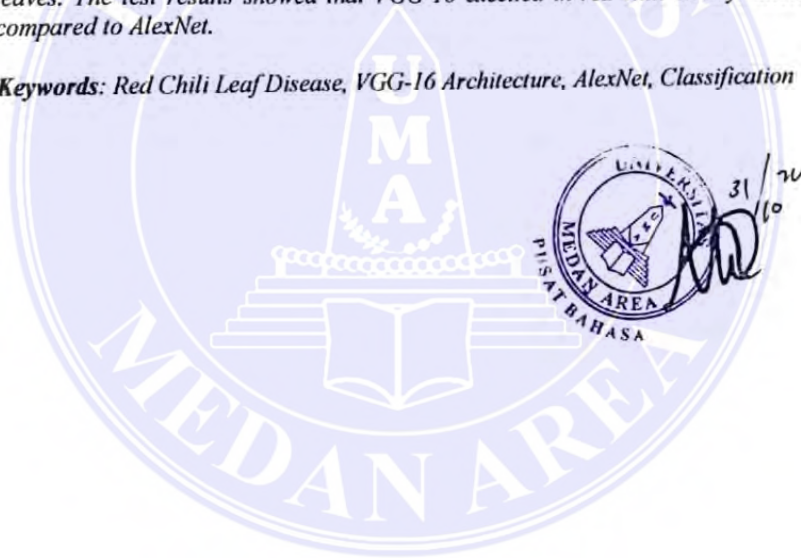
Arsitektur VGG-16 dan Alexnet merupakan model CNN yang terkenal karena keberhasilannya dalam kontes ImageNet Large Scale Visual Recognition Challenge, arsitektur ini menjadi model CNN yang mampu melakukan kontribusi besar terhadap perkembangan jaringan saraf konvolusional dan menjadi salah satu landasan bagi arsitektur-arsitektur lebih canggih yang diperkenalkan kemudian, model ini mampu memberikan kontribusi yang sangat baik dalam pengenalan gambar dengan menggunakan arsitektur yang mendalam. Penelitian ini ingin melakukan uji coba pelatihan dengan menentukan hasil akurasi dengan menggunakan metode evaluasi *Confusion matrix* F1-score dalam mengklasifikasi Penyakit daun cabai merah, data yang digunakan melalui pengambilan gambar langsung dengan pembagian data menjadi 3 yaitu Training, Testing dan Validasi untuk sampel data yang digunakan bercak, kekuningan, keriting, dan kutu kebul dengan pembagian 3600 data sampel 30% training, 30% testing, 40% validasi. Dalam hasil uji coba pelatihan ini, pada arsitektur *VGG-16* berhasil pada pengujian *epoch 10* mencapai tingkat akurasi sebesar 80%, dan pada pengujian *epoch 20* mencapai 82% sementara *AlexNet* mencapai akurasi sebesar 48% pada pengujian *epoch 10* dan pada *epoch 20* mencapai 56%. Evaluasi model menggunakan *confusion matrix* dan F1-score memberikan wawasan lebih lanjut tentang kinerja klasifikasi pada setiap kategori penyakit daun pada cabai merah. Hasil pengujian dalam menentukan akurasi pada ke 2 arsitektur ini, VGG16 unggul dalam klasifikasi cabai merah dibandingkan dengan AlexNet.

Kata kunci: Penyakit Daun cabai merah, arsitektur VGG-16, AlexNet, klasifikasi.

ABSTRACT

The VGG-16 and AlexNet architectures are well-known CNN models due to their success in the ImageNet Large Scale Visual Recognition Challenge. These architectures have made significant contributions to the development of convolutional neural networks and serve as a foundation for more advanced architectures introduced later. These models have shown excellent performance in image recognition using deep architecture. This research aimed to conduct training trials to determine accuracy results using the F1-score confusion matrix evaluation method in classifying diseases on red chili leaves. The data was collected by direct imaging, with data division into three parts: training, testing, and validation. The sample data used included spot, yellowing, curling, and whitefly diseases, with 3600 sample data divided into 30% for training, 30% for testing, and 40% for validation. In the training trial results, the VGG-16 architecture reached an accuracy level of 80% at epoch 10 and 82% at epoch 20, while AlexNet achieved an accuracy of 48% at epoch 10 and 56% at epoch 20. Model evaluation using the confusion matrix and F1-score provided further insights into classification performance for each disease category in red chili leaves. The test results showed that VGG-16 excelled in red chili classification compared to AlexNet.

Keywords: Red Chili Leaf Disease, VGG-16 Architecture, AlexNet, Classification



RIWAYAT HIDUP

Dicky Candid Saragih, lahir di Sipinggian pada tanggal 05 September 2001 dari pasangan orang tua ayah Adenan Saragih dan ibu Sinta Marni Purba. Penulis berkebangsaan Indonesia dan beragama Kristen Protestan, kini penulis beralamat di Kecamatan Dolok Panribuan, Kabupaten Simalungun, Provinsi Sumatera utara. Penulis merupakan anak ke 4 dari 5 bersaudara.

Penulis pertama sekali sekolah di SD N 091461 simpang kawat pada tahun 2007-2013 dan kemudian melanjutkan Sekolah Menengah Pertama di SMP N 1 Dolok Panribuan lulus pada 2016. Tahun 2019 Penulis lulus dari SMA N 1 Dolok Panribuan.

Kemudian penulis melanjutkan studi pada tahun 2019 sebagai mahasiswa Fakultas Teknik Prodi Informatika Universitas Medan Area. Selama mengikuti perkuliahan, penulis aktif untuk berdiskusi belajar dengan teman-teman.

KATA PENGANTAR

Puji syukur penulis panjatkan kepada Tuhan Yang Maha Esa atas berkat dan karunia-Nya. Sehingga penulis dapat menyelesaikan tugas akhir dengan judul “Klasifikasi penyakit daun tanaman cabai merah menggunakan model convolutional neural network dengan arsitektur *VGG16* dan *AlexNet* ini dengan sebaik-baiknya. Laporan tugas akhir/skripsi disusun berdasarkan hasil pengamatan dilapangan.

Penyusunan laporan tugas akhir/ skripsi ini merupakan syarat yang harus di tempuh untuk memenuhi kelulusan dalam menempuh Gelar Sarjana Jenjang Strata (S-1) sesuai dengan kurikulum Jurusan Teknik Informatika Universitas Medan Area yang berlaku pada saat ini.

Penyusunan laporan tugas akhir/skripsi ini tidak akan sesuai tanpa bimbingan, nasihat serta petunjuk dari berbagai pihak. Untuk itu, saya sebagai penulis menyampaikan ucapan terima kasih yang sebesar-besarnya kepada :

1. Bapak Prof. Dr. Dadan Ramdan, M.Eng, M.Sc selaku Rektor Universitas Medan Area.
2. Bapak Dr. Eng., Supriatno, S.T, M.T selaku Dekan Fakultas Teknik Universitas Medan Area.
3. Bapak Rizki Muliono, S.Kom, M.Kom selaku Kepala Program Studi Teknik Informatika.
4. Bapak Dosen Pembimbing yaitu Bapak Rizki Muliono, S.Kom, M.Kom, yang dengan sabar memberikan waktu, pemikiran dan tenaganya untuk memberikan bimbingan, dukungan serta semangat


kepada penulis dalam pengerjaan skripsi ini baik hingga tulisan ini dapat terselesaikan tepat pada waktunya.

5. Seluruh dosen dan staff Universitas Medan Area khususnya dosen prodi Teknik Informatika yang telah membagi ilmunya kepada penulis selama perkuliahan.
6. Keluarga terkasih yang selalu setia mendukung dan mendoakan penulis dengan penuh cinta selama menempuh pendidikan di Universitas Medan Area. Terutama kepada kedua orang tua yang senantiasa berharap dalam doa dan dengan cinta mengajari, mendukung, menyemangati dan memenuhi kebutuhan penulis dalam perkuliahan, Bapak Adenan Saragih yang menjadi panutan dalam hidup penulis dan Ibu Sinta Marni Purba yang dengan penuh kesabaran dan kelembutan menjadi ibu dalam keluarga, kepada abang dan kakak penulis, yang juga turut membantu penulis dalam mendukung dan mendoakan penulis dalam menyelesaikan tulisan ini. Penulis sangat berterimakasih kepadaNya karena menghadirkan mereka di dalam kehidupan penulis, Tuhan senantiasa memberikan keluarga kita.
7. Orang-orang terdekat yang terkasih yang telah banyak mendukung dan menyemangati serta berbagai suka duka bersama penulis selama belajar di Universitas Medan Area.
8. Teman-teman seperjuangan dalam Bimbingan Skripsi \ Bapak Rizki Muliono, S.Kom, M.Kom, yang telah berjuang bersama-sama dalam penyelesaian skripsi ini mulai dari proses awal hingga akhir.

9. Semua pihak yang tidak dapat disebut satu per satu yang juga mendukung, menyemangati dan mendoakan penulis agar menyelesaikan studi di Universitas Medan Area. Kiranya kasih setiaNya selalu menyertai kita dimana pun berada.

Dalam penulisan skripsi ini penulis juga menyadari keterbatasan penulis, oleh karena itu penulis memohon maaf atas segala kekurangan yang terdapat dalam tulisan ini. Penulis juga dengan terbuka mengharapkan kritik, saran dan sumbangan pemikiran. Agar tulisan ini menjadi lebih baik lagi, kiranya tulisan ini bermanfaat bagi pembaca. Tuhan Allah Senantiasa selalu memberkati.

Medan, 19 Juni 2024


Dicky Candid Saragih
198160061

DAFTAR ISI

LEMBAR PENGESAHAN	Error! Bookmark not defined.
HALAMAN PERNYATAAN	iii
ABSTRAK	vi
ABSTRACT	Error! Bookmark not defined.
KATA PENGANTAR	ix
RIWAYAT HIDUP	viii
DAFTAR TABEL	xiv
DAFTAR GAMBAR	xv
BAB I PENDAHULUAN	1
1.1 Latar belakang.....	1
1.2 Rumusan masalah.....	3
1.3 Batasan Masalah	4
1.4 Tujuan Penelitian.....	5
1.5 Manfaat penelitian.....	5
1.6 Sistematika Penulisan.....	6
BAB II LANDASAN TEORI	7
2.1 Klasifikasi Citra	7
2.2 Convolutional Neural Network (CNN)	7
2.3 Jenis hama dan Penyakit daun pada tanaman cabai merah	14
2.4 Arsitektur <i>VGG-16</i>	14
2.5 Arsitektur <i>AlexNet</i>	16
2.6 Google Collab	17
2.7 <i>Confusion Matrix</i>	18
2.8 Penelitian Terdahulu	21
BAB III METODE PENELITIAN	22
3.1 Prosedur Penelitian.....	22
3.2. Teknik pengumpulan data.....	23
3.3 input citra <i>grayscale (Preprocessing)</i>	23

3.4	Perancangan metode CNN.....	24
3.5	Pemodelan arsitektur <i>VGG16</i> dan <i>AlexNet</i>	25
3.6	Arsitektur <i>VGG-16</i>	25
3.7	arsitektur <i>Alexnet</i>	30
3.8	Uji evaluasi <i>Confusion Matrix</i>	35
3.8	Hasil klasifikasi.....	36
3.9	Alat dan bahan Penelitian.....	36
BAB IV HASIL DAN PEMBAHASAN		38
4.1.	Hasil Implementasi.....	38
4.1.1	Data Augmentasi <i>VGG-16 (Preprocessing)</i>	38
4.1.2	Implementasi arsitektur Model <i>VGG-16</i>	39
4.1.3	Hasil akurasi arsitektur <i>VGG-16</i>	39
4.1.4	Hasil evaluasi <i>Confusion Matrix</i> arsitektur <i>VGG-16</i>	41
4.1.5	Data Augmentasi <i>AlexNet (Preprocessing)</i>	44
4.1.6	Implementasi Model <i>Alexnet</i>	45
4.1.7	Hasil akurasi arsitektur <i>Alexnet</i>	46
4.1.8	Hasil evaluasi <i>Confusion Matrix</i> arsitektur <i>Alexnet</i>	49
4.2.	Pembahasan.....	53
4.2.1	Hasil akurasi penelitian Terdahulu.....	54
BAB V.....		55
KESIMPULAN DAN SARAN.....		55
5.1.	Kesimpulan.....	55
5.2.	Saran.....	55
DAFTAR PUSTAKA.....		57
LAMPIRAN.....		60

DAFTAR TABEL

Tabel 2.1 penelitian terdahulu.....	21
Tabel 3.1. Contoh data penelitian.....	23
Tabel 3.2. Tabel jaringan arsitektur <i>VGG-16</i>	26
Tabel 3.3 pengujian data.....	27
Tabel 3.4. Tabel jaringan arsitektur <i>AlexNet</i>	31
Tabel 3.5 Tabel pengujian dataset	31
Tabel 3.6 perhitungan <i>Confusion Matrix presisi,Recal,F1-score</i>	35
Tabel 3.7 Hardware yang digunakan.....	36
Tabel 3.8 Software yang digunakan	37
Tabel 4.1 pengujian arsitektur <i>VGG-16 dan AlexNet</i>	38
Tabel 4.2 Proses augmentasi arsitektur <i>VGG-16</i>	38
Tabel 4.3 Proses perancangan parameter pelatihan <i>VGG16</i>	40
Tabel 4.4 pelatihan ke-1 <i>VGG-16 Confusion matriks F1-Score epoch 10 batch size</i>	44
Tabel 4.5 pelatihan ke-1 <i>VGG-16 Confusion matriks F1-Score epoch 20 batch size 16</i>	44
Tabel 4.6 Hasil augmentasi arsitektur <i>AlexNet</i>	44
Tabel 4.7 Implementasi arsitektur <i>AlexNet</i>	46
Tabel 4.8 Hasil akurasi <i>AlexNet</i> dengan <i>Confusion Matrix F1-score ke-1 epoch 10 batch size 16</i>	52
Tabel 4.9 Hasil akurasi <i>AlexNet</i> dengan <i>Confusion Matrix F1-score percobaan ke-1 epoch 20 bacht size 16</i>	52
Tabel 4.10 Hasil akurasi <i>VGG-16 dan AlexNet</i> menggunakan <i>Confusion Matrix</i>	53
Tabel 4.11 hasil akurasi penelitian terdahulu dengan <i>VGG-16 dan AlexNet</i>	54

