

**IMPLEMENTASI *CONVOLUTIONAL NEURAL NETWORK* MENGGUNAKAN ARSITEKTUR *VISUAL GEOMETRY GROUP-19 (VGG-19)* UNTUK
KLASIFIKASI CITRA BUAH SALAK**

(Studi Kasus : Klasifikasi Citra Buah Salak Asal DI Yogyakarta)

TUGAS AKHIR

Diajukan Sebagai Salah Satu Syarat Untuk Memperoleh Gelar Sarjana Program
Studi Statistika



Disusun Oleh:

Indina Isyfi Annie Widyan

20611209

**PROGRAM STUDI STATISTIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS ISLAM INDONESIA
YOGYAKARTA
2024**

**HALAMAN PERSETUJUAN PEMBIMBING
TUGAS AKHIR**

Judul : Implementasi *Convolutional Neural Network*
Menggunakan Arsitektur *Visual Geometry Group-19 (VGG-19)* untuk Klasifikasi Citra Buah Salak
(Studi kasus: Klasifikasi Citra Buah Salak Asal DI Yogyakarta)

Nama Mahasiswa : Indina Isyfi Annie Widyan

NIM : 20611209

**TUGAS AKHIR INI TELAH DIPERIKSA DAN
DISETUIJUI UNTUK DIUJIKAN**

Mengetahui
Ketua Prodi Statistika

Yogyakarta, 13 Maret 2024
Dosen Pembimbing


Dr. Atina Ahlika, S.Si., M.Si


Dr. Atina Ahlika, S.Si., M.Si

**HALAMAN PENGESAHAN
TUGAS AKHIR**

**IMPLEMENTASI *CONVOLUTIONAL NEURAL NETWORK*
MENGUNAKAN ARSITEKTUR *VISUAL GEOMETRY GROUP-19*
(VGG-19) UNTUK KLASIFIKASI CITRA BUAH SALAK
(Studi Kasus : Klasifikasi Citra Buah Salak Asal DI Yogyakarta)**

Nama Mahasiswa : Indina Isyfi Annie Widyan

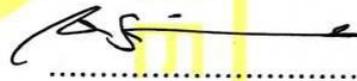
NIM : 20611209

**TUGAS AKHIR INI TELAH DIUJIKAN
PADA TANGGAL : 27 Maret 2024**

Nama Penguji

Tanda Tangan

1. Dr. RB. Fajriya Hakim, S.Si., M.Si



2. Ayundyah Kesumawati, S.Si., M.Si



3. Dr. Atina Ahdika, S.Si., M.Si



Mengetahui,

Dekan Fakultas Matematika dan Ilmu Pengetahuan Alam



Prof. Riyanto, S.Pd., M.Si., Ph.D



KATA PENGANTAR



Assalamu'alaikum Wr.Wb

Alhamdulillah rabbil'alamin puji serta syukur penulis panjatkan atas kehadiran Allah SWT yang telah melimpahkan rahmat serta hidayah – Nya sehingga Tugas Akhir ini dapat diselesaikan. Shalawat serta salam selalu tercurah limpahkan kepada Nabi Muhammad SAW kepada keluarganya, sahabatnya, serta kita selaku umatnya, semoga mendapat syafaatnya sampai akhir zaman Aamiin.

Penyusunan Tugas Akhir ini merupakan salah satu syarat yang harus dipenuhi untuk memperoleh gelar sarjana di Program Studi Statistika, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Islam Indonesia. Dalam proses penyusunan ini, tentu saja penulis diberikan bimbingan dan dukungan oleh berbagai pihak, baik dalam bentuk materi maupun non materi. Oleh karena itu, pada kesempatan ini, penulis ingin mengungkapkan rasa terima kasih kepada:

1. Orang tua tercinta Mamah Ida Widayanti dan Papap Winaja yang selalu memberi dukungan dari awal perkuliahan sampai saat ini, do'a yang terus mengalir mengiringi perjalanan perkuliahan penulis, dan memotivasi penulis sehingga dapat menyelesaikan Tugas Akhir ini dengan lancar.
2. Keluarga besar penulis yang turut memberikan dukungan, semangat, motivasi, dan doa kepada penulis selama proses perkuliahan.
3. Prof. Riyanto, S.Pd., M.Si., Ph.D. selaku Dekan Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Islam Indonesia.
4. Bapak Dr. Edy Widodo, S.Si., M.Si., selaku Ketua Jurusan Statistika, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Islam Indonesia beserta jajarannya.
5. Ibu Dr. Atina Ahdika, S.Si., M.Si., selaku Ketua Program Studi Statistika dan dosen pembimbing penulis yang telah memberi bimbingan dan arahan dalam penyusunan Tugas Akhir ini.
6. Seluruh Tenaga Pengajar dan Staf Program Studi Statistika yang telah banyak memberikan ilmu kepada penulis selama perkuliahan berlangsung.
7. Keluarga Mejimi kesebelasan, *my 911* selama *campus life* yang telah

menemani perjalanan penulis dari awal masuk hingga akhir perkuliahan, saling memotivasi, berbagi canda dan tawa, serta memberikan warna di masa perkuliahan penulis.

8. Manager, dan rekan INVISIO Gen 6 sampai Gen 9 yang telah bekerja sama, berbagi ilmu, serta berbagi canda dan tawa.
9. Teman-teman bimbingan TA Bu Atina yang sudah sama-sama berjuang, saling mengingatkan, dan memberi motivasi serta dorongan untuk menyelesaikan Tugas Akhir ini.
10. Serta semua pihak lainnya yang tidak bisa dituliskan penulis satu per satu yang telah membantu selama pembuatan Tugas Akhir ini.

Demikian Tugas Akhir ini, penulis mengucapkan terima kasih kepada semua pihak yang telah memberikan bantuan baik moril maupun materil sehingga Tugas Akhir ini dapat diselesaikan. Penulis menyadari bahwa Tugas Akhir ini masih jauh dari kata sempurna dan masih banyak kekurangan. Hal tersebut dikarenakan keterbatasan ilmu dan pengetahuan yang dimiliki penulis semata. Akhir kata, semoga Allah SWT senantiasa melimpahkan rahmat serta hidayah-Nya kepada kita semua, Amin amin ya robbal 'alamiin.

Wassalamualaikum Wr.Wb

Yogyakarta, 13 Maret 2024



Indina Isyfi Annie Widyan

DAFTAR ISI

HALAMAN PERSETUJUAN PEMBIMBING TUGAS AKHIR.....	ii
HALAMAN PENGESAHAN TUGAS AKHIR	iii
KATA PENGANTAR.....	iv
DAFTAR ISI	vi
DAFTAR TABEL	viii
DAFTAR GAMBAR.....	ix
PERNYATAAN	x
INTISARI	xi
ABSTRACT	xii
BAB I PENDAHULUAN	1
1.1. Latar Belakang Masalah.....	1
1.2. Rumusan Masalah	3
1.3. Batasan Masalah.....	4
1.4. Tujuan Penelitian	4
1.5. Manfaat Penelitian	5
BAB II TINJAUAN PUSTAKA	6
2.1. Penelitian Terdahulu	6
BAB III LANDASAN TEORI	11
3.1 Buah Salak.....	11
3.1 Citra Digital.....	13
3.2 Augmentasi Data.....	15
3.2.1 <i>Cropping</i>	15
3.2.2 <i>Resize</i>	16
3.2.3 <i>Padding dan Stride</i>	17
3.2.4 <i>Flipping dan Rotation</i>	17
3.3 <i>Deep Learning</i>	18
3.4 <i>Convolutional Neural Network (CNN)</i>	19
3.4.1 <i>Feature Extraction</i>	20
3.4.2 <i>Fully Connected Layer</i>	24
3.5 <i>Visual Geometry Group-19 (VGG-19)</i>	26
3.6 <i>Optimization Algorithm</i>	28
3.7 Matrik Evaluasi Klasifikasi.....	29
3.7.1 <i>Confusion Matrix</i>	29
3.7.2 Akurasi	30
3.7.3 Presisi	30
3.7.4 <i>Recall (Sensitivitas)</i>	31
3.7.5 <i>F1-score</i>	31
BAB IV METODOLOGI PENELITIAN.....	32
4.1. Populasi dan Sampel Penelitian	32
4.2. Tempat dan Waktu Penelitian	32
4.3. Definisi Operasional Variabel Penelitian.....	32
4.4. Alat dan cara organisir data.....	34
4.5. Metode Penelitian.....	34
BAB V HASIL DAN PEMBAHASAN	38
6.1 Pengumpulan dan Inputasi Data.....	38

BAB VI PENUTUP.....	49
DAFTAR PUSTAKA.....	51
LAMPIRAN	57

DAFTAR TABEL

Tabel 2.1 Tabel Penelitian Sebelumnya	8
Tabel 4.1 Definisi Variabel Penelitian	32
Tabel 5.1 Labelisasi <i>Class</i> Buah Salak	40
Tabel 5.2 <i>Output</i> Model CNN VGG-19	41
Tabel 5.3 Proses <i>Training</i> Model	43
Tabel 5.4 Proses <i>Testing</i> Model	44
Tabel 5.5 Evaluasi Model.....	45

DAFTAR GAMBAR

Gambar 3.1 Salak Pondoh	12
Gambar 3.2 Salak Madu	12
Gambar 3.3 Salak Gading	13
Gambar 3.4 Representasi Citra Digital (Haidaroh, 2016)	14
Gambar 3.5 Contoh Augmentasi Data (Ho, dkk 2019)	15
Gambar 3.6 Contoh <i>Cropping Image</i> (Kalambe, 2022)	16
Gambar 3.7 Arsitektur Penyusun <i>Deep Learning</i> (Bahi & Moamed, 2018).....	19
Gambar 3.8 Arsitektur <i>Convolutional Neural Network</i> (CNN) (Hendriyana & Maulana, 2020)	20
Gambar 3.9 Operasi <i>Input</i> dan Filter CNN (Dharmadi, 2018).....	21
Gambar 3.10 <i>Stride Convolutional Layer</i> (Chen, 2017)	22
Gambar 3.11 Fungsi Aktivasi ReLU (Sanjaya & Ayub, 2020).....	23
Gambar 3.12 <i>Max Pooling</i> (Sanjaya & Ayub, 2020)	24
Gambar 3.13 <i>Neural Network Layer</i> (Ognjanovski, 2019)	25
Gambar 3.14. Proses <i>Fully Connected Layer</i> (Nurhikmat, 2018).....	26
Gambar 3.15 Arsitektur VGG-19 (Xiao, dkk 2023)	27
Gambar 3.16 Operasi Matriks Lapisan Konvolusi (Parhusip, 2021)	27
Gambar 3.17 <i>Confusion Matriks</i> (Nugroho, 2019)	30
Gambar 4.1 <i>Flowchart</i> Penelitian	30
Gambar 5.1 Augmentasi Data Pada Salak Pondoh Sehat.....	39
Gambar 5.2 Pembagian Data.....	39
Gambar 5.3 Citra Sebelum dan Setelah Perubahan Piksel	40
Gambar 5.4 Grafik <i>Loss Value</i> Proses <i>Training</i> Model	43
Gambar 5.5 Grafik <i>Accuracy Value</i> Proses <i>Training</i> Model	44
Gambar 5.6 <i>Confusion Matrix</i> Model <i>Testing</i>	46

PERNYATAAN

Dengan ini saya menyatakan bahwa dalam Tugas Akhir ini tidak terdapat karya karya yang sebelumnya pernah diajukan untuk memperoleh gelar kesarjanaan di suatu Perguruan Tinggi dan sepanjang pengetahuan saya juga tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan orang lain, kecuali yang diacu dalam naskah ini dan disebutkan dalam daftar pustaka.

Yogyakarta, 13 Maret 2024



Indinalisyfi Annie Widyan

INTISARI

IMPLEMENTASI *CONVOLUTIONAL NEURAL NETWORK* MENGUNAKAN ARSITEKTUR *VISUAL GEOMETRY GROUP-19* (VGG- 19) UNTUK KLASIFIKASI CITRA BUAH SALAK (Studi Kasus : Klasifikasi Citra Buah Salak Asal DI Yogyakarta)

Indina Isyfi Annie Widyan
Program Studi Statistika, Fakultas Matematika dan Ilmu Pengetahuan Alam
Universitas Islam Indonesia

Penelitian ini bertujuan untuk mengimplementasikan metode *Convolutional Neural Network* (CNN) dengan menggunakan arsitektur *Visual Geometry Group-19* (VGG-19) untuk mengklasifikasikan varian buah salak yang berasal dari DI Yogyakarta. Enam kelas buah salak yang digunakan pada penelitian ini meliputi salak Gading busuk, Gading sehat, Madu busuk, Madu sehat, Pondoh busuk, dan Pondoh sehat. Menggunakan data primer berupa 1.200 citra buah salak, dengan setiap kelas memiliki representasi sebanyak 200 citra. Data citra dibagi menjadi tiga subset, yaitu 80% untuk data train, 10% untuk data validasi, dan 10% untuk data test untuk memastikan generalisasi model. Hasil penelitian ini memperoleh tingkat akurasi yang signifikan, dengan model mencapai 94,58% akurasi pada data *train*, 95,83% pada data validasi, dan 95,83% pada data *test*. Berdasarkan penelitian ini menunjukkan bahwa model CNN dengan arsitektur VGG-19 terbukti efektif dalam mengklasifikasikan varian buah salak di Yogyakarta. Tingkat akurasi yang konsisten pada ketiga subset data menegaskan kemampuan model dalam mengidentifikasi dengan tepat setiap kelas buah salak. Keberhasilan model ini dalam mengklasifikasikan varian buah salak dengan tingkat akurasi yang tinggi memberikan kontribusi positif dalam peningkatan efisiensi dan akurasi pengelolaan produksi salak di Yogyakarta. Penelitian ini menjadi landasan untuk pengembangan sistem otomatisasi yang dapat membantu meningkatkan pengenalan kualitas dan jenis buah salak secara cepat dan akurat dalam industri perkebunan.

Kata Kunci : Akurasi, Buah Salak, CNN, Klasifikasi Citra, VGG-19

ABSTRACT

THE IMPLEMENTATION OF CONVOLUTIONAL NEURAL NETWORK USING VISUAL GEOMETRY GROUP-19 (VGG-19) ARCHITECTURE FOR SNAKE FRUIT IMAGE CLASSIFICATION (CASE STUDY : IMAGE CLASSIFICATION OF SNAKE FRUIT FROM DI YOGYAKARTA)

Indina Isyfi Annie Widyan
Department of Statistics, Faculty of Mathematics and Natural Sciences
Universitas Islam Indonesia

This research aims to implement the Convolutional Neural Network (CNN) method using the Visual Geometry Group-19 (VGG-19) architecture to classify variants of snake fruit from DI Yogyakarta. The six classes of snake fruit used in this study include rotten Gading snake fruit, healthy Gading snake fruit, rotten Madu snake fruit, healthy Madu snake fruit, rotten Pondoh snake fruit, and healthy Pondoh snake fruit. We utilize primary data consisting of 1,200 images of snake fruit, with each class represented by 200 images. The image data is divided into three subsets: 80% for training data, 10% for validation data, and 10% for test data to ensure model generalization. The results of this research obtained a significant accuracy level, with the model reaching 95% accuracy on the training data, 96% on the validation data, and 96% on the test data. Based on this research, it is shown that the CNN model with the VGG-19 architecture proves to be effective in classifying variants of snake fruit in Yogyakarta. The consistent accuracy across all three data subsets confirms the model's ability to accurately identify each class of snake fruit. The success of this model in classifying variants of snake fruit with high accuracy contributes positively to improving the efficiency and accuracy of snake fruit production management in Yogyakarta. This research serves as a foundation for the development of automation systems that can help enhance the rapid and accurate recognition of the quality and types of snake fruit in the plantation industry.

Keywords: Accuracy, CNN, Image Classification, Snake fruit, VGG-19

BAB I

PENDAHULUAN

1.1. Latar Belakang Masalah

Salacca zalacca atau buah salak adalah jenis buah tropis yang berasal dari Indonesia. Berdasarkan data yang dirilis oleh Badan Pusat Statistik, pada tahun 2022 produksi salak di Indonesia mencapai 1.147.473 ton/tahun. Produsen utama buah salak di Indonesia berasal dari Provinsi Jawa Tengah dengan jumlah produksi sebesar 482.423 ton, Sumatera Utara sebesar 265.180, Jawa Timur sebesar 199.625, dan DI Yogyakarta sebesar 54.498 (Badan Pusat Statistik, 2022).

Berdasarkan informasi yang telah diuraikan di atas, provinsi DI Yogyakarta berperan sebagai salah satu produsen buah salak terbesar di Indonesia, menempati peringkat keempat dalam hal kontribusi produksi nasional buah salak. Meskipun bukan yang terbesar, kontribusi dari provinsi DI Yogyakarta terhadap produksi buah salak di Indonesia tetap signifikan. Selain memproduksi buah salak dalam skala yang cukup besar, Provinsi DI Yogyakarta juga berperan penting dalam distribusi buah salak ke pasar lokal maupun pasar ekspor. Kualitas buah salak dari DI Yogyakarta sering kali menjadi daya tarik tersendiri bagi pasar lokal dan juga memperoleh perhatian dari pasar ekspor (Bappeda Sleman, 2021). Dengan kontribusinya yang signifikan terhadap produksi buah salak di Indonesia, Provinsi DI Yogyakarta tidak hanya menjadi bagian integral dalam ekonomi lokal, tetapi juga berperan dalam menjaga keberlanjutan industri buah salak di tingkat nasional.

Dalam proses pendistribusian buah salak, memerlukan pemilahan yang hati-hati agar salak yang dikirimkan mempunyai kualitas yang baik dan tetap segar, terutama untuk pengiriman salak ke lokasi yang jauh atau untuk keperluan ekspor. Buah salak yang layak dikirim harus dipilih dengan teliti, memastikan buah salak tersebut tidak terlalu matang, bebas dari cacat seperti sobekan atau kerusakan, dan busuk. Salak yang siap dikirim memiliki sisik yang sedang, warna kulit yang merata antara gelap dan terang. Sementara salak yang sudah terlalu matang memiliki sisik yang lebih besar dan warna yang cenderung cerah (Anam, 2018).

Memahami kriteria ini penting untuk memastikan bahwa salak yang terdistribusi tetap segar dan berkualitas.

Di Provinsi DI Yogyakarta buah salak juga merupakan komoditas utama bagi para penduduk setempat, khususnya di Kabupaten Sleman. Jika dilihat berdasarkan varietasnya, ada beberapa varietas salak yang diproduksi di Provinsi DI Yogyakarta, seperti salak pondoh, salak madu, salak gading, salak gula pasir, dan lain sebagainya (Abdurrahman, 2023). Dari beberapa varietas salak ini, dapat dibedakan berdasarkan rasa dan visualnya. Seperti salak pondoh yang memiliki bentuk yang lebih oval, salak madu memiliki kandungan air yang relatif banyak dan memiliki ukuran yang lebih besar, salak gading yang memiliki warna kulit lebih muda dan lain sebagainya.

Untuk membedakan macam-macam varietas salak dan kualitas salak, salah satunya dapat memanfaatkan teknologi *object detection* yang mana dapat memudahkan dalam pengidentifikasian varietas salak dan kualitas salak tersebut, khususnya dalam membedakan varietas salak dan kualitas salak yang dapat dilihat secara visual. Melalui penggunaan teknik *deep learning* dan analisis citra, teknologi *object detection* memungkinkan identifikasi cepat dan akurat untuk mengidentifikasi hal tersebut.

