

**KLASIFIKASI KEMATANGAN BUAH NANAS BERDASARKAN WARNA
DENGAN METODE *DEEP LEARNING CONVOLUTIONAL NEURAL
NETWORK (CNN)***

SKRIPSI

MUHAMMAD SETIAWAN

191401108



PROGRAM STUDI S-1 ILMU KOMPUTER

FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI

UNIVERSITAS SUMATERA UTARA

MEDAN

2023

**KLASIFIKASI KEMATANGAN BUAH NANAS BERDASARKAN WARNA
DENGAN METODE *DEEP LEARNING CONVOLUTIONAL NEURAL
NETWORK (CNN)***

SKRIPSI

Ditujukan untuk memenuhi persyaratan penyelesaian studi dan mendapatkan gelar
Sarjana Ilmu Komputer

MUHAMMAD SETIAWAN

191401108



PROGRAM STUDI S-1 ILMU KOMPUTER

FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI

UNIVERSITAS SUMATERA UTARA

MEDAN

2023

PERSETUJUAN

PERSETUJUAN

Judul : KLASIFIKASI KEMATANGAN BUAH NANAS
BERDASARKAN WARNA DENGAN METODE
DEEP LEARNING CONVOLUTIONAL NEURAL
NETWORK (CNN)

Kategori : SKRIPSI

Nama : MUHAMMAD SETIAWAN

Nomor Induk Mahasiswa : 191401108

Program Studi : SARJANA (S-1) ILMU KOMPUTER

Fakultas : ILMU KOMPUTER DAN TEKNOLOGI
INFORMASI UNIVERSITAS SUMATERA UTARA

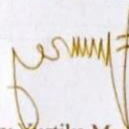
Telah diuji dan dinyatakan lulus di Medan, 27 Oktober 2023

Dosen Pembimbing II



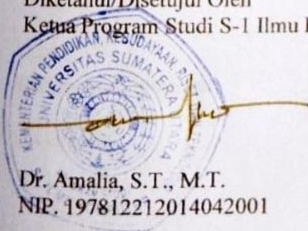
Prof. Dr. Syahril Efendi, S.Si., M.IT.
NIP. 196711101996021001

Dosen Pembimbing I



Fuzy Yustika Manik, S.Kom., M.Kom
NIP. 198710152019032010

Diketahui/Dijetujui Oleh
Ketua Program Studi S-1 Ilmu Komputer


Dr. Amalia, S.T., M.T.
NIP. 197812212014042001

Universitas Sumatera Utara

PERNYATAAN

KLASIFIKASI KEMATANGAN BUAH NANAS BERDASARKAN WARNA DENGAN METODE *DEEP LEARNING CONVOLUTIONAL NEURAL NETWORK (CNN)*

SKRIPSI

Saya ingin menegaskan bahwa skripsi ini adalah hasil dari upaya penelitian pribadi saya, dengan pengecualian beberapa kutipan dan ringkasan yang saya identifikasi sumbernya.

Medan, 19 September 2023



Muhammad Setiawan

191401108

KATA PENGANTAR

Dengan penuh rasa syukur yang mendalam dan ungkapan terima kasih yang tulus kepada Allah SWT atas berkat-Nya, penulis berhasil proyek tugas akhir judul “KLASIFIKASI KEMATANGAN BUAH NANAS BERDASARKAN WARNA DENGAN METODE *DEEP LEARNING CONVOLUTIONAL NEURAL NETWORK (CNN)*”.

Penelitian ini merupakan bagian integral dari persyaratan kelulusan dalam program sarjana Universitas Sumatera Utara. Meskipun menghadapi berbagai tantangan selama proses penulisan, berkat panduan dan dukungan yang diberikan oleh berbagai pihak, terutama pembimbing, penulis berhasil mengatasi semua hambatan dengan sukses.

Dalam penyusunan skripsi ini, perlu diakui bahwa ada kelemahan, baik dalam hal kualitas maupun jumlah materi penelitian yang telah disajikan. Hal ini mencerminkan batasan-batasan yang dihadapi oleh penulis. Penulis dengan sepenuh kesadaran menyadari bahwa skripsi ini masih jauh dari sempurna, sehingga dengan sungguh-sungguh penulis mengharapkan umpan balik konstruktif untuk guna perbaikan dan pengembangan lebih lanjut.

Selama penelitian ini dilakukan dalam rangka penyusunan skripsi, penulis menghargai dukungan dan kolaborasi yang telah diberikan oleh berbagai pihak yang memiliki peran yang sangat penting. Oleh karena itu, dengan tulus dan tulus hati, penulis ingin mengucapkan terima kasih yang mendalam kepada:

1. Rektor Universitas Sumatera Utara, Dekan Fakultas Ilmu Komputer dan Teknologi Informasi, dan Ketua Prodi S-1 Ilmu Komputer Universitas Sumatera Utara.
2. Keluarga, termasuk Ayah, Ibu, Adik-adik, dan Sanak Saudara yang selalu memberikan dukungan dalam berbagai bentuk serta mendoakan kelancaran penulisan skripsi ini.
3. Ibu Fuzy Yustika Manik, S.Kom.,M.Kom, selaku dosen pembimbing I, yang memberikan bimbingan dan arahan yang sangat berharga dalam penyusunan skripsi ini.

4. Bapak Prof. Dr. Syahril Efendi, S.Si., M.IT, selaku dosen pembimbing II, yang memberikan panduan dan bantuan dalam penyusunan skripsi ini.
5. Semua dosen program studi Ilmu Komputer Fakultas Ilmu Komputer yang berbagi pengetahuan dan wawasan berharga.
6. Teman-teman yang telah memberikan dukungan, semangat, dan berbagi pengalaman sepanjang penyelesaian skripsi ini.

Terakhir, semoga bantuan yang telah diberikan menjadi pahala yang diterima oleh Allah SWT. Semoga skripsi ini memberikan manfaat berkelanjutan dalam bidang ilmu komputer dan memiliki potensi untuk terus dikembangkan menjadi lebih baik di masa depan.

Medan, 19 September 2023

Penulis

ABSTRAK

Nanas (*Ananas Comocus L*), tumbuh subur di wilayah tropis dan sub-tropis. Buah nanas memiliki karakteristik unik, yaitu tekstur kenyal dan kaya akan vitamin C. Nanas termasuk kategori buah *Non Klimaterik*, proses klasifikasi kematangan buah nanas saat ini masih sangat bergantung pada analisis visual manusia. Faktor-faktor psikologis manusia dapat mempengaruhi hasil klasifikasi, mengakibatkan potensi kesalahan. penelitian ini bertujuan utama untuk mengklasifikasikan tingkat kematangan buah nanas berdasarkan warna, mencakup buah matang, setengah matang, dan belum matang. Tujuannya adalah memberikan solusi otomatis yang lebih akurat dan andal dalam proses klasifikasi, dengan menggunakan metode *Convolutional Neural Network (CNN)* dalam bidang *Deep Learning*, dengan menggunakan model VGG16 sebagai alat bantu. Sebanyak 598 data digunakan tahap pelatihan (*training*), sementara 40 data digunakan dalam tahap pengujian (*testing*), hasil menunjukkan tingkat akurasi 95%, dengan *precision* sekitar 93%, untuk *recall* sebesar 93% dan F1-score mencapai 93% . Dari temuan ini, dapat disimpulkan bahwa metode *Convolutional Neural Network* dengan model VGG16 mampu melakukan klasifikasi kematangan buah nanas berdasarkan warna secara akurat.

Kata kunci: *Convolutional Neural Network (CNN)*, Deep Learning, VGG16.

ABSTRACT

Pineapple (*Ananas Comocus L*) thrive in tropical and subtropical regions. Pineapples are characterized by their unique traits, including a firm texture and richness in Vitamin C. Pineapple belong to the *Non-Climacteric* fruit category, and currently, the maturity classification process for pineapple heavily relies on human visual analysis. Human psychological factors can influence the classification results, potentially leading to errors. The main objective of this research is to classify the ripeness level of pineapple based on color, encompassing ripe, semi-ripe, and unripe categories. The aim to provide a more accurate and reliable automated solution in the classification process, utikizing, with the VGG16 model as a tool. A total of 598 data points were used for the training phase, while 40 data points were used for testing. The results showed and accuracy rate 95%, with *precision* at around 93%, a *recall* of 93%, and an F1-score reaching 93%. From these findings, it can be concluded that the Convolutional Neural Network method with the VGG16 model is capable of accurately classifying the ripeness of pineapple based on color.

Key Word: *Convolutional Neural Network* (CNN), Deep Learning, VGG16.

DAFTAR ISI

PERSETUJUAN	iii
PERNYATAAN	iv
KATA PENGANTAR	v
ABSTRAK.....	vii
ABSTRACT	viii
DAFTAR ISI	ix
DAFTAR GAMBAR	xii
DAFTAR TABEL	xiii
BAB I PENDAHULUAN	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah	3
1.3 Batasan Masalah	3
1.4 Tujuan Penelitian.....	4
1.5 Manfaat Penelitian.....	4
1.6 Metode Penelitian	4
1.7 Penelitian Relevan	5
1.8 Sistematika Penulisan.....	6
BAB 1 PENDAHULUAN.....	6
BAB 2 LANDASAN TEORI.....	7
BAB 3 ANALISIS DAN PERANCANGAN SISTEM.....	7
BAB 4 IMPLEMENTASI DAN PENGUJIAN	7
BAB 5 KESIMPULAN DAN SARAN	7
BAB 2 TINJAUAN PUSTAKA	8
2.1 Nanas.....	8
2.2 Tingkat Kematangan Buah Nanas	8
2.3 Machine Learning.....	9
2.4 Deep Learning	9
2.5 Convolutional Neural Network	9
2.6 Arsitektur VGG	17

2.7 Neural Network.....	18
BAB 3 ANALISIS DAN PERANCANGAN SISTEM.....	19
3.1 Analisis Sistem	19
3.1.1 Analisis Masalah.....	19
3.1.2 Analisis Kebutuhan	20
3.1.3 Analisis Proses	21
3.1.4 Arsitektur Umum	23
3.2 Pemodelan Sistem	24
3.2.1 <i>Use Case diagram</i>	24
3.2.2 <i>Activity diagram</i>	24
3.2.3 <i>Sequence diagram</i>	26
3.3 Flowchart.....	27
3.4 Perancangan Antarmuka Sistem.....	28
BAB 4 IMPLEMENTASI DAN PENGUJIAN	30
4.1 Implementasi sistem	30
4.1.1 Perangkat Keras (<i>Hardware</i>).....	30
4.1.2 Perangkat Lunak (<i>Software</i>)	30
4.2 Pengumpulan <i>Dataset</i>	30
4.3 Persiapan dan Pelabelan <i>Dataset</i>	30
4.4 Ekstraksi Ciri	31
4.5 Implementasi Tahapan Antarmuka.....	32
4.6 Perancangan Model	34
4.6.1 Perancangan Model CNN	34
4.6.2 Perancangan Model VGG16	37
4.7 Penentuan Parameter Model	40
4.6.1 Pengaruh Banyaknya Epoch	40
4.8 Pengujian Sistem.....	41
4.7.1 Hasil Proses Training.....	41
4.7.2 Hasil Proses Testing	43
BAB 5 KESIMPULAN DAN SARAN	56
5.1 Kesimpulan.....	56

5.2 Saran	56
DAFTAR PUSTAKA	57

DAFTAR GAMBAR

Gambar 2.1 Tingkat Kematangan Buah Nanas.....	9
Gambar 2.2 Flowchart Algoritma CNN.....	10
Gambar 2.3 Arsitektur CNN	11
Gambar 2.4 Model Ilustrasi Arsitektur VGG-16	17
Gambar 2.5 Struktur dari Neural Network	18
Gambar 3.1 Fishbone Diagram	19
Gambar 3.2 Buah Nanas Matang	21
Gambar 3.3 Buah Nanas Setengah Matang	22
Gambar 3.4 Buah Nanas Belum Matang	22
Gambar 3.5 Ilustrasi Arsitektur Umum Sistem.....	23
Gambar 3.6 Use Case Diagram.....	24
Gambar 3.7 Activity Diagram.....	25
Gambar 3.8 Sequence Diagram	26
Gambar 3.9 Flowchart Sistem.....	27
Gambar 3.10 Halaman awal.....	28
Gambar 3.11 Halaman Prediksi	29
Gambar 4.1 Warna pada buah nanas matang, belum matang, dan setengah matang....	31
Gambar 4.2 Ekstraksi Fitur Warna RGB	32
Gambar 4.3 Tampilan Antarmuka	33
Gambar 4.4 Summary Model CNN	34
Gambar 4.5 Implementasi Model CNN	35
Gambar 4.6 Summary Model VGG	37
Gambar 4.7 Summary Model VGG	38
Gambar 4.8 Hasil Prediksi proses training.....	42
Gambar 4.9 Hasil akurasi proses training	42
Gambar 4.10 Grafik akurasi dan loss.....	43

DAFTAR TABEL

Tabel 3.1 Rincian Dataset	23
Tabel 4.1 Tabel Banyaknya Epoch	41
Tabel 4.2 Tabel Pengujian Model	44
Tabel 4.3 Tabel confusion matriks	53
Tabel 4.4 Tabel precision	54
Tabel 4.5 Tabel recall	55

BAB I

PENDAHULUAN

1.1 Latar Belakang

Nanas, yang dikenal dengan nama ilmiah (*Ananas Comocus L*), merupakan salah satu varietas tumbuhan yang biasanya tumbuh subur di daerah tropis dan subtropis. Nanas ialah buah yang mempunyai tekstur menarik dan kaya akan Vitamin C. Selain kandungan nutrisinya, nanas juga dapat dijadikan konsumsi langsung atau diolah menjadi berbagai produk dengan menggunakan buah nanas sebagai bahan dasar. Yang menarik, dalam proses pengolahan ini, ciri khas rasa dan kandungan gizi buah nanas tetap terjaga.