DAFTAR GAMBAR

Gambar 2.1. Tahapan pada metode CNN	8
Gambar 2.2. Layer pada <i>convolutional</i>	9
Gambar 2.3. Fungsi <i>ReLU</i>	10
Gambar 2.4. <i>Dropout regulation</i>	13
Gambar 2.5. Arsitektur <i>VGG 16</i>	15
Gambar 2.6. Arsitektur <i>AlexNet</i>	16
Gambar 2.7 Metode <i>Confusion matrix</i>	19
Gambar 3.1 Diagram alur proses penelitian	22
Gambar 3.2 Proses <i>preprocessing</i>	24
Gambar 3.3 Jaringan arsitektur CNN.....	25
Gambar 4.1 Hasil Augmentasi arsitektur <i>VGG-16</i>	39
Gambar 4.2. Diagram garis hasil visualisasi pelatihan <i>VGG-16</i> dengan <i>batch size 16 epoch 10</i>	40
Gambar 4.3. Diagram garis hasil visualisasi pelatihan <i>VGG-16</i> dengan <i>batch size 16 epoch 20</i>	41
Gambar 4.4 Hasil dari pengujian ke-1 akurasi <i>VGG-16</i> dengan <i>Confusion matrix epoch 10 batch size 16</i>	42
Gambar 4.5 Hasil dari pengujian ke-2 akurasi <i>VGG-16</i> dengan <i>Confusion matrix epoch 20 batch size 16</i>	43
Gambar 4.6 Hasil augmentasi arsitektur <i>AlexNet</i>	45
Gambar 4.7 Hasil visualisasi arsitektur <i>AlexNet epoch 10 batch size 16</i>	46
Gambar 4.8 Hasil visualisasi arsitektur <i>AlexNet epoch 20 batch size 16</i>	48
Gambar 4.9 Hasil akurasi <i>Confusion Matrix F1-score arsitektur AlexNet epoch 10 batch size 16</i>	50
Gambar 4.10 Hasil akurasi <i>Confusion Matrix F1-score arsitektur AlexNet epoch 20 batch size 16</i>	51



BAB I

PENDAHULUAN

1.1 Latar belakang

Cabai merah merupakan komoditas penting di Indonesia, namun seringkali terserang penyakit sehingga menimbulkan kerugian ekonomi yang besar. Klasifikasi penyakit cabai merah sangat penting karena memungkinkan identifikasi yang cepat dan akurat sehingga pengobatan dapat dilakukan dengan lebih tepat (Septariani dkk, 2019). Dalam dunia pertanian cabai pencegahan dan pengendalian penyakit perlu didalami untuk mencegah kerugian bagi para petani dengan memahami faktor-faktor penyebaran penyakit. hal ini juga mendukung penelitian dan pengembangan teknologi pertanian dalam meningkatkan ketahanan dan produktivitas tanaman cabai merah. (Marianah, 2020)

Seiring dengan kemajuan era digital saat ini, Perkembangan teknologi informasi telah meningkatkan kebutuhan akan sistem analisis data visual khususnya untuk klasifikasi citra. Pendekatan jaringan saraf *Convolutional neural network (CNN)* unggul karena dapat mengekstrak fitur secara otomatis dan efisien. CNN terdiri dari lapisan konvolusional dan pooling yang mendeteksi fitur lokal dalam suatu gambar, memungkinkan pengenalan pola yang sangat akurat dan kompleks. Keunggulan CNN mencakup akurasi tinggi dan kemampuan untuk belajar dari data dalam jumlah besar, menjadikannya berguna untuk aplikasi seperti pengenalan wajah, pengenalan objek, analisis medis, dan keamanan. Pendekatan CNN mempunyai

potensi untuk merevolusi komputasi visual dan memajukan bidang ilmu computer (Syahputra & Wibowo, 2020).

VGG16 adalah arsitektur *Convolutional neural network* (CNN) yang dikembangkan pada tahun 2014 oleh *Visual Geometry Group* (VGG) di Universitas Oxford. Dikenal karena kesederhanaan dan efektivitasnya, *VGG16* menggunakan lapisan konvolusional kecil (3x3) diikuti dengan lapisan pooling yang terdiri dari 13 lapisan konvolusional dan 3 lapisan fully connected. Arsitektur ini menunjukkan kinerja yang baik dalam kompetisi *ImageNet* dan mampu mengekstrak fitur-fitur utama dari gambar menggunakan prinsip "jaringan yang lebih dalam". Pendekatan modular membuatnya mudah untuk diterapkan dan dipahami, dan memberikan dasar bagi banyak model kompleks lainnya, berguna untuk aplikasi seperti klasifikasi gambar, deteksi objek, dan segmentasi gambar. Pada penelitian sebelumnya yang menggunakan arsitektur *VGG-16* dalam mengklasifikasi daun herbal menunjukkan hasil yang bagus dalam proses autentikasi dengan tingkat akurasi 93,62% untuk testing data secara offline dan 91,04 % untuk testing data secara online (Haryono dkk, 2020)

AlexNet diiluncurkan pada tahun 2012 oleh Alex Krizhevsky, Ilya Sutskever, dan Geoffrey Hinton, *AlexNet* merevolusi bidang visi komputer dengan memenangkan Kompetisi *ImageNet* 2012. Arsitektur ini terdiri dari delapan lapisan yang dapat dilatih, termasuk lima lapisan *convolutional* dan tiga lapisan *Fully conected*. Inovasi utama yang diperkenalkan oleh *AlexNet* mencakup penggunaan lapisan *ReLU* untuk mempercepat pelatihan, teknik *dropout* untuk mengurangi overfitting, dan penggunaan *GPU* untuk meningkatkan kecepatan komputasi. Keberhasilan *AlexNet* membuka jalan bagi pengembangan jaringan saraf

konvolusional yang lebih dalam dan kompleks, yang menjadi dasar berbagai aplikasi dalam visi komputer, termasuk klasifikasi gambar, deteksi objek, dan segmentasi gambar (Hendriyana & Yazid Hilman Maulana, 2020). Pendekatan *AlexNet* yang inovatif dan efisien menjadikan hal ini sebagai tonggak penting dalam pengembangan pembelajaran mendalam. Pada penelitian sebelumnya yang menggunakan model CNN dengan arsitektur *AlexNet* untuk mengklasifikasikan kanker kulit jinak dan ganas menggunakan arsitektur model *AlexNet* Hasil performa dievaluasi dengan data uji menggunakan *confusion matrix*. Dari hasil pelatihan didapatkan akurasi 96,37%. Pada pengujian hasil akurasi pada masing-masing kelas sebesar 90,6%, 62,3%, 94,3%, dan 92%. (Nisa, dkk, 2020).

Penelitian ini dilakukan untuk menguji akurasi algoritma dari arsitektur *VGG-16* dan *Alexnet* dalam mengklasifikasikan penyakit pada cabai merah, pengujian kedua arsitektur dilakukan dengan setiap parameter arsitektur dan layer yang berbeda untuk menentukan model dari kedua arsitektur ini yang lebih baik menentukan akurasi dalam mengklasifikasi penyakit tanaman cabai merah maka penelitian ini Mencoba melakukan analisis klasifikasi untuk membandingkan hasil akurasi terbaik dari kedua arsitektur CNN yaitu *VGG-16* dan *AlexNet* dengan judul “Klasifikasi Citra Penyakit daun pada tanaman cabai merah menggunakan Model *Convolutional Neural Network* dengan Arsitektur *VGG-16* dan *AlexNe*”.

1.2 Rumusan masalah

Permasalahan yang diidentifikasi dalam penelitian ini yaitu bagaimana analisis klasifikasi citra penyakit daun pada Tanaman Cabai merah dalam menggunakan Algoritma *Convolutional Neural Network* dengan 2 arsitektur, *VGG-*

16 dan AlexNet untuk mengklasifikasi hasil akurasi dari ke-2 arsitektur dengan tujuan menentukan tingkat akurasi tertinggi pada citra Penyakit daun cabai merah.

1.3 Batasan Masalah

Beberapa batasan permasalahan dalam peneletian ini adalah:

1. Penelitian ini hanya berfokus untuk klasifikasi jenis penyakit daun pada tanaman cabai merah, seperti kekuningan, keriting daun, bercak dan kutu kebul diluar dari kategori penyakit lain tidak akan diperhitungkan
2. Dataset yang digunakan dalam penelitian ini diambil dari sumber lapangan dengan menggunakan kamera hp sebagai penangkap objek pada permukaan daun yang terkena penyakit untuk kualitas dan ukuran dataset diatur untuk memastikan konsistensi dalam pelatihan dan pengujian model Penelitian ini bertujuan untuk membandingkan performa dari arsitektur CNN, yaitu *VGG-16* dan *AlexNet*. Diluar dari model lain tidak akan dieksplorasi
3. *Preprocessing* data citra akan dibatasi pada *resizing* atau augmentasi data dan akan dibatasi pada teknik teknik standar pada pelatihan CNN
4. Evaluasi model pada pelatihan ini menggunakan *confusion matrix* hanya sampai *F1-score* diluar dari itu tidak akan dianalisis
5. Proses penelitian ini menggunakan model *Deep learning* untuk klasifikasi citra dengan fokus pada implementasi bahasa pemograman tertentu (misalnya *python* dengan *tensorflow*).
6. Penelitian in dibatasa pada periode sumber daya yang tersedia yang dapat mempengaruhi perubahan ukuran dan jumlah pelatihan pada model.

1.4 Tujuan Penelitian

Adapun tujuan dilakukannya penelitian ini yaitu untuk mengklasifikasi citra penyakit pada daun cabai merah dengan menggunakan 2 arsitektur CNN yaitu *VGG-16* dan *AlexNet* dalam menguji seberapa efektif kedua arsitektur ini untuk menentukan akurasi dalam mengklasifikasi kasus pada penyakit tanaman Cabai dan menentukan akurasi terbaik dari kedua arsitektur CNN.

1.5 Manfaat penelitian

1. Bagi peneliti:

- a. Memberikan pengalaman buat penulis dalam melakukan penelitian sebuah karya ilmiah yang melibatkan menambahnya pengetahuan dalam bidang *Deep Learning*
- b. mengimplementasikan teori dan ilmu yang di dapat semasa perkuliahan dalam bidang *Deep learning* dalam mengklasifikasi sebuah gambar

2. Bagi pihak lain:

- a. Menjadi sebuah referensi untuk peneliti selanjutnya yang akan Menambah ilmu pengetahuan mengenai penerapan Algoritma *Convolutional Neural Network* dalam menganalisis klasifikasi citra penyakit daun pada tanaman cabai.
- b. Memberikan dorongan dan kesadaran di kalangan petani pentingnya deteksi dini penyakit tanaman, serta manfaat penggunaan teknologi modern dalam pertanian.

1.6 Sistematika Penulisan

Sistematika penulisan pada penelitian ini ditunjukkan kepada pembaca untuk lebih mudah memahami isi dari penelitian ini. Berikut sistematika penulisan penelitian ini secara garis besar, yaitu:

BAB I PENDAHULUAN

Pada bagian pendahuluan membahas seputar latar belakang dari penelitian ini, dimana untuk mengetahui sebab penelitian ini dilakukan dan selanjutnya membahas rumusan masalah, Batasan masalah, tujuan masalah dan manfaat penelitian.

BAB II TINJAUAN PUSTAKA

Pada tinjauan pustaka membahas hal – hal yang mendasar berisi teori – teori yang berkaitan dengan Metode CNN dengan 2 arsitektur VGG-16 dan AlexNet dalam mengklasifikasi penyakit pada tanaman cabai merah

BAB III METODE PENELITIAN

Pada bagian ini membahas tentang sumber data dan tahapan yang dilakukan dalam melakukan penelitian ini serta gambaran umum sistem yang akan dikerjakan.

BAB IV :PEMBAHASAN DAN HASIL

Pada bagian ini dilakukan pembahasan dan menampilkan hasil dari penelitian yang telah dilakukan.

BAB V KESIMPULAN DAN SARAN

Pada bagian ini peneliti menarik kesimpulan dari hasil penelitian yang telah dilakukan dan memberikan beberapa saran – saran yang nantinya, diharapkan penelitian selanjutnya dapat dikembangkan.