Deep learning merupakan teknik modern terbaru dalam pemrosesan gambar dan analisis data, dengan potensi besar dan hasil yang akurat. *Deep learning* telah berhasil diimplementasikan di berbagai bidang, dan akhir-akhir ini *deep learning* juga telah diimplementasikan di sektor pertanian (Kamilaris & Prenafeta-Boldu, 2018). *Deep Learning* merupakan sub bidang dari *machine learning* yang terinspirasi oleh struktur otak manusia (Purwins, dkk 2019). Sebelum *deep learning* muncul, *machine learning* telah ada sejak lama. *Machine learning* adalah suatu sistem komputer yang dapat memperoleh pengetahuan dari pengalaman, sehingga semakin banyak pengalaman yang diberikan ke dalam sistem *machine learning* tersebut, maka kinerjanya akan semakin meningkat. Namun, dalam kenyataannya, algoritma *machine learning* tidak akan mengalami peningkatan kinerja yang signifikan setelah mencapai tingkat tertentu, terlepas dari seberapa banyak data yang diberikan. Oleh karena itu, *deep learning* dikembangkan untuk

mengatasi permasalahan performa yang timbul pada algoritma *machine learning* ketika volume data semakin besar (Brownlee, 2020).

Deep learning telah mengalami kemajuan pesat dalam beberapa tahun terakhir dan telah memberikan kontribusi signifikan dalam berbagai bidang, termasuk *face recognition* dan *computer vision*. Salah satu contoh arsitektur *deep learning* yang bisa digunakan untuk melakukan klasifikasi gambar adalah *Visual Geometry Group-19* (VGG-19). VGG-19 adalah salah satu model *neural network* yang dikembangkan oleh tim peneliti dari *Visual Geometry Group* (VGG) di *Oxford University*. Sesuai namanya, model ini terdiri dari 19 lapisan termasuk lapisan konvolusi dan lapisan *fully connected* (Simonyan & Zisserman, 2015). Arsitektur VGG-19 terdiri dari serangkaian lapisan konvolusi dengan filter kecil (3x3) dan menggunakan aktivasi ReLU di antara lapisan-lapisannya yang diikuti oleh beberapa lapisan *fully connected* untuk klasifikasi akhir. Arsitektur ini telah menjadi salah satu referensi utama dalam pemrosesan citra dengan *deep learning*.

Dalam konteks produksi buah salak, penerapan VGG-19 dapat membantu dalam mendeteksi varietas buah salak dan mengidentifikasi kualitas buah salak secara akurat. Dengan menggunakan data citra dari berbagai varietas buah salak, VGG-19 dapat mempelajari pola visual yang berbeda pada citra tersebut. Sebagai hasilnya, sistem dengan menggunakan VGG-19 dapat mengidentifikasi potensi masalah produksi buah salak khususnya dalam membedakan varietas buah salak dan kualitas buah salak tersebut. Hal ini merupakan salah satu implementasi bagaimana teknologi *deep learning* dan arsitektur seperti VGG-19 dapat digunakan untuk meningkatkan produktivitas produksi buah salak dan pendistribusian buah salak ke pasar lokal maupun pasar ekspor.

Berdasarkan uraian latar belakang diatas, maka penulis akan melakukan penelitian dengan judul “Implementasi *Convolutional Neural Networks* Menggunakan Arsitektur *Visual Geometry Group-19* (VGG-19) untuk Klasifikasi Citra Buah Salak” dengan studi kasus klasifikasi citra buah salak asal DI Yogyakarta.

1.2. Rumusan Masalah

Berdasarkan pada uraian latar belakang di atas, terdapat beberapa persoalan utama yang berkaitan dengan implementasi *Convolutional Neural Networks* untuk

mengklasifikasikan citra buah salak menggunakan VGG-19, sehingga dapat dilakukan perumusan masalah sebagai berikut:

1. Bagaimana karakteristik visual yang membedakan varietas dan kualitas buah salak?
2. Bagaimana implementasi arsitektur VGG-19 dalam pengklasifikasian citra buah salak berdasarkan varietas dan kualitasnya?
3. Bagaimana hasil akurasi pengklasifikasian citra buah salak berdasarkan varietas dan kualitasnya?

1.3. Batasan Masalah

Adapun batasan masalah yang menjadi acuan penulis dalam penyusunan Tugas Akhir ini adalah:

1. Menggunakan data primer untuk bahan *training model*, sebagai *validation*, dan *testing data* dengan format .jpg.
2. Menggunakan sebanyak 1.200 *dataset* yang dibagi menjadi 80% data *train*, 10% data *validation*, dan 10% data *test*.
3. Pengkodean algoritma menggunakan Bahasa pemrograman Python.
4. Arsitektur yang digunakan merupakan model CNN VGG-19.
5. Buah salak diklasifikasikan menjadi enam kelas yaitu salak pondoh sehat, salak madu sehat, salak gading sehat, salak pondoh busuk, salak madu busuk, dan salak gading busuk.
6. Citra digital yang digunakan memuat satu objek buah salak dari setiap citra.
7. Objek yang akan diklasifikasikan berdasarkan busuk dan sehatnya dilihat berdasarkan yang nampak secara visualnya.

1.4. Tujuan Penelitian

Berdasarkan rumusan masalah yang telah dipaparkan, penelitian ini memiliki tujuan adalah sebagai berikut:

1. Untuk mengetahui karakteristik visual yang membedakan varietas dan kualitas buah salak.

2. Untuk mengetahui implementasi arsitektur VGG-19 dalam pengklasifikasian citra buah salak berdasarkan varietas dan kualitasnya.
3. Untuk mengetahui hasil akurasi pengklasifikasian citra buah salak berdasarkan varietas dan kualitasnya.

1.5. Manfaat Penelitian

Adapun manfaat yang bisa didapatkan dari penelitian ini adalah untuk mengembangkan informasi dan pengetahuan mengenai implementasi *deep learning Convolutional Neural Networks* untuk mengklasifikasikan gambar, khususnya citra buah salak dan kualitas buah salak menggunakan arsitektur VGG-19. Harapannya bisa membantu para produsen buah salak dalam mengidentifikasi dan klasifikasi buah salak berdasarkan varietas dan kualitasnya dengan cepat dan akurat. Sehingga dapat meningkatkan produktivitas dan kualitas hasil panen, serta memberikan solusi yang lebih efisien dalam pengelolaan sumber daya.

BAB II

TINJAUAN PUSTAKA

Penelitian terdahulu telah menjadi panduan dan acuan penulis dalam proses penulisan tugas akhir. Dengan mengacu pada hasil penelitian sebelumnya, penulis memiliki dasar yang kuat untuk mengembangkan penelitian tersebut. Harapannya, melalui penelitian ini, hasil-hasil yang telah ditemukan sebelumnya dapat diperluas, ditingkatkan, dan diimplementasikan dalam kerangka teori yang menjadi inti dari penelitian saat ini. Dengan cara ini, penelitian terdahulu tidak hanya berfungsi sebagai referensi, tetapi juga sebagai sumber inspirasi untuk mengembangkan pemahaman penulis tentang suatu topik dan meningkatkan kontribusi ilmiah yang dapat penulis berikan.

2.1. Penelitian Terdahulu

Mengacu pada Tabel 2.1, penelitian pertama ditulis oleh Simonyan dan Zisserman (2015) yang berjudul “*Very Deep Convolutional Networks for Large-Scale Image Recognition*”. Tujuan penelitian ini untuk mengetahui kedalaman konfigurasi layer *convolution* terhadap nilai akurasi yang didapatkan. Hasil dari penelitian ini adalah peningkatan akurasi yang signifikan dapat dicapai dengan konfigurasi mencapai 16-19 layer *convolution*.

Penelitian kedua ditulis oleh Rianto dan Harjoko (2017) yang berjudul “Penentuan Kematangan Buah Salak Pondoh di Pohon Berbasis Pengolahan Citra Digital”. Tujuan penelitian ini untuk mendeteksi kematangan buah salak dengan menggunakan citra digital buah salak tersebut. Hasil dari penelitian ini adalah dengan menggunakan metode *backpropagation* diperoleh nilai akurasi sebesar 92% dan menggunakan metode *k-Nearest Neighbor* memperoleh nilai akurasi sebesar 93%.

Penelitian ketiga ditulis oleh Xiao, dkk (2020) yang berjudul “*Application of a Novel and Improved VGG-19 Network in the Detection of Workers Wearing Masks*”. Tujuan penelitian ini untuk mendeteksi penggunaan masker saat Covid-19 di wilayah kantor dan padat penduduk dengan menggunakan metode VGG-19. Hasil dari penelitian ini adalah deteksi penggunaan masker ditempat kerja dan

daerah padat penduduk menggunakan VGG-19 menghasilkan nilai presisi model sebesar 97,62% dan *recall* sebesar 96,31%.

Penelitian keempat ditulis oleh Rahman (2020) yang berjudul “Implementasi *Convolutional Neural Networks* menggunakan VGG-19 Net pada *Image Depth Data Hand Posture*”. Tujuan penelitian ini adalah untuk mengetahui implementasi, tingkat akurasi, serta filter yang paling efektif dalam arsitektur VGG-19 menggunakan *Image Depth Data Hand Posture*. Hasil dari penelitian ini adalah dengan menggunakan model VGG-19 dapat mengklasifikasikan *Image Depth Data Hand Posture* dengan nilai akurasi sebesar 99,76%.

Penelitian kelima ditulis oleh Rismayanati dan Luthfiarta (2021) yang berjudul “Klasifikasi Kualitas Buah Salak dengan *Transfer Learning* Arsitektur VGG-16”. Tujuan penelitian ini adalah untuk membedakan kualitas buah salak dengan *machine learning*. Salak diklasifikasikan menjadi dua kelas, kelas bagus dan jelek. Hasil dari penelitian ini adalah akurasi tertinggi yang diperoleh dari penelitian ini adalah 95,83%. Akurasi ini diperoleh dengan menggunakan *learning rate* = 0,0001 dan *momentum* 0,9. Presisi dan perolehan model ini adalah 97,2 dan 94,6.

Penelitian keenam ditulis oleh Meena, dkk (2022) yang berjudul “*Sentiment Analysis from Images using VGG19 based Transfer Learning Approach*”. Tujuan penelitian ini adalah untuk membandingkan tiga metode klasifikasi dengan menggunakan dataset yang sama. Hasil dari penelitian ini adalah klasifikasi dengan menggunakan VGG-19 menghasilkan nilai akurasi paling tinggi yaitu sebesar 94% untuk dataset JAFFE, 65,41% untuk dataset FER2013, dan 99,47% untuk dataset CK+.

Penelitian ketujuh ditulis oleh Andrew dan Santoso (2022) yang berjudul “*Compare VGG19, ResNet50, Inception-V3 for review food rating*”. Tujuan penelitian ini adalah untuk mengklasifikasikan rating makanan yang dijual di aplikasi *online* berdasarkan gambar yang ditampilkan menggunakan tiga metode yang berbeda. Hasil dari penelitian ini adalah klasifikasi gambar makanan berdasarkan review rating di aplikasi *online* dengan menggunakan VGG-19 menghasilkan nilai akurasi sebesar 97,86%, dengan ResNet50 menghasilkan nilai akurasi 97,29%, dan Inception-v3 menghasilkan nilai akurasi sebesar 97,57%.

Penelitian kedelapan ditulis oleh Thomkaew dan Intaskosum (2022) yang berjudul “*Improvement Classification Approach in Tomato Leaf Disease using Modified Visual Geometry Group (VGG)-InceptionV3*”. Tujuan penelitian ini adalah untuk meningkatkan nilai akurasi dalam mengklasifikasikan penyakit pada daun tomat dengan menggunakan VGG-InceptionV3. Hasil dari penelitian ini adalah dengan menggunakan metode VGG-InceptionV3 mampu mengurangi jumlah parameter dan waktu komputasi serta memperoleh nilai akurasi sebesar 97,27%.

Penelitian kesembilan ditulis oleh Rizky (2023) yang berjudul “Klasifikasi Penyakit Pada Daun Padi Menggunakan Convolutional Neural Network (CNN) dengan Model VGG-19”. Tujuan penelitian ini adalah untuk membantu para petani dalam mengkategorikan penyakit-penyakit yang terdapat pada daun padi. Hasil dari penelitian ini adalah dengan menggunakan model VGG-19 dapat mengklasifikasikan penyakit daun padi dengan memperoleh nilai akurasi sebesar 98%.

Penelitian kesepuluh ditulis oleh Shinta, dkk (2023) yang berjudul “Klasifikasi Citra Penyakit Daun Tanaman Padi Menggunakan CNN dengan Arsitektur VGG-19”. Tujuan penelitian ini adalah untuk membandingkan hasil akurasi pengujian dari model yang menggunakan augmentasi dan tanpa augmentasi data. Hasil dari penelitian ini adalah dengan menggunakan arsitektur VGG-19 dan augmentasi data dapat meningkatkan nilai akurasi menjadi 94,31%.

Tabel 2.1 Tabel Penelitian Sebelumnya

Tahun	Nama	Judul	Hasil Penelitian
2015	Karen Simonyan dan Andrew Zisserman	<i>Very Deep Convolutional Networks for Large-Scale Image Recognition</i>	Peningkatan akurasi yang signifikan dapat dicapai dengan konfigurasi mencapai 16-19 layer <i>convolution</i> .
2017	Pawit Rianto dan Agus Harjoko	Penentuan Kematangan Buah Salak Pondoh di Pohon Berbasis Pengolahan Citra Digital	Kematangan buah salak di deteksi dengan menggunakan citra buah salak tersebut. Dengan menggunakan metode <i>backpropagation</i> diperoleh nilai akurasi sebesar 92% dan menggunakan metode <i>k-Nearest Neighbor</i> memperoleh nilai akurasi sebesar 93%.

Tahun	Nama	Judul	Hasil Penelitian
2020	Jian Xiao, Jia Wang, Shaozong Cao, and Bilong Li	<i>Application of a Novel and Improved VGG-19 Network in the Detection of Workers Wearing Masks</i>	Deteksi penggunaan masker di tempat kerja dan daerah padat penduduk menggunakan VGG-19 menghasilkan nilai presisi model sebesar 97,62% dan <i>recall</i> sebesar 96,31%.
2020	Fatur Rahman	Implementasi <i>Convolutional Neural Networks</i> menggunakan VGG-19 Net pada <i>Image Depth Data Hand Posture</i>	Model VGG-19 dapat mengklasifikasikan <i>Image Depth Data Hand Posture</i> dengan nilai akurasi sebesar 99,76%.
2021	Rismayanati dan Ardytha Luthfiarta	Klasifikasi Kualitas Buah Salak dengan Transfer Learning Arsitektur VGG-16	Dengan menggunakan arsitektur VGG-16 didapatkan nilai akurasi untuk membedakan kualitas salak yang bagus dan jelek sebesar 95,83 % dan nilai presisi sebesar 97,2.
2022	Gaurav Meena, Krishna Kumar Mohbey, Ajay Indian, dan Sunil Kumar	<i>Sentiment Analysis from Images using VGG19 based Transfer Learning Approach</i>	Klasifikasi dengan menggunakan VGG-19 menghasilkan nilai akurasi paling tinggi yaitu sebesar 94% untuk dataset JAFFE, 65,41% untuk dataset FER2013, dan 99,47% untuk dataset CK+.
2022	Andrew dan Handri Santoso	<i>Compare VGG19, ResNet50, Inception-V3 for review food rating</i>	Klasifikasi gambar makanan berdasarkan review rating di aplikasi <i>online</i> dengan menggunakan VGG-19 menghasilkan nilai akurasi sebesar 97,86%, dengan ResNet50 menghasilkan nilai akurasi 97,29%, dan Inception-v3 menghasilkan nilai akurasi sebesar 97,57%.
2022	Jiarpon Thomkaew and Sarun Intaskosum	<i>Improvement Classification Approach in Tomato Leaf Disease using Modified Visual Geometry Group (VGG)-InceptionV3</i>	Klasifikasi penyakit daun tomat dengan menggunakan model VGG-16 dan disempurnakan menggunakan model VGG-InceptionV3 menghasilkan nilai akurasi yang tinggi yaitu sebesar 97,27%
2023	Dyah Ayu Dewi Rizky	Klasifikasi Penyakit Pada Daun Padi Menggunakan <i>Convolutional Neural Network (CNN)</i> Dengan	Model VGG-19 dapat mengklasifikasikan penyakit daun padi dengan memperoleh nilai akurasi sebesar 98%.

Tahun	Nama	Judul	Hasil Penelitian
		Model VGG-19	
2023	Rahma Shinta, Jasril, Muhammad Irsyad, Febi Yanto, dan Suwanto Sanjaya	Klasifikasi Citra Penyakit Daun Tanaman Padi Menggunakan CNN dengan Arsitektur VGG-19	Dengan menggunakan arsitektur VGG-19 dan augmentasi data dapat meningkatkan nilai akurasi menjadi 94,31%.

Berdasarkan penelitian terdahulu yang disajikan pada **Tabel 2.1** diperoleh beberapa persamaan penelitian ini dengan penelitian terdahulu tersebut. Pertama, penelitian ini membahas tentang klasifikasi pada suatu citra. Kedua, implementasi metode *convolutional neural network* (CNN) menggunakan arsitektur *visual geometry group-19* (VGG-19) untuk mengklasifikasikan suatu citra. Ketiga mengklasifikasikan kualitas buah salak dengan *machine learning*.

Adapun perbedaan dari penelitian ini dengan penelitian terdahulu tersebut adalah, arsitektur CNN yang digunakan menggunakan arsitektur VGG-19, klasifikasi citra pada buah salak terdiri dari tiga varietas salak yaitu salak pondoh, salak madu, dan salak gading. Dari ketiga varietas salak ini juga dibedakan atas salak yang baik dan salak yang busuk. Dataset yang digunakan milik penulis atau merupakan data primer.

BAB III

LANDASAN TEORI

3.1 Buah Salak

Salacca zalacca dalam bahasa latin berarti buah salak, termasuk dalam buah tropis dan tergabung dalam famili *Palmae* yang memiliki kulit berduri dan daging buah yang manis. Kulitnya yang kasar dan berduri melindungi daging buahnya yang berwarna putih-transparan kekuningan. Buah salak memiliki biji yang besar di dalamnya. Rasa buah salak bisa bervariasi dari manis sampai sedikit masam tergantung pada varietasnya (Ashari, 2013).

Salak termasuk ordo Aricales, atau serumpun dengan kelapa. Memiliki ciri khas batang tidak terlihat karena tertutup oleh pelepah daun yang tersusun rapat dan berduri. Adapun pada batang yang berduri tersebut tumbuh bunga buah salak dengan jumlah yang relatif banyak (Haryanto & Priyanto, 2018).

Varietas buah salak yang tumbuh dan berkembang di Indonesia biasanya dibedakan dan diberi nama sesuai dengan daerah dimana buah salak tersebut tumbuh dan berkembang. Seperti salak bali yang berasal dari Kabupaten Karangasem Bali, salak condet berasal dari Kelurahan Condet Jakarta Timur, salak pondoh berasal dari Kabupaten Sleman DI Yogyakarta, dan lain sebagainya. Berikut ini beberapa varietas buah salak yang akan menjadi objek penelitian pada tugas akhir ini yang berasal dari Provinsi DI Yogyakarta.