Salah satu ciri khas nanas adalah teksturnya yang kenyal dan kandungan pektinnya. Tekstur kenyal ini memberikan pengalaman yang unik saat mengonsumsi buah nanas. Pektin, yang banyak terkandung dalam nanas, adalah sejenis serat larut yang memberikan tekstur tersebut.

Nanas juga termasuk dalam kategori buah Non Klimaterik. Artinya, setelah mencapai tingkat kematangan buah tertentu, nanas tidak dapat matang lebih lanjut. Oleh karena itu, buah nanas harus dipetik langsung dari pohonnya saat sudah matang. Jika dipetik saat sebelum matang, nanas tidak akan mencapai tingkat kematangan yang diinginkan meskipun dibiarkan di tempat penyimpanan.

Proses klasifikasi nanas menjadi krusial. Mengidentifikasi nanas yang sudah matang memastikan buah memiliki rasa dan kualitas yang diinginkan. Memetik nanas saat masih asam tidak akan menghasilkan buah yang manis, karena tidak ada pematangan lanjutan setelah dipanen.

Pada saat ini, langkah klasifikasi buah nanas masih bergantung pada metode manual yang melibatkan pengamatan visual secara langsung pada buah-buah tersebut. Meskipun demikian, pendekatan ini memiliki kekurangan karena sangat bergantung pada penilaian subjektif dari operator yang melakukan penyortiran. Situasi ini berpotensi menimbulkan ketidaksesuaian dalam proses klasifikasi, terutama ketika operator memiliki penilaian yang beragam.

Mengembangkan metode klasifikasi nanas yang lebih objektif dan konsisten telah menjadi perhatian utama dalam usaha meningkatkan kualitas dan efisiensi dalam pemrosesan nanas.

Beberapa penelitian sebelumnya telah menerapkan algoritma Deep Learning *Convolutional Neural Network* (CNN) untuk mengidentifikasi dan mengklasifikasi enam jenis buah yang berbeda yang menggabungkan CNN dengan model VGG16 kelayakan model ini, rasio yang dihasilkan mencapai 100% N. Abu-Jamie, Tanseem.(2022).

Berdasarkan penelitian Lestari, Sri. (2022) dalam pengenalan penyakit pada daun kentang, berhasil membuktikan keberhasilan klasifikasi menggunakan metode *Deep Learning* dengan arsitektur CNN. Klasifikasi ini melibatkan tiga kelas, yaitu daun yang sehat, daun dengan penyakit *early blight*, dan daun dengan penyakit *late blight*. Pada *epoch* ke-10 dengan ukuran batch size 20 hasil training akurasi mencapai akurasi sebesar 95% dan akurasi validasi mencapai 94%.

Berdasarkan penelitian M. AL-Masawabe, Marah. (2021) untuk klasifikasi buah pepaya jika sudah matang atau matang sebagian atau belum matang. Model terlatih mencapai akurasi 100% pada serangkaian pengujian yang diadakan, yang menunjukkan kelayakan model klasifikasi VGG16 mencapai akurasi 100% dan waktu latihan 112 detik.

Berdasarkan penelitian B. MacEachern., Craig. (2023) dalam mendeteksi kematangan blueberry liar, yang dikembangkan dengan YOLOv4 yang memiliki presisi rata-rata masing-masing 79,79 % dan 88,12 %, YOLOv4 meraih skor F1 tertinggi sebesar 0,82. YOLOV4-Tiny melakukan yang terbaik dari perspektif beban komputasi memiliki inferensi waktu rata-rata 7,8 ms dan penggunaan memori rata-rata 1,6 GB untuk 1280 gambar 736 pixel.

Mengacu pada latar belakang dan penelitian sebelumnya, penelitian ini akan fokus pada pengembangan klasifikasi kematangan buah nanas berdasarkan warna dengan menggunakan metode *Deep Learning Convolutional Neural Network* (CNN). Tujuan utama dari penelitian ini adalah untuk mengidentifikasi buah nanas yang telah matang dengan memanfaatkan informasi warna pada kulit buah nanas.

1.2 Rumusan Masalah

Saat ini, tantangan muncul dalam proses mengklasifikasikan tingkat kematangan buah nanas, yang masih bergantung pada tindakan manusia dalam melakukan analisis visual. Namun, metode ini dapat terpengaruh oleh aspek psikologis pengamat, berpotensi menyebabkan kesalahan dalam mengidentifikasi tingkat kematangan yang sesuai pada buah nanas.

Sebagai solusi, diperlukan pengembangan aplikasi berbasis web yang memanfaatkan komputer dan algoritma *Deep Learning Convolutional Neural Network* (CNN) untuk melakukan klasifikasi tingkat kematangan buah nanas berdasarkan warna. Dengan pendekatan ini, proses klasifikasi akan menjadi lebih objektif dan akurat, mengurangi ketergantungan pada penilaian subjektif manusia.

1.3 Batasan Masalah

Dalam lingkup penelitian ini, akan ada sejumlah batasan masalah di antaranya:

1. Menggunakan algoritma *Deep Learning Convolutional Neural Network* dan model *VGG16*.
2. Penelitian ini hanya mendeteksi kematangan buah nanas.
3. Output yang dihasilkan adalah hasil klasifikasi buah nanas matang, setengah matang, atau belum matang sesuai dengan ketentuan standar tingkat kematangan buah nanas.

Indikator yang digunakan untuk mengidentifikasi tingkat kematangan buah nanas adalah sebagai berikut:

1. Buah nanas matang (kulit buah berwarna kuning dan terdapat warna jingga kecoklatan).
 2. Buah nanas setengah matang (kulit buah memiliki perpaduan warna kuning, warna jingga dan terdapat warna hijau).
 3. Buah nanas belum matang (kulit buah berwarna hijau).
3. Sistem dirancang berbasis website dengan menggunakan Bahasa pemrograman python.

1.4 Tujuan Penelitian

Maksud dari penelitian ini adalah untuk merancang dan mengembangkan sebuah aplikasi web yang memiliki kemampuan untuk melakukan klasifikasi kematangan buah nanas berdasarkan warna, menggunakan metode *Deep Learning Convolutional Neural Network (CNN)*. Tambahan dari itu, penelitian ini juga memiliki tujuan untuk mengukur sejauh mana tingkat akurasi CNN dalam melakukan klasifikasi tingkat kematangan buah nanas.

1.5 Manfaat Penelitian

Dalam penelitian ini, diperoleh beberapa manfaat, yaitu:

1. Membantu dalam proses pengelompokan tingkat kematangan buah nanas.
2. Menyediakan alternatif dalam pengklasifikasian kematangan buah nanas selain dari pendekatan manual, yaitu melalui penerapan teknologi.
3. Menyediakan potensi sebagai acuan bagi penelitian mendatang yang terkait dengan pengkategorian tingkat kematangan buah nanas menggunakan warna kulit, menggunakan metode *Deep Learning* melalui *Convolutional Neural Network (CNN)*.

1.6 Metode Penelitian

Dalam penelitian ini, pendekatan yang diterapkan adalah sebagai berikut:

1. Studi Pustaka

Pada bagian ini, penelitian akan dimulai dengan mengumpulkan beberapa referensi dari berbagai sumber yang dianggap andal, dan dilanjutkan dengan melakukan tinjauan literatur melalui berbagai media seperti buku, jurnal, e-book, artikel ilmiah, makalah, dan situs web yang relevan dengan topik *Deep Learning*, Pengolahan Citra, dan *Convolutional Neural Network (CNN)*, dengan tujuan untuk mengidentifikasi parameter kematangan buah nanas.

2. Analisis dan Perancangan

Dengan merujuk pada batasan masalah penelitian, peneliti akan melakukan analisis terhadap komponen-komponen yang diperlukan dalam penelitian. Analisis menjadi dasar untuk merancang berbagai elemen seperti *flowchart*, *fishbone diagram*, *activity diagram*, *use case diagram*, *sequence diagram*, dan desain *interface*. Dengan

pendekatan ini, peneliti dapat menghasilkan bahwa semua aspek penelitian tercakup dengan baik dalam rancangan visual yang sesuai.

3. Implementasi

Pada tahap ini, langkah selanjutnya melibatkan pembangunan sistem yang akan dikonstruksi sesuai dengan diagram alur yang telah disusun sebelumnya. Selanjutnya, proses ini akan dilanjutkan menggunakan Bahasa pemrograman Python sebagai dasar dalam pembuatan situs web yang akan dibangun.

4. Pengujian

Dalam tahap ini, sistem yang telah dirancang akan menjalani pengujian pada implementasi yang sebenarnya, dengan maksud untuk memverifikasi bahwa sistem beroperasi sesuai dengan ekspektasi yang telah ditetapkan. Jika masih terdapat kesalahan atau kekurangan pada website, perbaikan akan dilakukan untuk mengatasi masalah tersebut. Langkah ini dimaksudkan untuk memverifikasi bahwa sistem yang dikembangkan berjalan dengan efektif dan memenuhi dengan baik tujuan serta kebutuhan yang telah ditentukan sebelumnya.

5. Dokumentasi

Pada langkah ini, seluruh proses penelitian yang sudah dilakukan akan dilanjutkan dokumentasi secara komprehensif, dimulai dari tahap analisis hingga tahap akhir yaitu pengujian. Dokumentasi ini akan diwujudkan dalam bentuk skripsi tugas akhir, yang merupakan laporan tertulis yang menyajikan semua aspek penelitian secara terstruktur dan rinci.

1.7 Penelitian Relevan

Dibawah ini disajikan sejumlah penelitian yang memiliki kaitan dengan topik penelitian ini.

1. Berdasarkan penelitian (N. Abu-Jamie, Tanseem., et al. 2022) yang berjudul “*Six Fruits Using Deep Learning*” untuk mengidentifikasi dan mengklasifikasi enam jenis buah yang berbeda yang menggabungkan CNN dengan model VGG16 kelayakan model ini, rasio yang dihasilkan mencapai 100%.
2. Berdasarkan penelitian (Lestari, Sri., et al. 2022) yang berjudul “Implementasi Deteksi Objek Penyakit Daun Kentang Dengan Metode Convolutional Neural

Network” dalam mendeteksi penyakit pada daun kentang, klasifikasi yang dilakukan dengan tiga kelas berupa *daun sehat*, *early blight*, dan *late blight* menggunakan metode *Deep Learning* dengan arsitektur *Convolutional Neural Network* dianggap berhasil karena pada *epoch* ke 10 dengan batch size 20 menghasilkan training akurasi 95% dan akurasi validasi 94%.

3. Berdasarkan penelitian (M. AL-Masawabe, Marah., et al. 2021) yang berjudul “Papaya maturity Classification Using Deep Convolutional Neural Networks” untuk klasifikasi status papaya jika sudah matang atau matang sebagian atau belum matang. Model terlatih mencapai akurasi 100% pada serangkaian pengujian yang diadakan, yang menunjukkan kelayakan model klasifikasi VGG16 mencapai akurasi 100% dan waktu latihan 112 detik.
4. Berdasarkan penelitian (B. MacEachern., Craig., et al. 2023) yang berjudul “Detection of Fruit Maturity Stage and Yield Estimation in Wild Blueberry using deep learning convolutional neural networks” dalam mendeteksi kematangan blueberry liar, yang dikembangkan dengan YOLOv4 yang memiliki presisi rata-rata masing-masing 79,79% dan 88,12%, YOLOv4 meraih skor F1 tertinggi sebesar 0,82. YOLOv4-Tiny melakukan yang terbaik dari perspektif beban komputasi memiliki inferensi waktu rata-rata 7,8ms dan penggunaan memori rata-rata 1,6 GB untuk 1280 gambar 736 pixel.