BAB II

LANDASAN TEORI

2.1 Klasifikasi Citra

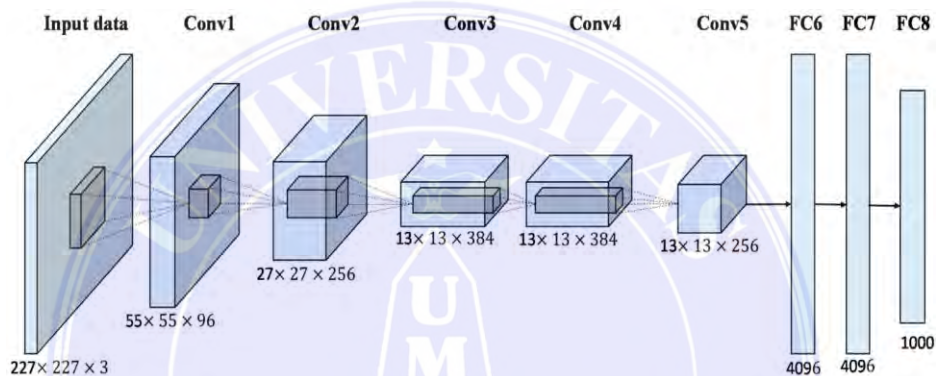
Klasifikasi citra adalah proses pengelompokan citra digital ke dalam kategori atau kelas tertentu berdasarkan informasi yang terkandung dalam citra tersebut. Ini merupakan bagian penting dari pengolahan citra dan computer vision, dengan aplikasi luas di berbagai bidang, termasuk pertanian, kesehatan, keamanan, dan pengenalan objek. Objek yang akan digunakan dalam proses klasifikasi menggunakan banyak (Septariani dkk., 2019)

Klasifikasi gambar (citra) adalah cara alternatif dalam pemecahan suatu masalah klasifikasi pada gambar. Tujuan mengklasifikasikan gambar adalah untuk menduplikasikan atau meniru gambar kemampuan manusia untuk memahami informasi citra digital dengan cara yang dapat dilakukan oleh komputer dengan Mengkategorikan objek berupa gambar mirip manusia. permasalahan yang ditemukan di Klasifikasi citra gambar adalah proses desain fitur yang cukup terbatas pada kumpulan data tertentu, Ini karena setiap gambar memiliki perspektif dan skala yang berbeda-beda (Ratna, 2020)

2.2 *Convolutional Neural Network (CNN)*

Jaringan saraf convolutional adalah jenis algoritma terawasi yang bekerja dengan menerima masukan berupa gambar (citra). CNN sangat baik dalam mengolah input berupa gambar. CNN adalah metode pembelajaran mesin yang dikembangkan oleh Multi Layer Perceptron dalam memproses dan memproses data 2D,CNN dapat

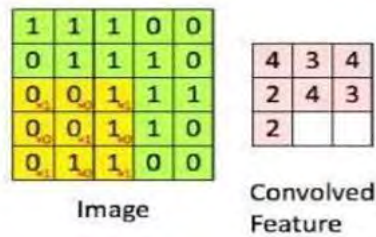
mempelajari fitur gambar yang kompleks. Struktur CNN terutama terdiri dari lapisan input, lapisan konvolusi, pooling, lapisan penghubung penuh, dan lapisan keluaran, Fungsi pooling layer yaitu mengurangi dimensi peta fitur dan menjaga skala invariant. Algoritma back propagation digunakan untuk mengatur parameter bobot jaringan syaraf. Parameter optimasi utama adalah parameter kernel convolutional, bobot lapisan penyatuan, bobot lapisan yang terhubung penuh dan parameter bias (Nugroho dkk., 2020)



Gambar 2.1. Tahapan pada metode CNN (Nugroho dkk., 2020)

2.3.1 Convolutional Layer

Lapisan convolution terdiri dari neuron yang disusun dengan membentuk filter pada pixel yang panjang dan tinggi. Layer ini melakukan operasi konvolusi pada output dari layer sebelumnya Tujuan dari layer ini adalah untuk mengekstrak fitur dari citra pada gambar dan secara linier mengubah data input berdasarkan informasi dalam data. Filter ini akan bergeser dari kiri atas ke kanan bawah. Outputan dari convolution layer ini berupa feature map (Hendriyana & Yazid Hilman Maulana, 2020)



Gambar 2.2. Layer pada *convolutional*
 Sumber: (Hendriyana & Yazid Hilman Maulana, 2020)

$$O = \frac{I - F + 2P}{S} + 1 \quad (2.1)$$

Keterangan:

O : Ukuran output (panjang atau lebar peta fitur setelah konvolusi)

I : Ukuran input (panjang atau lebar peta fitur atau gambar input)

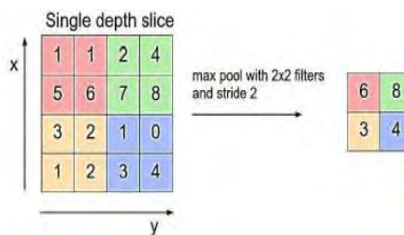
F : Ukuran filter (panjang atau lebar kernel konvolusi)

P : Padding (jumlah piksel yang ditambahkan di tepi gambar untuk menjaga dimensi)

S : Stride (langkah pergeseran kernel konvolusi)

2.3.2 Fungsi Aktivasi *ReLU* (Rectified Linear Unit)

Fungsi aktivasi ReLU dilakukan setelah melakukan proses convolution layer. Nilai pada feature map akan di lakukan seleksi dimana nilai negative akan dijadikan 0 (Fadlia & Kosasih, 2019)



Gambar 2.3. Fungsi ReLU
Sumber: (Fadlia & Kosasih, 2019)

$$f(x) = \max(0, x) \quad (2.2)$$

$$f(X) = \begin{cases} x, & \text{jika } x > 0 \\ 0, & \text{jika } x \leq 0 \end{cases} \quad (2.3)$$

Keterangan :

$f(x)$: Ini adalah output dari fungsi aktivasi. Fungsi aktivasi digunakan untuk memperkenalkan non-linearitas dalam jaringan saraf. Non-linearitas ini memungkinkan jaringan untuk belajar berbagai pola yang kompleks.

x : Input yang diterima oleh fungsi aktivasi, biasanya merupakan hasil dari perkalian bobot dan input, serta penambahan bias, yaitu $W \times I + b$

$\max(0, x)$: Fungsi ini mengambil nilai maksimum antara 0 dan x , artinya:

Jika $x > 0$, maka output $f(x) = x$ (nilai x dipertahankan).

Jika $x \leq 0$, maka output $f(x) = 0$ (nilai x dipotong menjadi 0).

$x > 0$: Jika input x lebih besar dari 0, fungsi ReLU mengeluarkan output yang sama dengan input x . Dalam kasus ini, tidak ada modifikasi terhadap input.

$x \leq 0$: Jika input x lebih kecil dari atau sama dengan 0, maka output dari fungsi ReLU adalah 0. Ini berarti bahwa input negatif atau nol di-nonaktifkan, memberikan keluaran nol.

2.3.3 Pooling layer

Pooling layer terdiri dari filter dengan ukuran dan grade tertentu dan filter bergerak melalui area peta. pooling yang umum digunakan adalah maximum pooling dan average pooling. Tujuannya untuk memperkecil ukuran peta objek peta

(downsampling).sehingga mempercepat perhitungan karena lebih sedikit parameter yang diperbarui dan over-fitting(Rasywir dkk., 2020)

$$O = \frac{I-F}{S} + 1 \quad (2.4)$$

Keterangan :

O: Ukuran output dari layer konvolusi, biasanya dalam bentuk panjang atau lebar (height/width) dari feature map hasil setelah filter diterapkan. Nilai ini menunjukkan berapa banyak neuron atau piksel yang ada di layer hasil (output).

I: Ukuran dari input, yang biasanya adalah dimensi panjang atau lebar dari gambar input atau layer sebelumnya. Jika input adalah gambar, ini akan merujuk pada ukuran height atau width dari gambar tersebut.

F: Ukuran filter atau kernel yang digunakan dalam proses konvolusi. Filter ini adalah matriks kecil yang "digeser" melintasi gambar untuk mengekstrak fitur. Contoh filter berukuran $3 \times 3 \times 3$ berarti $F = 3$

S: Langkah perpindahan filter saat melakukan konvolusi. Stride menentukan seberapa jauh filter akan bergerak setiap kali dalam proses konvolusi. Jika stride bernilai 1, filter akan bergeser satu piksel setiap kali. Jika stride bernilai 2, filter akan melompati satu piksel setiap kali.

+1: Bagian tambahan dalam rumus ini digunakan untuk memperhitungkan elemen output pertama dari konvolusi, karena meskipun filter di-shift, output harus dimulai dari indeks 1, bukan 0.

2.3.4 Fully connected layer

Peta fitur yang dihasilkan dari ekstraksi fitur masih berupa array multidimensi, sehingga merupakan peta fitur harus diubah menjadi vektor untuk digunakan

sebagai input dari fully connected layer. ada node yang saling terhubung antara node depan dan node belakang atau bisa disebut juga forward propogation dan back propogation . Node ini bertindak sebagai pengklasifikasi objek pada gambar Item tersebut didasarkan pada probabilitas yang bergantung kepada setiap epocnya (Syahputra & Wibowo, 2020)

$$O=W \times I+b \quad (2.5)$$

Keterangan :

O : Output dari fully connected layer (hasil prediksi atau peta fitur yang diproses).

W : (**Bobot / Weights**): Matriks atau vektor yang berisi bobot dari setiap input ke neuron. Bobot menentukan seberapa besar pengaruh input tertentu terhadap output neuron.

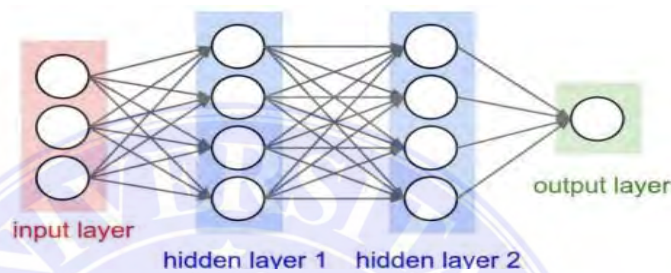
M adalah jumlah neuron di lapisan saat ini dan N adalah jumlah neuron di lapisan sebelumnya.

I (Input): Vektor input yang diterima oleh neuron, bisa berupa data asli (misalnya gambar, teks) atau output dari layer sebelumnya.

b : Bias, ilai tambahan yang tidak bergantung pada input dan ditambahkan ke hasil dari $W \times I$.

2.3.5 Dropout Regularization

Dropout adalah teknik regularisasi jaringan saraf yang secara acak memilih beberapa sejumlah neuron dan tidak menggunakan neuron ini selama pelatihan. dropout berfungsi untuk sebagai pencegahan terjadinya overfitting dan mempermudah prose pengerjaan citra(Charli dkk., 2020)



Gambar 2.4. Dropout regulation
Sumber: (Charli dkk., 2020)

$$y_{dropout} = \begin{cases} x, & \text{dengan probabilitas } 1 - p \\ 0, & \text{dengan probabilitas } p \end{cases} \quad (2.6)$$

Keterangan:

x : adalah input atau output dari neuron sebelum dropout diterapkan.

p : adalah probabilitas dropout, yaitu probabilitas neuron tersebut di-nonaktifkan (diatur menjadi nol).

$(1 - p)$: adalah probabilitas neuron tetap aktif dan meneruskan outputnya.

Penjelasan:

Dengan probabilitas $(1 - p)$, neuron akan tetap aktif dan meneruskan outputnya

x

Dengan probabilitas p_1 , neuron akan di-nonaktifkan dan outputnya diatur menjadi 0. Rumus ini adalah representasi formal bagaimana dropout bekerja dalam sebuah layer jaringan saraf tiruan (neural network).

2.3 Jenis hama dan Penyakit daun pada tanaman cabai merah

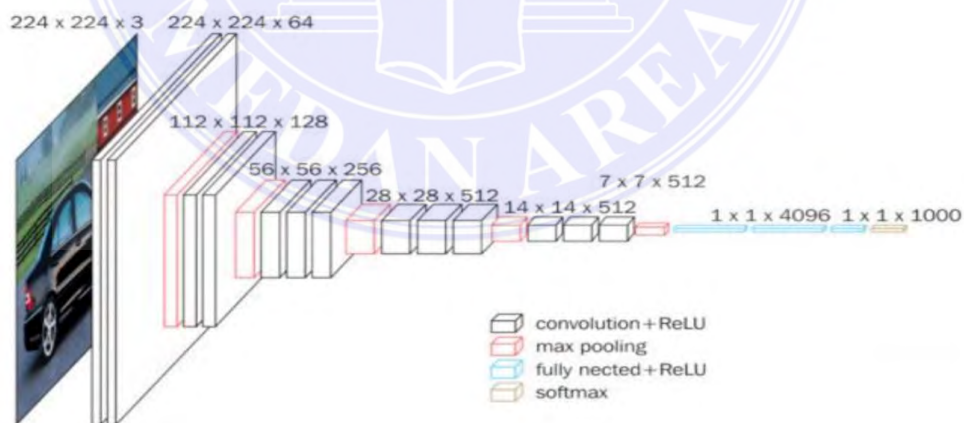
Penyakit yang pada daun Tanaman cabai merah dapat disebabkan dari beberapa faktor yang umumnya terjadi karena adanya serangan dari virus, serangan hama dan perubahan suhu atau curah hujan yang berlebih menyebabkan tanaman cabai merah rusak, adapun tanda yang menunjukkan tanaman cabai mengalami penyakit yaitu pada bentuk daun yang mulai mengering/mengulung kedalam seperti, timbul bercak keemasan dan keriting dengan corak gelap pada lengkungan daun yang keriting serta dengan serangan hama seperti kutu putih menunjukkan gejala permukaan daun yang memutih akibat serbuk yang disebabkan hama kutu putih. sehingga serangan ini mempengaruhi kualitas serta tanaman cabai, dan menyebabkan kerugian buat para petani jika tidak ditindak lanjuti secepatnya (Marianah, 2020)

Pada penelitian kali ini peneliti berfokus untuk meneliti sebuah objek 2D pada penyakit daun pada tanaman cabai merah untuk analisis klasifikasi citra gambar penyakit daun pada tanaman cabai merah menggunakan model Convolutional Neural Network dengan arsitektur VGG-16 dan Alexnet.

2.4 Arsitektur VGG-16

Salah satu arsitektur CNN dalam proses penelitian ini adalah arsitektur VGG-16 dan LeNet. Arsitektur VGG-16 yang menggunakan CNN yang memiliki

kelebihan dalam melakukan proses fitur yang efisien dalam memproses citra cenderung lebih banyak *Convolutional* layer yang di tumpukan pada filter yang lebih sedikit *VGG-16* adalah arsitektur CNN dengan convolutional layer dengan 5 *convolutional block* layer dengan maxpooling layer sebagai fitur ekstraksi dan fully connected layer, *bump normalization*, *drop* dan *density layer* sebagai classifier arsitektur *VGG-16* merupakan salah satu jenis dari arsitektur CNN yang baik dalam mengklasifikasi citra gambar dengan akurasi yang terbaik yang memiliki hidden layer berjumlah 16 buah Pada *VGG-16*, proses konvolusi dilakukan dengan menggunakan filter dengan jumlah 3x3, stride sejumlah 1 dan max pooling sejumlah 2x2. Input citra yang digunakan pada *VGG-16* yaitu berukuran 224x224x3 dan akan melalui reduksi pada blok awal yang menghasilkan 224x224x64 hingga blok akhir sejumlah 7x7x512. Keunggulan model ini menampilkan arsitektur yang sangat homogen yang hanya melakukan convolutional 3x3 dan pooling 2x2 dari awal hingga akhir (Sandhopi dkk., 2020).