1. Salak pondoh adalah salah satu varietas salak berasal dari Provinsi DI Yogyakarta. Salak pondoh memiliki ciri visual yaitu kulit berwarna coklat dengan susunan sisiknya tampak lebih rapat menciptakan penampilan yang khas. Bentuk buah salak pondoh ini bulat atau oval dengan ujung yang agak tumpul memberikan identitas yang mudah dikenali. Buah ini dikenal dengan daging buahnya yang manis, dan sedikit lebih renyah dibandingkan dengan varietas salak lainnya. Daging buahnya putih-transparan dengan biji besar didalamnya.



Gambar 3.1 Salak Pondoh

2. Salak madu juga berasal dari Provinsi DI Yogyakarta memiliki tingkat produktivitas yang tinggi dengan kualitas yang cukup baik, daging buahnya tebal dan teksturnya lembut, serta memiliki rasa manis yang khas seperti madu. Salak madu membedakan dirinya dari salak pondoh melalui ciri kulitnya yang memiliki sisik tersusun rapi, cenderung lebih renggang dan besar dengan membentuk garis lurus dari bagian bawah buah hingga ke ujung pada salah satu sisi buahnya. Daging salak madu memiliki kandungan air yang lebih banyak dibandingkan dengan salak lainnya. Kelebihan salak madu, di samping rasanya yang *juicy*, harganya relatif lebih mahal dan populasinya masih sedikit. Kekurangannya, market untuk salak madu belum terbentuk seperti salak pondoh.



Gambar 3.2 Salak Madu

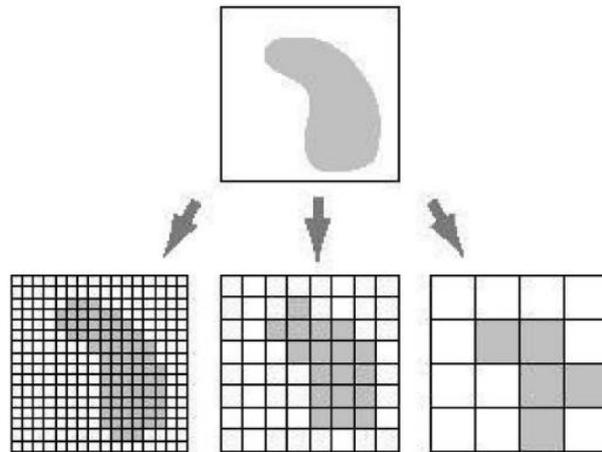
3. Salak gading yang juga berasal dari Provinsi DI Yogyakarta memiliki warna kulit yang lebih muda dibandingkan salak lainnya, yaitu berwarna gading atau coklat muda. Daging dari salak gading ini relatif tebal dan memiliki rasa yang khas yaitu sepat dan sedikit masam. Salak gading ini mempunyai banyak manfaat untuk kesehatan, salah satunya untuk asam urat.



Gambar 3.3 Salak Gading

3.1 Citra Digital

Citra adalah sebuah representasi visual, kesamaan, atau replikasi dari suatu objek. Citra dinyatakan sebagai sebuah fungsi dengan dua variabel misalnya $\alpha(x, y)$ di mana α menggambarkan tingkat amplitudo, seperti kecerahan, pada titik koordinat tertentu (Hendriyana & Maulana, 2020). Citra digital $\alpha[m, n]$ adalah hasil dari pengambilan sampel dari citra analog $\alpha(x, y)$ dalam ruang kontinyu 2D melalui suatu proses yang dikenal sebagai digitalisasi. Proses ini mengubah citra analog menjadi citra digital dalam ruang diskrit 2D. Dengan kata lain, citra digital $\alpha[m, n]$ merupakan representasi diskrit dari citra analog $\alpha(x, y)$ yang berasal dari ruang kontinyu 2D melalui proses sampling (Hidayatullah, 2017).



Gambar 3.4 Representasi Citra Digital (Haidaroh, 2016)

Dalam lingkup yang lebih umum, pengolahan citra digital adalah istilah yang merujuk pada manipulasi data 2D. Ini merupakan cabang ilmu yang memfokuskan pada teknik-teknik pengolahan citra, yang mencakup gambar diam (foto) maupun gambar bergerak (video). Aspek kunci dari pengolahan citra digital adalah bahwa proses manipulasi gambar dilakukan melalui perangkat komputer dalam format digital (Sanjaya & Ayub, 2020). Informasi yang terdapat dalam citra dapat dianalisis dengan mempertimbangkan atribut dimensi dan warna. Citra dapat direpresentasikan dalam bentuk matriks yang memiliki M baris N kolom, dan titik-titik perpotongannya disebut piksel atau elemen gambar. Piksel atau *picture element* merupakan komponen paling kecil dalam citra, yang memiliki dua parameter utama, yaitu posisi koordinat dan nilai warna. Citra digital diberikan dalam bentuk fungsi dua dimensi $f(x, y)$, di mana nilai yang terkandung pada x dan y mewakili koordinat serta tingkat kecerahan pada titik-titik tersebut (Haidaroh, 2016). Citra digital dalam bentuk matriks dapat ditulis sebagai berikut.

$$f(x, y) = \begin{bmatrix} f(0,0) & f(0,1) & \dots & f(0, N-1) \\ f(1,0) & f(1,1) & \dots & f(1, N-1) \\ \vdots & \vdots & \ddots & \vdots \\ f(M-1,0) & f(M-1,1) & \dots & f(M-1, N-1) \end{bmatrix} \quad (3.1)$$

Berdasarkan matriks di atas, citra dengan fungsi $f(x, y)$ dapat dituliskan ke dalam fungsi matematis adalah sebagai berikut (Kusmanto & Tompunu, 2011):

$$\begin{aligned} 0 &\leq x \leq M - 1 \\ 0 &\leq y \leq N - 1 \\ 0 &\leq f(x, y) \leq G - 1 \end{aligned}$$

di mana:

M : Jumlah piksel baris pada *array* citra

N : Jumlah piksel kolom pada *array* citra

G : Nilai skala keabuan (*grayscale*)

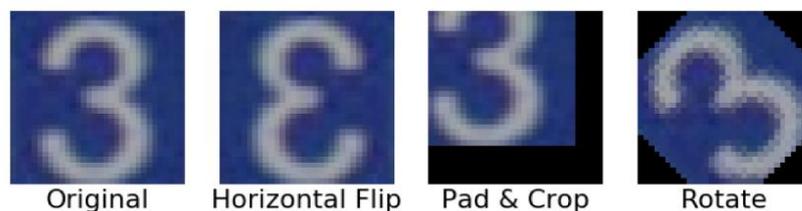
Nilai M, N dan G umumnya merupakan hasil dari pendekatan kuadrat, seperti yang tergambar dalam persamaan berikut:

$$M = 2m; N = 2n; G = 2k \quad (3.2)$$

Di mana m, n , dan k adalah bilangan positif dengan rentang $(0, G)$ dikenal sebagai skala keabuan (*grayscale*). Besar nilai G bergantung pada metode digitalisasi yang digunakan. Secara umum, keabuan 0 (nol) mewakili intensitas warna hitam, sementara 1 (satu) mewakili intensitas warna putih. Untuk citra 8 bit, nilai G setara dengan 2 pangkat 8, yaitu 256 warna (derajat keabuan) (Nurhikmat, 2018).

3.2 Augmentasi Data

Augmentasi data adalah strategi yang memungkinkan praktisi untuk secara signifikan meningkatkan variasi data yang dapat digunakan dalam melatih model, tanpa perlu mengumpulkan data tambahan. Teknik augmentasi data seperti *cropping*, *resizing*, *padding*, dan *flipping* seringkali digunakan dalam melatih jaringan saraf yang besar. Meskipun demikian, sebagian besar pendekatan yang digunakan dalam pelatihan jaringan saraf hanya memanfaatkan jenis augmentasi yang dasar, sementara arsitektur jaringan saraf telah mendapatkan perhatian mendalam.



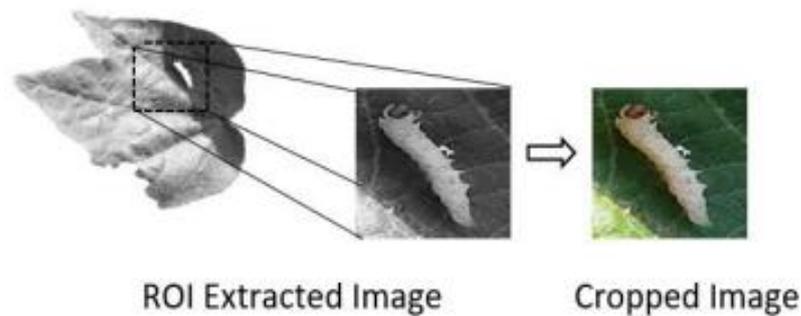
Gambar 3.5 Contoh Augmentasi Data (Ho, dkk 2019)

3.2.1 *Cropping*

Preprocessing data citra awal bertujuan untuk meningkatkan mutu citra dan mempersempit batasan pada bagian gambar yang ingin diamati, contohnya dengan menghilangkan gangguan seperti *noise*, meningkatkan ketajaman tepi objek, dan menghapus efek buram. *Noise* adalah informasi tidak berguna yang terdapat dalam citra dan tidak dibutuhkan dalam analisis (Yuhandri, 2019). *Noise* dapat

ditemukan di dalam area objek yang akan diamati atau pun di luar area objek yang diamati. Terutama untuk *noise* yang berada di luar wilayah objek yang sedang diamati, satu metode untuk menghilangkannya adalah dengan melakukan *cropping* (Saifullah, 2016).

Cropping adalah tindakan memperkecil dimensi citra dengan memotong citra pada koordinat tertentu di dalam citra (Saifullah, 2016). Dalam proses *cropping*, akan dihasilkan sebuah objek hasil pemotongan dari citra asli atau bagian dari gambar dengan dimensi tertentu. Proses *cropping* dapat dilakukan dengan menggunakan *library* OpenCV pada Python.



Gambar 3.6 Contoh *Cropping Image* (Kalambe, 2022)

3.2.2 *Resize*

Resize merupakan salah satu langkah dalam pemrosesan gambar yang melibatkan penyesuaian ukuran gambar agar sesuai dengan dimensi yang dapat diinterpretasikan oleh suatu metode. Metode tersebut mengharapkan gambar memiliki ukuran yang konsisten, oleh karena itu perlu dilakukan penyesuaian ukuran agar gambar tersebut dapat diolah. Dalam *resize*, gambar diperkecil atau diperbesar sesuai dengan kebutuhan analisis atau model yang akan digunakan (Lorentius, 2019).

Langkah *resizing* ini menjadi suatu prasyarat penting dalam rangka memenuhi persyaratan sistem. Karena metode yang digunakan dalam penelitian ini memerlukan *input* gambar dengan dimensi sebesar 224×224 piksel, untuk mengatasi hal tersebut proses *resizing* dapat dilakukan dengan memanfaatkan *library* OpenCV pada Python (Efendi, 2022). Sehingga semua gambar yang digunakan pada penelitian ini akan diubah ukurannya menjadi 224x224 piksel agar sesuai dengan persyaratan sistem.

3.2.3 *Padding dan Stride*

Padding adalah proses menambahkan dimensi piksel dengan nilai tertentu di sekitar data *input* untuk memastikan bahwa hasil dari bidang reseptif tidak menjadi terlalu kecil, sehingga menghindari kehilangan banyak informasi. Biasanya, nilai yang ditambahkan adalah nol, sehingga ini dikenal sebagai *zero padding*. Hasil dari bidang reseptif ini merupakan representasi tunggal dari informasi. Dimensi *output* dapat disesuaikan dengan dimensi *input* atau tidak mengalami banyak pengurangan. Penggunaan *padding* pada lapisan konvolusi dapat memberikan manfaat pada tingkat kedalaman dengan mengekstrak banyak fitur, yang pada gilirannya dapat meningkatkan kinerja model. Hasil dari konvolusi ini selanjutnya digunakan sebagai *input* untuk langkah berikutnya dalam proses yang dikenal sebagai langkah (*stride*) untuk ilustrasi susunan yang lebih besar (Sentosa, dkk 2022).

Stride adalah parameter yang menentukan seberapa banyak filter bergerak dari satu lokasi ke lokasi lainnya. Perpindahan piksel secara horizontal dan vertikal pada filter sesuai dengan nilai *stride* yang ditentukan. Semakin kecil nilai *stride*, informasi yang diperoleh akan semakin banyak, namun waktu komputasi akan meningkat. Meskipun penggunaan nilai *stride* yang kecil dapat menghasilkan lebih banyak informasi, tidak selalu menghasilkan performa model yang optimal (Yang, dkk 2023).

Pemilihan nilai *stride* yang tepat juga mempertimbangkan keseimbangan antara akurasi dan efisiensi komputasi. Pada beberapa kasus, penggunaan nilai *stride* yang lebih besar dapat mempercepat proses komputasi dengan mengorbankan sebagian informasi spasial. Oleh karena itu, dalam pengaturan model, perlu dilakukan eksperimen dan penyesuaian secara cermat untuk menentukan nilai *stride* yang optimal, sesuai dengan kebutuhan spesifik dataset yang dihadapi. Dengan demikian, pemahaman mendalam terhadap karakteristik *stride* dapat memberikan kontribusi signifikan dalam pengembangan model konvolusional yang efisien dan berkinerja tinggi (Sanjaya, 2020).

3.2.4 *Flipping dan Rotation*

Flipping merupakan salah satu teknik augmentasi data yang umum digunakan dalam pelatihan model *machine learning*. *Flipping* melibatkan pembalikan gambar secara horizontal atau vertikal. Secara khusus, *horizontal flipping* melibatkan pencerminan gambar secara horizontal, seperti mengubah gambar dari

kiri ke kanan atau sebaliknya, sedangkan *vertical flipping* melibatkan pencerminan gambar secara vertikal, mengubah gambar dari atas ke bawah atau sebaliknya. Penerapan *flipping* pada data augmentasi bertujuan untuk meningkatkan variasi dataset pelatihan, membantu model untuk lebih baik mengenali dan menangani variasi dalam data uji yang mungkin berbeda arah atau orientasi. Selain itu, *flipping* dapat membantu mengurangi *overfitting*, di mana model dapat terlalu spesifik terhadap data pelatihan.

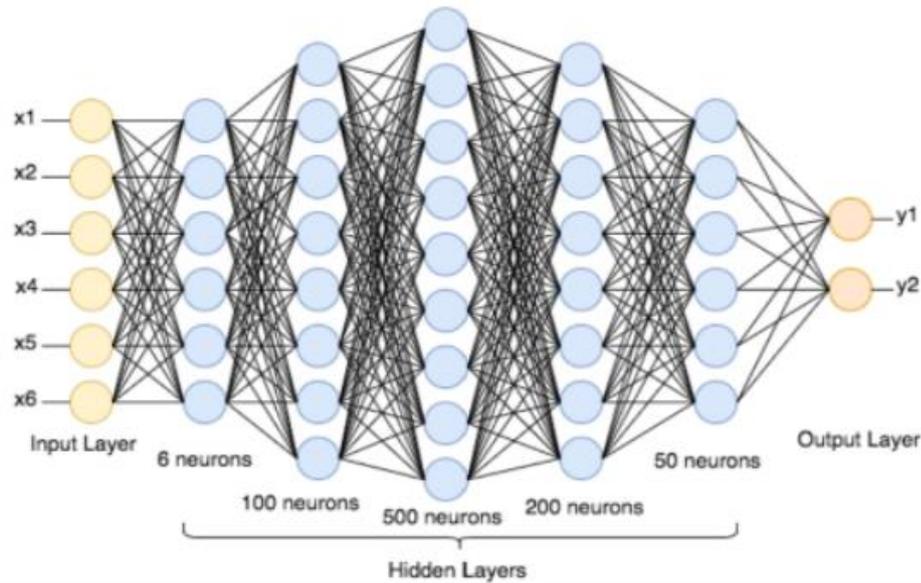
Rotation adalah teknik augmentasi data yang melibatkan putaran gambar sekitar sumbu tertentu. Umumnya *rotation* dilakukan secara acak untuk menciptakan variasi dalam dataset pelatihan. Penerapan *rotation* pada gambar membantu model untuk belajar dan beradaptasi dengan variasi orientasi objek dalam dunia nyata. Dengan mengintegrasikan *rotation* sebagai bagian dari strategi augmentasi data, model dapat menjadi lebih *robust* dan dapat mengenali objek dengan berbagai orientasi, memperbaiki kinerja generalisasi model pada data uji yang mungkin memiliki variasi orientasi yang tidak terlihat pada data pelatihan.

3.3 Deep Learning

Deep Learning merupakan sub bidang dari *machine learning* yang terinspirasi oleh struktur otak manusia (Purwins, dkk 2019). Dalam domain *machine learning* yang mengadopsi jaringan saraf tiruan sebagai model komputasi, yang terdiri dari sejumlah lapisan proses yang digunakan untuk mengekstrak nilai-nilai dari data yang digunakan, konsep ini dikenal sebagai *Deep Learning*. Salah satu elemen sentral dalam *machine learning* adalah pemanfaatan *Feature Engineering*, yang berfungsi untuk mengekstrak pola dari data dengan tujuan mempermudah identifikasi perbedaan di antara kelas-kelas data tersebut. Di dalam *deep learning*, lapisan-lapisan pada tingkat awal akan mempelajari pola-pola yang umum pada dataset, dan semakin spesifik ketika mencapai lapisan yang lebih dalam (Rochman & Junaedi, 2020).

Deep learning merupakan teknik modern terbaru dalam pemrosesan gambar dan analisis data, dengan potensi besar dan hasil yang akurat. Arsitektur dalam *deep learning* juga merupakan alat yang sangat efektif dalam konteks *Supervised Learning*. Dalam *supervised learning*, arsitektur ini mampu mengenali dan memahami hubungan antara *input data* dan *output* yang diinginkan, memungkinkan model untuk melakukan prediksi yang tepat berdasarkan pola yang telah dipelajari. Dengan demikian, *deep learning* adalah pendekatan yang

sangat kuat dan bermanfaat dalam pengawasan pembelajaran yang berkaitan dengan arsitektur jaringan saraf tiruan yang kompleks (Kamilaris & Prenafeta-Boldu, 2018).



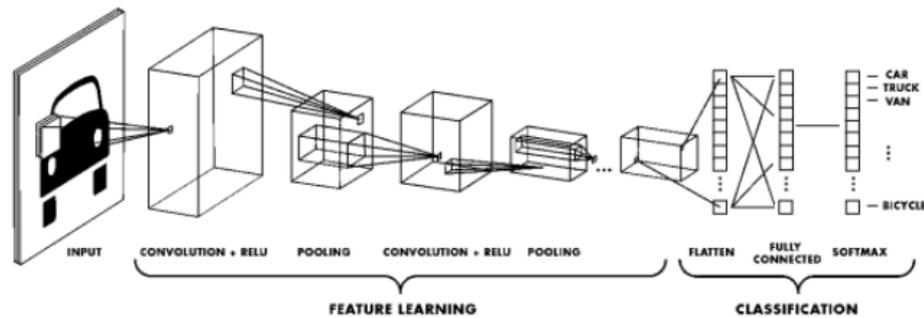
Gambar 3.7 Arsitektur Penyusun *Deep Learning* (Bahi & Moamed, 2018)

Arsitektur ini adalah kerangka dasar yang membentuk jaringan saraf tiruan, yang terdiri dari berbagai lapisan dan *neuron*. Setiap lapisan memiliki peran spesifik dalam pemrosesan data, dan setiap koneksi antar *neuron* memiliki bobot dan bias yang disesuaikan. Fungsi aktivasi memberikan komponen non-linear ke dalam jaringan, sementara fungsi nilai *error* digunakan untuk mengukur sejauh mana prediksi jaringan berkaitan dengan target yang diinginkan.