1.8 Sistematika Penulisan

Penyusunan kerangka skripsi ini mengikuti beberapa bagian yang diatur secara berurutan, seperti yang dijabarkan di bawah ini:

BAB 1 PENDAHULUAN

Pada bagian ini, akan dijelaskan mengenai konteks awal dari masalah, pengidentifikasian masalah, batasan lingkup masalah, dampak dari hasil penelitian, langkah-langkah yang diambil dalam penelitian, penelitian sebelumnya yang relevan, dan pengaturan struktur tulisan.

BAB 2 LANDASAN TEORI

Pada bab ini, dijelaskan tentang beberapa teori mengenai deep learning, tingkat kematangan buah nanas, dan algoritma *Convolutional Neural Network* (CNN).

BAB 3 ANALISIS DAN PERANCANGAN SISTEM

Bagian ini mencakup analisis proses, perancangan sistem antarmuka dari website klasifikasi kematangan buah nanas berdasarkan warna dengan menggunakan algoritma *Convolutional Neural Network* (CNN)

BAB 4 IMPLEMENTASI DAN PENGUJIAN

Pada bagian ini, akan diuraikan mengenai pelaksanaan implementasi sistem dan hasil pengujian yang telah berhasil diselesaikan.

BAB 5 KESIMPULAN DAN SARAN

Bagian ini melibatkan rangkuman temuan dari penelitian serta rekomendasi yang diberikan oleh peneliti yang dapat memberikan kontribusi pada penelitian yang akan datang.

BAB 2

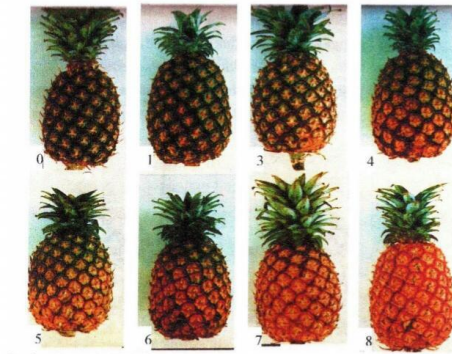
TINJAUAN PUSTAKA

2.1 Nanas

Nanas (*Ananas Comocus L*) merupakan salah satu jenis tanaman varietas yang umumnya ditemui di wilayah tropis dan subtropis. Asal-usul tanaman nanas berasal dari benua Amerika, khususnya Brazil dan Peru, dan kemudian menyebar ke berbagai wilayah termasuk Indonesia, tanaman nanas dapat tumbuh mulai dari ketinggian dataran rendah hingga dataran tinggi. Beberapa daerah yang terkenal sebagai produsen buah nanas di Indonesia meliputi Subang, Bogor, Palembang, dan Pekanbaru. Nanas merupakan jenis tanaman yang beradaptasi dengan kondisi kering dan memiliki kemampuan menyimpan air dalam jumlah yang cukup, sehingga tanamman ini memiliki kemampuan bertahan di lingkungan yang cenderung kering. Selain itu, buah nanas juga sering disebut sebagai “ratu buah” dalam beberapa kasus penamaan. (Sarkar, 2018).

2.2 Tingkat Kematangan Buah Nanas

Beberapa faktor yang digunakan dalam menilai kualitas buah nanas adalah kriteria klasifikasi yang diterapkan. Ketika buah nanas dipanen, salah satu metode umum dalam klasifikasinya adalah berdasarkan perubahan warna pada buah tersebut. Petani nanas sering melakukan identifikasi dan klasifikasi berdasarkan tingkat kematangan buah, yang biasanya dikelompokkan menjadi tiga kategori utama: buah yang masih mentah, buah nanas yang sudah matang, dan buah yang berada pada tahap setengah matang. Pendekatan ini memungkinkan petani untuk mengelompokkan buah-buah berdasarkan karakteristik warna dan kematangan, yang pada gilirannya dapat membantu dalam pengaturan waktu panen yang tepat. (Yanto, 2021).



Gambar 2.1 Tingkat Kematangan Buah Nanas

2.3 Machine Learning

Machine Learning berperan penting dalam mengelola dan meramalkan kumpulan data dengan mengimplementasikan algoritma untuk pembelajaran. Dengan menggunakan *Machine Learning*, komputer mampu melakukan proses pembelajaran secara otonom melalui analisis data yang telah diberikan. (Danukusumo, 2017) Sejalan dengan itu, pandangan dari (Shukla, 2018) menggambarkan *Machine Learning* sebagai suatu paradigm di mana perangkat lunak dapat belajar dari pengalaman sebelumnya. Program komputer akan diberi kemampuan untuk mengidentifikasi pola dan menerapkan pengetahuan yang diperoleh dari masa lalu.

2.4 Deep Learning

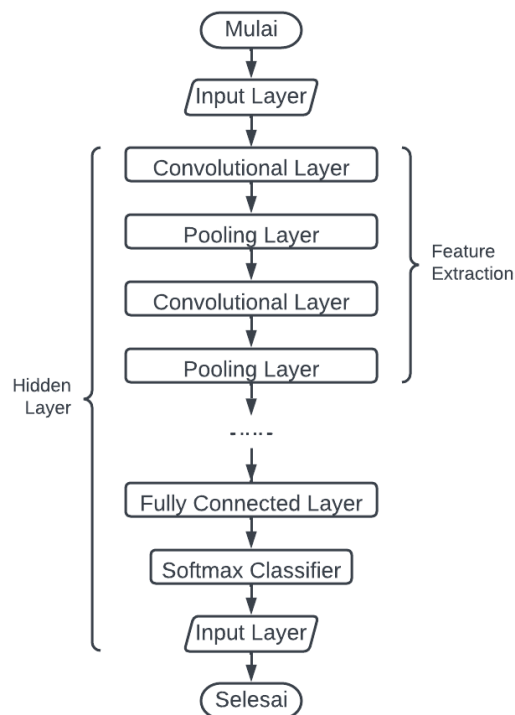
Deep Learning memfasilitasi pengembangan model komputasi yang terdiri dari beberapa tahap pemrosesan, memungkinkannya untuk menggali informasi dari data dengan beragam tingkat abstraksi. Melalui penggunaan algoritma *Convolutional Neural Network (CNN)*, *Deep Learning* mampu mengidentifikasi struktur yang kompleks dalam himpunan data besar. CNN memungkinkan mesin untuk mengatur parameter internal guna menghasilkan representasi pada tiap lapisan, yang didasarkan pada representasi lapisan sebelumnya. (LeCun, 2015).

2.5 Convolutional Neural Network

Convolutional Neural Network (CNN) mewakili struktur inti *Deep Learning*. Arsitektur CNN terdiri dari berbagai lapisan representasi yang mencakup fitur-fitur penting. Karakteristik yang mendalam ini memungkinkan CNN untuk secara otomatis

mengidentifikasi representasi khusus dari data melalui transformasi nonlinier dan pemodelan fungsi nonlinier.

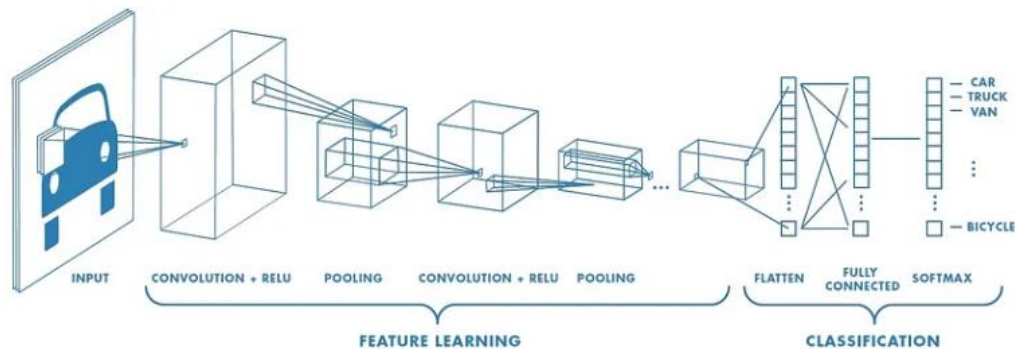
Struktur CNN memiliki komponen yang terdiri dari lapisan ekstraksi fitur. Proses ini terdiri dari lapisan konvolusi yang biasanya diikuti oleh *lapisan polling*, serta diakhiri dengan lapisan klasifikasi *softmax*. Pada lapisan konvolusi, fitur-fitur diambil dari citra gambar melalui operasi konvolusi. Lalu, pada *lapisan polling*, dimensi data dikurangi sehingga mengurangi kebutuhan komputasi. Kelebihan arsitektur ini adalah kemampuannya dalam menciptakan regulasi internal. Fitur-fitur yang diekstraksi oleh *konvolusi* dan *polling* dimasukkan ke dalam lapisan *softmax* untuk melanjutkan proses klasifikasi. (You, 2017).



Gambar 2.2 Flowchart Algoritma CNN

Proses dalam algoritma CNN, terbagi menjadi dua tahap pemrosesan utama, yaitu pada tahap pembelajaran fitur (*feature learning*) dan klasifikasi (*classification*). Pada tahap pembelajaran fitur, gambar dimasukkan mengalami proses ekstraksi fitur untuk mengidentifikasi nilai-nilai yang terkandung dalam gambar tersebut. Selanjutnya, hasil

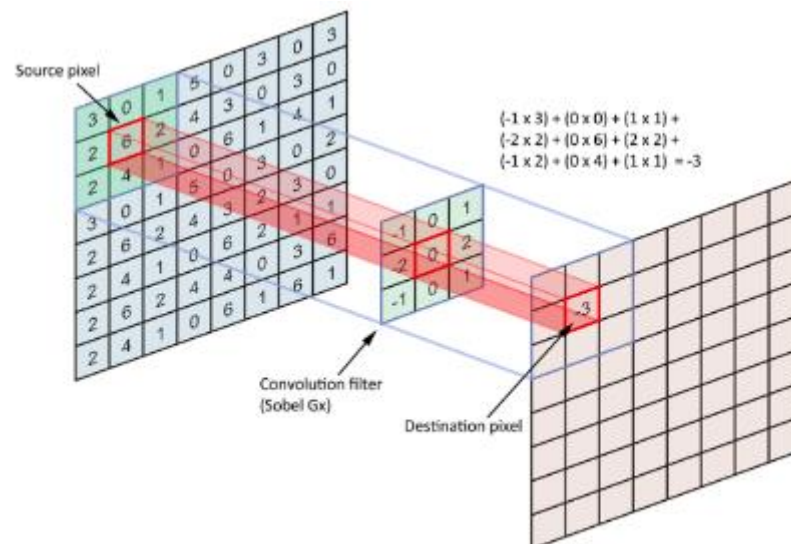
fitur tersebut dipelajari oleh algoritma. Contoh arsitektur CNN yang sesuai dengan konsep ini dapat ditemukan dalam: **Gambar 2.3**.



Gambar 2.3 Arsitektur CNN

2.5.1 Convolution Layer

Convolution Layer adalah lapisan yang terdiri dari neuron-neuron yang membentuk filter dengan dimensi panjang dan lebar tertentu. Input yang masuk ke lapisan ini akan melewati filter, menghasilkan peta fitur (*feature map*) sebagai output.



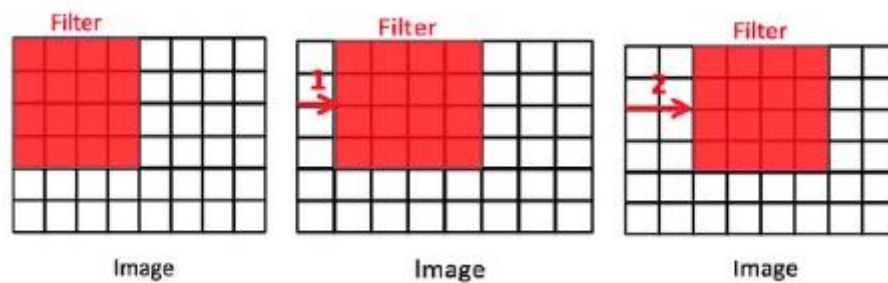
Sumber gambar: towardsdatascience.com

Gambar 2. 4 Convolution Layer

Hasil dari convolution layer akan diteruskan ke Pooling layer untuk mengurangi dimensinya.

2.5.1.1 Stride

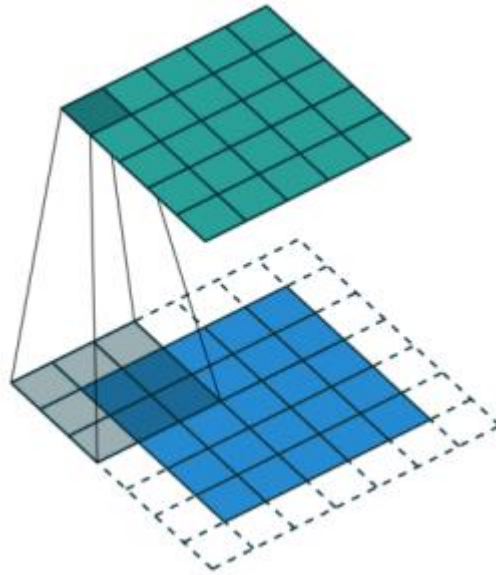
Stride adalah parameter pada convolution layer yang digunakan untuk mengatur seberapa jauh filter akan bergeser saat proses konvolusi. Semakin besar nilai stride, semakin besar jarak pergeseran filter, yang dapat mengurangi waktu komputasi, tetapi juga mengakibatkan informasi yang diambil menjadi lebih terbatas.