Gambar 2.5. Arsitektur VGG 16
Sumber: (Sandhopi dkk., 2020)

2.5 Arsitektur AlexNet

Arsitektur AlexNet dengan besar margin yang cukup besar. Jaringan ini terbukti, untuk pertama kalinya bahwa fitur yang diperoleh dengan *deep learning* dapat melampaui fitur desain secara manual, memecahkan paradigma sebelumnya dalam visi komputer. Secara garis besar, cara kerja sistem arsitektur Alexnet dibagi menjadi dua kelompok layer. Pertama adalah layer ekstraksi fitur yang tersusun dari layer *konvolusi dan pooling layer*, dan kedua adalah *classification layer* (Herwanto et al., 2020). Arsitektur AlexNet terdiri dari 5 lapisan convolution, 3 Lapisan pooling, 2 lapisan dropouts dan 3 fully conected. Konvolusi adalah istilah matematika yang berarti berulang kali menggunakan fungsi ke output fungsi lain dan membuat sehingga menghasilkan feature mapping. fungsi Konvolusi itu sendiri berguna untuk mengekstrak fitur citra pada gambar. Pooling atau subsampling cara untuk mengurangi ukuran matriks. terdapat 2 kombinasi berbeda sering digunakan yaitu avarage *pooling* dan max pooling. sedangkan dropout adalah metode untuk mengurangi *overfitting* pada neural network.cara Ini dilakukan dengan mencegah overfitting data pada pelatihan. *fully conected layer* adalah kumpulan proses pada *konvolusi* (Ridha dkk., 2019)



Gambar 2.6. Arsitektur AlexNet

Sumber: (Ridha dkk., 2019)

2.5.1 Struktur Arsitektur AlexNet

Struktur Layer Tumpang Tindih: *AlexNet* terdiri dari 5 lapisan konvolusi (convolutional layers) dan 3 lapisan *fully connected* (FC layers), yang membentuk total 8 lapisan utama. Dalam struktur ini, konvolusi dan pengecilan resolusi (subsampling) dilakukan secara bergantian untuk mengambil fitur-fitur berbeda dalam citra (D.K. et al., 2020).

ReLU Activation dan Normalisasi Batch: *AlexNet* menggunakan fungsi aktivasi *ReLU* (Rectified Linear Unit) untuk membantu dan menghindari masalah gradien yang menghilang. Selain itu, lapisan normalisasi batch digunakan untuk meminimalisir overfitting dan memudahkan proses konvergensi (Sabilla, 2020).

Pengecilan Resolusi Melalui Max-Pooling: Lapisan *max-pooling* digunakan untuk mereduksi resolusi spasial dari fitur-fitur yang dihasilkan oleh konvolusi. Ini membantu dalam mengurangi jumlah parameter dan waktu komputasi. (Lubis dkk., 2019)

Data Augmentation: yaitu berfungsi mengatasi masalah overfitting, *AlexNet* menggunakan dan memanfaatkan data *augmentation* dengan mengubah sedikit citra latih seperti rotasi, pergeseran, dan pemutaran citra untuk menghasilkan akurasi yang baik (Alamsyah & Pratama, 2020).

2.6 Google Collab

google collab atau "*google Colaboratory*" yaitu salah satu produk *Google Research* yang mengizinkan siapa saja membuat tulisan dan menjalankan kode dengan bahasa pemrograman *Python* apa pun dari browser dan bisa digunakan

untuk machine learning, dan menganalisis data, Google collab berfungsi Colab untuk memberikan layanan teknologi Jupyter yang dihosting yang dapat digunakan tanpa penginstalan dan menawarkan akses gratis ke *resource* komputasi, termasuk *GPU*, *Colab Resources* tidak dijamin dan dibatasi oleh sifatnya, dan batas penggunaannya bervariasi dari waktu ke waktu. Hal ini diperlukan agar Colab dapat menyediakan resource secara gratis. Pengguna yang menginginkan akses yang lebih andal ke resource menjadi lebih baik menggunakan *Colaboratory Pro*(Sengkey dkk., 2020). Peluncuran Colab Pro adalah langkah pertama yang diambil Google dalam melayani pengguna yang ingin mengakses lebih banyak fungsi *google collab* bertujuan untuk proses jangka panjang Google terus menawarkan Colab versi gratis. Sementara pihak perusahaan terus mengembangkan fitur-fitur terbaru untuk memenuhi segala kebutuhan para pengguna(Guan dkk., 2019).

2.7 *Confusion Matrix*

Evaluasi data *Confusion matrix* adalah proses yang digunakan untuk mengukur atau menghitung nilai akurasi pada image citra. Tabel pada proses evaluasi confusion matrix terdiri dari baris data yang akan uji diprediksi true (benar) dan true (tidak benar) suatu data yang di proses pada saat klasifikasi (Imron, 2019). Seperti pada contoh gambar di bawah:

	<i>Predicted Class</i>	
<i>True Class</i>	<i>Positive</i>	<i>Negative</i>
<i>Positive</i>	TP	FN

Gambar 2.7 Metode *confusion matrix*

Sumber : (Imron, 2019)

Keterangan :

TP (True Positive) : Memprediksi nilai positif dan hasilnya benar.

TN (True Negative) : Memprediksi nilai negative dan hasilnya benar.

FP (False Positive) : Memprediksi nilai positif dan hasilnya salah.

FN (False Negative) : Memprediksi nilai negatif dan hasilnya Salah.

1. Akurasi

Akurasi mempresentasikan seberapa akurat sistem dapat mengklasifikasi data secara benar. Nilai akurasi adalah perbandingan antara data yang terklasifikasi benar dengan jumlah keseluruhan data (AGUSTINA et al., 2022). Nilai akurasi dapat diperoleh dengan Persamaan berikut.

$$Accuracy = \frac{TN+TP}{TN+FP+TP+FN} \quad (2.7)$$

2. Presisi

Presisi adalah perbandingan nilai dari jumlah data kategori terklasifikasi benar dengan total keseluruhan data kategori terklasifikasi benar (AGUSTINA dkk., 2022). Untuk mencari nilai presisi dapat digunakan Persamaan berikut

$$Presicion = \frac{TP}{TP+FP} \quad (2.8)$$

3. Recal

Recall dilakukan untuk mengetahui perbandingan jumlah data kategori terklasifikasi benar oleh sistem dengan jumlah data kategori terklasifikasi benar dan salah (AGUSTINA dkk., 2022). Persamaan berikut dapat digunakan untuk memperoleh nilai recall.

$$Recall = \frac{TP}{TP+FN} \quad (2.9)$$

4. F1-score

F1-Score merupakan penggabungan dari presisi dan recal (AGUSTINA dkk., 2022)l. Nilai f1-score dapat diperoleh dengan Persamaan berikut.

$$F1 = \frac{2*Presicion*Recall}{Presicion+Recall} \quad (2.10)$$

2.8 Penelitian Terdahulu

Tabel 2.1 penelitian terdahulu

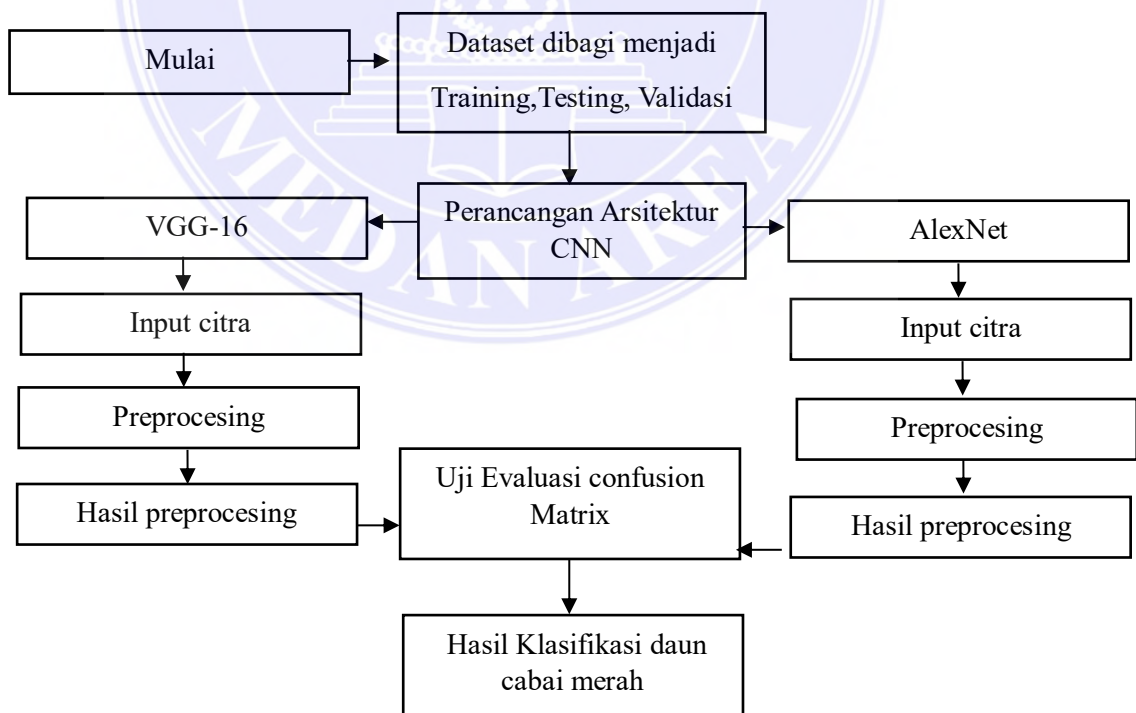
No	Judul	Metode	Keterangan
1.	(Dwi Fitriana Sari & Swanjaya, 2020) Implementasi Convolutional Neural Network Untuk Identifikasi Penyakit Daun Gambas Dwi	VGG-16	identifikasi 3 jenis penyakit pada tanaman gambas yaitu Embun Bulu, Kumbang Daun, dan Ulat Daun memiliki akurasi terbaik pada epoch 25 dan learning rate 0,001 dengan akurasi training senilai 92% dan akurasi cross-validation 91,1% dan akurasi testing senilai 90%. Kata
2.	(Jadhav et al., 2019) Convolutional neural networks for leaf image-based plant disease classification Sachin	AlexNet	Hasil ini didapatkan dalam percobaan: (1) Model AlexNet memiliki akurasi 90.2%; (2) Model ResNet memiliki akurasi 97.3%; dan
3.	(Sriyati dkk., 2020) Literature Review: Pengenalan Wajah Menggunakan Algoritma Convolutional Neural Network	VGG16	Berdasarkan hasil pengujian, model yang dihasilkan mencapai akurasi tertingginya yaitu 85.96% pada saat menggunakan epoch 50 dan skenario data 90:10, sedangkan akurasi terendah sebesar 77.19% dicapai ketika menggunakan epoch 17 dan skenario data 80:20.
4.	(Ilham & Rochmawati, 2020) Transliterasi Aksara Jawa Tulisan Tangan ke Tulisan Latin Menggunakan CNN	AlexNet,	nilai recall dan precision yang dihasilkan dari pengujian ini adalah 72.2% dengan rata-rata untuk waktu pencarian adalah 0.8089 seconds.
5.	(Akbar dkk., 2020) Perancangan Program Pengenalan Isyarat Tangan Dengan Metode Convolutional Neural Network (Cnn)	AlexNet	Nilai akurasi validasi arsitektur AlexNet besar, sedang, dan kecil terhadap performa terbaik adalah masing-masing 96,25%, 95,6%, dan 95,6%.
6.	(Dewi dkk., 2019) Klasifikasi diabetes retinopati menggunakan metode Convolutional neural network	VGG-16	Nilai akurasi tertinggi diperoleh dari sistem yang menggunakan perbandingan 3:2 data latih dan data uji, yaitu 81,25% dengan waktu komputasi 14,518 detik. 4.

BAB III

METODE PENELITIAN

3.1 Prosedur Penelitian

Pada Gambar 3.1 merupakan diagram alur proses klasifikasi penyakit daun pada cabai merah yang meliputi dengan beberapa prosedur yang diawali dengan, persiapan dataset yang akan diuji kemudian data akan dibagi menjadi 3 data dengan data training, testing, dan validasi selanjutnya data akan dilakukan preproccesing setelah melalui preproccesing data akan diolah dengan menggunakan arsitektur CNN dengan mengawali rancangan *VGG-16* kemudian *AlexNet* dengan menentukan akurasi menggunakan Confusion Matrix selanjutnya hasil dari pengujian akurasi pada ke 2 arsitektur di analisa untuk menentukan akurasi terbaik.







Gambar 3.1 Diagram alur proses penelitian

3.2. Teknik pengumpulan data

Data pada penelitian ini menggunakan data gambar yang di ambil langsung melalui kamera handphone dengan format data gambar berbentuk PNG dengan bentuk citra daun cabai yang terkena penyakit seperti: keriting daun, bercak daun, kutu kebul, dan kekuningan, Proses dalam penelitian ini menggunakan 3 proses yaitu proses training testing, dan validasi dalam proses training, dari data yang akan di analisis adapun model uji coba data yang digunakan sebanyak 3600 data gambar. Model proses evaluasi digunakan untuk penelitian yaitu dengan membagi data menjadi 30% data dijadikan sebagai data training dan untuk 30% data dijadikan testing kemudian 40% untuk dijadikan data validasi. Fungsi dari model evaluasi ini digunakan untuk menguji fungsi dari alrsitektur *VGG-16* dan *AlexNet* dengan menggunakan hasil evaluasi *Confusion matrix* presisi *F1-score* pada penelitian.