3.4 *Convolutional Neural Network (CNN)*

Convolutional Neural Network (CNN) adalah suatu algoritma yang beroperasi dengan menerima gambar sebagai *input*. CNN merupakan suatu metode dalam bidang *machine learning* yang berkembang dari *MultiLayer Perceptron (MLP)*, dirancang khusus untuk mengolah data berdimensi 2D, seperti gambar dan citra. Dengan menggunakan filter dan lapisan konvolusi, CNN dapat mengekstrak fitur-fitur penting dari gambar dan memahami pola-pola yang ada di dalamnya, menjadikannya sangat efektif dalam tugas-tugas pengenalan gambar dan pemrosesan visual (Hawari, dkk 2022). *Convolutional Neural Networks (CNN)* termasuk dalam kategori *Deep Neural Network* karena sering digunakan dalam pengolahan data citra dan memiliki arsitektur jaringan yang sangat dalam. Keunggulan utama CNN dalam pemrosesan citra terletak pada kemampuannya

untuk memahami hierarki fitur-fitur visual dalam citra dengan berbagai lapisan konvolusi. Dengan memanfaatkan kedalaman jaringan yang tinggi, CNN mampu mengenali pola-pola kompleks dan informasi detail dalam citra, membuatnya sangat efektif dalam tugas-tugas seperti klasifikasi gambar, deteksi objek, dan pengenalan wajah (Cahya, dkk 2021).



Gambar 3.8 Arsitektur *Convolutional Neural Network* (CNN) (Hendriyana & Maulana, 2020)

Dalam struktur arsitektur CNN, terdapat serangkaian proses yang cukup kompleks sebelum akhirnya mencapai tahap klasifikasi gambar. Secara keseluruhan CNN melibatkan dua tahap utama, yaitu ekstraksi fitur (*feature extraction*) dan *fully connected layer*.

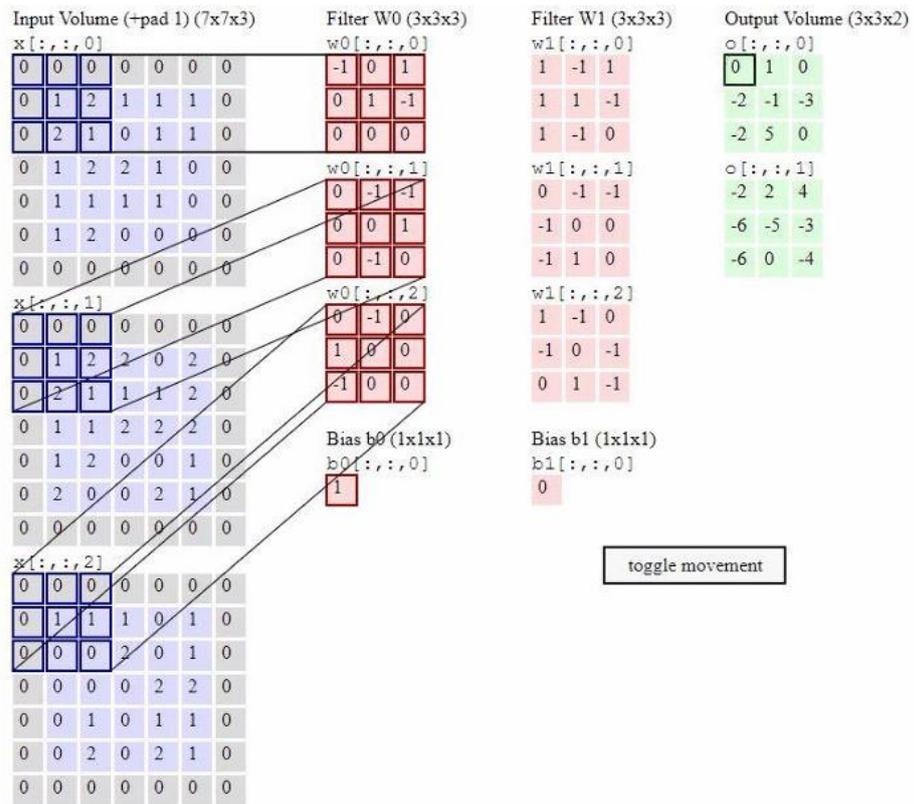
3.4.1 *Feature Extraction*

Ekstraksi fitur dalam *Convolutional Neural Networks* (CNN) adalah proses di mana CNN mengidentifikasi dan mengekstrak fitur-fitur penting dari gambar atau data visual. Dalam proses ini, lapisan-lapisan yang membentuk tahap ekstraksi fitur bekerja bersama untuk mengidentifikasi dan mengekstraksi fitur-fitur yang unik dari gambar. Dalam tahap ekstraksi fitur, terdapat beberapa komponen penting yang berperan dalam mengolah gambar dan mengungkap informasi penting, seperti tepi, tekstur, atau pola (Hidayat & Putra, 2019). Berikut ini adalah penjelasan lebih rinci mengenai lapisan-lapisan yang membentuk tahap ekstraksi fitur.

1. *Convolutional Layer*

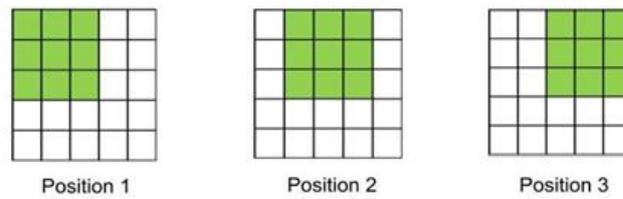
Lapisan yang paling terkenal dalam CNN adalah lapisan konvolusi (*convolutional layer*). Bagian ini adalah *layer* pertama pada algoritma CNN dan berperan dalam melakukan operasi konvolusi pada data *input* dari lapisan sebelumnya dengan menggunakan filter yang digeser untuk menghasilkan *feature map* (Putra, dkk 2019). *Convolutional layer* adalah

filter yang terdiri dari *neuron-neuron* dengan dimensi pixel yang memiliki panjang dan tinggi tertentu. Filter ini digeser melalui seluruh area gambar, dan pada setiap perpindahan, terjadi operasi perkalian titik antara *input* dan filter (Hendriyana & Maulana, 2020).



Gambar 3.9 Operasi *Input* dan Filter CNN (Dharmadi, 2018)

Berdasarkan Gambar 3.9 untuk menghasilkan *feature map*, langkah pertama adalah melakukan operasi perkalian matriks antara citra *input* dan filter. Proses perkalian dimulai dari sudut kiri atas dan berlanjut ke arah kanan bawah citra. Hasil dari setiap perkalian ini akan diakumulasikan untuk membentuk elemen baru pada *feature map*. Setelah menyelesaikan satu baris dari kiri atas, filter akan digeser ke arah kanan. Jumlah pergeseran ini ditentukan oleh parameter "*stride*". Jika *stride* bernilai 1, maka filter akan bergeser satu pixel ke kanan hingga selesai, dan kemudian akan beralih ke pergeseran vertikal ke bawah untuk melanjutkan operasi serupa



Gambar 3.10 *Stride Convolutional Layer* (Chen, 2017)

Jika ukuran *stride* tidak sesuai dengan ukuran matriks *input*, maka dapat dilakukan penambahan *padding* pada matriks *input*. *Padding* adalah parameter yang mengontrol jumlah piksel yang akan ditambahkan pada sisi matriks *input* (Learning, 2020). Tujuannya adalah untuk memanipulasi dimensi *output* dari *feature map* agar dimensinya sama dengan *input*. *Convolutional layer* akan mengekstrak pola-pola dari sebuah gambar dan mewakili pola-pola tersebut dalam bentuk sejumlah himpunan angka. Proses konvolusi dapat dijelaskan dengan persamaan berikut. Sebagai contoh jika memiliki matriks *input* 5x5 dan ingin menerapkan operasi konvolusi dengan filter 3x3 dengan menggunakan *stride* sebesar 2, maka akan menghasilkan *feature map* berukuran 2x2. Namun, jika menyertakan *zero padding* sebanyak 1, maka *feature map* yang dihasilkan akan memiliki berukuran 3x3 (Lina, 2019). Untuk menghitung dimensi *feature map* dapat menggunakan persamaan berikut.

$$Output = \frac{W - F + 2p}{S} + 1 \quad (3.3)$$

di mana:

W : Ukuran gambar

F : Ukuran filter/kernel

P : *Padding*

S : *Stride*

2. *Rectified Linear Unit* (ReLU)

Fungsi aktivasi non-linear yang sering digunakan dalam jaringan saraf multilayer atau jaringan saraf mendalam adalah *Rectified Linear Unit* (ReLU). Biasanya, dalam *Convolutional Neural Network* (CNN), *Rectified Linear Unit* (ReLU) digunakan sebagai fungsi aktivasi untuk menghindari hilangnya gradien dan meningkatkan kecepatan pelatihan (Qiu, dkk 2017). Fungsi ReLU didefinisikan sebagai:

$$f(x) = \max(0, x) \quad (3.4)$$

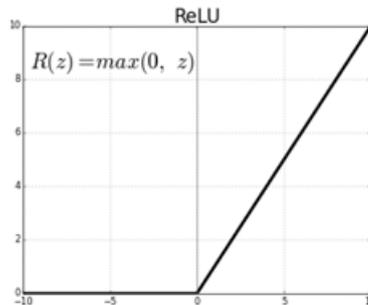
Output dari fungsi ReLU adalah nilai maksimum antara nol dan nilai *input* yang diberikan. Ketika nilai *input* bernilai negatif, maka *output* menjadi nol, sementara jika nilai *input* positif, maka *output* akan sama dengan nilai *input* tersebut. Oleh karena itu, persamaan dapat disederhanakan menjadi:

$$f(x) = \begin{cases} 0, & x < 0 \\ x, & x \geq 0 \end{cases} \quad (3.5)$$

di mana:

x : *Input* dari konvolusi

Sehingga akan membentuk grafik sebagai berikut:



Gambar 3.11 Fungsi Aktivasi ReLU (Sanjaya & Ayub, 2020)

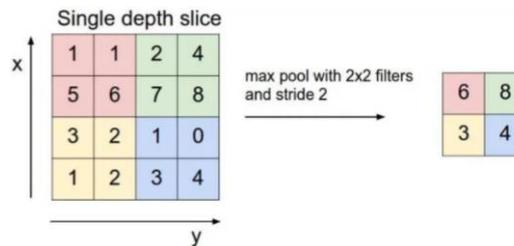
Gambar 3.11 merupakan representasi grafis dari fungsi ReLU. Berdasarkan gambar tersebut, bahwa nilai untuk setiap *input* x negatif akan menghasilkan *output* 0. Namun, jika *input* x positif maka fungsi ini menjadi aktif dan mengembalikan nilai *input* itu sendiri sebagai *output*-nya. Dengan kata lain, ReLU adalah fungsi yang aktif hanya ketika nilainya positif.

3. *Pooling Layer*

Langkah berikutnya dalam arsitektur CNN adalah *pooling layer*. *Pooling* adalah komponen yang umumnya terdapat dalam struktur CNN. Konsep utama di balik *pooling layer* adalah untuk mengumpulkan fitur yang dihasilkan oleh filter dengan cara tertentu di atas gambar (Dharmadi, Mengenal Convolutional Neural Network, 2018). *Pooling layer* biasanya ditempatkan setelah lapisan konvolusi. Secara esensial, *pooling layer* terdiri dari suatu filter dengan ukuran tertentu yang digerakkan melintasi seluruh area *feature map* dengan jarak tertentu. Dua jenis *pooling* yang

umum digunakan adalah *Max Pooling* dan *Average Pooling*. Tujuan penggunaan *pooling layer* adalah mengurangi dimensi dari *feature map* (*downsampling*) dengan tujuan untuk mempercepat komputasi, karena jumlah parameter yang perlu diperbarui menjadi lebih sedikit, dan juga untuk mengatasi masalah *overfitting* (Lina, 2019).

Suatu hal yang paling penting dalam merancang model CNN adalah memilih jenis *pooling layer* yang sesuai. Pemilihan ini dapat signifikan mempengaruhi kinerja model. *Pooling layer* bekerja pada setiap tumpukan *feature map* dan bertujuan untuk mengurangi dimensinya (Yu Lee, dkk 2015). Contoh umum dari *pooling layer* adalah menggunakan filter berukuran 2×2 dengan pergeseran (*stride*) sebesar 2, yang diterapkan ke setiap bagian dari *input*. Dalam contoh ini, *pooling layer* dapat mengurangi ukuran *feature map* hingga 75% dari ukuran aslinya. Operasi *Max Pooling* adalah salah satu contoh yang umum digunakan dalam proses ini.



Gambar 3.12 *Max Pooling* (Sanjaya & Ayub, 2020)

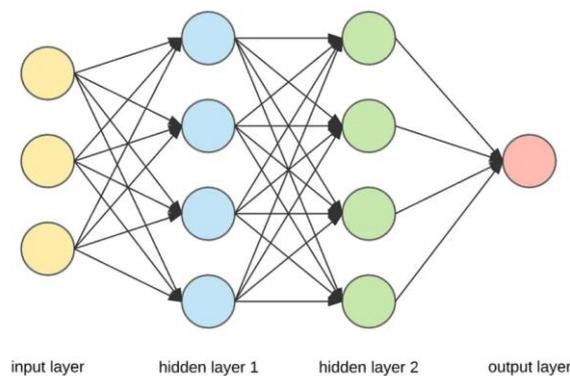
Pooling layer akan melakukan operasinya secara berurutan pada setiap lapisan kedalaman dari volume *input*. Seperti yang terlihat Gambar 3.12, *pooling layer* menggunakan salah satu metode yang paling umum, yaitu operasi maksimum. Dapat dilihat operasi ini dilakukan dengan *stride* sebesar 2 dan menggunakan filter berukuran 2×2 . Dari *input* berukuran 4×4 , diambil nilai maksimal pada setiap set keempat angka dalam *input*, yang kemudian menghasilkan *output* baru berukuran 2×2 .

3.4.2 Fully Connected Layer

Fully Connected Layer adalah jenis lapisan di mana setiap *neuron* aktivasi dari lapisan sebelumnya memiliki koneksi langsung ke semua *neuron* di lapisan berikutnya. Sebelum dapat dihubungkan ke semua *neuron* dalam lapisan *Fully Connected*, setiap aktivitas dari lapisan sebelumnya harus diubah menjadi representasi satu dimensi. Secara umum, lapisan ini sering digunakan dalam

Multilayer Perceptron (MLP) dengan tujuan untuk mengubah dimensi data sehingga data dapat diklasifikasikan secara linear (Fuadah, dkk 2022).

Tugas utama dari *Fully Connected Layer* adalah memproses data agar dapat dijadikan objek klasifikasi. Kelebihan utama dari *Fully Connected Layer* adalah kemampuannya untuk menghubungkan semua *neuron* secara menyeluruh. Sebelum data diproses oleh lapisan *Fully Connected*, langkah awalnya adalah mengubah matriks menjadi vektor, proses ini sering disebut sebagai "flatten". Flatten mengubah bentuk matriks menjadi vektor tunggal. Hasil dari lapisan ini akan menjadi *input* bagi *output layer* (Lina, 2019). Komponen-komponen dalam pembentukan jaringan saraf terdiri dari *input layer*, *hidden*, dan *output*.



Gambar 3.13 *Neural Network Layer* (Ognjanovski, 2019)

Hidden layer sering kali terdiri dari sejumlah besar lapisan yang digunakan dalam pelatihan model yang lebih kompleks. *Neuron-neuron* di lapisan tersebut menggunakan fungsi aktivasi untuk memproses nilai *input* x dengan mengalikan koefisien bobot w , dan menghasilkan nilai *output* z (Kimura, dkk 2019). Variabel-variabel ini dijelaskan sebagai berikut.

$$y_j = b_j + \sum_{i=1}^n w_{i,j}x_i \quad (3.6)$$

$$z_j = f(y_j) \quad (3.7)$$

di mana:

y_j : *Input hidden layer ke-j*

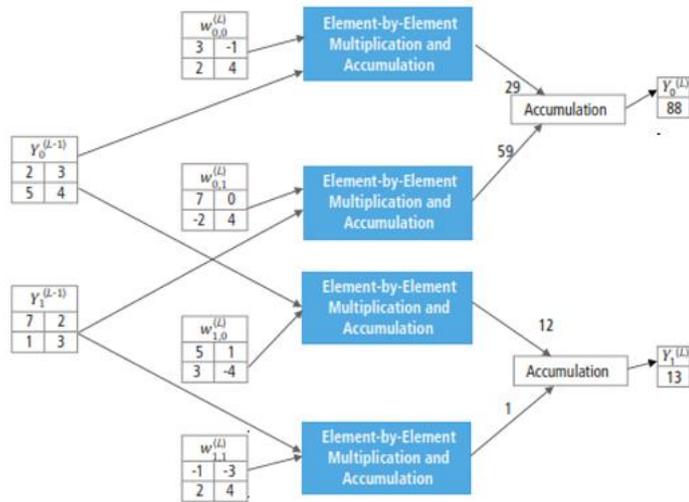
b_j : *Baris ke-j*

w : *Bobot*

x : *Input*

z : *Output hidden layer*

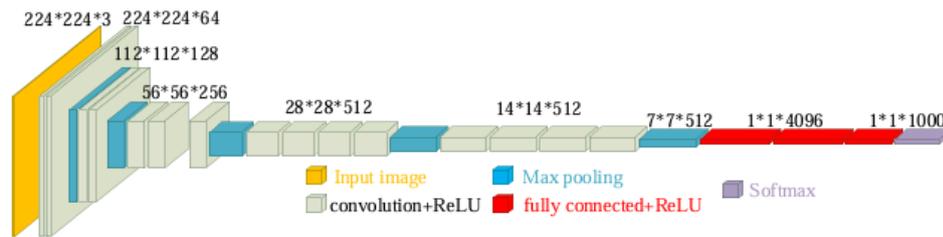
Perbedaan antara *Fully Connected Layer* dan *Connected Layer* biasa terletak pada koneksi antara *neuron*. *Connected Layer* hanya terhubung ke area tertentu pada *input*, sementara *Fully Connected Layer* memiliki koneksi yang menyeluruh. Meskipun demikian, baik *Connected Layer* maupun *Fully Connected Layer* tetap menggunakan operasi produk titik (*dot product*), sehingga prinsip dasar fungsinya tidak jauh berbeda. Berikut adalah ilustrasi proses *Fully Connected Layer*.



Gambar 3.14. Proses *Fully Connected Layer* (Nurhikmat, 2018)

3.5 Visual Geometry Group-19 (VGG-19)

VGG-19 adalah salah satu model *neural network* yang dikembangkan oleh tim peneliti dari *Visual Geometry Group* (VGG) di *Oxford University* Sesuai namanya, model ini terdiri dari 19 lapisan (Simonyan & Zisserman, 2015) yang terdiri dari 16 lapisan *convolutional layer* dengan filter kecil (3x3) dan aktivasi ReLU, serta 3 lapisan *fully connected layer* juga dengan aktivasi ReLU. Pada setiap lapisan *convolutional* terakhir, terdapat lapisan *max pooling* yang digunakan untuk mereduksi dimensi spasial dari lapisan *convolutional* sebelumnya. Lapisan *input* menggunakan resolusi gambar 224 x 224 piksel, dan lapisan *output* menggunakan fungsi aktivasi *softmax* untuk menampilkan hasil klasifikasi (Shinta, dkk 2023). Berikut ini gambar arsitektur VGG-19.