Gambar 2. 5 Stride

2.5.1.2 Padding

Padding digunakan untuk meningkatkan dimensi lapisan konvolusi (*convolution layer*). Penggunaan padding bertujuan agar setiap pixel dalam lapisan konvolusi dikunjungi dengan jumlah yang seragam selama proses konvolusi.



Gambar 2. 6 Padding

Penggunaan padding dapat membantu mengurangi resiko overfitting karena pixel-pixel tersebut dikunjungi secara sama, tetapi harus diingat bahwa penggunaan padding juga akan menghasilkan peta fitur (*feature map*) yang lebih besar sebagai output.

$$\text{Output} = \frac{W - N + 2P}{S} + 1$$

Dimana:

W = Panjang/Tinggi Input

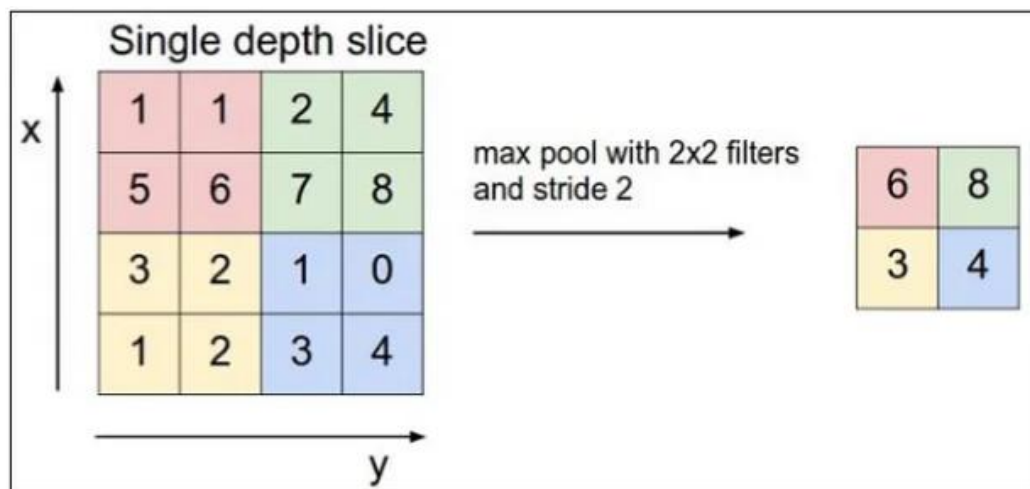
N = Panjang/Tinggi Filter

P = Zero Padding

S = Stride

2.5.2 Pooling Layer

Pooling layer memiliki prinsip kerja yang mirip dengan lapisan konvolusi (*convolution layer*). Yang membedakannya adalah bahwa filter dalam *lapisan Pooling* tidak digunakan untuk mendeteksi fitur, melainkan untuk melakukan operasi seperti mencari nilai maksimum (*Max Pooling*) atau nilai rata-rata (*Average Pooling*) dalam area tertentu.



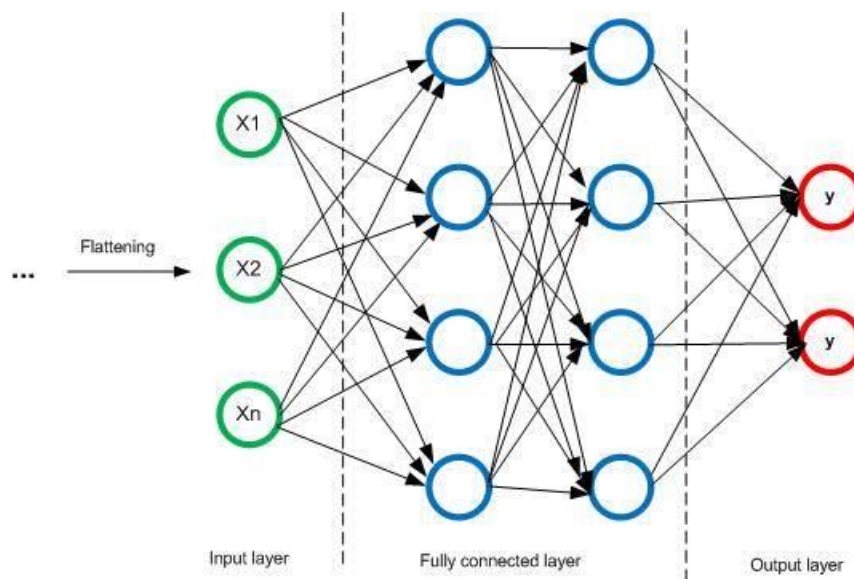
Gambar 2. 7 Pooling Layer

2.5.3 Fully-Connected Layer

Fully Connected layer adalah komponen dalam jaringan saraf tiruan yang memiliki peran klasifikasi data berdasarkan hasil yang diperoleh dari lapisan konvolusi (*convolution layer*). Lapisan *Fully Connected* menerima input

berbentuk *ID array*, sehingga hasil dari lapisan sebelumnya perlu diubah menjadi *ID array* terlebih dahulu.

Setiap neuron dalam lapisan *fully connected* terdiri dari dua elemen utama: fungsi regresi, yang berfungsi untuk melakukan prediksi output, dan fungsi aktivasi, yang digunakan untuk menentukan output yang akan disampaikan ke neuron berikutnya.



Gambar 2. 8 Fully-Connected Layer

$$y = \sum_{i=0}^n w^i x^i + b$$

Dimana:

W = Weight

X = Input dari neuron sebelumnya

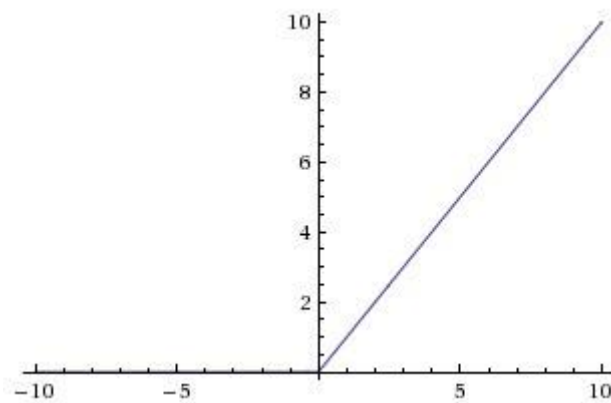
B = Bias

Ada beberapa jenis fungsi aktivasi yang umum digunakan dalam lapisan *fully connected*, di antaranya *sigmoid*, *softmax*, dan *ReLU* (Rectified Linear Unit).

2.5.4 Softmax

Softmax berperan dalam mengkonversi nilai-nilai logis menjadi tiga atau lebih output yang cocok untuk digunakan sebagai fungsi aktivasi pada dataset yang memiliki tiga kategori atau lebih. Selain itu, *softmax* juga menghasilkan output dalam bentuk probabilitas di mana total probabilitas semua output adalah 1 (satu).

2.5.5 ReLu



Gambar 2. 9 ReLu Function

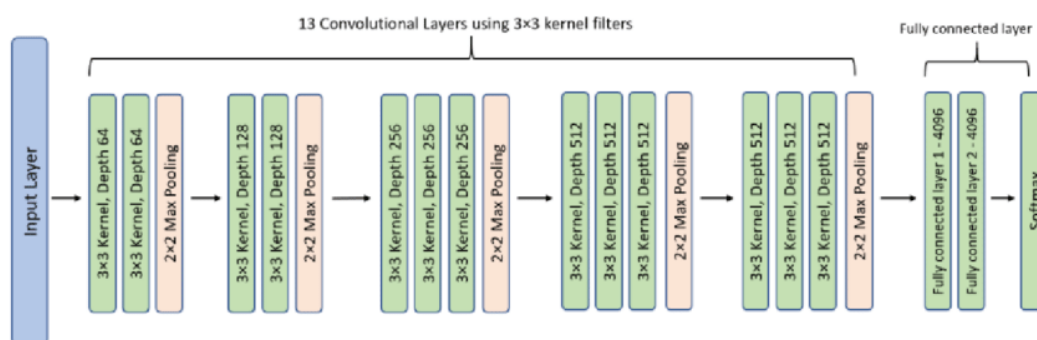
Rectified Linear Unit (ReLU) merupakan salah satu fungsi aktivasi yang populer dan banyak digunakan. ReLu umumnya digunakan pada lapisan konvolusi maupun lapisan *fully connected* dalam jaringan saraf. Fungsi ReLu mengubah output negative menjadi 0, sehingga hanya menghasilkan output positif. Keunggulan ReLu meliputi kemudahan pemahaman serta kinerja yang cepat, sehingga seringkali menjadi pilihan yang disukai dalam pelatihan model kecerdasan buatan (AI).

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

2.6 Arsitektur VGG

Jaringan VGG yang dikembangkan oleh *Karen Simonyan* dan *Andrew Zisserman* dari *Oxford Visual Group* memiliki dua versi utama, yaitu VGG-16 dengan 16 lapisan dan VGG-19 dengan 19 lapisan. Struktur jaringan ini terdiri dari lapisan konvolusi dengan ukuran 3x3 dan lapisan max pooling untuk mengurangi dimensi data. Bagian akhir jaringan terdiri dari dua lapisan *fully connected* dengan masing-masing 4096 neuron, diikuti oleh lapisan *softmax* untuk tujuan klasifikasi. Dalam tahap preprocessing, langkah yang diambil pada input adalah mengurangi nilai rata-rata dari setiap komponen warna (R,G,B) pixel, dengan nilai rata-rata tersebut dihitung dari data pelatihan.. Proses *max pooling* menggunakan jendela *2x2 pixel* dan langkah (stride) sebanyak 2. Aktivasi ReLu diterapkan pada setiap lapisan tersembunyi dari jaringan. Jumlah filter pada lapisan konvolusi meningkat sesuai dengan kedalaman VGG yang digunakan.

VGG-16 adalah model yang terdiri dari 16 lapisan, yang terdiri dari kombinasi lapisan konvolusi dan *fully connected*, yang sering digunakan dalam tugas pengenalan dan klasifikasi image (gambar). Struktur VGG-16 mencakup 13 lapisan konvolusi yang dilengkapi dengan lapisan max pooling untuk mengurangi dimensi data. Selain itu, VGG-16 memiliki dua lapisan *fully connected* yang terletak dalam lapisan tersembunyi, dengan total 4096 unit di setiap lapisannya. Lalu, lapisan-lapisan ini diikuti oleh lapisan dense dengan 1000 unit, dimana setiap unit mewakili satu kategori gambar dalam dataset ImageNet. (shanmugamani, 2018). Ilustrasi dari arsitektur VGG-16 dapat ditemukan dalam gambar **Gambar 2.4**.



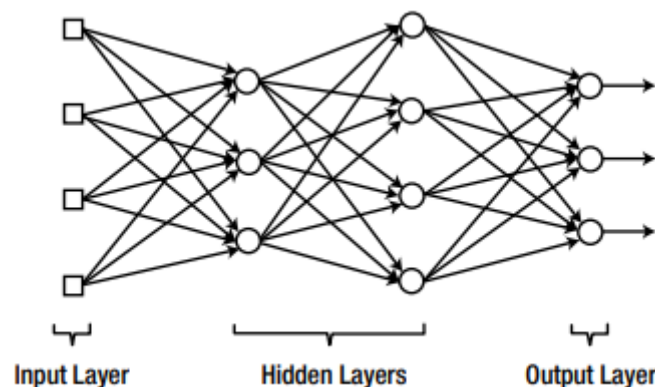
Gambar 2.10 Model Ilustrasi Arsitektur VGG-16

2.7 Neural Network

Istilah Jaringan syaraf (*Neural Network*) pertama kali digunakan oleh McCulloch dan Pitts pada tahun 1943 ketika mereka mencoba mengembangkan representasi matematis untuk menjelaskan proses informasi dalam sistem biologis Jaringan saraf (*neural network*) dalam konteks ini merujuk kepada simpul (node) dalam jaringan yang meniru struktur neuron yang ditemukan dalam otak makhluk hidup. Tiap simpul melakukan perhitungan dengan menggabungkan jumlah nilai masukan yang dikalikan dengan bobot yang spesifik. kemudian hasil diproses melalui lapisan tersembunyi dan di aktivasi menggunakan fungsi tertentu sebelum menghasilkan output dengan bobot yang sesuai.