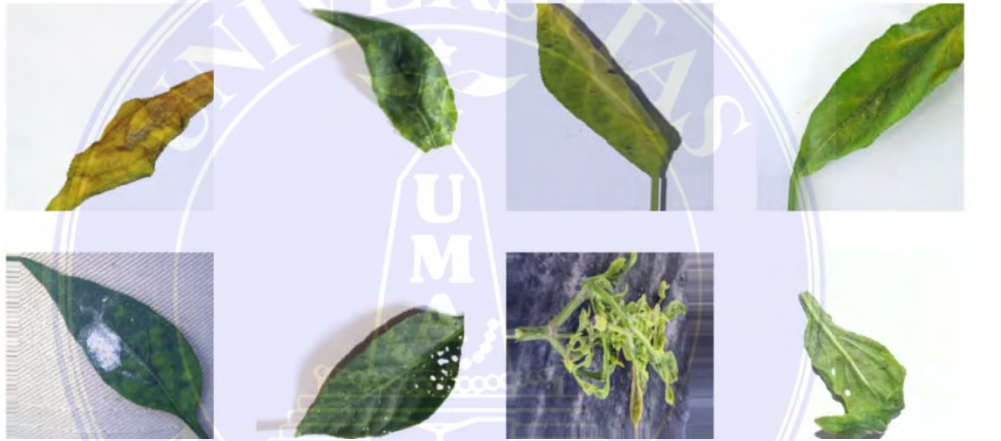
Tabel 3.1. Contoh data penelitian

Jenis Penyakit pada tanaman cabai Merah			
Bercak	Kekuningan	Keritig Daun	Kutu Kebul
			

3.3 input citra grayscale (*Preprocessing*)

Pada model pelatihan penelitian ini menggunakan proses preprocessing proses dimana data akan di seleksi melalui beberapa tahap sebelum di proses mulai dari

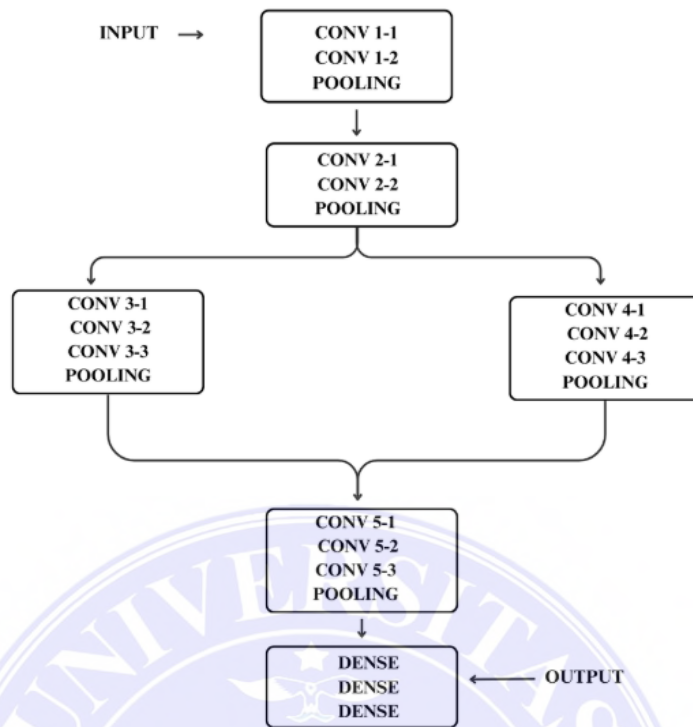
tahap, pembersihan data (*Cleaning*) data yang sudah ada akan di seleksi kembali untuk mencengah kesalah pahaman data yang akan di proses penghapusan data yang tidak lengkap dan tidak relevan untuk mengatasi ke tidak salah pahaman data yang akan di proses ketika pengujian citra. Setelah data dalam tahap pembersihan dilanjutkan dengan tahapan data yang akan diolah akan disatukan dengan format yang sama tujuannya agar mempermudah proses analisis tahap terakhir mengurangi jumlah data dengan cara pengurangan dimensi, pengurangan jumlah, dan data kompresi data agar mempermudah proses pada tahap akhir.



Gambar 3.2 Proses preprocessing

3.4 Perancangan metode CNN

Perancangan metode CNN pada penelitian ini dilakukan mulai dari tahapan seperti, menyusun proses pelatihan data yang akan digunakan dimulai dari menghitung jumlah lapisan yang digunakan, menentukan filter, terakhir menetapkan fungsi aktivasi dan menentukan ukuran *pooling*. Sebagai contoh ilustrasi jaringan CNN



Gambar 3.3 Jaringan arsitektur CNN

3.5 Pemodelan arsitektur VGG16 dan AlexNet

Pemodelan arsitektur pada penelitian menggunakan dua jenis arsitektur CNN yaitu *VGG 16* dan *AlexNet* dengan membandingkan akurasi tertinggi antara kedua arsitektur CNN dalam melakukan klasifikasi dalam menentukan nilai akurasi evaluasi menggunakan Confusion matrix Presisi *F-1 Score*.

3.6 Arsitektur VGG-16

Metode deep learning CNN dengan tahapan proses pemodelan arsitektur *VGG-16* melakukan penginputan citra dengan menghitung akurasi yang akan dihasilkan. Pada penelitian ini berfokus untuk memiliki lapisan konvolusi filter 3x3 dengan langkah 1 dan menggunakan *padding* yang sama dan lapisan *maxpool filter* 2x2 langkah 2 seperti aturan pada arsitektur *VGG-16* pada tahapanya penelitian ini menerapkan mode *sequential* agar semua lapisan yang akan disusun secara berurutan. Sedikit modifikasi dalam menentukan akurasi terbaik diperlukan saat

menggunakan metode ini untuk mencapai akurasi yang baik, dengan mengubah parameter seperti batch size dan epoch yang akan digunakan sehingga menghasilkan akurasi terbaik yang di hasilkan *VGG16*, Adapun tahapan contoh proses pada image data generator data training dan testing fungsi dari image generator untuk mengimpor data dengan label dengan mudah ke dalam model setelah itu proses penelitian ini menggunakan aktivasi *RELU* (Rectified Linear Unit) pada setiap lapisan, sehingmengurangi nilai negatif agar tidak merambat pada lapisan selanjutnya. Setelah itu baru bisa dilakukan penentuan akurasi menggunakan *Evaluasi Confusion* untuk menentukan nilai akurasi *presisi F1* dalam klasifikasi Citra Penyakit Tanaman Cabai dengan *VGG16* Contoh tabel jaringan arsitektur *VGG-16*:

Tabel 3.2. Tabel jaringan arsitektur VGG-16

VGG-16						
	Layer	Feature map	Size	Karnel size	Stride	Activatio n
input	Image	1	224 x 224 x 3	-	-	-
1	2 x convolutional	64	224 x 224 x 64	3 x 3	1	ReLU
	Max Pooling	64	112 x 112x 64	3 x 3	2	ReLU
3	2 x convolutional	128	112 x 112 x 128	3 x 3	1	ReLU
	Max Pooling	128	56 x 56 x 128	3 x 3	2	ReLU
5	2 x convolutional	256	56 x 56 x 256	3 x 3	1	ReLU
	Max Pooling	256	28 x 28 x 256	3 x 3	2	ReLU
7	2 x convolutional	512	28 x 28 x 512	3 x 3	1	ReLU
	Max Pooling	512	14 x 14 x 512	3 x 3	2	ReLU
10	2 x convolutional	512	14 x 14 x 512	3 x 3	1	ReLU
	Max Pooling	512	7 x 7 x 512	3 x 3	2	ReLU
13	Fully connected	-	25088	-		ReLU

	Fully connected	-	4096	-		ReLU
	Fully connected	-	4096	-		ReLU
	Fully connected	-	1000	-		Softmax

3.6.1 Rumus perhitungan pada VGG-16

Dikondisikan dengan studi kasus pembagian data 30% data training dan 30 data testing kemudian 40% data validasi dari pembagian 3600 citra data

Tabel 3.3 pengujian data

kelas	Pembagian data	Jumlah citra gambar
Bercak,Kekuningan Keriting,Kutu kebul	<i>Training =30%</i>	1.080 gambar
Bercak,Kekuningan Keriting,Kutu kebul	<i>Testing =30%</i>	1.080 gambar
Bercak,Kekuningan Keriting,Kutu kebul	<i>Validasi=40%</i>	1.440 gambar

1. Blok 1:

2 lapisan pada konvolusi menggunakan filter 3x3, 64 filter, *stride 1, padding* "same".

Rumus ukuran output *konvolusi*:

$$O = \frac{I - F + 2P}{s} + 1$$

$$I = 224, F = 3, P = 1, \text{ dan } s = 1:$$

$$O = \frac{224 - 3 + 2(1)}{1} + 1 = 224$$

Max Pooling 2x2 dengan stride 2. Selanjutnya dilakukanganya *max pooling*, pada ukuran output:

$$O = \frac{224-2}{1} + 1 = 112$$

Maka hasil outputnya menjadi = 112x112x64.

2. Blok 2:

2 Lapisan Konvolusi dengan filter 3x3, 128 filter, stride 1, padding "same".

Ukuran input sekarang 112x112, jadi setelah dua lapisan konvolusi, ukuran output tetap 112x112, dengan 128 filter, sehingga dimensinya menjadi 112x112x128.

Max Pooling 2x2 dengan stride 2.

Setelah *max pooling*:

$$O = \frac{112-2}{2} + 1 = 56$$

Maka output = 56x56x128.

3. Blok 3:

3 Lapisan Konvolusi dengan filter 3x3, 256 filter, stride 1, padding "same". tiga lapisan konvolusi, ukuran *output* tetap 56x56, digunakan 256 filter, sehingga dimensinya menjadi 56x56x256.

Max Pooling 2x2 dengan stride 2:

$$O = \frac{56-2}{2} + 1 = 28$$

Maka outputnya = 28x28x256

4. Blok 4:

3 Lapisan Konvolusi dengan filter 3x3, 512 filter, stride 1, padding "same".

Setelah tiga lapisan konvolusi, ukuran *output* tetap 28x28, dengan 512 filter, sehingga dimensinya menjadi 28x28x512.

Max Pooling 2x2 dengan stride 2:

$$O = \frac{28-2}{2} + 1 = 14$$

Maka outputnya = 14x14x512

5. Blok 5:

3 Lapisan Konvolusi dengan filter 3x3, 512 filter, stride 1, padding "same".

Setelah tiga lapisan konvolusi, ukuran output tetap 14x14, dengan 512 filter, sehingga dimensinya menjadi 14x14x512.

Max Pooling 2x2 dengan stride 2:

$$O = \frac{14-2}{2} + 1 = 7$$

Maka outputnya = 7x7x512

6. Lapisan Fully Connected:

Misalkan input I dari layer sebelumnya adalah vektor yang dihasilkan setelah proses flattening dari lapisan konvolusi terakhir. Ukuran input ini adalah $7 \times 7 \times 512$ (setelah pooling), yang jika di-flatten menjadi vektor berdimensi 25088 (*karena* $7 \times 512 = 25088$).

Input (I): Vektor dengan ukuran 25088

Bobot (W): Matriks dengan ukuran 4096×25088 . Setiap neuron di *fully connected* layer terhubung dengan semua nilai di input *vektor*, sehingga membutuhkan bobot sebanyak 4096×25088 .

Bias (b): Vektor bias dengan ukuran 4096 (satu bias untuk setiap neuron di FC1).

Perhitungan:

1. Ukuran Input: I adalah vektor dengan panjang 25088 (hasil *flatten* dari layer sebelumnya).

2. Perkalian Bobot dan Input: Setiap neuron di layer *fully connected* menghitung output sebagai perkalian matriks bobot dengan vektor input:

$$W \times I$$

Misalkan I adalah vektor $[x_1, x_2, \dots, x_{25088}]$ dan W adalah matriks $W_{4096 \times 25088}$ maka output neuron pertama O_1 dihitung sebagai:

$$O_1 = W_{11} \cdot x_1 + W_{12} \cdot x_2 + \dots + W_{1,25088} \cdot x_{25088} + b_i$$

Di mana b_i adalah bias untuk neuron pertama.

Penambahan Bias: Setelah melakukan perkalian, bias ditambahkan untuk setiap neuron:

$$O_1 = (W_i \times I) + b_i$$

Di mana $i = 1, 2, \dots, 4096$ (karena ada 4096 neuron di FC1).

3.7 arsitektur *Alexnet*

Pada tahapan pemrosesan perhitungan jenis arsitektur *Alexnet* dengan melakukan fungsi konvolusi untuk mengekstrak fitur pada citra kemudian akan dilakukan *pooling* dan *subssampling* pada citra untuk mengurangi jumlah nilai matriks ada 2 macam jenis pooling yang akan digunakan yaitu dengan menentukan jumlah *avarage pooling* dan *max polling* langkah selanjutnya yaitu melakukan fungsi dropout untuk mengurangi overfitting untuk mencegah adaptasi berlebihan pada data pelatihan salam proses penentuan akurasi terbaik yang akan diperoleh pada penelitian ini sedikit modifikasi dalam arsitektur *AlexNet* akan dilakukan jika dalam pemodelan pada tabel *AlexNet* dilakukan tidak menentukan akurasi yang tepat dan munculnya adaptasi berlebihan karna beberapa faktor maka akan di

lakukan proses modifikasi dengan struktur *AlexNet*. Kemudian hasil akhirnya akan ditentukan dengan menghitung hasil akhir dari fungsi conected layer untuk mengumpulkan proes pada *konvolusi*. bentuk detail dari arsitektur *alexNet* dilihat pada tabel 3.3:

Tabel 3.4. Tabel jaringan arsitektur *AlexNet*

AlexNet						
Layer		Feature map	Size	Karnel size	Stri de	Activation
input	Image	1	227 x 227 x 3	-	-	-
1	Convolution	96	56 x 55 x 96	11 x 11	4	ReLU
	Max Pooling	96	27 x 27 x 96	3 x 3	2	ReLU
2	Convolution	256	27 x 27 x 256	3 x 3	1	ReLU
	Max Pooling	256	13 x 13 x 256	3 x 3	2	ReLU
3	Convolution	384	13 x 13 x 384	3 x 3	1	ReLU
4	Convolution	384	13 x 13 x 384	3 x 3	1	ReLU
5	Convolution	256	13 x 13 x 256	3 x 3	1	ReLU
	Max Pooling	256	6 x 6 x 256	3 x 3	2	ReLU
6	Fully connected	-	9216	-		ReLU
7	Fully connected	-	4096	-		ReLU
8	Fully connected	-	4096	-		ReLU
output	Fully connected	-	1000	-		Softmax

3.6.1 Rumus perhitungan AlexNet secara manual

Dikondisikan dengan studi kasus pembagian data 30% data training dan 30 data testing kemudian 40% data validasi dari pembagian 3600 citra data

kelas	Pembagian data	Jumlah citra gambar
Bercak,Kekuningan Keriting,Kutu kebul	<i>Training = 30% x 3600</i>	=1.080 gambar
Bercak,Kekuningan Keriting,Kutu kebul	<i>Testing = 30% x 3600</i>	=1.080 gambar

Bercak,Kekuningan	Validasi=40%x3600	=1.440 gambar
Keriting,Kutu kebul		

Tabel 3.5 tabel pengujian dataset

3.6.2 Perhitungan pada arsitektur Alexnet

1. Input layer

Ukuran layer gambar pada arsitektur alexnet: 227x227x3

Setiap gambar pada dataset memiliki ukuran 227x227 piksel memiliki jenis gambar (RGB)

2. Layer konvolusi 1(conv1)

Filter: 96 filter dengan ukuran 11x11

Stride: 4

Padding: 0

Input: gambar dengan ukuran 227x227x3

Rumus pada konvolusi :

$$O = \frac{I - F + 2P}{s} + 1$$

Dimana :

$I = 227$ (ukuran input)

$F = 11$ (ukuran filter)

$P = 0$ (padding)

$S = 4$ stride

Dengan perhitungan:

$$O = \frac{227 - 11 + 2(0)}{4} + 1 = 55$$

Ukuran output lapisan konvolusi pertama adalah 55x55 dengan 96 filter, jadi *output layer Conv1* adalah 55x55x96.