Gambar 3.15 Arsitektur VGG-19 (Xiao, dkk 2023)

Berikut penjelasan dari setiap lapisan arsitektur VGG-19

1. Lapisan *input*: Lapisan *input* CNN terdiri dari kumpulan data. Data *input* akan dijadikan representasi dalam bentuk matriks 3x3
2. Lapisan konvolusi: Lapisan ini memanfaatkan filter untuk mempelajari informasi dari bagian data *input* yang lebih kecil, dengan tujuan untuk mengekstrak fitur dari gambar tersebut. Berikut ini contoh operasi matriks lapisan konvolusi untuk memfilter matriks input data dengan menggunakan matriks identitas 2x2. Sehingga diperoleh output dengan matriks yang lebih kecil.

$$\begin{bmatrix} 1 & 1 & 1 & 3 \\ 4 & 6 & 4 & 8 \\ 30 & 0 & 1 & 5 \\ 0 & 2 & 2 & 4 \end{bmatrix} \otimes \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 7 & 5 & 9 \\ 4 & 7 & 9 \\ 32 & 2 & 5 \end{bmatrix}$$

Gambar 3.16 Operasi Matriks Lapisan Konvolusi (Parhusip, 2021)

Cara kerja matriks lapisan operasi konvolusi melibatkan proses konvolusi antara matriks input data dan matriks filter. Proses ini dilakukan dengan mengalikan matriks input data dengan matriks filter secara bergantian dan bergerak ke arah kanan.

3. *Pooling layer*: Fungsi dari lapisan ini adalah untuk mengurangi dimensi gambar, sehingga dapat mengurangi daya pemrosesan yang diperlukan untuk lapisan selanjutnya.
4. *Max pooling*: Dalam proses penguraian input, yang dipilih adalah piksel dengan nilai maksimum, dan piksel ini akan disalurkan ke dalam output. Pendekatan ini adalah yang sering digunakan dibandingkan dengan metode pooling rata-rata.
5. *Flatten*: Hasil *pooling* diratakan ke FC

6. *Fully Connected Layer*: Lapisan ini terletak di bagian akhir dari CNN dan memiliki kemampuan untuk mengidentifikasi fitur yang memiliki hubungan yang kuat dengan kelas *output*. Hasilnya adalah sebuah vektor satu dimensi yang dihasilkan dengan meratakan *output* dari lapisan pooling.
7. *Softmax layer*: Lapisan akhir dari jaringan yang berperan dalam pengklasifikasian gambar *input* individu dari sekumpulan data ke dalam berbagai kelas, tergantung pada karakteristik yang telah dipelajari oleh jaringan.
8. Lapisan *output*: Yang menyimpan hasil klasifikasi akhir (Thakur, 2019).

3.6 Optimization Algorithm

Optimization Algorithm (Algoritma Optimisasi) adalah algoritma untuk menemukan nilai x sehingga dapat menghasilkan $f(x)$ yang bernilai minimum atau maksimum untuk fungsi f yang diberikan. Masalah optimisasi muncul dalam beragam disiplin ilmu, seperti matematika, ilmu komputer, ilmu data, ekonomi, dan berbagai bidang lainnya (Hasad, 2018). Tujuan dari *Optimization Algorithm* adalah untuk menemukan solusi terbaik dalam konteks pembatasan yang ada. Solusi optimal dapat berupa nilai yang memaksimalkan atau meminimalkan suatu fungsi objektif, seperti memaksimalkan profit bisnis, meminimalkan biaya produksi, atau menemukan parameter model yang menghasilkan prediksi terbaik dalam *machine learning* (Hastomo, dkk 2022).

Di dalam *Convolutional Neural Network* (CNN) juga terdapat algoritma optimasi yang digunakan untuk menyesuaikan parameter dalam jaringan saraf konvolusi selama proses pelatihan. Tujuannya adalah untuk mencari nilai-parameter yang menghasilkan kinerja terbaik dalam tugas klasifikasi atau tugas lain yang dihadapi oleh CNN. Berikut ini beberapa algoritma optimasi yang dapat digunakan pada CNN:

1. *Stochastic Gradient Descent* (SGD)

Algoritma optimasi yang digunakan dalam *machine learning* dan *deep learning* untuk meminimalkan fungsi kerugian (*loss function*) saat melatih model. SGD digunakan dalam pelatihan model, termasuk dalam jaringan saraf seperti *Convolutional Neural Networks* (CNN) dan *Recurrent Neural Networks* (RNN) dengan dataset yang besar. Biasanya digunakan dalam pengenalan gambar, pemrosesan bahasa alami, dan berbagai aplikasi

machine learning lainnya. SGD mengatasi *Gradient Descent* dengan melakukan pembaruan parameter menggunakan hanya satu rekaman. Tetapi SGD memiliki konvergensi yang lambat karena memerlukan propagasi maju dan mundur untuk setiap rekaman (Mayanglambam, 2020).

2. *Adaptive Moment Estimation (ADAM)*

Algoritma optimasi yang digunakan dalam *machine learning*, khususnya dalam pelatihan model *neural networks* dan *deep learning*. Adam adalah algoritma yang menggabungkan konsep dari algoritma optimasi lain seperti Momentum dan RMSprop, yang memungkinkan penyesuaian adaptif dari tingkat pembelajaran dan memberikan konvergensi yang cepat (Tilawah, 2020). Adam memiliki perbedaan dengan *stochastic gradient descent*, *stochastic gradient descent* mempertahankan *single learning rate* (alpha) yang sama untuk seluruh pembaruan parameter, dan tingkat pembelajaran ini tetap tidak berubah selama proses pelatihan. Algoritma menghitung *learning rate* yang disesuaikan secara adaptif untuk setiap parameter berdasarkan perkiraan dari momen pertama dan kedua dari gradien (Mayanglambam, 2020).

3.7 Matrik Evaluasi Klasifikasi

Matriks evaluasi klasifikasi adalah matriks yang digunakan untuk mengukur kinerja dari sebuah model klasifikasi dalam *machine learning*. Matriks ini menyajikan informasi mengenai berapa banyak prediksi benar dan salah yang dibuat oleh model terhadap suatu dataset. Matrik evaluasi yang biasa digunakan dalam pemodelan klasifikasi adalah sebagai berikut:

3.7.1 *Confusion Matrix*

Confusion matrix sering disebut sebagai *error matrix*, pada dasarnya memberikan perbandingan antara hasil klasifikasi yang diberikan oleh sistem (model) dengan hasil klasifikasi yang sebenarnya. *Confusion matrix* disajikan dalam tabel matriks yang menggambarkan kinerja model klasifikasi pada dataset pengujian di mana nilai sebenarnya diketahui. Berikut adalah gambar *confusion matrix* yang menampilkan empat kombinasi nilai prediksi dan nilai aktual yang berbeda.

		Actual Values	
		1 (Positive)	0 (Negative)
Predicted Values	1 (Positive)	TP (True Positive)	FP (False Positive) <i>Type I Error</i>
	0 (Negative)	FN (False Negative) <i>Type II Error</i>	TN (True Negative)

Gambar 3.17 *Confusion Matriks* (Nugroho, 2019)

Berdasarkan Gambar 3.17 terdapat beberapa nilai dalam matriks, termasuk "True Positive" (TP) yaitu data positif yang diprediksi benar oleh model, "True Negative" (TN) yaitu data negatif yang diprediksi benar oleh model, "False Positive" (FP) data negatif namun diprediksi sebagai data positif oleh model, dan "False Negative" (FN) data positif namun diprediksi sebagai data negatif oleh model. *False Positive* (FP) biasa disebut sebagai *type I error* dan *False Negative* (FN) disebut sebagai *type II error*. Nilai-nilai yang sudah didapatkan pada *confusion matrix* ini dapat digunakan untuk menghitung akurasi, presisi, *recall*, dan *f1-score*.

3.7.2 Akurasi

Akurasi adalah salah satu metrik evaluasi yang digunakan untuk mengukur kinerja model klasifikasi dalam *machine learning*. Ini mengukur sejauh mana model klasifikasi mampu melakukan prediksi yang benar dibandingkan dengan total jumlah prediksi yang dibuat.

Secara matematis, akurasi dihitung dengan rumus berikut:

$$Akurasi = \frac{TP + TN}{P + N} \quad (3.8)$$

Akurasi memberikan gambaran tentang sejauh mana model mampu mengklasifikasikan data dengan benar (Afifah, 2023).

3.7.3 Presisi

Presisi adalah salah satu metrik evaluasi yang digunakan dalam klasifikasi dalam *machine learning*. Presisi mengukur sejauh mana model klasifikasi mampu mengidentifikasi secara benar kasus positif dari semua kasus yang diprediksi sebagai positif.

Secara matematis, akurasi dihitung dengan rumus berikut:

$$Precision = \frac{TP}{TP + FP} \quad (3.9)$$

Presisi mengindikasikan sejauh mana model dapat memberikan prediksi yang tepat untuk peristiwa positif dalam rangkaian prediksi. Selain mempertimbangkan presisi dan akurasi, untuk melihat lebih mendalam tentang kinerja sistem, kita juga dapat mempertimbangkan *recall* atau sensitivitas sistem terhadap kelas tertentu (Nugroho, 2019).

3.7.4 Recall (Sensitivitas)

Recall adalah metrik yang relevan dalam situasi untuk menghindari *missed detection* atau gagal mendeteksi kasus positif yang sebenarnya. Misalnya, dalam sistem deteksi penyakit atau keamanan, tingginya *recall* adalah kunci untuk meminimalkan risiko melewatkan kasus positif yang penting.

Secara matematis, *recall* dihitung dengan rumus berikut:

$$Recall = \frac{TP}{TP + FN} \quad (3.10)$$

Recall mengukur kemampuan model untuk mengingat atau mendeteksi sebagian besar kasus positif. Namun, perlu diingat bahwa *recall* seringkali memiliki *trade-off* dengan presisi. Meningkatkan *recall* mungkin mengorbankan presisi, dan sebaliknya. Oleh karena itu, penting untuk memahami dan mempertimbangkan keseimbangan antara *recall* dan presisi tergantung pada kebutuhan spesifik dari aplikasi klasifikasi (Arthana, 2019).

3.7.5 F1-score

F1-score adalah matrik evaluasi yang mengukur kinerja model klasifikasi dalam *machine learning* dengan memadukan *precision* dan *recall*. *F1-score* memberikan gambaran tentang keseimbangan antara kemampuan model untuk mengidentifikasi kasus positif secara tepat (*precision*) dan kemampuan model untuk mengidentifikasi semua kasus positif yang sebenarnya (*recall*). Sehingga *f1-score* didefinisikan dalam bentuk matematis sebagai berikut.

$$F1 - Score = \frac{2 \times Presisi \times Recall}{Presisi + Recall} \quad (3.11)$$

F1-score sangat berguna dalam konteks klasifikasi untuk menjaga presisi yang tinggi tetapi juga memastikan bahwa sebagian besar kasus positif yang sebenarnya dapat diidentifikasi (Nugroho, 2019).

BAB IV METODOLOGI PENELITIAN

4.1. Populasi dan Sampel Penelitian

Populasi dalam penelitian ini adalah citra buah salak yang berasal dari Kabupaten Sleman, Provinsi DI Yogyakarta. Data diambil pada bulan Desember 2023 dengan cara memotret langsung objek tersebut. Adapun sampel yang digunakan pada penelitian ini adalah citra buah salak dengan tiga jenis atau tiga varietas yaitu salak pondoh, salak madu, dan salak gading. Dari ketiga varietas buah salak ini dibedakan atas kualitas visual nya yaitu salak yang sehat (bagus dan matang) dan salak yang sudah busuk (terlalu matang). Sehingga pada penelitian ini akan terdapat 6 *class* klasifikasi buah salak. Data yang akan digunakan pada penelitian ini merupakan data primer (milik peneliti). Total citra yang digunakan untuk sampel ini sebanyak 1.200 citra, dengan masing-masing *class* sebanyak 200 citra buah salak. Alasan penggunaan data sebanyak 1.200 citra dalam penelitian ini bertujuan untuk mendapatkan sampel yang cukup representatif dan memadai untuk melatih dan menguji model klasifikasi dengan tingkat keakuratan yang tinggi.

4.2. Tempat dan Waktu Penelitian

Penelitian dilaksanakan di Gedung Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Islam Indonesia mulai tanggal 2 Oktober 2023 s.d 31 Desember 2023.

4.3. Definisi Operasional Variabel Penelitian

Pada penelitian ini variabel yang akan digunakan adalah citra buah salak dengan enam *class* yang disajikan pada **Tabel 4.1** berikut.

Tabel 4.1 Definisi Variabel Penelitian

No	Class	Contoh Citra
1	Salak Gading Busuk	

No	<i>Class</i>	Contoh Citra
2	Salak Gading Sehat	
3	Salak Madu Busuk	
4	Salak Madu Sehat	
5	Salak Pondoh Busuk	

No	Class	Contoh Citra
6	Salak Pondoh Sehat	

4.4. Alat dan cara organisir data

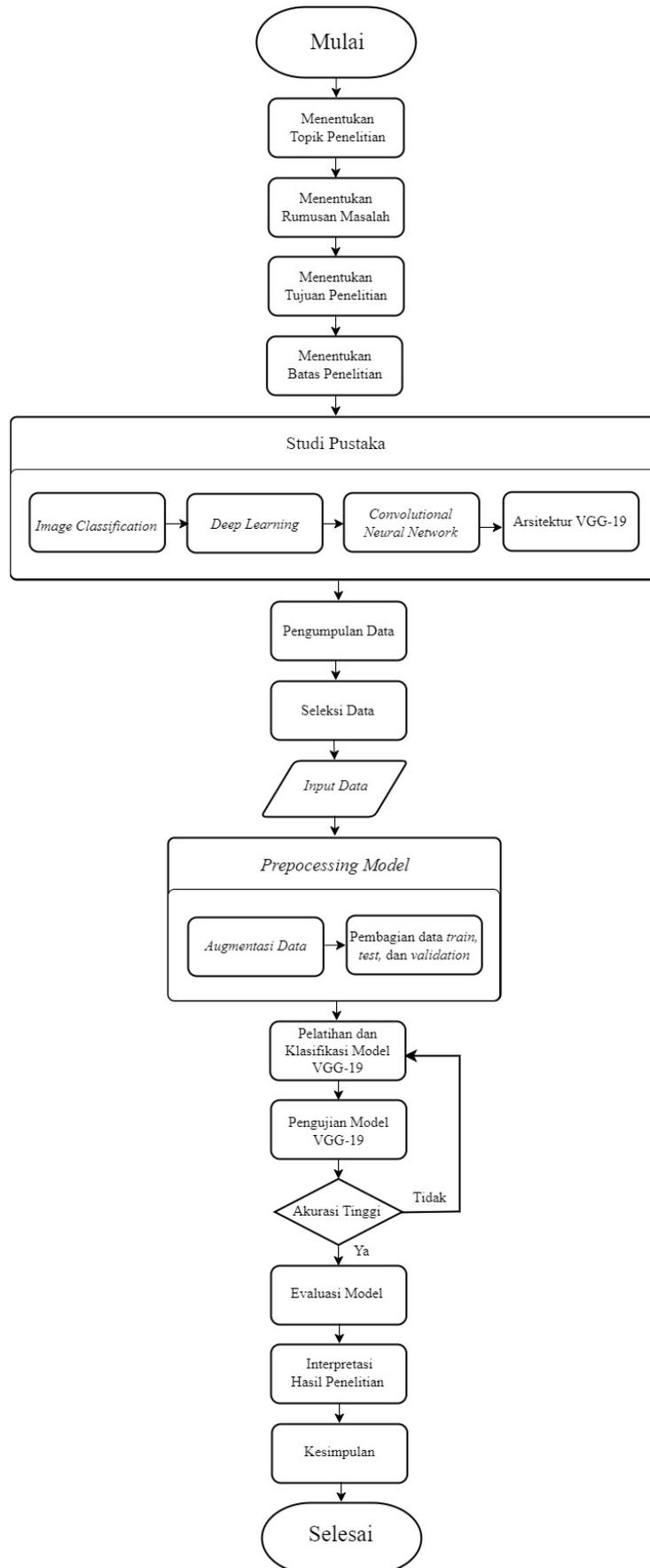
Alat yang digunakan untuk pengambilan data pada penelitian ini adalah kamera *handphone* dengan merek iPhone 11 yang memiliki kapasitas kamera 12 MP untuk melakukan pengambilan data citra. Pada tahap komputasi menggunakan perangkat keras (*hardware*) berupa laptop MSI Modern 14 B11MOU dengan *processor* 11th Gen Intel® Core™ i7-1195G7 @ 2.90GHz, 2918 Mhz, 4 Core(s) dan *storage* SSD 256 GB. Adapun perangkat lunak (*software*) yang digunakan pada penelitian ini adalah *Windows* 11 sebagai sistem operasi, *Visual Studio Code* sebagai editor teks, dan Python sebagai bahasa pemrograman. Setelah data terkumpul, kemudian dibagi menjadi 80% untuk data *train*, 10% untuk data *test*, dan 10% untuk data valid.

4.5. Metode Penelitian

Penelitian ini mengimplementasikan metode *Convolutional Neural Network* (CNN) dengan menggunakan arsitektur *Visual Geometry Group-19* (VGG-19) sebagai kerangka kerja utama dalam tugas mengklasifikasikan buah salak. Alasannya menggunakan metode dan arsitektur ini karena telah terbukti efektif dalam klasifikasi gambar dan pengenalan citra dengan pola yang kompleks. VGG-19 memiliki kedalaman yang cukup besar dengan banyak lapisan konvolusi dan lapisan terkait, sehingga dapat mengekstraksi fitur-fitur yang lebih abstrak dan mendalam dari gambar (Meena, 2022). Selain itu, keunggulan lain dari VGG-19 adalah kemampuannya untuk belajar dari data yang besar dan kompleks seperti dalam penelitian ini dengan menggunakan 1.200 citra, sehingga dapat memberikan hasil klasifikasi yang akurat. Hasil akurasi yang akan didapatkan akan bernilai besar, sebagaimana yang telah dilakukan pada penelitian-penelitian

terdahulu. Dengan membandingkan metode CNN dengan berbagai metode *transfer learning*-nya, telah terbukti bahwa metode CNN dengan arsitektur VGG-19 mampu menghasilkan nilai akurasi yang lebih tinggi. Untuk penjelasan dan informasi lebih lengkap nya terkait perbandingan nilai akurasi pada metode *transfer learning* CNN ini tertelah terurai pada Tabel 2.1 di atas.

Dalam mengimplementasikan model ini, penelitian menggunakan *module TensorFlow* dan *Keras* dari Bahasa pemrograman Python, yang merupakan alat bantu yang sangat berguna dalam pembuatan dan prediksi model citra buah salak. Penerapan metode ini memungkinkan analisis citra yang mendalam dan akurat untuk identifikasi buah salak. Alur penelitian ini disajikan dalam bentuk *flowchart* pada Gambar 4.1 yang memberikan gambaran visual mengenai alur keseluruhan penelitian.