Neural Network telah mengalami perkembangan dari arsitektur yang sederhana menjadi struktur yang menjadi kompleks. Pada awalnya, model *neural network* memiliki arsitektur yang sangat sederhana dengan hanya lapisan masukan (*input*) dan lapisan keluaran (*output*), disebut sebagai jaringan saraf *single-layer*. Namun, kemudian evolusi ini mengarah pada jaringan saraf multi-layer dengan inklusi lapisan tersembunyi (*hidden layer*) yang ditempatkan antara lapisan input dan output. Gambaran ini dapat dilihat pada **Gambar 2.5** dalam referensi (Kim, 2017).

Untuk mendapatkan hasil akhir (*output*) dari neuron, nilai dari neuron sbelumnya (*input*) dikalikan dengan bobot yang sesuai (w), kemudian ditambahkan denhan nilai bias (b). Selanjutnya, hasil ini diaktivasi menggunakan fungsi aktivasi (g), yang akan mempengaruhi neuron-neuron pada lapisan berikutnya (*output*).



Gambar 2.11 Struktur dari Neural Network

BAB 3

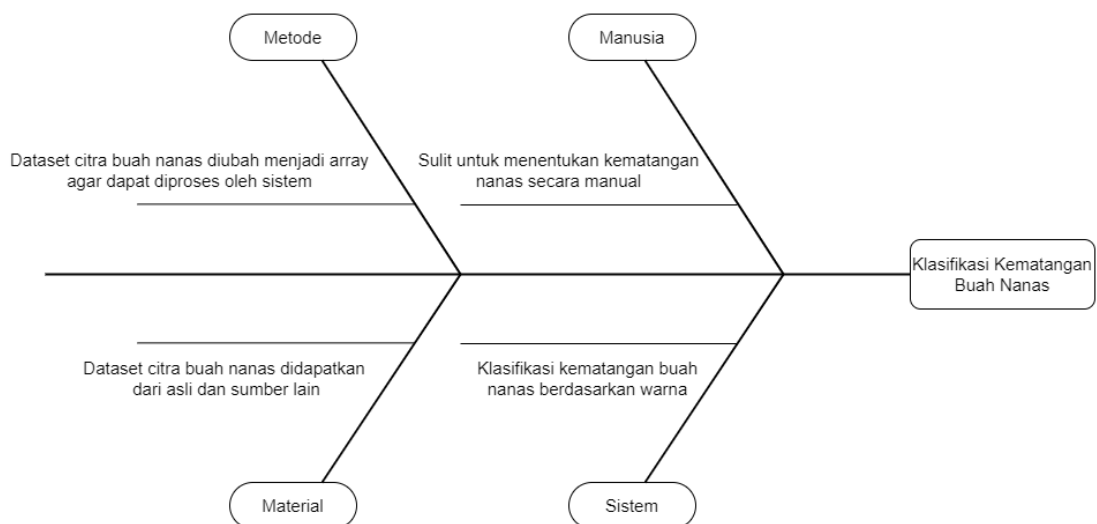
ANALISIS DAN PERANCANGAN SISTEM

3.1 Analisis Sistem

Dalam penelitian ini, analisis digunakan sebagai landasan untuk menghimpun rekomendasi dan mengatasi tantangan yang ada dalam upaya merencanakan dan mengembangkan sistem secara bertahap. Tujuan dari analisis tersebut adalah untuk memberikan solusi atau pemecahan masalah dengan menggunakan komponen yang terstruktur dan berurutan. Ada beberapa langkah yang dilakukan dalam proses analisis sistem, yang meliputi tahap-tahap berikut:

3.1.1 Analisis Masalah

Langkah ini adalah poin kritis dalam upaya mengidentifikasi akar penyebab dari masalah yang dihadapi. Pada penelitian ini, untuk menjelaskan permasalahan yang ada, digunakanlah Diagram Ishikawa (*Fishbone Diagram*) untuk menggambarkan akar masalah dalam 4 kategori yaitu, manusia, sistem, metode, dan material yang digambarkan sebagai berikut.



Gambar 3.1 Fishbone Diagram

Gambar 3.1 menampilkan empat masalah yang perlu diatasi. Oleh karena itu, diperlukan sistem untuk mengklasifikasikan tingkat kematangan buah nenas

3.1.2 Analisis Kebutuhan

Ini adalah proses yang ditujukan untuk mengenali dan memahami kebutuhan yang diperlukan untuk merancang dan membangun suatu sistem. Analisis kebutuhan membantu memastikan bahwa solusi yang dikembangkan akan sesuai dengan tujuan dan harapan pengguna, serta mampu memecahkan permasalahan yang ada. Analisis kebutuhan umumnya dapat dibagi menjadi dua bagian utama:

1. Kebutuhan Fungsional

Persyaratan yang berkaitan dengan fungsionalitas atau tindakan yang harus dilakukan oleh sistem yang sedang dirancang. Dalam penelitian ini, kebutuhan fungsional akan merinci tindakan-tindakan yang diharapkan dilakukan oleh sistem dalam mengklasifikasikan buah nanas berdasarkan warna kulitnya. Berikut adalah contoh-contoh kebutuhan fungsional untuk penelitian ini:

- a. Sistem harus dapat mengklasifikasikan buah nanas ke dalam kategori kematangan berdasarkan warna kulitnya, seperti matang, setengah matang, dan belum matang.
- b. Sistem harus menyediakan antarmuka yang memungkinkan pengguna untuk mengunggah citra buah nanas, melihat hasil klasifikasi kematangannya.

2. Persyaratan Non-Fungsional

Adalah aspek-aspek yang melengkapi dan melibatkan performa serta kualitas sistem yang akan dibangun dalam penelitian ini. Persyaratan ini tidak berkaitan dengan fungsionalitas murni, tetapi lebih kepada bagaimana sistem harus berforma dan berinteraksi dengan lingkungan serta pengguna. Berikut adalah beberapa contoh persyaratan non-fungsional untuk penelitian ini.

- a. Sistem harus memiliki tingkat akurasi yang tinggi dalam mengklasifikasikan warna kulit buah nanas. Akurasi dapat diukur dengan membandingkan hasil klasifikasi dengan label kematangan.

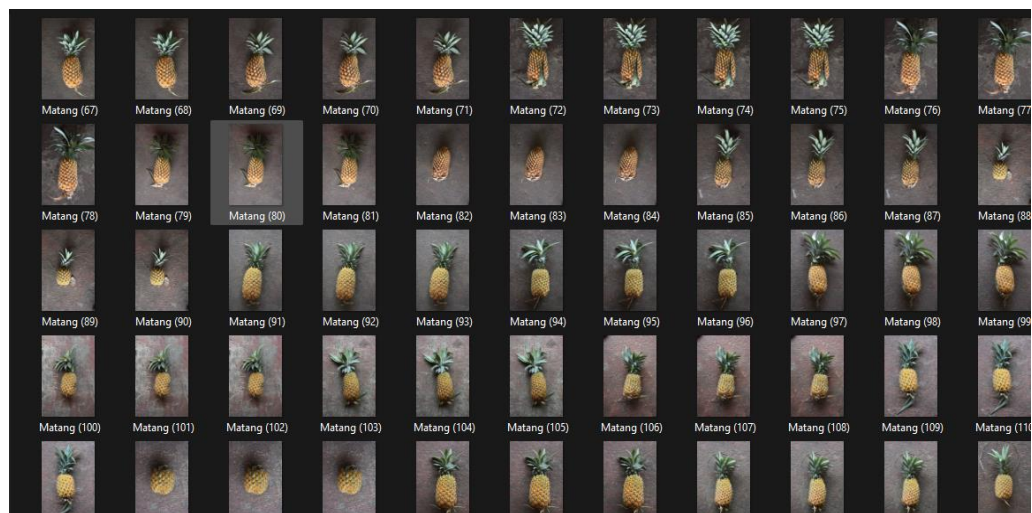
- b. Sistem harus mampu menangani penambahan pengguna atau gambar buah nanas yang lebih besar di masa depan tanpa mengorbankan kinerja atau akurasi.
- c. Sistem harus mampu melakukan analisis klasifikasi kematangan buah nanas dengan cepat dan akurat, memberikan hasil dalam waktu yang wajar bahkan saat menghadapi volume gambar yang cukup besar.

3.1.3 Analisis Proses

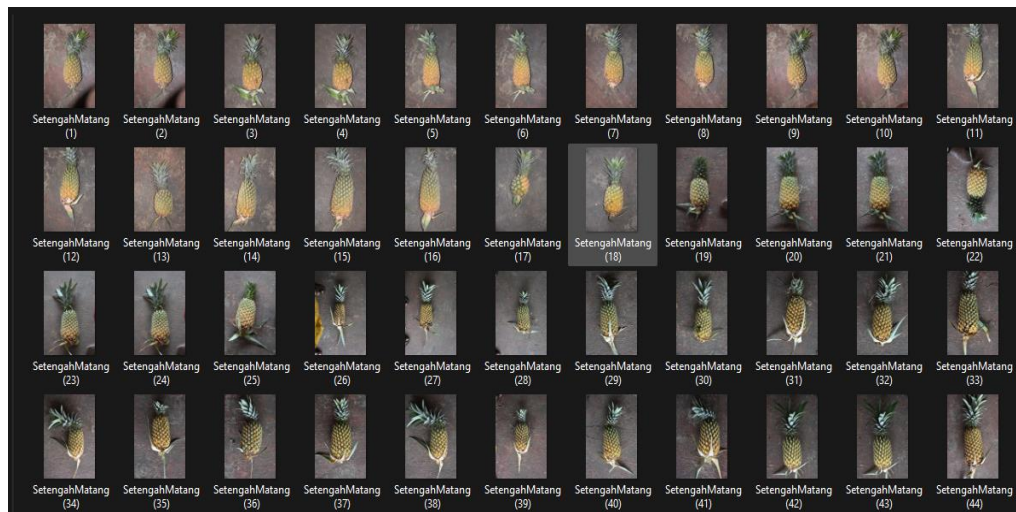
Di bawah ini disajikan serangkaian langkah-langkah yang dilakukan pada penelitian ini untuk membangun sistem klasifikasi kematangan buah nanas berdasarkan warna menggunakan *Convolutional Neural Network* dalam bidang *Deep Learning*:

1. Pengumpulan Data

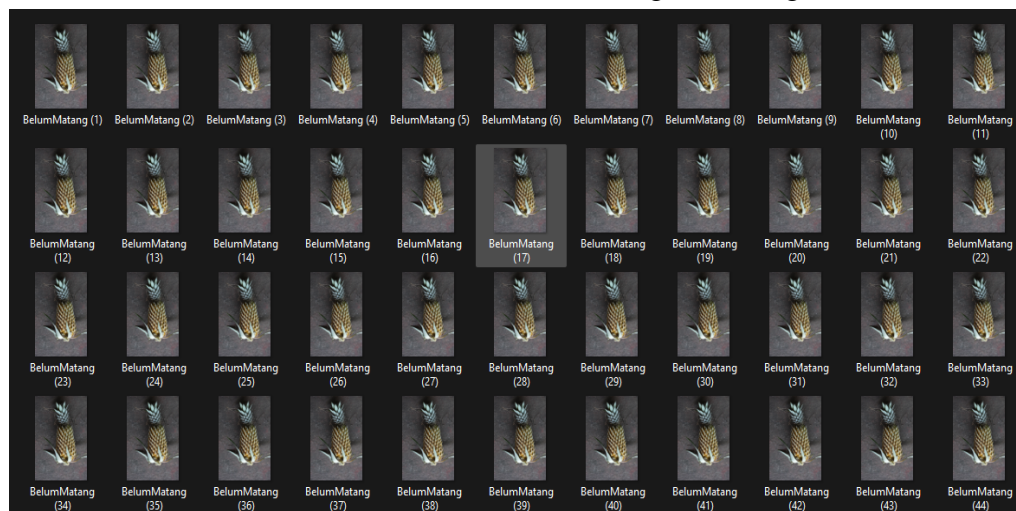
Dalam tahap pengumpulan data dalam penelitian ini, peneliti akan mengambil foto atau gambar buah nanas yang memenuhi sejumlah kriteria kematangan, seperti matang, setengah matang, dan belum matang. Berikut adalah contoh foto atau gambar buah nanas yang sesuai dengan klasifikasi yang telah ditentukan:



Gambar 3.2 Buah Nanas Matang



Gambar 3.3 Buah Nanas Setengah Matang



Gambar 3.4 Buah Nanas Belum Matang

2. Menentukan Dataset

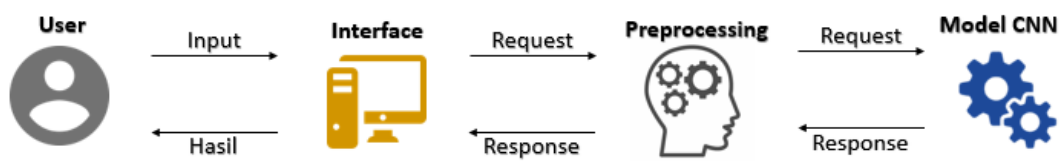
Berdasarkan informasi yang telah dikumpulkan, terdapat total 638 citra buah nanas. Pembagian *dataset* menjadi bagian pelatihan (*training*) dan bagian pengujian (*testing*) adalah langkah penting dalam membangun dan menguji model *Convolutional Neural Network* (CNN). Berikut adalah rincian pembagian dataset berdasarkan citra buah nanas yang telah dikumpulkan.