3. Max polling:

filter *pooling* 3x3

stride 2

input 55x55x96

rumus pada *max pooling*:

$$O = \frac{I-F}{S} + 1$$

Dimana;

$I = 55$ (ukuran input)

$F = 3$ (ukuran filter pooling)

$S = 2$ (*stride*)

Dengan perhitungan:

$$O = \frac{55-3}{2} + 1 = 27$$

Ukuran output setelah pooling adalah $27 \times 27 \times 96$.

4. *layer konvolusi 2(conv 2)*:

filter: 256 dengan ukuran 5×5

stride: 1

padding: 2

input: $27 \times 27 \times 96$

menghitung *output konvolusi* kedua:

$$O = \frac{I-F+2P}{2} + 1$$

$$O = \frac{27-5+2(2)}{1} + 1 = 27$$

Maka ukuran pada output tetap 27×27 dengan 256 filter maka output conv2 adalah $27 \times 27 \times 256$

5. *Max pooling 2*

Filter pooling: 3×3

Stride: 2

Input: $27 \times 27 \times 256$

Rumus *max pooling* kedua:

$$O = \frac{27-3}{2} + 1 = 13$$

Maka hasil pooling kedua adalah $13 \times 13 \times 256$

6. *Layer konvolusi 3(conv3)*:

Filter: 384 filter dengan ukuran 3×3

Stride: 1

Padding: 1

Input: 13x13x256

Perhitungan rumus konvolusi ketiga:

$$O = \frac{I-F+2P}{s} + 1$$
$$O = \frac{13-3+2(1)}{1} + 1 = 13$$

Maka hasil *konvolusi* ketiga adalah 13x13x384

7. Layer *konvolusi 4(conv4)*

filter: 384 filter dengan ukuran 3x3

stride: 1

padding: 1

input: 13x13x384

perhitungan *konvolusi* keempat:

$$O = \frac{I-F+2P}{s} + 1$$
$$O = \frac{13-3+2(1)}{1} + 1 = 13$$

Maka ukuran output tetap 13x13 dengan 284 filter, hasil konvolusi keempat adalah 13x13x384

8. Layer *konvolusi 5(conv5)*

Filter 256 filter dengan ukuran 3x3

Stide: 1

Peding: 1

Input: 13x13x384

Perhitungan *konvolusi* kelima

$$O = \frac{I-F+2P}{s} + 1$$
$$O = \frac{13-3+2(1)}{1} + 1 = 13$$

Maka ukuran output tetap 13x13 dengan filter 256, output conv5 adalah 13x13x256

9. *Max pooling 3*

Filter pooling: 3x3

Stride: 2

Input: 13x13x256

Perhitungan *pooling* ketiga :

$$O = \frac{13-3}{2} + 1 = 6$$

Maka hasil output dari *pooling* ketiga 6x6x256

10. Flattening

Tahapan selanjutnya setelah dilakukannya konvolusi dan pooling terakhir maka peta fitur pada ukuran 6x6x256 dihitung vector dengan panjang:

$$6 \times 6 \times 256 = 9216$$

11. Fully connected layers

FC1: 4096 neuron

FC2: 4096 neuron

FC3(output layer): sama seperti jumlah kelas yang di uji (jumlah kelas 4 maka lapisan output juga 4)

3.8 Uji evaluasi *Confusion Matrix*

Proses uji evaluasi klasifikasi data pada penelitian ini dengan *Confusion Matrix* dengan menghitung akurasi, *precision*, *recall* dan *F1-score* dalam menghitung nilai dan menentukan hasil dari pengujian TP dan FP pada data yang diolah menggunakan arsitektur *VGG-16* dan *AlexNet*, berikut tampilan dari ilustrasi *Confusion Matrix*:

Tabel 3.6 perhitungan *Confusion Matrix* presisi, *Recal*, *F1-score*

True Positive (TP)	False Positive(FP)	True Negative(TN)	False Negative(FN)
150	30	50	20

Precision (P) dihitung sebagai $\frac{TP}{TP+FP}$	Recal(R) dihitung sebagai $\frac{TP}{TP+FN}$	F1-score dihitung sebagai $\frac{2 \times P \times R}{P+R}$
---	--	---

$P = \frac{150}{150+30} = \frac{150}{180} = 0.833$	$R = \frac{150}{150+20} = \frac{150}{170} = 0.882$	$FI = \frac{2 \times 0.833 \times 0.882}{0.833 + 0.882} = 0.857$
--	--	--

3.8 Hasil klasifikasi

Hasil klasifikasi akan dirangkum dalam satu kesimpulan dengan hasil akurasi terbaik dari kedua arsitektur *VGG-16* dan *AlexNet*, dengan melampirkan nilai dan proses akurasi pada tiap arsitektur yang terlebih dahulu sudah di olah menggunakan metode *evaluasi Confusion Matrix FI-score* akhir di antara kedua arsitektur *VGG-16* dan *AlexNet* yang mampu mencapai hasil nilai tertinggi maka dinyatakan baik dalam proses penelitian klasifikasi citra pada penyakit daun pada tanaman Cabai merah pada penelitian ini.

3.9 Alat dan bahan Penelitian

Adapaun alat bahan yang digunakan untuk membantu proses penelitian ini yaitu sebagai berikut :

3.9.1 Hardware

Pada tabel 3.5 ditunjukkan tampilan spesifikasi hardware yang digunakan dalam proses pengujian arsitektur *VGG-16* dan *AlexNet*.

Tabel 3.7 Hardware yang digunakan

No	Hardware	Keterangan
1.	Laptop	VivobookvivoBook ASUSLaptop X415DAP M415D4
2.	Processor	AMD Ryzen 3 3250U dengan Radeon Graphics 2.60 GHz
3.	SSD	300 GB
4.	RAM	8.00 GB

3.9.2. Software

Pada tabel 3.6 ditampilkan software yang akan memproses hasil pengujian dari arsitektur *VGG-16* dan *AlexNet*.

Tabel 3.8 Software yang digunakan

No	Softwere	keterangan
1.	Sistem Operasi	Windows 11 Home Single Language
2.	Google Colab	-RAM : 12.7 GB -Disk : 107.7 GB
3.	RAM	8 GB



BAB V

KESIMPULAN DAN SARAN

5.1. Kesimpulan

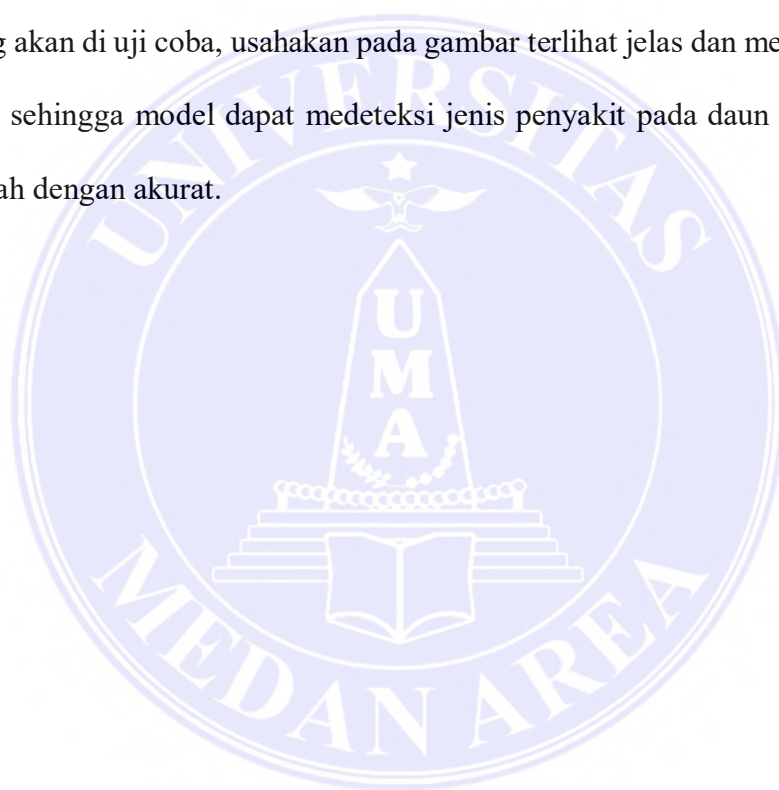
Berdasarkan hasil penelitian yang sudah dilakukan maka kesimpulan yang diambil dalam melakukan penelitian klasifikasi penyakit tanaman cabai merah berhasil dilakukan dengan menentukan hasil akurasi evaluasi *Confusion Matrix F1-score* dengan metode transfer learning untuk menguji 2 arsitektur CNN yaitu *VGG-16* dan *AlexNet* dengan melakukan modifikasi untuk mencapai hasil akurasi dari kedua arsitektur tergolong baik, dengan penambahan lapisan *fully connected* pada ke 2 arsitektur serta melakukan beberapa pengujian dengan mengubah *epoch* dan *batch size* untuk setiap arsitektur untuk memperoleh nilai akurasi tertinggi. Sehingga hasil dari akurasi tertinggi *VGG-16* mencapai 82% dan hasil akurasi yang dicapai menggunakan arsitektur *AlexNet* mencapai 56%

5.2. Saran

1. Menambahkan beberapa hyperparameter seperti jumlah layer, batch size, kernal size dan stride dan penambahan epoch yang akan dipanggil sebagai pembanding untuk memperoleh arsitektur CNN yang menghasilkan akurasi yang lebih baik lagi.
2. Selain itu penting juga untuk melakukan peningkatan Augmentasi dan metode evaluasi data hasil augmentasi agar hasil data yang di uji coba dapat lebih

maksimal dalam menggambarkan kelas serta meningkatkan performa model pelatihan yang telah dibangun.

3. Pemilihan fungsi model pada arsitektur juga penting untuk menentukan hasil akhir nilai akurasi pada pengujian baiknya menggunakan Sequential karena pada proses ini data akan di proses satu persatu untuk menghasilkan data yang dipanggil ditemukan atau tidak
4. Kemudian pentingnya kualitas gambar pada dataset mempengaruhi hasil akurasi yang akan di uji coba, usahakan pada gambar terlihat jelas dan memiliki kualitas baik sehingga model dapat medeteksi jenis penyakit pada daun tanaman cabai merah dengan akurat.