Gambar 4.1 Flowchart Penelitian

Berikut ini, penjelasan pada *flowchart* (diagram alur) penelitian yang telah dipaparkan pada Gambar 4.1 di atas:

1. Menentukan topik, rumusan masalah, tujuan dan batasan penelitian. Mendapatkan topik klasifikasi menggunakan CNN dengan arsitektur VGG-19.
2. Mencari studi pustaka terkait *image classification*, CNN, *Deep learning*, dan VGG-19.
3. Mengumpulkan data yaitu dengan cara memotret buah salak, sehingga didapatkan enam *class* citra buah salak.
4. Seleksi data dengan memisahkan data terbaik yang tidak menyebabkan *error* saat komputasi.
5. *Input data* ke dalam teks editor *Visual Studio Code*
6. Melakukan *preprocessing model* dengan *augmentasi data* yang terdiri dari *cropping* dan *resize* kemudian melakukan *split dataset* menjadi *training*, *testing*, dan *validation*.
7. Melakukan pelatihan model klasifikasi VGG-19 dilanjutkan dengan pengujian model sehingga akan mendapatkan *training accuracy*.
8. Akurasi *training* ini akan menentukan apakah model harus dilatih ulang atau digunakan berdasarkan tinggi rendahnya nilai akurasi model.
9. Apabila mencapai tingkat akurasi yang diharapkan, penelitian akan dilanjutkan dengan menginterpretasikan hasil penelitian.
10. Kemudian melakukan prediksi pada gambar lain (*testing data*) untuk melihat akurasi prediksi yang akan digunakan sebagai kesimpulan.

BAB V

HASIL DAN PEMBAHASAN

Penelitian ini menggunakan metode klasifikasi *Convolutional Neural Network* (CNN) dengan arsitektur *Visual Geometry Group-19* (VGG-19) dengan mengklasifikasikan 6 *class* buah salak yaitu salak gading busuk, gading sehat, madu busuk, madu sehat, pondoh busuk, dan pondoh sehat. Tahapan pengklasifikasian buah salak menggunakan model CNN arsitektur VGG-19 seperti pada *flowchart* Gambar 4.1 dimana ada beberapa tahapan yang harus dilakukan yaitu pengumpulan data, inputasi data, *preprocessing model* berupa augmentasi data dan split data, pelatihan dan klasifikasi model, pengujian model, evaluasi model, terakhir interpretasi hasil dan kesimpulan. Pengimplementasian model CNN VGG-19 untuk mengklasifikasikan buah salak ini menggunakan Bahasa pemrograman Python dengan Visual Studio Code sebagai teks editor nya. Berikut langkah-langkah dan kode untuk proses klasifikasi buah salak.

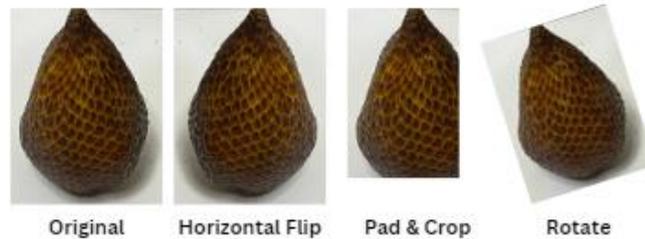
7.1 Pengumpulan dan Inputasi Data

Peneliti mengumpulkan dan menyusun dataset yang terdiri dari enam varietas citra buah salak. Proses pengumpulan data dilakukan dengan mengambil citra buah salak secara langsung dari enam kelas yang berbeda (data primer). Hasilnya adalah sebuah dataset sebanyak 1.200 citra buah salak, dengan setiap varietas memiliki 200 citra yang representatif. Proses pengambilan gambar secara langsung memastikan keakuratan dan keberagaman dataset, yang menjadi dasar untuk analisis dan penelitian lebih lanjut dalam pengklasifikasian buah salak. Setelah berhasil mengumpulkan dataset, langkah selanjutnya adalah memulai analisis dengan melakukan inputasi data ke dalam *software* Python. Sebelum melanjutkan ke tahap *preprocessing data*, diperlukan memanggil atau mengimpor *library* atau *package-package* yang akan digunakan dan mendukung selama penelitian.

7.2 Preprocessing Data

Dalam tahap *preprocessing data* ini, diterapkan metode augmentasi data untuk memperluas variasi dataset dan dilakukan pembagian data (*split data*) untuk memisahkan dataset menjadi subset *train*, validasi, dan *test*. Augmentasi adalah teknik yang memperkenalkan sampel baru klasifikasi dengan menerapkan transformasi pada sampel nyata (Sentosa, 2022). Proses augmentasi data bertujuan

untuk meningkatkan keberagaman dataset dengan cara memperkenalkan variasi pada data yang ada. Berikut ini hasil augmentasi data menggunakan salah satu *class* data.



Gambar 5.1 Augmentasi Data Pada Salak Pondoh Sehat

Pembagian data merupakan salah satu langkah penting dalam pengembangan model, di mana dataset utama dibagi menjadi subset yang digunakan untuk *train*, validasi, dan *test*. Proses ini bertujuan untuk memastikan bahwa model yang dikembangkan memiliki kinerja yang baik dan dapat digeneralisasi dengan baik pada data yang belum pernah digunakan sebelumnya (Anrel, 2019). Berikut ini *output* yang dihasilkan untuk pembagian 1.200 dataset yang terdiri dari 960 untuk data *train*, 120 untuk data validasi, dan 120 untuk data *test*.

```
Found 960 images belonging to 6 classes.  
Found 120 images belonging to 6 classes.  
Found 120 images belonging to 6 classes.
```

Gambar 5.2 Pembagian Data

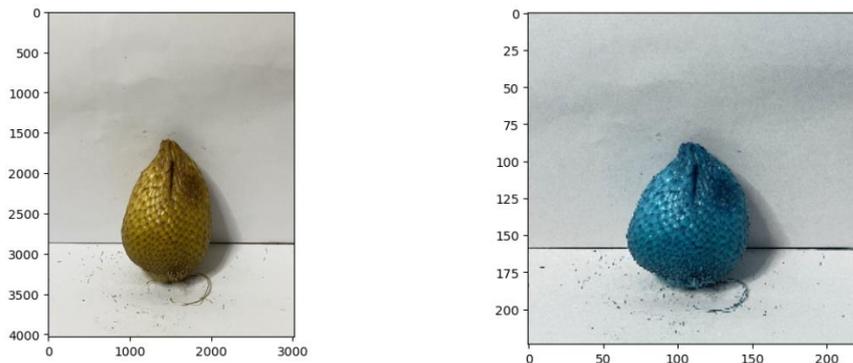
Selain itu, dalam proses *preprocessing data*, peneliti juga melakukan labelisasi pada enam kelas berbeda untuk buah salak. Buah salak Gading busuk diberi label 0, salak Gading sehat diberi label 1, salak Madu busuk diberi label 2, salak Madu sehat diberi label 3, salak Pondoh busuk diberi label 4, dan salak Pondoh sehat diberi label 5. Proses labelisasi ini secara otomatis dilakukan oleh *software* Python yang disusun berdasarkan abjad pertama dari nama buah salak tersebut. Labelisasi ini dapat membantu mengenali dan membedakan dengan jelas varietas dan kualitas buah salak, yang pada akhirnya akan berkontribusi pada keakuratan dan kinerja model dalam memproses data pada *software* Python.

Tabel 5.1 Labelisasi *Class* Buah Salak

Jenis Kelas Buah Salak	Pelabelan
Gading Busuk	0
Gading Sehat	1
Madu Busuk	2
Madu Sehat	3
Pondoh Busuk	4
Pondoh Sehat	5

Aspek penting dalam tahap *preprocessing data* adalah *resizing data*, yang menjadi prasyarat utama untuk memenuhi kebutuhan sistem. Dikarenakan metode CNN ini memerlukan *input* gambar berdimensi 224x224 piksel, penyelesaian atas kendala tersebut dapat dilakukan melalui proses *resizing* menggunakan *library* OpenCV dalam bahasa pemrograman Python (Efendi, 2022). Dengan demikian, semua gambar yang digunakan dalam penelitian ini akan diubah ukurannya menjadi 224x224 piksel agar sesuai dengan persyaratan sistem.

Proses perubahan ini dilakukan untuk memastikan konsistensi ukuran dan resolusi citra dalam dataset. Dengan ukuran yang seragam, data dapat disesuaikan dengan model pengklasifikasian dengan model CNN untuk keperluan *training*, validasi dan *testing*. Berikut ini adalah contoh citra sebelum dan setelah perubahan resolusi piksel.



Gambar 5.3 Citra Sebelum dan Setelah Perubahan Piksel

Setelah berhasil me-*resize* data, langkah berikutnya adalah mengubah data gambar yang telah di-*input* ke dalam bentuk *array*, dengan memanfaatkan *library* Numpy. Konversi gambar ke dalam bentuk *array* dengan menggunakan *library* Numpy memberikan kemudahan dalam melakukan operasi matematika dan transformasi lainnya pada data gambar.

Data gambar yang telah dikonversi menjadi bentuk *array* selanjutnya akan dinormalisasi dengan mengubah rentang nilai piksel. Proses ini melibatkan pembagian setiap elemen matriks piksel dalam setiap dataset dengan nilai 255.0. Tujuannya adalah untuk memastikan bahwa nilai-nilai piksel dalam *array* berada dalam rentang standar [0, 1].

Dengan normalisasi ini, model yang dibuat harapannya dapat belajar dengan lebih baik karena nilai yang lebih kecil dapat membantu *perceptron* dan *layer-layer neural network* bekerja secara lebih efisien. Normalisasi ini juga dapat membantu menghindari masalah numerik yang mungkin timbul dalam pelatihan model.

7.3 Pelatihan Model Klasifikasi

Setelah menyelesaikan tahap *preprocessing data*, langkah selanjutnya melakukan pelatihan menggunakan model CNN dengan arsitektur VGG-19. Dalam proses pelatihan model ini, peneliti menggunakan fungsi kode VGG19 dari *TensorFlow* dengan *input shape* (224, 224, 3), yang mengindikasikan ukuran 224x224 piksel dan 3 sebagai representasi warna RGB. Selain itu peneliti juga menggunakan *weight* (bobot) *imagenet* yang telah dilatih sebelumnya untuk pengenalan gambar dengan menggunakan dataset ImageNet. Untuk parameter-parameter yang digunakan pada pelatihan ini melibatkan *dense* 6, fungsi aktivasi *softmax*, *loss Sparse Categorical Crossentropy*, *optimizer* Adam, jumlah *Epoch* 5, dan ukuran *batch* 32. Proses pelatihan ini bertujuan untuk mengoptimalkan model agar dapat mengenali dan mengklasifikasikan berbagai varietas buah salak dengan akurasi yang tinggi. Berikut ini hasil model yang didapatkan.

Tabel 5.2 Output Model CNN VGG-19

Layer	Output Shape	Params
input_1 (InputLayer)	[(None, 224, 224, 3)]	0
block1_conv1 (Conv2D)	(None, 224, 224, 64)	1792
block1_conv2 (Conv2D)	(None, 224, 224, 64)	36928
block1_pool (MaxPooling2D)	(None, 112, 112, 64)	0
block2_conv1 (Conv2D)	(None, 112, 112, 128)	73856
block2_conv2 (Conv2D)	(None, 112, 112, 128)	147584
block2_pool (MaxPooling2D)	(None, 56, 56, 128)	0
block3_conv1 (Conv2D)	(None, 56, 56, 256)	295168

Layer	Output Shape	Params
block3_conv2 (Conv2D)	(None, 56, 56, 256)	590080
block3_conv3 (Conv2D)	(None, 56, 56, 256)	590080
block3_conv4 (Conv2D)	(None, 56, 56, 256)	590080
block3_pool (MaxPooling2D)	(None, 28, 28, 256)	0
block4_conv1 (Conv2D)	(None, 28, 28, 512)	1180160
block4_conv2 (Conv2D)	(None, 28, 28, 512)	2359808
block4_conv3 (Conv2D)	(None, 28, 28, 512)	2359808
block4_conv4 (Conv2D)	(None, 28, 28, 512)	2359808
block4_pool (MaxPooling2D)	(None, 14, 14, 512)	0
block5_conv1 (Conv2D)	(None, 14, 14, 512)	2359808
block5_conv2 (Conv2D)	(None, 14, 14, 512)	2359808
block5_conv3 (Conv2D)	(None, 14, 14, 512)	2359808
block5_conv4 (Conv2D)	(None, 14, 14, 512)	2359808
block5_pool (MaxPooling2D)	(None, 7, 7, 512)	0
flatten (Flatten)	(None, 25088)	0
dense (Dense)	(None, 6)	150534
Total Params: 20.174.918		
Trainable params: 150.534		
Non-trainable params: 20.024.384		

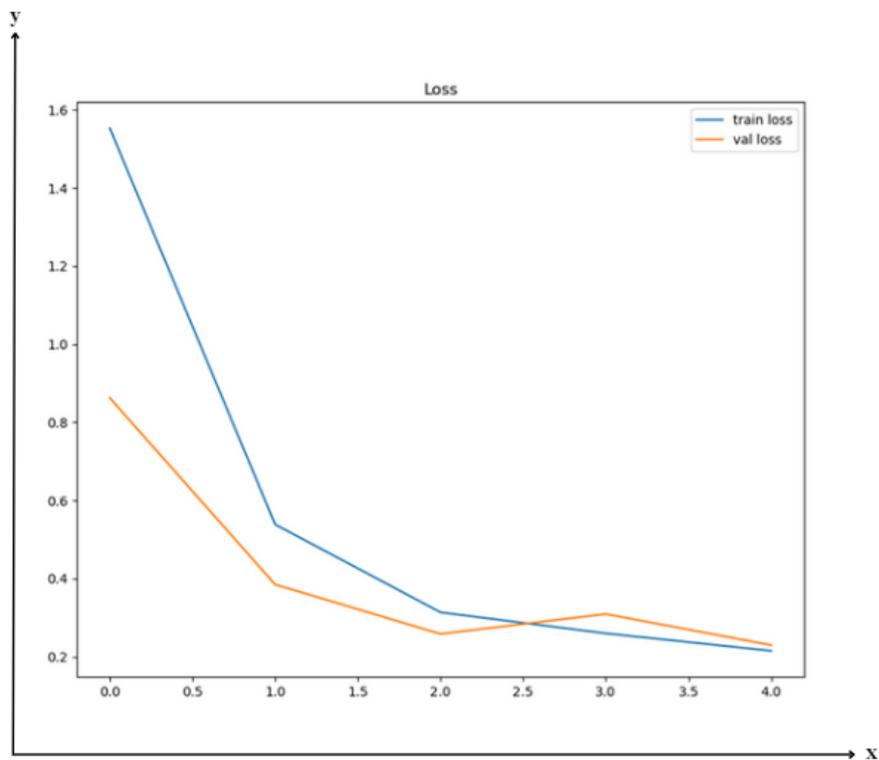
Berdasarkan pada Tabel 5.2 telah berhasil membangun model VGG-19 dan memperoleh model dengan total parameter 20.174.918 yang merupakan 150.534 *trainable* parameter dan 20.024.384 *non-trainable* parameter. Artinya, hanya sebagian kecil dari total parameter yang dapat diubah atau disesuaikan melalui proses pelatihan model, sementara mayoritas parameter lainnya tetap konstan atau tidak dapat diubah.

Model CNN VGG-19 untuk mengklasifikasikan varietas dan kualitas buah salak telah didapatkan, selanjutnya dilakukan pelatihan model tersebut dengan menggunakan data *train* dan validasi dengan menggunakan parameter-parameter yang telah dipaparkan di atas, didapatkan hasil sebagai berikut.

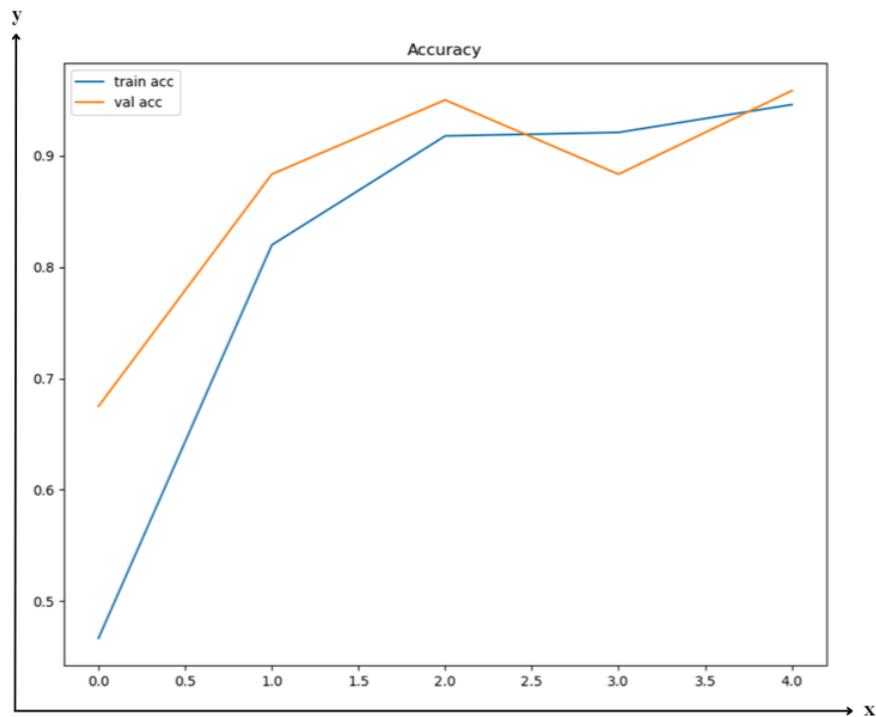
Tabel 5.3 Proses *Training Model*

Epoch	Time	Train Loss	Train Accuracy	Val Loss	Vall Accuracy
1/5	302s	1,552	0,4667	0,8622	0,6750
2/5	329s	0,5386	0,8198	0,3849	0,8833
3/5	320s	0,3137	0,9177	0,2583	0,9500
4/5	319s	0,2598	0,9208	0,3094	0,8833
5/5	315s	0,2152	0,9458	0,2298	0,9583

Setelah melakukan pelatihan model sebanyak lima kali perulangan menggunakan data *train* dan validasi, diperoleh nilai akurasi untuk data *train* sebesar 0,9458, atau setara dengan 94,58% dan nilai *loss* sebesar 0,2152 atau setara dengan 21,52% sedangkan nilai akurasi untuk data valid sebesar 0,9583, atau setara dengan 95,83% dan nilai *loss* sebesar 0,2298 atau setara dengan 22,98%. *Output* di atas juga dapat divisualisasikan ke dalam bentuk grafik sehingga didapatkan hasil sebagai berikut.



Gambar 5.4 Grafik *Loss Value* Proses *Training Model*



Gambar 5.5 Grafik *Accuracy Value* Proses *Training Model*

Berdasarkan Gambar 5.4 dan 5.5 menunjukkan hasil dari setiap perulangan ke perulangan berikutnya tampak terjadi peningkatan kinerja model, yang ditunjukkan oleh penurunan *loss* dan peningkatan akurasi pada kedua dataset, baik pada data *train* maupun data validasi. Selain itu, model mencapai akurasi yang lebih tinggi pada data validasi (*val_accuracy*) daripada pada data *train (accuracy)*, hal ini menunjukkan bahwa model tidak mengalami *overfitting* (terlalu memfokuskan pada data pelatihan).

Setelah menyelesaikan pelatihan model selama lima kali perulangan, langkah berikutnya adalah mengevaluasi performa model dengan menggunakan data *test*. Evaluasi ini memberikan gambaran tentang seberapa baik model yang telah dilatih dapat menggeneralisasi dan memberikan prediksi yang baik pada data baru yang belum pernah digunakan sebelumnya. Dan dihasilkan *output* seperti pada Gambar berikut.