Tabel 3. 1 Rincian Dataset

Tingkat Kematangan	Dataset Training	Dataset Testing	Total Dataset
Matang	300	20	320
Setengah Matang	156	10	166
Belum Matang	142	10	152

3.1.4 Arsitektur Umum

Gambar di bawah ini adalah representasi visual dari arsitektur keseluruhan sistem yang diterapkan dalam penelitian ini.

**Gambar 3.5** Ilustrasi Arsitektur Umum Sistem

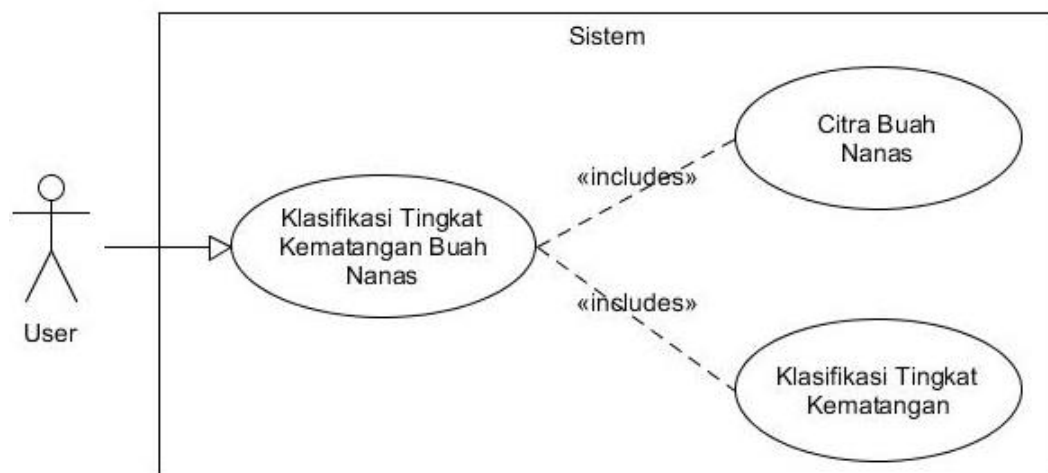
- Pengguna mengunggah gambar buah nanas ke aplikasi klasifikasi kematangan buah nanas.
- Gambar yang diunggah diterima oleh modul pengelola file dalam sistem.
- Modul preprocessing menerima gambar dan melakukan tahap preprocessing, seperti resize.
- Data gambar yang telah dipreprocessing diteruskan ke modul CNN dan VGG16.
- Modul CNN dan VGG16 akan melakukan proses training dan akan menghasilkan file model.h5. yang akan digunakan pada proses klasifikasi tingkat kematangan buah nanas.
- Modul klasifikasi menghasilkan prediksi tingkat kematangan buah nanas berdasarkan analisis fitur dari gambar.

- g. Hasil klasifikasi dikirim kembali ke aplikasi dan ditampilkan kepada pengguna.
- h. Pengguna dapat melihat hasil klasifikasi pada antarmuka aplikasi, yang menampilkan informasi tentang kematangan buah nanas berdasarkan gambar yang di unggah.

Arsitektur ini mencakup modul pengelola file, preprocessing, modul CNN, klasifikasi, dan antarmuka aplikasi yang bekerja bersama-sama untuk menerima gambar buah nanas, mengolahnya, menganalisis fitur, melakukan klasifikasi, dan menampilkan hasil kepada pengguna.

3.2 Pemodelan Sistem

3.2.1 Use Case diagram

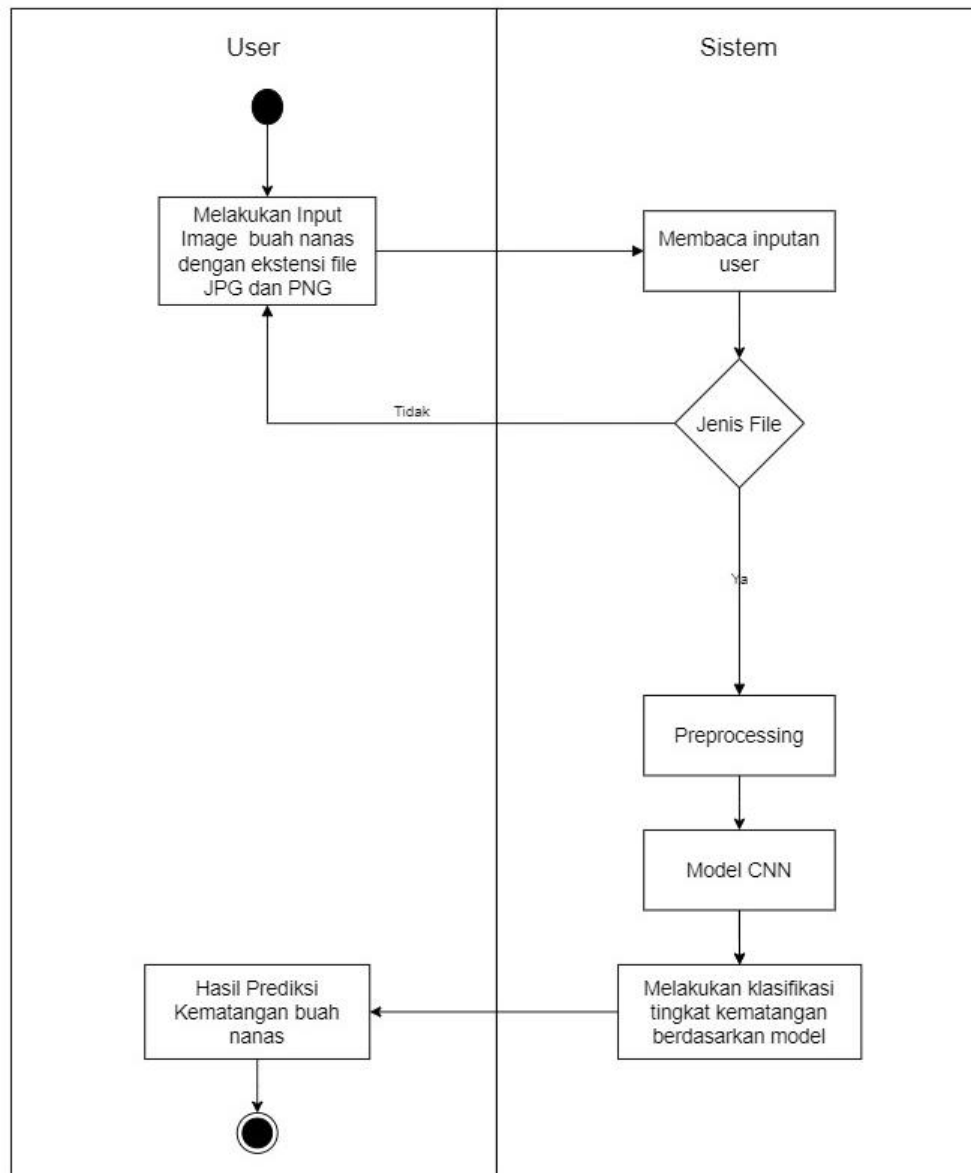


Gambar 3.6 Use Case Diagram

Di dalam **Gambar 3.6** sistem ini akan beroperasi ketika *user* melakukan upload citra buah nanas. Berdasarkan file yang di upload oleh user berupa file dengan ekstensi file JPG dan PNG akan menghasilkan prediksi kematangan buah nanas berdasarkan citra yang di upload ke sistem.

3.2.2 Activity diagram

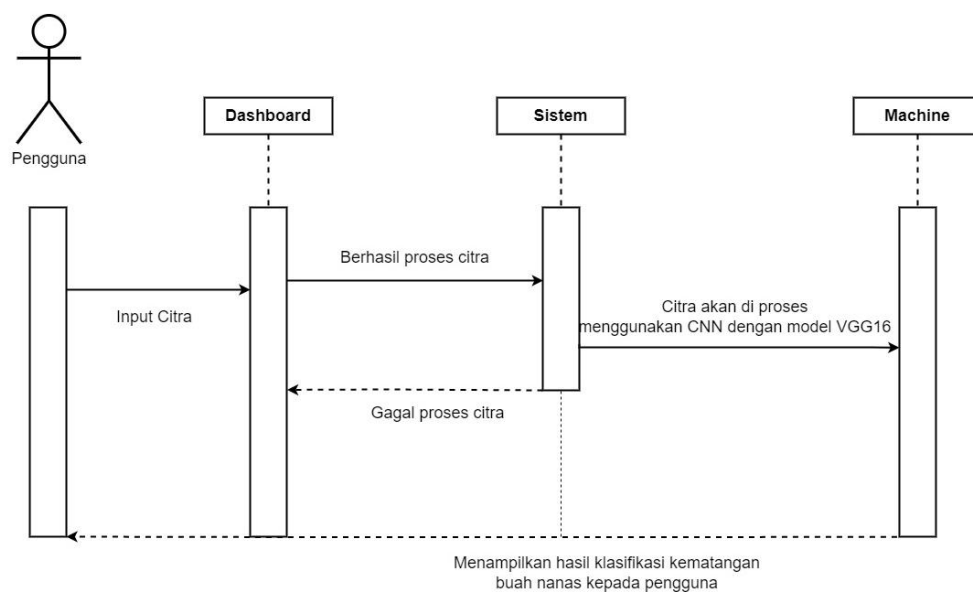
Pada **Gambar 3.7**, tahap ini akan mengilustrasikan urutan alur kerja atau aktivitas pada sistem klasifikasi kematangan buah nanas berdasarkan warna kulitnya. Gambar yang menggambarkan langkah-langkah ini dapat ditemukan di bawah ini.



Gambar 3.7 Activity Diagram

3.2.3 Sequence diagram

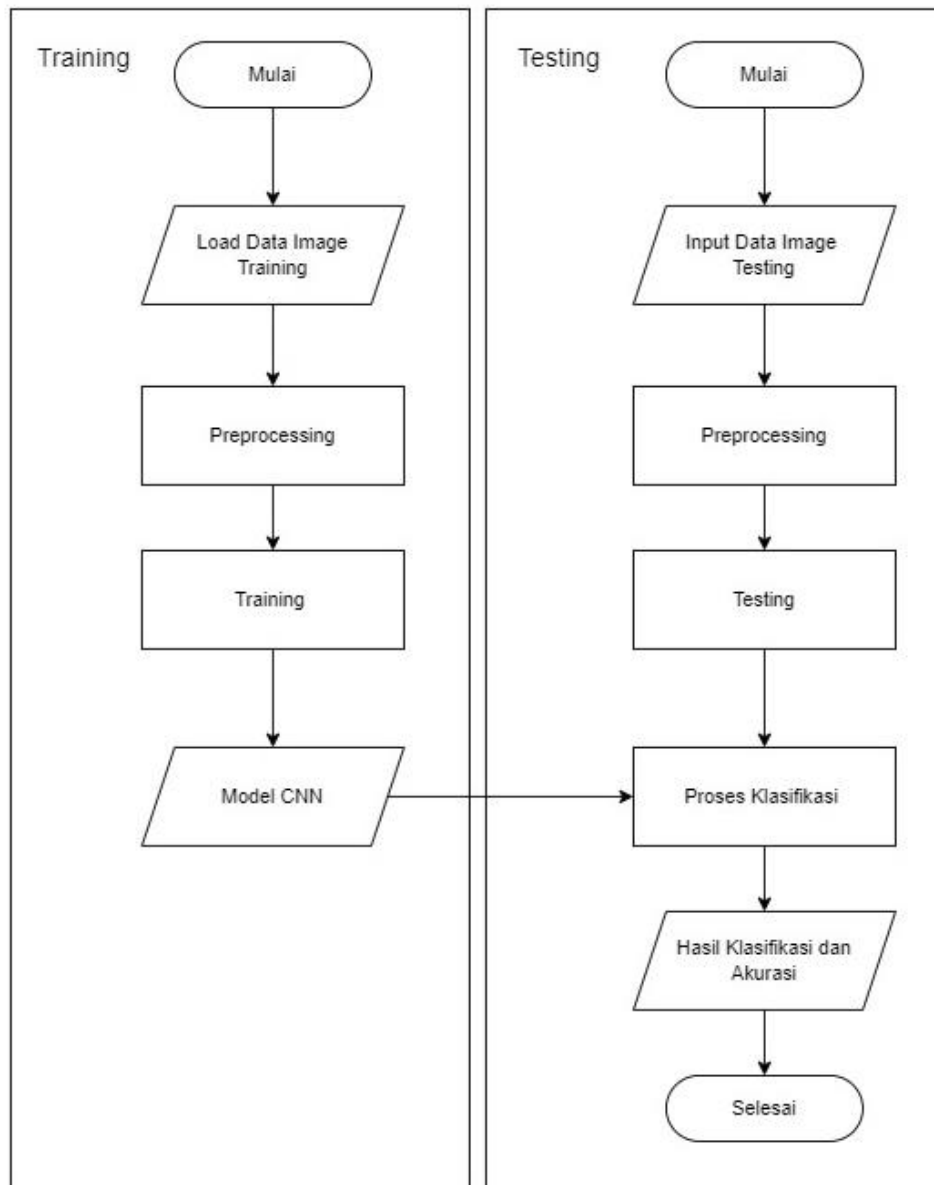
Sequence diagram adalah suatu bentuk diagram yang digunakan dalam rekayasa perangkat lunak dan desain sistem untuk menggambarkan cara objek atau komponen berinteraksi dalam suatu sistem. Dalam **Gambar 3.8**, terlihat penggunaan *sequence diagram* dalam mengklasifikasikan kematangan buah nanas berdasarkan warna kulitnya.