DAFTAR PUSTAKA

- AGUSTINA, R., MAGDALENA, R., & PRATIWI, N. K. C. (2022). Klasifikasi Kanker Kulit menggunakan Metode Convolutional Neural Network dengan Arsitektur VGG-16. *ELKOMIKA: Jurnal Teknik Energi Elektrik, Teknik Telekomunikasi, & Teknik Elektronika*, 10(2), 446.
- Akbar, F., Hidayatno, A., & Triwiyatno, A. (2020). Perancangan Program Pengenalan Isyarat Tangan Dengan Metode Convolutional Neural Network (Cnn). *Transient: Jurnal Ilmiah Teknik Elektro*, 9(1), 26–36.
- Alamsyah, D., & Pratama, D. (2020). implementasi CNN untuk klasifikasi ekspresi citra wajah pada FER-2013 DATASET. *Jurnal Teknologi Informasi*, 4(2), 350–355.
- Charli, F., Syaputra, H., Akbar, M., Sauda, S., & Panjaitan, F. (2020). Implementasi Metode Faster Region Convolutional Neural Network (Faster R-CNN) Untuk Pengenalan Jenis Burung Lovebird. *Journal of Information Technology Ampera*, 1(3), 185–197.
- D.K., H. B., Fadillah, M. A., Rachmawati, L. M., Jahja, M., & Suhendi, A. (2020). Convolutional Neural Networks Based on Raspberry Pi for a Prototype of Vocal Cord Abnormalities Identification. *Jurnal Infotel*, 12(3), 82–88.
- Dewi, C., Wulandari, R., Wibowo, S. A., Novamizanti, L., & Si, S. (2019). Klasifikasi Diabetes Retinopati Menggunakan Metode Statistical Region Merging Dan Convolutional Neural Network Classification of Diabetic Retinopathy Using Statistical Region Merging and Convolutional Neural Network. *E-Proceeding of Engineering*, 6(2), 4109.
- Dwi Fitriana Sari, & Swanjaya, D. (2020). Implementasi Convolutional Neural Network Untuk Identifikasi Penyakit Daun Gambas. *Seminar Nasional Inovasi Teknologi*, 04(03), 137–142.
- Eldianto, M. N. D. (2018). Implementasi Deep Learning Pada Sistem Klasifikasi Penyakit Paru Berdasarkan Foto Rontgen Menggunakan Metode Convolutional Neural Network (Cnn).
- Fadlia, N., & Kosasih, R. (2019). Klasifikasi Jenis Kendaraan Menggunakan Metode Convolutional Neural Network (Cnn). *Jurnal Ilmiah Teknologi Dan Rekayasa*, 24(3), 207–215.
- Guan, Q., Wang, Y., Ping, B., Li, D., Du, J., Qin, Y., Lu, H., Wan, X., & Xiang, J. (2019). Deep convolutional neural network VGG-16 model for differential diagnosing of papillary thyroid carcinomas in cytological images: A pilot study. *Journal of Cancer*, 10(20), 4876–4882.
- Haryono, Khairul Anam, & Azmi Saleh. (2020). Autentikasi Daun Herbal Menggunakan Convolutional Neural Network dan Raspberry Pi. *Jurnal*

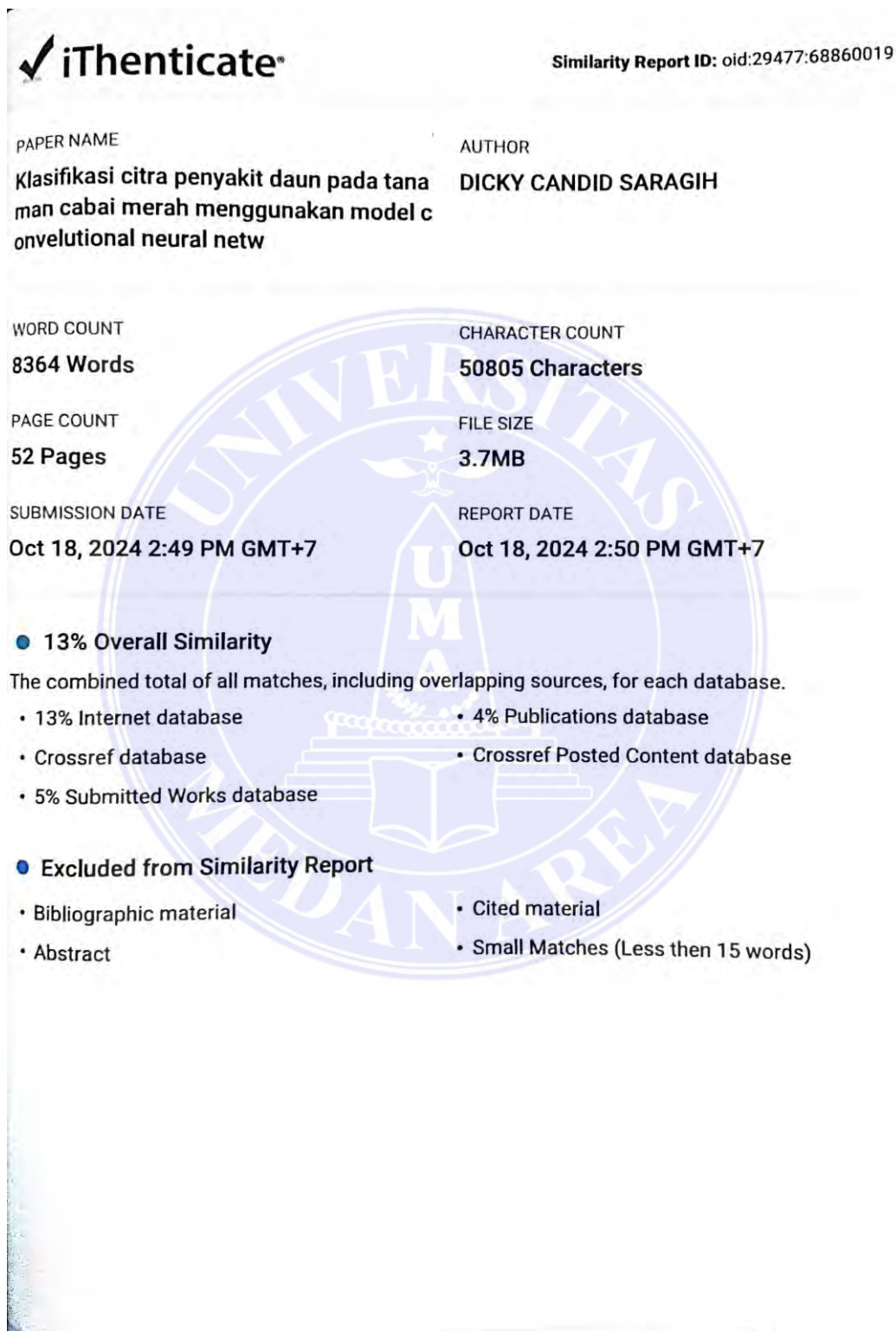
- Nasional Teknik Elektro Dan Teknologi Informasi*, 9(3), 278–286.
- Hendriyana, H., & Yazid Hilman Maulana. (2020). Identification of Types of Wood using Convolutional Neural Network with Mobilenet Architecture. *Jurnal RESTI (Rekayasa Sistem Dan Teknologi Informasi)*, 4(1), 70–76.
- Herwanto, R., Gunadi, K., & Setyati, E. (2020). Pengenalan Golongan Jenis Kendaraan Bermotor pada Ruas Jalan Tol Menggunakan CNN. *Jurnal Infra Petra*, 8(1), 196–202.
- Ilham, F., & Rochmawati, N. (2020). Transliterasi Aksara Jawa Tulisan Tangan ke Tulisan Latin Menggunakan CNN. *Journal of Informatics and Computer Science (JINACS)*, 1(04), 200–208.
- Imron, A. (2019). *Kabupaten Rembang Menggunakan Metode Naive Bayes Classifier*.
- Jadhav, S. B., Udipi, V. R., & Patil, S. B. (2019). Convolutional neural networks for leaf image-based plant disease classification. *IAES International Journal of Artificial Intelligence*, 8(4), 328–341.
- Lubis, A. A., Sirait, P., -, A., Tanrisono, A., & -, A. (2019). Klasifikasi Citra Multi Wajah Menggunakan Domain Adaptive Faster Region Convolutional Neural Network. *Jurnal SIFO Mikroskil*, 20(2), 159–168.
- Marianah, L. (2020). Serangga Vektor dan Intensitas Penyakit Virus pada Tanaman Cabai Merah Insect Vector and Virus Disease Intensity on Red Chili Plants. *AgriHumanis: Journal of Agriculture and Human Resource Development Studies*, 1(2), 127–134.
- Nisa', C., Puspaningrum, E. Y., & Maulana, H. (2020). Seminar Nasional Informatika Bela Negara (SANTIKA) Penerapan Metode Convolutional Neural Network untuk Klasifikasi Penyakit Daun Apel pada Imbalanced Data. *Seminar Nasional Informatika Bela Negara (SANTIKA)*, 1, 1–7.
- Nugroho, P. A., Fenriana, I., & Arijanto, R. (2020). Implementasi Deep Learning Menggunakan Convolutional Neural Network (CNN) Pada Ekspresi Manusia. *Algor*, 2(1), 12–21.
- Rasywir, E., Sinaga, R., & Pratama, Y. (2020). Analisis dan Implementasi Diagnosis Penyakit Sawit dengan Metode Convolutional Neural Network (CNN). *Paradigma - Jurnal Komputer Dan Informatika*, 22(2), 117–123.
- Ratna, S. (2020). Pengolahan Citra Digital Dan Histogram Dengan Phyton Dan Text Editor Phycharm. *Technologia: Jurnal Ilmiah*, 11(3), 181.
- Ridha, M. N., Setyati, E., & Kristian, Y. (2019). Identifikasi Foto Wanita Berhijab dari Majalah Untuk Pembuatan Katalog Busana Muslim Otomatis Memanfaatkan Convolutional Neural Network. *Journal of Intelligent System and Computation*, 1(2), 85–91.
- Sabilla, I. A. (2020). Arsitektur Convolutional Neural Network (Cnn) Untuk Klasifikasi Jenis Dan Kesegaran Buah Pada Neraca Buah. *Tesis, 201510370311144*, 1–119.

- Sandhopi, Lukman Zaman P.C.S.W, & Yosi Kristian. (2020). Identifikasi Motif Jepara pada Ukiran dengan Memanfaatkan Convolutional Neural Network. *Jurnal Nasional Teknik Elektro Dan TeSknologi Informasi*, 9(4), 403–413.
- Sengkey, D. F., Kambey, F. D., Lengkong, S. P., Joshua, S. R., & Kainde, H. V. F. (2020). Pemanfaatan Platform Pemrograman Daring dalam Pembelajaran Probabilitas dan Statistika di Masa Pandemi CoVID-19. *Jurnal Informatika*, 15(4), 217–224.
- Septariani, D. N., Herawati, A., & Mujiyo, M. (2019). Pemanfaatan Berbagai Tanaman Refugia Sebagai Pengendali Hama Alami Pada Tanaman Cabai (*Capsicum annum L.*). *PRIMA: Journal of Community Empowering and Services*, 3(1), 1.
- Sriyati, S., Setyanto, A., & Luthfi, E. E. (2020). Literature Review: Pengenalan Wajah Menggunakan Algoritma Convolutional Neural Network. *Jurnal Teknologi Informasi Dan Komunikasi (TIKomSiN)*, 8(2).
- Syahputra, M. I., & Wibowo, A. T. (2020). Klasifikasi Genus Tanaman Anggrek berdasarkan Citra Kuntum Bunga Menggunakan Metode Convolutional Neural Network (CNN). *E-Proceeding of Engineering*, 7(2), 8015–8023.



LAMPIRAN

- **CEK TURNITIN**



The image is a screenshot of an iThenticate similarity report. At the top left is the iThenticate logo. At the top right is the Similarity Report ID: oid:29477:68860019. The report is divided into several sections: Paper Name and Author, Word Count and Character Count, Page Count and File Size, Submission Date and Report Date. Below these is a section for Overall Similarity (13%) and a list of sources. Another section lists items excluded from the similarity report.

iThenticate Similarity Report ID: oid:29477:68860019

PAPER NAME	AUTHOR
Klasifikasi citra penyakit daun pada tanaman cabai merah menggunakan model convolutional neural network	DICKY CANDID SARAGIH


WORD COUNT	CHARACTER COUNT
8364 Words	50805 Characters

PAGE COUNT	FILE SIZE
52 Pages	3.7MB

SUBMISSION DATE	REPORT DATE
Oct 18, 2024 2:49 PM GMT+7	Oct 18, 2024 2:50 PM GMT+7

- **13% Overall Similarity**
The combined total of all matches, including overlapping sources, for each database.
 - 13% Internet database
 - 4% Publications database
 - Crossref database
 - Crossref Posted Content database
 - 5% Submitted Works database
- **Excluded from Similarity Report**
 - Bibliographic material
 - Cited material
 - Abstract
 - Small Matches (Less than 15 words)

- SK PEMBIMBING



UNIVERSITAS MEDAN AREA

FAKULTAS TEKNIK

Kampus I : Jalan Kolam Nomor 1 Medan Estate/Jalan PBSI Nomor 1 (061) 7366878, 7360168, 7364348, 7366781, Fax. (061) 7366998 Medan 20223
Kampus II : Jalan Seliabudi Nomor 79 / Jalan Sei Serayu Nomor 70 A, (061) 8225602, Fax. (061) 8226331 Medan 20122
Website: www.teknik.uma.ac.id E-mail: univ_medanarea@uma.ac.id

Nomor : 833/FT.6/01.10/XII/2023 29 Desember 2023
Lamp : -
Hal : **Perpanjangan SK Pembimbing Tugas Akhir**

Yth. Pembimbing Tugas Akhir
Rizki Muliono, S. Kom, M. Kom
di
Tempat

Dengan hormat,
Sehubungan telah berakhirnya waktu masa berlaku SK pembimbing nomor 473/FT.6/01.10/VI/2023 tertanggal 16 Juni 2023 maka perlu diterbitkan kembali SK Pembimbing Skripsi baru atas nama mahasiswa berikut :

N a m a : Dicky Candid Saragih
N P M : 198160061
Jurusan : Teknik Informatika

Oleh karena itu kami mengharapkan kesediaan saudara :


Rizki Muliono, S. Kom, M. Kom (Sebagai Pembimbing)

Adapun Tugas Akhir Skripsi berjudul :

"Klasifikasi Citra Penyakit Daun pada Tanaman Cabai Merah Menggunakan Model Convolutional Neural Networks (CNN) dengan VGG-16 dan AlexNet"

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. Engi Saragih, ST, MT

• SK IZIN PENELITIAN DAN PENGAMBILAN DATA



UNIVERSITAS MEDAN AREA
FAKULTAS TEKNIK

Kampus I : Jalan Kolam Nomor 1 Medan Estate/Jalan PUSI Nomor 150 (051) 7366878, 7360168, 7364348, 7366781, Fax (051) 7366888 Medan 20222
Kampus II : Jalan Sebelahdi Nomor 79 / Jalan Sei Saraya Nomor 70 A, (051) 8225602, Fax (051) 8226331 Medan 20122
Website: www.ifttek.uma.ac.id E-mail: umt.pendataran@uma.ac.id

Nomor : 16 /FT.6/01.10/1/2024 5 Januari 2024
Lamp : -
Hal : Penelitian Dan Pengambilan Data Tugas Akhir

Yth. Wakil Rektor Bid. Pengembangan SDM & Adm. Keuangan
Jln. Kolam No.1
Di
Medan

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

NO	N A M A	N P M	PRODI
1	Dicky Candid Saragih	198160061	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 :

Klasifikasi Citra Penyakit Daun pada Tanaman Cabai Merah Menggunakan Model *Convolutional Neural Networks (CNN)* dengan *VGG-16* dan *AlexNet*.

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

Atas perhatian dan kerja sama yang baik diucapkan terima kasih.