Tabel 5.4 Proses *Testing Model*

Epoch	Time	Test Loss	Test Accuracy
4/4	34s	0,1828	0,9583

Berdasarkan Tabel 5.4 diperoleh nilai akurasi untuk data *test* sebesar 0,9583, atau setara dengan 95,83% dan nilai *loss* sebesar 0,1827 atau setara dengan 18,27%. Hasil ini menegaskan bahwa model CNN dengan arsitektur VGG-19

telah memberikan performa yang sangat baik dalam proses klasifikasi varietas dan kualitas buah salak dengan menghasilkan prediksi yang akurat dan konsisten.

7.4 Evaluasi Model

Dalam proses evaluasi model menggunakan data test, peneliti memanfaatkan parameter seperti *precision*, *recall*, dan *f1-score* sebagai metrik evaluasi kinerja. Metrik-metrik ini memberikan pemahaman yang lebih komprehensif tentang kemampuan model dalam mengidentifikasi dan memprediksi dengan akurat, dan mempertimbangkan keseimbangan antara presisi dan *recall* dalam konteks tugas klasifikasi. Berikut ini hasil yang telah didapatkan.

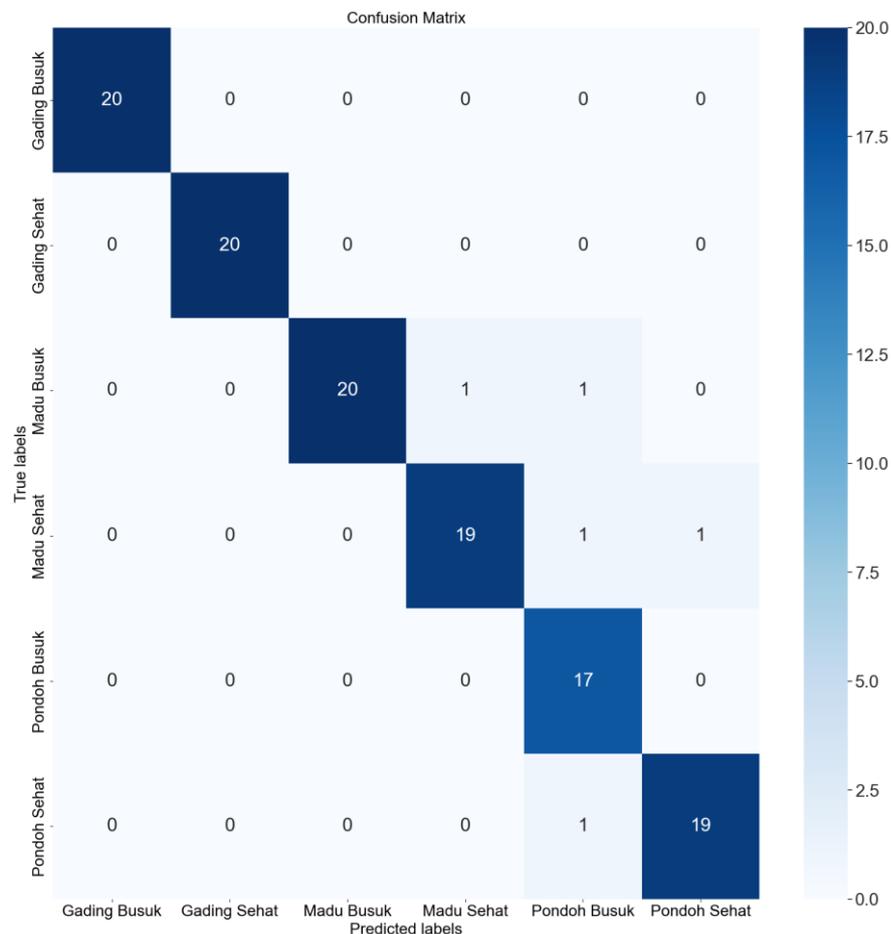
Tabel 5.5 Evaluasi Model

	Precision	Recall	F1-Score	Support
0	1,00	1,00	1,00	20
1	1,00	1,00	1,00	20
2	1,00	0,91	0,95	22
3	0,95	0,90	0,93	21
4	0,85	1,00	0,92	17
5	0,95	0,95	0,95	20
Accuracy			0,96	120
Macro Avg	0,96	0,96	0,96	120
Weighted Avg	0,96	0,96	0,96	120

Berdasarkan Tabel 5.5, *precision* untuk kelas 0, 1, dan 2 adalah sempurna (1,00), menunjukkan bahwa model sangat baik dalam mengidentifikasi kasus positif untuk kelas-kelas tersebut. Sedangkan *precision* untuk kelas 3, 4, dan 5 juga terkategori besar, artinya model memiliki kecenderungan kecil untuk memberikan *false positive*. *Recall* untuk kelas 0, 1, dan 4 adalah sempurna (1,00), menunjukkan bahwa model sangat baik dalam mengidentifikasi seluruh kasus positif untuk kelas-kelas tersebut sedangkan untuk kelas 2, 3, dan 5 adalah 0,91, 0,90, dan 0,95 menunjukkan bahwa model kehilangan sekitar 9%, 10%, dan 5% kasus positif yang seharusnya diidentifikasi. Untuk *f1-score* adalah *harmonic mean* dari presisi dan *recall*. *F1-score* yang tinggi (mendekati 1,00) untuk setiap kelas menunjukkan keseimbangan yang baik antara presisi dan *recall*. Adapun akurasi keseluruhan model adalah 0,96, yang mencerminkan 96% model dapat mengidentifikasi dengan benar seluruh kelas pada keseluruhan dataset.

7.5 Interpretasi Hasil

Hasil evaluasi yang didasarkan pada *confusion matrix* memberikan gambaran yang lebih lengkap mengenai keberhasilan model dalam mengidentifikasi *True Positives*, *False Positives*, *True Negatives*, dan *False Negatives* untuk masing-masing kelas, memungkinkan analisis yang lebih terperinci terhadap keakuratan dan ketepatan model pada setiap aspek klasifikasi. Berikut ini *confusion matrix* dengan menggunakan data *test* yang diperoleh.



Gambar 5.6 *Confusion Matrix Model Testing*

Berdasarkan *confusion matrix* 6x6 pada Gambar 5.6, 20 data citra buah salak Gading busuk, Gading sehat, dan Madu busuk berhasil diklasifikasikan dengan benar. 19 data citra salak Madu sehat berhasil diklasifikasikan dengan benar dan 1 data citra salak Madu sehat diklasifikasikan sebagai salak Madu busuk. 17 data citra salak Pondoh busuk berhasil diklasifikasikan dengan benar sedangkan 3 citra lainnya diklasifikasikan sebagai salak Pondoh sehat, Madu sehat, dan Madu busuk. Sedangkan 19 data citra salak Pondoh sehat berhasil diklasifikasikan dengan benar sedangkan 1 citra lainnya diklasifikasikan sebagai salak Madu sehat.

Selain itu, secara visual terdapat perbedaan yang signifikan antara salak Gading busuk, salak Gading sehat, dan Madu busuk, yang dapat dilihat dari bentuk dan warna kulitnya. Dengan demikian, ketiga salak tersebut berhasil diklasifikasikan dengan tepat. Sedangkan untuk klasifikasi salak Madu sehat, terdapat 1 kesalahan yaitu salak Madu sehat diklasifikasikan sebagai salak Madu busuk. Hal ini dikarenakan secara visual kedua salak tersebut cukup mirip dari segi bentuk dan warnanya, dan ada kemungkinan satu salak Madu sehat ini berwarna lebih gelap sehingga terklasifikasikan sebagai salak Madu busuk. Kemudian untuk klasifikasi salak Pondoh busuk, terdapat tiga kesalahan yaitu salak Pondoh busuk diklasifikasikan sebagai salak Pondoh sehat, Madu sehat, dan Madu busuk. Dari ketiga kesalahan ini, secara visual salak Pondoh busuk dan Pondoh sehat cukup mirip dari segi bentuk dan warnanya, kemungkinan satu salak Pondoh busuk ini berwarna lebih terang dan tidak terdapat corak busuk sehingga terklasifikasikan sebagai salak Pondoh sehat. Untuk salak Pondoh busuk dan Madu sehat, secara visual bentuk dan warnanya cukup berbeda, tetapi sepertinya satu salak Pondoh busuk ini memiliki ukuran yang lebih besar warna yang lebih terang, sehingga terklasifikasikan sebagai salak Madu sehat. Selain itu untuk salak Pondoh busuk dan Madu busuk juga secara visual bentuknya cukup berbeda, tetapi sepertinya satu salak Pondoh busuk ini memiliki ukuran yang lebih besar sehingga terklasifikasikan sebagai salak Madu busuk. Terakhir untuk klasifikasi pada salak Pondoh sehat terdapat satu kesalahan yaitu salak Pondoh sehat diklasifikasikan sebagai salak Madu sehat. Karena secara visual kedua salak tersebut memiliki warna yang cukup mirip, sehingga kemungkinan terdapat satu salak Pondoh sehat yang memiliki ukuran lebih besar sehingga terklasifikasikan sebagai salak Pondoh sehat.

Berdasarkan hasil *confusion matrix* yang telah dijelaskan sebelumnya, dapat disimpulkan bahwa model CNN dengan arsitektur VGG-19 menunjukkan kinerja yang cukup baik dalam mengklasifikasikan berbagai varietas dan kualitas buah salak di Yogyakarta. Keberhasilan model ini memberikan kontribusi yang signifikan bagi petani salak. Dengan menggunakan teknologi *object detection* berbasis *deep learning* seperti CNN VGG-19, petani dapat dengan cepat dan akurat mengidentifikasi kondisi kesehatan buah salak mereka. Selain itu, dengan kemampuan untuk membedakan varietas salak, petani dapat memilih varietas yang paling sesuai dengan kondisi lingkungan dan permintaan pasar. Secara

keseluruhan, penerapan CNN VGG-19 dalam pengenalan varietas dan kondisi kesehatan buah salak tidak hanya meningkatkan efisiensi produksi, tetapi juga membantu petani salak dalam mengoptimalkan hasil panen dan meningkatkan kualitas produk mereka. Teknologi ini dapat menjadi alat penting dalam meningkatkan daya saing dan keberlanjutan industri perkebunan salak di Yogyakarta.

BAB VI

PENUTUP

6.1 Kesimpulan

Berdasarkan analisis dan pembahasan dalam kasus ini, penulis dapat menarik kesimpulan bahwa:

1. Pada penelitian ini, karakteristik visual yang membedakan varietas dan kualitas buah salak dapat diidentifikasi melalui bentuk dan warna kulit dari setiap varietas. Salak Gading sehat memiliki bentuk oval kecil dengan kulit berwarna gading atau coklat muda kekuningan. Sebaliknya, Salak Gading busuk memiliki warna yang lebih gelap, dan terdapat spot coklat menandakan kebusukan. Salak Madu sehat memiliki bentuk besar dan bulat dengan kulit coklat keemasan, sisik tersusun rapi, cenderung renggang, dan membentuk garis lurus. Salak Madu busuk memiliki warna lebih gelap, terkadang pada bagian ujung atas kulit hancur dan berair. Salak Pondoh sehat berbentuk oval dengan kulit coklat yang sisiknya tampak lebih rapat. Sementara Salak Pondoh busuk memiliki warna yang lebih gelap dan ujung kulit bagian atas terkadang berwarna hitam.
2. Implementasi metode *Convolutional Neural Network* (CNN) dengan arsitektur *Visual Geometry Group-19* (VGG-19) untuk klasifikasi varietas dan kualitas buah salak asal DI Yogyakarta menunjukkan bahwa model dapat dengan tepat mengenali perbedaan visual antara Salak Gading, Salak Madu, dan Salak Pondoh, serta mengklasifikasikan kualitas buah berdasarkan karakteristik kulitnya yaitu busuk dan sehat.
3. Nilai akurasi yang didapatkan setelah melakukan pelatihan model sebanyak lima kali perulangan menggunakan data *train* dan validasi, diperoleh nilai akurasi untuk data *train* sebesar 94,58% sedangkan nilai akurasi untuk data valid sebesar 95,83%. Model mencapai akurasi yang lebih tinggi pada data validasi daripada pada data *train*, hal ini menunjukkan bahwa model tidak mengalami *overfitting*. Sedangkan nilai akurasi untuk data *test* sebesar 95,83%, hal ini membuktikan bahwa model memiliki performa yang sangat baik dalam proses klasifikasi varietas dan kualitas buah salak.

6.2 Saran

Berdasarkan kesimpulan yang telah dipaparkan, maka peneliti memberikan saran sebagai berikut:

1. Diharapkan data citra yang digunakan dalam penelitian selanjutnya hanya memuat objek yang diteliti, dengan memperhatikan kondisi cahaya saat pengambilan citra digital. Hal ini bertujuan untuk menghindari bayangan dari objek maupun bayangan peneliti yang mungkin muncul selama proses pemotretan, sehingga nilai akurasi dapat ditingkatkan secara signifikan.
2. Diharapkan citra yang digunakan pada penelitian selanjutnya memuat video dari objek tersebut, tidak hanya berbentuk gambar. Sehingga model yang digunakan juga dapat dikembangkan tidak hanya *image classification* saja tetapi masuk ke *object detection*.
3. Diharapkan data yang digunakan dapat lebih banyak dengan kategori yang lebih banyak dan menarik.
4. Perbandingan antara parameter-parameter seperti *epoch*, fungsi aktivasi, *optimizer*, dan lain sebagainya yang digunakan bisa menjadi penelitian lebih lanjut untuk melihat bagaimana pengaruhnya terhadap tingkat akurasi yang diperoleh.

DAFTAR PUSTAKA

- Abdurrahman. (2023, Oktober 20). *Salak Pondoh Manis Gurih Asli Sleman*. Diambil kembali dari belibakpia.com: <https://belibakpia.com/salak-pondoh-jogja/>
- Afifah, L. (2023). *Apa itu Confusion Matrix di Machine Learning?* Diambil kembali dari ilmudatapy: <https://ilmudatapy.com/apa-itu-confusion-matrix/>
- Ahmad, A. (2017). Mengenal Artificial Intelligence, Machine Learning, Neural Network, dan Deep Learning. *Yayasan Cahaya Islam, Jurnal Teknologi Indonesia*, 1-5.
- Anam, I. C., Dermasetiawan, D. N., & Nugroho, D. E. (2018). *Budidaya Tanaman Salak*. Surabaya: PT. Revka Petra Media.
- Anrel, P. (2019, Oktober 28). *Training Pada Machine Learning*. Diambil kembali dari Medium: <https://medium.com/@anrelputranew/proses-data-testing-dan-training-pada-machine-learning-cdf8fcd93355>
- Arthana, R. (2019). Mengenal Accuracy, Precision, Recall dan Specificity serta yang diprioritaskan dalam Machine Learning. *Medium*.
- Ashari, P. I. (2013). *Salak The Snake Fruit*. Malang: UB Press.
- Bahi, M., & Moamed, B. (2018). Deep Learning for Ligand-Based Virtual Screening in Drug Discovery. *International Conference on Pattern Analysis and Intelligent Systems*.
- Brownlee, J. (2020). *Deep Learning for Computer Vision Image Classification, Object Detection, Face Recognition in Python*. Machine Learning Mastery.
- Cahya, F. N., Hardi, N., & Dwiza Riana, S. H. (2021). Klasifikasi Penyakit Mata Menggunakan Convolutional Neural Network (CNN). *Jurnal Sistem Informasi*, 618-626.
- Chen, T. H. (2017). What is “stride” in Convolutional Neural Network? *Medium*.
- Davies, a. P. (2004). *Database System Third Edition*. New York: Palgrave macmillan.
- Dharmadi, R. (2018). Mengenal Convolutional Layer Dan Pooling Layer. *Medium*, 1-7.
- Dharmadi, R. (2018). Mengenal Convolutional Neural Network. *Medium*.

- Efendi, D., Jasril, Suwanto, S., Syafria, F., & Budianita, E. (2022). Penerapan Algoritma Convolutional Neural Network Arsitektur ResNet-50 untuk Klasifikasi Citra Daging Sapi dan Babi. *Jurnal Riset Komputer*, 607-614.
- Ferentinos, K. P. (2018). Deep learning models for plant disease detection and diagnosis. *Computers and Electronics in Agriculture, Volume 145*, 311-318.
- Feurer, M., & Hutter, F. (2019). Hyperparameter Optimization. *Auto Machine Learning*, 1-32.
- Fuadah, Y. N., Ubaidillah, I. D., Ibrahim, N., Taliningsing, F. F., & Nindaaan Khofiya Sy, M. A. (2022). Optimasi Convolutional Neural Network dan K-Fold Cross Validation pada Sistem Klasifikasi Glaukoma. *Elkomika*, 728-741.
- Haidaroh, A. (2016). Inroduction Citra Digital. Dalam A. Haidaroh, *Citra Digital* (hal. 1-38). Bandung: Artha Buana.
- Halim, C., & Prasetyo, H. (2018). Penerapan Artificial Intelligence dalam Computer Aided Instructure (CAI). *Jurnal Sistem Cerdas 2018 Volume 01*, 45-51.
- Han, J., Kamber, M., & Pei, J. (2012). *Data Mining Concepts and Techniques Third Edition*. Waltham USA: Morgan Kaufmann.
- Harrington, P. (2012). *Machine Learning in Action*. New York: Manning Publication.
- Haryanto, D., & Priyanto, E. M. (2018). *Potensi Buah Salak Sebagai Suplemen Obat dan Pangan*. Surakarta: UMS Press.
- Hasad, A. (2018). *Algoritma Optimasi dan Aplikasinya*. Bogor: wordpress.
- Hastie, T., Tibshirani, R., & Friedman, J. (2008). *The Elements of Statistical Learning: Data mining, Inference, and Prediction*. New York: Springer.
- Hastomo, W., Karno, A. S., & Bakti, I. (2022). Optimasi Convolutional Neural Network Untuk Deteksi Covid-19. *Jurnal Peradaban Sains, Rekayasa, dan Teknologi*, 317-331.
- Hawari, F. H., Fadillah, F., Alviandi, M. R., & Arifin, T. (2022). Klasifikasi Penyakit Tanaman Padi Menggunakan Algoritman CNN (Convolutional Neural Network). *Riset Sains dan Informatika*, 184-189.

- Hendriyana, & Maulana, Y. H. (2020). Identifikasi Jenis Kayu menggunakan Convolutional Neural Network dengan Arsitektur Mobilenet. *Rekayasa Sistem dan Teknologi Informasi*, 70-76.
- Hidayat, B. M., & Putra, R. E. (2019). Penerapan CNN dengan Filter Gabor sebagai feature extractor untuk Content-Based Image Retrieval. *Journal of Informatics and Computer Science*, 16-25.
- Hidayatullah, P. (2017). *Pengolahan Citra Digital Teori dan Aplikasinya*. Bandung: Informatika.
- Ho, D., Liang, E., & Liaw, R. (2019, June 7). *1000x faster Data Augmentation*. Diambil kembali dari Berkeley Artificial Intelligence Research: https://bair.berkeley.edu/blog/2019/06/07/data_aug/
- Hogg, R., & Craig, A. (2005). *Introduction to Mathematical Statistics (6th ed.)*. New Jersey: Pearson Prentice Hall.
- James, G., Witten, D., Hastie, T., & Tibshirani, R. (2017). *An Introduction to Statistical Learning with Applications in R*. New York: Springer.
- Kalambe, K., Awachat, S., & Raipure, S. (2022). Design of Deep Recursive CNN Model for Detecting and Classifying Peston Plant. *Internationa; Journal of Health Sciences*, 10537-10545.
- Kamilaris, A., & Prenafeta-Boldu, F. X. (2018). Deep learning in agriculture: A survey. *Computers and Electronics in Agriculture*, 147, 70 - 90.
- Kimura, N., Yoshinaga, I., Sekijima, K., Azechi, I., & Baba, D. (2019). Convolutional Neural Network Coupled with a Transfer-Learning Approach for Time-Series Flood Predictions. *MDPI Journals Awaded Impact Factor*.
- Kusmanto, R., & Tomponu, A. N. (2011). Pengolahan Citra Digital untuk Mendeteksi Obyek Menggunakan Pengolahan Warna Model Normalisasi RGB. *Seminar Nasional Teknologi Informasi & Komunikasi Terapan*, 321-328.
- Learning, D. I. (2020, Agustus). *Padding and Stride*. Diambil kembali dari Dive Into Deep Learning: https://d2l.ai/chapter_convolutional-neural-networks/padding-and-strides.html
- Lina, Q. (2019). Apa itu Convolutional Neural Network? *Medium*.