Gambar 3.8 Sequence Diagram

Gambar 3.8 Menyediakan gambaran visual tentang aliran pesan yang dikomunikasikan antara objek dan rangkaian langkah eksekusi dalam sistem selama proses klasifikasi tingkat kematangan buah nanas.

3.3 Flowchart



Gambar 3.9 Flowchart Sistem

Pada **Gambar 3.9** Langkah awal melibatkan memuat dataset gambar buah nanas (load file directory) untuk tahap *training*, langkah selanjutnya adalah melakukan *preprocessing*, tujuan dari tahap preprocessing ini adalah untuk menyiapkan data dan melakukan resize pada gambar menjadi 150x150 pixel sebelum masuk ke tahap pelatihan model. Proses pengolahan data, konversi nilai,serta pemisahan data *training*

data split. Setelah data berhasil di olah, Pada tahap pelatihan dan pengujian, model CNN dengan arsitektur yang dikenal sebagai VGG16 digunakan.

Selanjutnya, data yang sudah melalui seluruh tahap tersebut akan digunakan untuk *training* model CNN. Proses ini melibatkan proses *training* dengan menggunakan data yang sudah diolah sebelumnya. Setelah proses *training* selesai, model CNN (Model.h5) akan dihasilkan. Selanjutnya, model CNN akan digunakan untuk proses *testing* yang dimana setelah melakukan input image, selanjutnya gambar tersebut akan di proses pada tahap preprocessing setelah itu akan dilakukan proses klasifikasi dengan menggunakan file model.h5 yang dihasilkan pada tahap training. Ketika selesai di proses klasifikasi dengan menggunakan model maka, akan menghasilkan prediksi dan akurasi klasifikasi kematangan buah nanas berdasarkan warna.

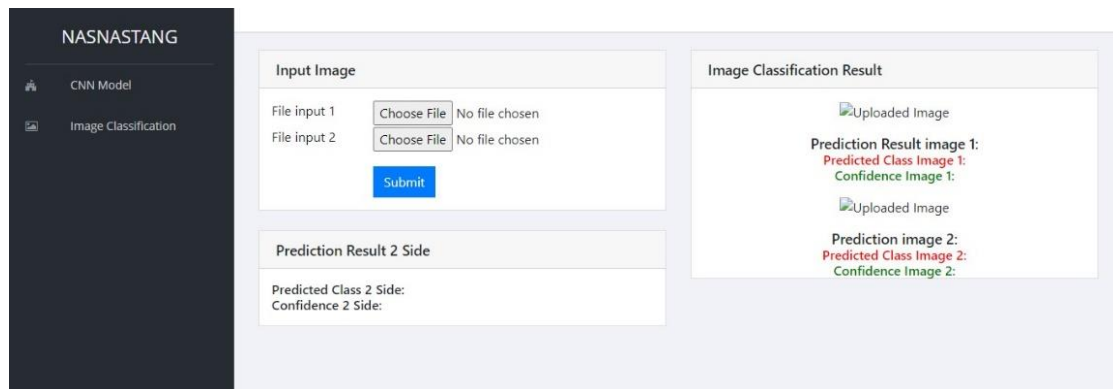
3.4 Perancangan Antarmuka Sistem

Tampilan antarmuka, atau sering disebut sebagai “interface”, mengacu pada cara informasi dan kontrol dipresentasikan kepada pengguna dalam suatu sistem atau perangkat lunak. Ini melibatkan pengaturan elemen-elemen seperti pengaturan tata letak, elemen grafis, tombol, teks, ikon, warna, dan elemen visual lainnya yang digunakan untuk berinteraksi dengan perangkat lunak atau perangkat keras tertentu.

Sebagai contoh, pada **Gambar 3.10**, dapat dilihat bahwa halaman beranda atau dashboard adalah tampilan yang diperlihatkan kepada pengguna ketika mereka menggunakan situs web yang dikembangkan dalam penelitian ini.



Gambar 3.10 Halaman awal



Gambar 3.11 Halaman Prediksi

Di dalam **Gambar 3.11**, kita melihat Halaman Prediksi, yang merupakan lokasi di mana proses klasifikasi tingkat kematangan buah nanas berdasarkan warna dilakukan menggunakan metode Convolutional Neural Network (CNN) dalam konteks deep learning. Input data telah dimasukkan sebelumnya. Hasil prediksi akan menentukan apakah buah nanas tersebut masuk ke dalam kategori matang, setengah matang, atau belum matang.

BAB 4

IMPLEMENTASI DAN PENGUJIAN

4.1 Implementasi sistem

Deskripsi perangkat keras yang disebutkan dalam kerangka penelitian ini mencakup persyaratan minimal yang diperlukan agar sistem yang akan dibangun dapat beroperasi. Berikut adalah komponen perangkat keras yang disebutkan:

4.1.1 Perangkat Keras (*Hardware*)

1. Intel Core i5 atau setara
2. Minimal 4 GB (Giga Byte) DDR4 RAM (Random Access Memory)
(setidaknya 8 GB atau lebih untuk memastikan kelancaran proses pelatihan model)

4.1.2 Perangkat Lunak (*Software*)

1. Visual Studio Code
2. Sistem operasi Windows 11 Pro 64-bit

4.2 Pengumpulan *Dataset*

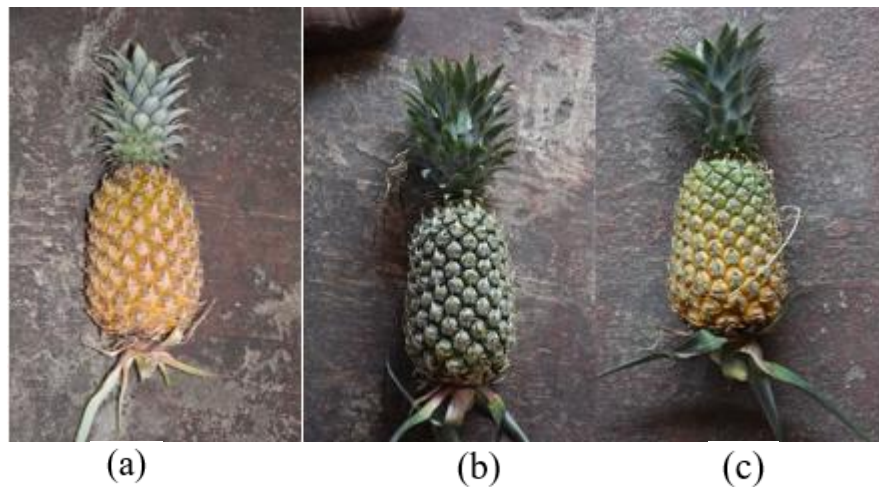
Pengumpulan dataset penelitian ini dilakukan dengan 2 (dua) cara berbeda, yaitu:

1. Mengambil foto buah nanas menggunakan kamera *smartphone* Xiaomi Redmi 12.
2. Mengambil *dataset* yang tersedia di website.

4.3 Persiapan dan Pelabelan *Dataset*

Pelabelan data adalah proses penting dalam pengolahan data dalam penelitian klasifikasi kematangan buah nanas. Setelah dataset dikumpulkan, langkah selanjutnya melakukan pelabelan atau mengklarifikasi data sesuai dengan kategori yang telah ditetapkan. Dalam penelitian ini, ada tiga kategori kematangan buah nanas yaitu: matang, belum matang dan setengah matang.

Proses pelabelan data ini memiliki tujuan utama, yaitu untuk menciptakan *ground truth* (referensi utama) yang akan menjadi landasan untuk melatih model klasifikasi.

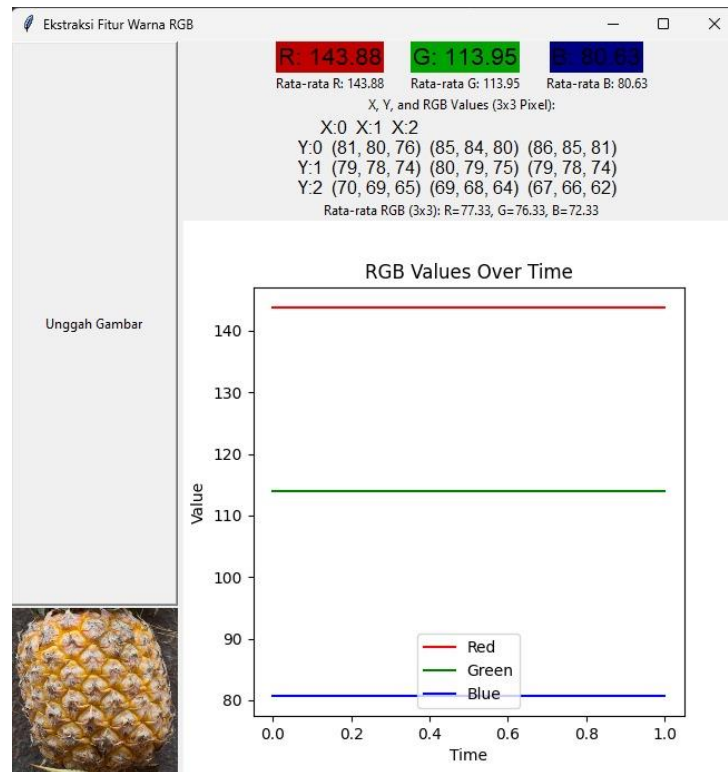


Gambar 4.1 Warna pada buah nanas matang, belum matang, dan setengah matang

- a. **Buah Nanas Matang** : adalah buah nanas yang telah mencapai tahap kematangan optimal untuk dikonsumsi. Warna kulit nanas matang biasanya memiliki warna yang cerah, seperti kuning atau jingga. Warna yang merata dan cerah adalah tanda bahwa buah ini telah matang dengan baik
- b. **Buah Nanas Belum Matang** : adalah buah nanas yang belum mencapai tahap kematangan optimal untuk dikonsumsi. Warna kulit nanas yang belum matang cenderung memiliki warna yang lebih hijau daripada kuning atau jingga yang cerah. Warna hijau lebih dominan pada nanas yang belum matang.
- c. **Buah Nanas Setengah Matang** : adalah nanas yang telah melewati tahap awal pematangan, tetapi belum mencapai tahap kematangan penuh. Warna kulit nanas setengah matang mungkin tidak lagi sangat hijau seperti nanas yang benar-benar belum matang, tetapi juga belum mencapai warna kuning atau jingga cerah yang menunjukkan kematangan penuh.

4.4 Ekstraksi Ciri

Pada **Gambar 4.2** dibawah ini adalah hasil citra yang sudah di lakukan rata-rata dari RGB nya yang berguna untuk mengetahui nilai dari setiap pixelnya.



Gambar 4.2 Ekstraksi Fitur Warna RGB

4.5 Implementasi Tahapan Antarmuka

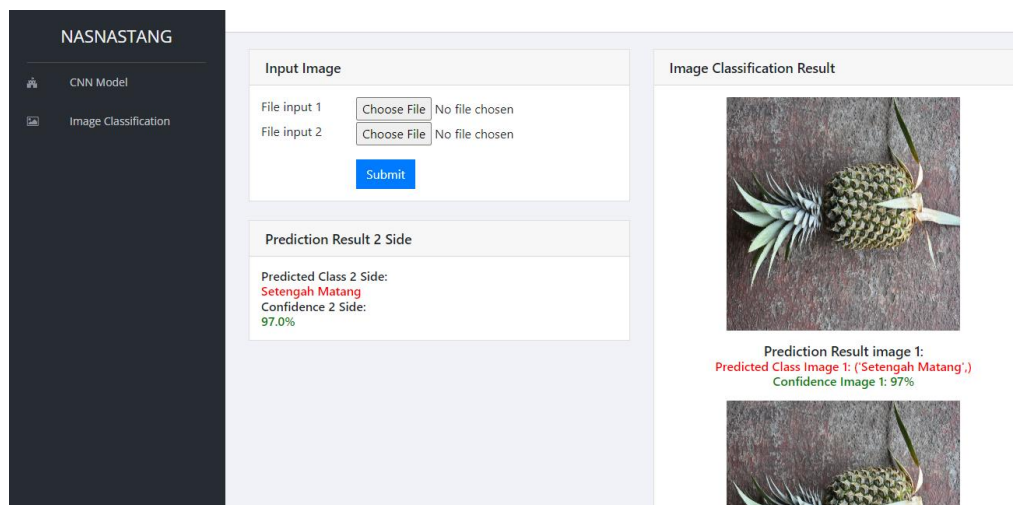
Ini adalah tampilan yang muncul ketika proses klasifikasi dimulai, mulai dari penginputan gambar buah nanas hingga hasil prediksi diperoleh.