Dekan
Dr. Edy M. H. Sano, ST, MT
FAKULTAS TEKNIK

Tembusan :
1. Ka. BAMA
2. Mahasiswa
3. File

• SK SELESAI RISET



UNIVERSITAS MEDAN AREA

Kampus I : Jalan Kolam Nomor 1 Medan Estate ☎ (061) 7360168, 7366878, 7364348 ☎ (061) 7368012 Medan 20223
Kampus II : Jalan Setia Budi Nomor 79 / Jalan Sei Serayu Nomor 70 A ☎ (061) 8225602 ☎ (061) 8226331 Medan 20122
Website: www.uma.ac.id E-Mail: univ_medanarea@uma.ac.id

- SURAT KETERANGAN
Nomor : 188/UMA/B/01.7/11/2024

Rektor Universitas Medan Area dengan ini menerangkan bahwa :

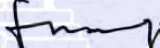
Nama : Dicky Candid Saragih
No.Pokok Mahasiswa : 198160061
Fakultas : Teknik
Program Studi : Teknik Informatika

Benar telah selesai Pengambilan Data Tugas Akhir di Laboratorium Komputer Universitas Medan Area dengan Judul Skripsi "**Klasifikasi Citra Penyakit Daun pada Tanaman Cabai Merah Menggunakan Model *Convolutional Neural Networks* (CNN) dengan VGG-16 dan Alexnet**".

Dan kami harapkan Data tersebut kiranya dapat membantu yang bersangkutan dalam penyusunan skripsi dan dapat bermanfaat bagi mahasiswa khususnya Fakultas Teknik Universitas Medan Area.

Demikian surat ini diterbitkan untuk dapat digunakan seperlunya

Medan, 05 Februari 2024.
an Rektor
Wakil Rektor Bidang Mutu SDM &
Perekonomian,


Prof. Dr. Ir. Suswati, MP

CC:
- Arsip



```
+ Code + Text

[ ] import tensorflow as tf
    from tensorflow.keras.preprocessing.image import ImageDataGenerator
    from tensorflow.keras.applications import VGG16
    from tensorflow.keras.layers import Flatten, Dense
    from tensorflow.keras.models import Model
    from sklearn.metrics import classification_report, confusion_matrix, accuracy_score, recall_score, precision_score, f1_score
    import numpy as np
    import matplotlib.pyplot as plt
    import seaborn as sns
    from google.colab import drive

▶ # Mount Google Drive
drive.mount('/content/drive')
import os

# Path ke folder utama di Google Drive
drive_path = '/content/drive/My Drive/'

# Nama folder utama

↳ Mounted at /content/drive

▶ # Path ke folder train, val, dan test
train_folder = os.path.join(drive_path, main_folder, 'training')
val_folder = os.path.join(drive_path, main_folder, 'validasi')
test_folder = os.path.join(drive_path, main_folder, 'testing')

# List file dan folder dalam setiap direktori
train_files = os.listdir(train_folder)
val_files = os.listdir(val_folder)
test_files = os.listdir(test_folder)

# Tampilkan isi folder
print(f"Isi folder 'training': {train_files}")
print(f"Isi folder 'validasi': {val_files}")
print(f"Isi folder 'testing': {test_files}")

↳ Isi folder 'training': ['Kutu_kebul', 'Kekuningan', 'Bercak', 'Keriting']
Isi folder 'validasi': ['Kekuningan', 'Bercak', 'Kutu_Kebul', 'Keriting']
Isi folder 'testing': ['Kutu_Kebul', 'Keriting', 'Bercak', 'Kekuningan']

[ ] # Pengaturan generator data
train_datagen = ImageDataGenerator(
    rescale=1./255,
    rotation_range=10,
```

```

▶ train_datagen = ImageDataGenerator(
    rescale=1./255,
    rotation_range=10,
    width_shift_range=0.2,
    height_shift_range=0.2,
    shear_range=0.2,
    zoom_range=0.2,
    horizontal_flip=True,
    fill_mode='nearest')

val_datagen = ImageDataGenerator(rescale=1./255)
test_datagen = ImageDataGenerator(rescale=1./255)

[ ] # Pengaturan generator data
train_datagen = ImageDataGenerator(
    rescale=1./255,
    rotation_range=10,
    width_shift_range=0.2,
    height_shift_range=0.2,
    shear_range=0.2,
    zoom_range=0.2,
    horizontal_flip=True,
    fill_mode='nearest')

val_datagen = ImageDataGenerator(rescale=1./255)
test_datagen = ImageDataGenerator(rescale=1./255)

train_generator = train_datagen.flow_from_directory(
    train_folder,
    target_size=(150, 150),
    batch_size=16,
    class_mode='binary') # Sesuaikan dengan tipe masalah (binary atau categorical)

# Menampilkan beberapa sampel gambar dari generator pelatihan
!pip install matplotlib
import matplotlib.pyplot as plt

for i in range(2): # Menampilkan 5 sampel gambar
    images, labels = next(train_generator) # Use next(train_generator) instead of train_generator.next()
    plt.imshow(images[0]) # Menampilkan hanya gambar pertama dari setiap batch
    plt.title(f'Label: {labels[0]}')
    plt.show()

```



```

▶ from tensorflow.keras.preprocessing.image import ImageDataGenerator

# Define data directories
train_dir = train_folder
validation_dir = val_folder
test_dir = test_folder

# Create ImageDataGenerator for training
train_datagen = ImageDataGenerator(rescale=1./255,
                                   rotation_range=10,
                                   width_shift_range=0.2,
                                   height_shift_range=0.2,
                                   shear_range=0.2,
                                   zoom_range=0.2,
                                   horizontal_flip=True,
                                   fill_mode='nearest')

# Create ImageDataGenerator for validation and testing (only rescaling)
val_datagen = ImageDataGenerator(rescale=1./255)
test_datagen = ImageDataGenerator(rescale=1./255)

# Define batch size
batch_size = 16

# Create data generators
train_generator = train_datagen.flow_from_directory(
    train_dir, target_size=(224, 224), batch_size=batch_size, class_mode='categorical', shuffle=True)

validation_generator = val_datagen.flow_from_directory(
    validation_dir, target_size=(224, 224), batch_size=batch_size, class_mode='categorical', shuffle=True)

test_generator = test_datagen.flow_from_directory(
    test_dir, target_size=(224, 224), batch_size=batch_size, class_mode='categorical', shuffle=False)

Found 1200 images belonging to 4 classes.
Found 1195 images belonging to 4 classes.
Found 1200 images belonging to 4 classes.

| num_classes = 4 # Pastikan ini sesuai dengan jumlah kelas di dataset Anda
img_height, img_width = 224, 224

base_model = VGG16(weights='imagenet', include_top=False, input_shape=(img_height, img_width, 3))
x = Flatten()(base_model.output)
x = Dense(1024, activation='relu')(x)
predictions = Dense(num_classes, activation='softmax')(x)

predictions = Dense(num_classes, activation='softmax')(x)
model = Model(inputs=base_model.input, outputs=predictions)

for layer in base_model.layers:
    layer.trainable = False

model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'])

📄 Downloading data from https://storage.googleapis.com/tensorflow/keras-applications/vgg16/vgg16\_weights\_t58889256/58889256 1s 0us/step

[ ] history = model.fit(train_generator, epochs=10, validation_data=validation_generator)

```

```
[ ] # Evaluasi model
test_generator.reset()
predictions = model.predict(test_generator)
predicted_classes = np.argmax(predictions, axis=1)
true_classes = test_generator.classes
class_labels = list(test_generator.aclass_indices.keys())

# Menghitung confusion matrix
conf_matrix = confusion_matrix(true_classes, predicted_classes)
print('Confusion Matrix')
print(conf_matrix)

# Menghitung metrik evaluasi
accuracy = accuracy_score(true_classes, predicted_classes)
recall = recall_score(true_classes, predicted_classes, average='macro')
precision = precision_score(true_classes, predicted_classes, average='macro')
f1 = f1_score(true_classes, predicted_classes, average='macro')

print(f'Akurasi: {accuracy:.4f}')
print(f'Recall: {recall:.4f}')
print(f'Presisi: {precision:.4f}')
print(f'Skor F1: {f1:.4f}')

# Laporan klasifikasi untuk setiap kelas
print('\nLaporan Klasifikasi:\n')
print(classification_report(true_classes, predicted_classes, target_names=class_labels))

# Mencetak nama setiap kelas
print("\nLabel Kelas:")
print(class_labels)

# Membuat heatmap dari confusion matrix
plt.figure(figsize=(10, 8))
sns.heatmap(conf_matrix, annot=True, fmt='g', cmap='Blues', xticklabels=class_labels, yticklabels=class_labels)
plt.xlabel('Prediksi')
plt.ylabel('Sebenarnya')
plt.title('Heatmap Confusion Matrix')
plt.show()
```

```
▶ # Ensure all necessary modules are imported
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Conv2D, MaxPooling2D, Flatten, Dense, BatchNormalization, Dropout
!pip install matplotlib
import matplotlib.pyplot as plt
from google.colab import drive
import os
```

```
[ ] # Mount Google Drive
drive.mount('/content/drive')
import os

# Path ke folder utama di Google Drive
drive_path = '/content/drive/My Drive/'

# Nama folder utama
main_folder = 'cabeku'

→ Mounted at /content/drive

▶ # Pengaturan generator data
train_datagen = ImageDataGenerator(
    rescale=1./255,
    rotation_range=20,
    width_shift_range=0.2,
    height_shift_range=0.2,
    shear_range=0.2,
    zoom_range=0.2,
    horizontal_flip=True,
    fill_mode='nearest')

val_datagen = ImageDataGenerator(rescale=1./255)
test_datagen = ImageDataGenerator(rescale=1./255)

# Define batch size
batch_size = 16

# Path ke folder train, val, dan test
train_folder = os.path.join(drive_path, main_folder, 'training')
val_folder = os.path.join(drive_path, main_folder, 'validasi')
test_folder = os.path.join(drive_path, main_folder, 'testing')
```

```

# Define batch size
batch_size = 16
# Create data generators
train_generator = train_datagen.flow_from_directory(
    train_folder, target_size=(224, 224), batch_size=batch_size, class_mode='categorical', shuffle=True)

validation_generator = val_datagen.flow_from_directory(
    val_folder, target_size=(224, 224), batch_size=batch_size, class_mode='categorical', shuffle=False)

test_generator = test_datagen.flow_from_directory(
    test_folder, target_size=(224, 224), batch_size=batch_size, class_mode='categorical', shuffle=False)

# Definisikan jumlah kelas dan ukuran gambar
num_classes = 4 # Sesuaikan dengan jumlah kelas di dataset Anda
img_height, img_width = 224, 224

```

```

[ ] # train data generator

import matplotlib.pyplot as plt
# Display a few sample images from the training dataset
sample_images, sample_labels = next(train_datagen.flow_from_directory(
    train_folder, target_size=(224, 224), batch_size=25, class_mode='categorical', shuffle=True))

plt.figure(figsize=(20, 20))
for i in range(25):
    plt.subplot(5, 5, i + 1)
    plt.imshow(sample_images[i])
    plt.axis('off')
plt.show()

```

```

# Mendefinisikan struktur model AlexNet
def alexnet(input_shape, num_classes):
    model = Sequential([
        Conv2D(96, kernel_size=(11, 11), strides=(4,4), input_shape=input_shape, activation='relu'),
        BatchNormalization(),
        MaxPooling2D(pool_size=(3, 3), strides=(2,2)),
        Conv2D(256, kernel_size=(5, 5), padding='same', activation='relu'),
        BatchNormalization(),
        MaxPooling2D(pool_size=(3, 3), strides=(4,4)),
        Conv2D(384, kernel_size=(3, 3), padding='same', activation='relu'),
        Conv2D(384, kernel_size=(3, 3), padding='same', activation='relu'),
        Conv2D(256, kernel_size=(3, 3), padding='same', activation='relu'),
        MaxPooling2D(pool_size=(3, 3), strides=(4,4)),
        Flatten(),
        Dense(4096, activation='relu'),
        Dropout(0.5),
        Dense(4096, activation='relu'),
        Dropout(0.5),
        Dense(num_classes, activation='softmax')
    ])
    return model

```

```

# Gunakan model AlexNet
model = alexnet((img_height, img_width, 3), num_classes)

# Kompilasi model
model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'])

# Fit the model and add try-except block to catch exceptions during training
try:
    history = model.fit(
        train_generator,
        steps_per_epoch=train_generator.samples // batch_size,
        epochs=10,
        validation_data=validation_generator,
        validation_steps=validation_generator.samples // batch_size,
    )

```

```
[ ] # Evaluasi model
test_generator.reset()
predictions = model.predict(test_generator)
predicted_classes = np.argmax(predictions, axis=1)
true_classes = test_generator.classes
class_labels = list(test_generator.aclass_indices.keys())

# Menghitung confusion matrix
conf_matrix = confusion_matrix(true_classes, predicted_classes)
print('Confusion Matrix')
print(conf_matrix)

# Menghitung metrik evaluasi
accuracy = accuracy_score(true_classes, predicted_classes)
recall = recall_score(true_classes, predicted_classes, average='macro')
precision = precision_score(true_classes, predicted_classes, average='macro')
f1 = f1_score(true_classes, predicted_classes, average='macro')

print(f'Akurasi: {accuracy:.4f}')
print(f'Recall: {recall:.4f}')
print(f'Presisi: {precision:.4f}')
print(f'Skor F1: {f1:.4f}')

# Laporan klasifikasi untuk setiap kelas
print('\nLaporan Klasifikasi:\n')
print(classification_report(true_classes, predicted_classes, target_names=class_labels))

# Mencetak nama setiap kelas
print("\nLabel Kelas:")
print(class_labels)

# Membuat heatmap dari confusion matrix
plt.figure(figsize=(10, 8))
sns.heatmap(conf_matrix, annot=True, fmt='g', cmap='Blues', xticklabels=class_labels, yticklabels=class_labels)
plt.xlabel('Prediksi')
plt.ylabel('Sebenarnya')
plt.title('Heatmap Confusion Matrix')
plt.show()
```