- Lorentius, C. A., Adipranata, R., & Tjondrowiguna, A. (2019). Pengenalan Aksara Jawa dengan Menggunakan Metode Convolutional Neural Network. *Infra*, 23-32.
- Mayanglambam, G. (2020). Deep Learning Optimizers SGD with momentum, Adagrad, Adadelata, Adam optimizer. *Medium*.
- Meena, G., Mohbey, K. K., Indian, A., & Kumar, S. (2022). Sentiment Analysis from Images using VGG19 based Transfer Learning Approach. *Telematika*, 37-48.
- Nugroho, K. S. (2019). Confusion Matrix untuk Evaluasi Model pada Supervised Learning. *Medium*.
- Nurhikmat, T. (2018). Implementasi Deep Learning Untuk Image Classification Menggunakan Algoritma Convolutional Neural Network (CNN) Pada Citra Wayang Golek. *ResearchGate*, 16.
- Ognjanovski, G. (2019). Everything you need to know about Neural Networks and Backpropagation — Machine Learning Easy and Fun. *Medium*.
- Parhusip, P. J. (2021). *Identifikasi Obat-Obatan Menggunakan Metode CNN (Convolutional Neural Network) Berbasis Android*. Medan: Repositori Institusi USU.
- Pujoseno, J. (2018). *Implementasi Deep Learning Menggunakan Convolutional Neural Network Untuk Klasifikasi Alat Tulis*. Yogyakarta: Universitas Islam Indonesia.
- Purwins, H., Li, B., Virtanen, T., Schluter, J., Chang, S.-y., & Sainath, T. (2019). Deep Learning for Audio Signal Processing. *OURNAL OF SELECTED TOPICS OF SIGNAL PROCESSING, VOL. 13*, 206-2019.
- Putra, F. A., Irawan, B., & Dinimaharawati, A. (2023). Pengenalan Ekspresi Wajah Menggunakan Arsitektur Xception pada Algoritma CNN. *e-Proceeding of Engineering*, 651-660.
- Qiu, S., Xu, X., & Cai, B. (2017). FReLU: Flexible Rectified Linear Units for Improving Convolutional Neural Networks. *Computer Vision and Pattern Recognition*, 297-304.
- Raharjo, B. (2017). *Belajar Otodidak Flask (Framework Python Untuk Pengembangan Aplikasi Web)*. Bandung: INFORMATIKA.
- Rahutomo, R., Perbangsa, A. S., Lie, Y., Cenggoro, T. W., & Pardamen, B. (2019). Artificial Intelligence Model Implementation in Web-Based

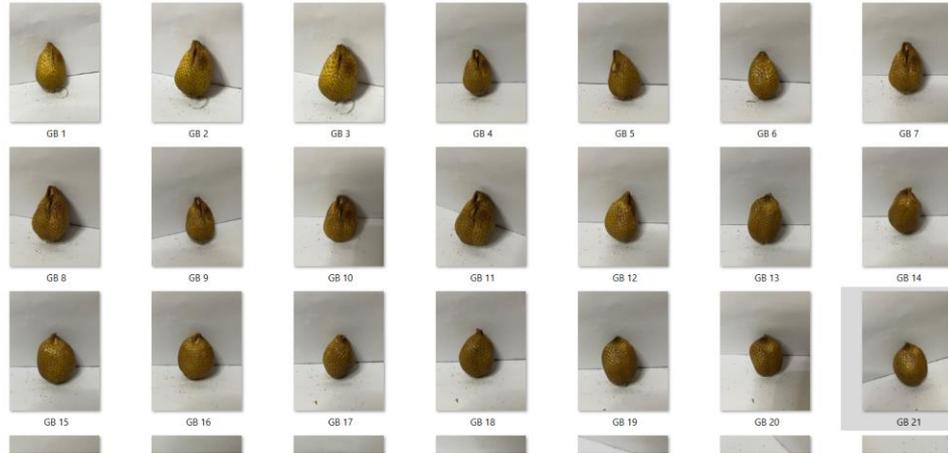
- Application for Pineapple Object Counting. *2019 International Conference on Information Management and Technology (ICIMTech)*, 525-530.
- Rale, N., Solanki, R., Bein, D., Andro-Vasko, J., & Bein, W. (2019). Prediction of Crop Cultivation. *2019 IEEE 9th Annual Computing and Communication Workshop and Conference (CCWC)* (hal. 0227-0232). Las Vegas, NV, USA, USA: IEEE.
- Ramadhan, M. M., Sitanggang, I. S., Nasution, F. R., & Ghifari, A. (2017). Parameter Tuning in Random Forest Based on Grid Search Method for Gender Classification Based on Voice Frequency. *International Conference on Computer, Electronics and Communication Engineering 2017*, (hal. 625).
- Rasyad, R. (2013). *Metode Statistik Deskriptif untuk Umum*. Jakarta: Grasindo.
- Rianto, P., & Agus, H. (2017). Penentuan Kematangan Buah Salak Pondoh di Pohon Berbasis Pengolahan Citra Digital. *IJCCS*, 143-154.
- Rochman, F., & Junaedi, H. (2020). Implementasi Transfer Learning untuk Identifikasi Ordo Tumbuhan Melalui Daun. *Syntax Admiration*, 672-679.
- Saifullah, S., Sunardi, & Anton, Y. (2016). Analisis Perbandingan Pengolahan Citra Asli dan Hasil Cropping untuk Identifikasi Telur. *Teknik Informatika dan Sistem Informasi*, 341-350.
- Sanjaya, J., & Ayub, M. (2020). Augmentasi Data Pengenalan Citra Mobil Menggunakan Pendekatan Random Crop, Rotate, dan Mixup. *Jurnal Teknik Informatika dan Sistem Informasi* , 311-323.
- Sentosa, E., Mulyana, D. I., Cahyana, A. F., & Pramuditasari, N. G. (2022). Implementasi Image Classification Pada Batik Motif Bali Dengan Data Augmentation dan Convolutional Neural Network. *Jurnal Pendidikan Tambusai*, 1451-1463.
- Shinta, R., Jasril, Irsyad, M., Yanto, F., & Sanjaya, S. (2023). Klasifikasi Citra Penyakit Daun Tanaman Padi Menggunakan CNN dengan Arsitektur VGG-19. *JURNAL SAINS DAN INFORMATIKA* , 37-45.
- Simonyan, K., & Zisserman, A. (2015). Very Deep Convolutional Networks for Large Scale Image Recognition. *ICLR*, 1-14.
- Singh, M., Verma, A., Parasher, A., Chauhan, N., & Budhiraja, G. (2019). Implementation of Database Using Python Flask Framework.

- International Journal of Engineering and Computer Science*, 24894-24899.
- Sleman, B. P. (2021, November 21). *Upaya Meningkatkan Ekspor Salak Sleman*. Diambil kembali dari Bappeda Sleman: <https://bappeda.slemankab.go.id/upaya-menguatkan-ekspor-salak-sleman.slm>
- Statistik, B. P. (2022). *Produksi Tanaman Buah-buahan 2022*. Diambil kembali dari Badan Pusat Statistik: <https://www.bps.go.id/indicator/55/62/1/produksi-tanaman-buah-buahan.html>
- Taylor, O., Ezekiel, P., & Deedam-Okuchaba, F. (2019). A Model to Detect Heart Disease using Machine Learning Algorithm. *International Journal of Computer Sciences and Engineering, Volume-7, Issue-11*, 1-5.
- Thakur, R. (2019). Step by step VGG16 implementation in Keras for beginners. *Medium*, 1-8.
- Tilawah, S. (2020). Adam Optimizer. *Medium*.
- Turban, E. (2005). *Decision Support Systems and Intelligent System*. Yogyakarta: Penerbit Andi.
- Wahyono, T. (2018). *Fundamental of Python for Machine Learning (Dasar-dasar Pemrograman Python untuk Machine Learning dan Kecerdasan Buatan)*. Yogyakarta: Penerbit Gava Media.
- Xiao, J., Wang, J., Cao, S., & Li, B. (2023). Application of a Novel and Improved VGG-19 Network in the Detection of Workers Wearing Masks. *Journal of Physics: Conference Series*, 1-6.
- Yang, N., Zhong, L., Huang, F., Yuan, D., & Bao, W. (2023). Random Padding and Stride Data Augmentation. *Computer Vision and Pattern Recognition*, 1-8.
- Yu Lee, C., Gallagher, P. W., & Tu, Z. (2015). Generalizing Pooling Functions in Convolutional Neural Networks: Mixed, Gated, and Tree. *Machine Learning*.
- Yuhandri. (2019). Perbandingan Metode Cropping Pada Sebuah Citra untuk Pengambilan Motif Tertentu pada Kain Songket Sumatera Barat. *KomputerTeknologiInformasi*, 96-105.

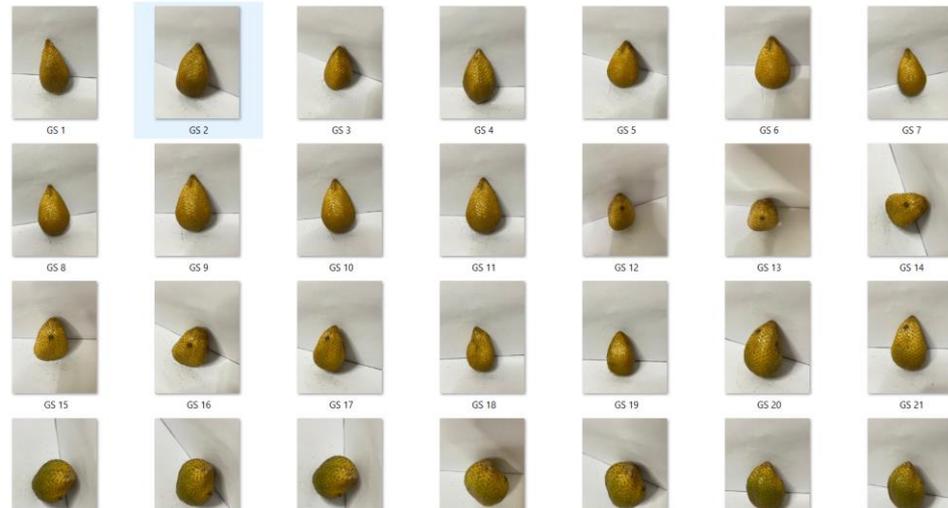
LAMPIRAN

Lampiran 1 Contoh Data Gambar Salak

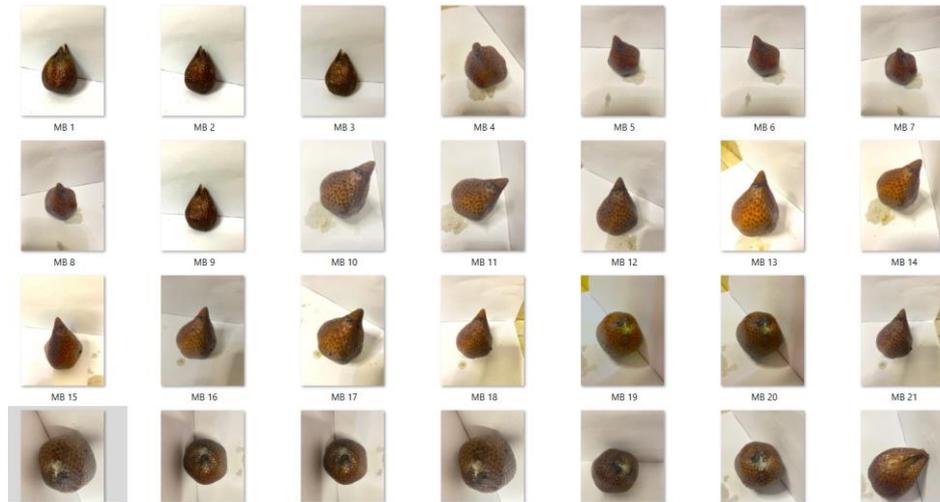
a. Salak Gading Busuk



b. Salak Gading Sehat



c. Salak Madu Busuk



d. Salak Madu Sehat



e. Salak Pondoh Busuk



f. Salak Pondoh Sehat



Lampiran 2 Program Python

```
#Import Package
import os
import cv2
import pickle
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import matplotlib.image as mpimg
import keras
import tensorflow
from tensorflow.keras.models import Model
from tensorflow.keras.utils import plot_model
from tensorflow.keras.models import Sequential
from tensorflow.keras.applications import VGG19
from tensorflow.keras.callbacks import EarlyStopping
from tensorflow.keras.preprocessing.image import ImageDataGenerator
```

```

from tensorflow.keras.layers import Input, Lambda, Dense, Flatten,
Dropout, BatchNormalization, Activation
from sklearn.metrics import confusion_matrix, classification_report,
accuracy_score, recall_score, precision_score, f1_score

#Input data
train_path = r"D:\Semester 7\TA\Data\Train"
test_path = r"D:\Semester 7\TA\Data\Test"
val_path = r"D:\Semester 7\TA\Data\Valid"

#Menampilkan 2 data pertama
for folder in os.listdir(train_path):
    sub_path = train_path + "/" + folder

    print(folder)
    for i in range(2):
        temp_path = os.listdir(sub_path)[i]
        temp_path = sub_path + "/" + temp_path
        img = mpimg.imread(temp_path)
        imgplot = plt.imshow(img)
        plt.show()

#Membaca data di directory dan mengubah ukuran
def imagearray(path, size):
    data = []
    for folder in os.listdir(path):
        sub_path=path+"/"+folder

        for img in os.listdir(sub_path):
            image_path=sub_path+"/"+img
            img_arr=cv2.imread(image_path)
            img_arr=cv2.resize(img_arr, size)
            data.append(img_arr)

    return data

#Mengubah ukuran gambar menjadi 224x224 piksel dan kedalam bentuk array
size = (224,224)

train = imagearray(train_path, size)
test = imagearray(test_path, size)
val = imagearray(val_path, size)

#Menampilkan gambar pertama dari array train_array
plt.imshow(train[0])
plt.show()

#Mengubah data menjadi array numpy
x_train = np.array(train)
x_test = np.array(test)
x_val = np.array(val)

#Mengubah rentang nilai piksel dalam array gambar (matriks)
x_train = x_train/255.0
x_test = x_test/255.0
x_val = x_val/255.0

```

```

#normalisasi data
def data_class(data_path, size, class_mode):
    datagen = ImageDataGenerator(rescale = 1./255)
    classes = datagen.flow_from_directory(data_path,
                                         target_size = size,
                                         batch_size = 32,
                                         class_mode = class_mode)

    return classes

#data_class u/ tugas klasifikasi multikelas, sparse u/ melabelkan mnjd
bilangan bulat
train_class = data_class(train_path, size, 'sparse')
test_class = data_class(test_path, size, 'sparse')
val_class = data_class(val_path, size, 'sparse')

#Class yang bersesuaian dengan x train
y_train = train_class.classes
y_test = test_class.classes
y_val = val_class.classes

train_class.class_indices

y_train.shape,y_test.shape,y_val.shape

# Model VGG-19
vgg = VGG19(input_shape = (224, 224, 3), weights = 'imagenet',
include_top = False)
for layer in vgg.layers:
    layer.trainable = False

x = Flatten() (vgg.output)
prediction = Dense(6, activation='softmax')(x)

model = Model(inputs=vgg.input, outputs=prediction)
model.summary()
model.compile(
    loss='sparse_categorical_crossentropy',
    optimizer="adam",
    metrics=['accuracy']
)

early_stop = EarlyStopping(monitor = 'val_loss', mode='min', verbose = 1,
patience = 5)

history = model.fit(x_train, y_train, validation_data = (x_val, y_val),
epochs = 5, callbacks=[early_stop], batch_size = 32)

plt.figure(figsize=(10, 8))
plt.plot(history.history['accuracy'], label='train acc')
plt.plot(history.history['val_accuracy'], label='val acc')
plt.legend()
plt.title('Accuracy')
plt.show()

plt.figure(figsize=(10, 8))
plt.plot(history.history['loss'], label='train loss')
plt.plot(history.history['val_loss'], label='val loss')
plt.legend()
plt.title('Loss')
plt.show()

#evaluasi model dengan data test
model.evaluate(x_test, y_test, batch_size=32)

```

```

#prediksi dengan bentuk probabilitas
y_pred = model.predict(x_test)

#mengubah hasil prediksi yang awalnya dalam bentuk probabilitas menjadi
label kelas yang sesuai.
y_pred=np.argmax(y_pred,axis=1)

print(classification_report(y_pred,y_test))

cm = confusion_matrix(y_pred,y_test)

plt.figure(figsize=(20, 20))
ax = plt.subplot()
sns.set(font_scale=2.0)
sns.heatmap(cm, annot=True, fmt='g', cmap="Blues", ax=ax);

# labels, title and ticks
ax.set_xlabel('Predicted labels', fontsize=20);ax.set_ylabel('True
labels', fontsize=20);
ax.set_title('Confusion Matrix Using The Test Data', fontsize=20);
ax.xaxis.set_ticklabels(['Gading Busuk', 'Gading Sehat', 'Madu Busuk',
'Madu Sehat', 'Pondoh Busuk', 'Pondoh Sehat'], fontsize=20);
ax.yaxis.set_ticklabels(['Gading Busuk', 'Gading Sehat', 'Madu Busuk',
'Madu Sehat', 'Pondoh Busuk', 'Pondoh Sehat'], fontsize=20);

f1_score(y_test, y_pred, average='micro')

recall_score(y_test, y_pred, average='weighted')

precision_score(y_test, y_pred, average='micro')

```

Lampiran 3 Publikasi

The screenshot shows a web browser window displaying the review page for a submission in the journal 'Knowledge Engineering and Data Science'. The page is titled '#50340 Review' and shows the submission details, peer review history, and editor decision.

Submission Details:

Authors	Indina Isyfi Annie Widyan, Atina Ahdika
Title	Decoding Yogyakarta's Snake Fruit: Exploring Image Classification with VGG-19 Convolutional Neural Networks
Section	Articles
Editor	Aji Wibawa

Peer Review:

Round 1	
Review Version	50340-165061-1-RV/DOC 2024-01-26
Initiated	2024-02-16
Last modified	2024-02-17
Uploaded file	None

Editor Decision:

Decision	—
Notify Editor	Editor/Author Email Record No Comments
Editor Version	None
Author Version	None
Upload Author Version	<input type="button" value="Pilih File"/> Tidak ada file yang dipilih <input type="button" value="Upload"/>

The page also includes a navigation menu (HOME, ABOUT, USER HOME, SEARCH, CURRENT, ARCHIVES) and a sidebar with links to various journal resources like Editorial Board, Aim & Scope, Author Guidelines, etc.