Dimana terdapat beberapa tahapan untuk menggunakan sistem ini :

1. Pada sisi sebelah kiri sistem terdapat menu CNN Model (Dashboard) dan Image Classification. Pada menu *image classification* digunakan untuk melakukan prediksi kematangan buah nanas.
2. Pengguna harus melakukan upload gambar buah nanas pada File Input 1 dan 2 (*Choose File*).
3. Setelah dilakukan upload gambar buah nanas maka bisa dilakukan *Submit* untuk proses gambar yang telah di upload pada tempat inputan, apakah satu sisi buah nanas ataupun kedua sisi buah nanas.
4. Hasil akhir dari pemrosesan gambar buah nanas akan ditampilkan pada sisi kanan yaitu *Image Classification Image*. Gambar yang telah di upload dan di proses

akan ditampilkan disini dan akan di diperlihatkan yang dimana *Prediction Result Image* 1 dan 2 dengan berwarna merah, dan untuk yang berwarna hijau adalah *Confidence* dari hasil prediksi masing masing gambar.

5. Tepat dibawah Input image (*Prediction Result 2 Side*) adalah hasil prediksi dari kedua gambar yang telah di inputkan sebelumnya. Pada **Gambar 4.2** adalah contoh tampilan antarmuka (*interface*) pada sistem.



Gambar 4.3 Tampilan Antarmuka

4.6 Perancangan Model

Setelah dilakukan pre-processing pada dataset selanjutnya adalah perancangan model yang akan digunakan yaitu CNN dan VGG16.

4.6.1 Perancangan Model CNN

Model: "sequential_9"		
Layer (type)	Output Shape	Param #
conv2d_67 (Conv2D)	(None, 148, 148, 32)	896
max_pooling2d_35 (MaxPooling2D)	(None, 74, 74, 32)	0
conv2d_68 (Conv2D)	(None, 72, 72, 64)	18496
max_pooling2d_36 (MaxPooling2D)	(None, 36, 36, 64)	0
conv2d_69 (Conv2D)	(None, 34, 34, 128)	73856
max_pooling2d_37 (MaxPooling2D)	(None, 17, 17, 128)	0
flatten_9 (Flatten)	(None, 36992)	0
dense_32 (Dense)	(None, 128)	4735104
dense_33 (Dense)	(None, 64)	8256
dense_34 (Dense)	(None, 32)	2080
dense_35 (Dense)	(None, 3)	99
Total params: 4838787 (18.46 MB)		
Trainable params: 4838787 (18.46 MB)		
Non-trainable params: 0 (0.00 Byte)		

Gambar 4.4 Summary Model CNN

```

# Membuat objek model Sequential
model = models.Sequential()

# Layer Convolutional pertama
model.add(layers.Conv2D(32, (3, 3), activation='relu', input_shape=(150, 150, 3)))
model.add(layers.MaxPooling2D((2, 2)))

# Layer Convolutional kedua
model.add(layers.Conv2D(64, (3, 3), activation='relu'))
model.add(layers.MaxPooling2D((2, 2)))

# Layer Convolutional ketiga
model.add(layers.Conv2D(128, (3, 3), activation='relu'))
model.add(layers.MaxPooling2D((2, 2)))

# Layer Flatten untuk mengubah output menjadi vektor 1D
model.add(layers.Flatten())

# Layer Dense (fully connected)
model.add(layers.Dense(128, activation='relu'))
model.add(layers.Dense(64, activation='relu'))
model.add(layers.Dense(32, activation='relu'))
model.add(layers.Dense(3, activation='softmax')) # Misalnya, 10 kelas output

# Tentukan learning rate sesuai kebutuhan Anda, contoh: 0.0001
learning_rate = 0.0001

# Compile model
optimizer = Adam(learning_rate=learning_rate)
model.compile(optimizer=optimizer, loss='categorical_crossentropy', metrics=['accuracy'])

# Tampilkan ringkasan model
model.summary()

```

Gambar 4.5 Implementasi Model CNN

Desain model CNN yang disajikan dalam **Gambar 4.4** dan **Gambar 4.5** terdiri dari total 9 lapisan, termasuk 3 lapisan konvolusi (*convolutional layer*), 3 lapisan pengambilan nilai maksimum (*max pooling layer*), 1 lapisan perataan (*flatten layer*), dan 2 lapisan berdimensi penuh (*dense layer*). Lapisan konvolusi pertama juga berfungsi sebagai lapisan masukan (*input layer*) dengan dimensi 150x150 pixel dan 3 channel warna (RGB).

Setiap lapisan konvolusi diikuti oleh lapisan pengambilan nilai maksimum (*max pooling*) dengan ukuran jendela (*matrix*) 2x2 dan langkah (*stride*) 2 untuk mengurangi dimensi feature map sebelumnya. Lapisan berdimensi penuh (*dense layer*) hanya dapat menerima masukan dalam bentuk vector 1 (satu) dimensi, oleh

karena itu, terdapat lapisan perataan (*flatten layer*) yang mengubah output dari lapisan konvolusi menjadi vector 1 dimensi.

Lapisan berdimensi penuh (*dense layer*) memiliki jumlah neuron sebanyak 128, 64, 32, dan 3 secara berurutan. Pemilihan jumlah neuron ini didasarkan pada eksperimen dan penyesuaian. Jumlah neuron 128, 64, dan 32 merupakan pilihan umum dalam model CNN, sementara 3 neuron dalam lapisan terakhir digunakan untuk output klasifikasi, penggunaan *softmax* sesuai dengan jumlah kategori kelas pada dataset yang terdiri dari 3 kelas.

Untuk melatih model, digunakan fungsi kerugian (*loss*) *categorical crossentropy* karena dataset memiliki 3 kelas yang berbeda. Optimizer yang digunakan adalah *Adam*, dengan *learning rate* 0,0001 yang efektif dalam menyesuaikan laju pembelajaran selama pelatihan. Performa model akan di evaluasi berdasarkan akurasi (*accuracy*) untuk mengukur sejauh mana model dapat mengklasifikasikan data dengan benar.

4.6.2 Perancangan Model VGG16

Model: "sequential_10"

Layer (type)	Output Shape	Param #
conv2d_70 (Conv2D)	(None, 150, 150, 32)	896
conv2d_71 (Conv2D)	(None, 150, 150, 32)	9248
max_pooling2d_38 (MaxPooling2D)	(None, 75, 75, 32)	0
conv2d_72 (Conv2D)	(None, 75, 75, 64)	18496
conv2d_73 (Conv2D)	(None, 75, 75, 64)	36928
max_pooling2d_39 (MaxPooling2D)	(None, 37, 37, 64)	0
conv2d_74 (Conv2D)	(None, 37, 37, 128)	73856
conv2d_75 (Conv2D)	(None, 37, 37, 128)	147584
conv2d_76 (Conv2D)	(None, 37, 37, 128)	147584
max_pooling2d_40 (MaxPooling2D)	(None, 18, 18, 128)	0
conv2d_77 (Conv2D)	(None, 18, 18, 256)	295168
conv2d_78 (Conv2D)	(None, 18, 18, 256)	590080
conv2d_79 (Conv2D)	(None, 18, 18, 256)	590080
max_pooling2d_41 (MaxPooling2D)	(None, 9, 9, 256)	0
conv2d_80 (Conv2D)	(None, 9, 9, 512)	1180160
conv2d_81 (Conv2D)	(None, 9, 9, 512)	2359808
conv2d_82 (Conv2D)	(None, 9, 9, 512)	2359808
max_pooling2d_42 (MaxPooling2D)	(None, 4, 4, 512)	0
flatten_10 (Flatten)	(None, 8192)	0
dense_36 (Dense)	(None, 128)	1048704
dropout_8 (Dropout)	(None, 128)	0
dense_37 (Dense)	(None, 128)	16512
dropout_9 (Dropout)	(None, 128)	0
dense_38 (Dense)	(None, 3)	387

=====

Total params: 8875299 (33.86 MB)
 Trainable params: 8875299 (33.86 MB)
 Non-trainable params: 0 (0.00 Byte)

Gambar 4.6 Summary Model VGG

```

def build_vgg16(input_shape=(150, 150, 3), num_classes=3):
    model = Sequential()

    # Block 1
    model.add(Conv2D(32, (3, 3), activation='relu', padding='same', input_shape=input_shape))
    model.add(Conv2D(32, (3, 3), activation='relu', padding='same'))
    model.add(MaxPooling2D((2, 2), strides=(2, 2)))

    # Block 2
    model.add(Conv2D(64, (3, 3), activation='relu', padding='same'))
    model.add(Conv2D(64, (3, 3), activation='relu', padding='same'))
    model.add(MaxPooling2D((2, 2), strides=(2, 2)))

    # Block 3
    model.add(Conv2D(128, (3, 3), activation='relu', padding='same'))
    model.add(Conv2D(128, (3, 3), activation='relu', padding='same'))
    model.add(Conv2D(128, (3, 3), activation='relu', padding='same'))
    model.add(MaxPooling2D((2, 2), strides=(2, 2)))

    # Block 4
    model.add(Conv2D(256, (3, 3), activation='relu', padding='same'))
    model.add(Conv2D(256, (3, 3), activation='relu', padding='same'))
    model.add(Conv2D(256, (3, 3), activation='relu', padding='same'))
    model.add(MaxPooling2D((2, 2), strides=(2, 2)))

    # Block 5
    model.add(Conv2D(512, (3, 3), activation='relu', padding='same'))
    model.add(Conv2D(512, (3, 3), activation='relu', padding='same'))
    model.add(Conv2D(512, (3, 3), activation='relu', padding='same'))
    model.add(MaxPooling2D((2, 2), strides=(2, 2)))

    # Fully connected layers with dropout
    model.add(Flatten())
    model.add(Dense(128, activation='relu'))
    model.add(Dropout(0.5)) # Menambahkan dropout dengan tingkat 0.5 (sangat umum)
    model.add(Dense(128, activation='relu'))
    model.add(Dropout(0.5)) # Menambahkan dropout di sini juga
    model.add(Dense(num_classes, activation='softmax'))

    return model

# Test the model
input_shape = (150, 150, 3)
num_classes = 3
vgg16_model = build_vgg16(input_shape, num_classes)
vgg16_model.summary()

```

Gambar 4.7 Summary Model VGG

Perancangan model VGG16 dapat dilihat pada **Gambar 4.5** dan **Gambar 4.6**

1. Block 1

- a. *Convolutional Layer* (Conv2D): Lapisan pertama adalah *Convolutional Layer* dengan 32 filter, masing-masing menggunakan kernel 3x3, aktivasi ReLu, dan padding “same” untuk mempertahankan ukuran gambar input. Ini diikuti oleh lapisan Conv2D kedua yang serupa.

- b. *MaxPooling Layer* (MaxPooling2D): Setelah dua lapisan Conv2D, selanjutnya *MaxPooling2D* memiliki ukuran (2x2) dan *strides* (2x2) untuk mengurangi dimensi gambar.
 - c. Jumlah Layer dalam Block 1: 4.
- 2. Block 2
 - a. *Convolutional Layer* (Conv2D): Blok kedua juga terdiri dari dua lapisan Conv2D dengan 64 filter dan aktivasi ReLu, yang diikuti oleh lapisan *MaxPooling2D* dengan ukuran (2x2) dan *strides* (2x2).
 - b. Jumlah Layer dalam Block 2: 3.
- 3. Block 3
 - a. *Convolutionan Layer* (Conv2D): Blok ketiga memiliki tiga lapisan Conv2D dengan 128 filter, aktivasi ReLu, dan *padding* “same”.
 - b. *MaxPooling Layer* (MaxPooling2D): *MaxPooling2D* dengan ukuran (2x2) dan *strides* (2x2).
 - c. Jumlah Layer dalam Block 3: 4.
- 4. Block 4
 - a. *Convolutional Layer* (Conv2D): Blok keempat memiliki tiga lapisan Conv2D dengan 256 filter, aktivasi ReLu, dan *padding* “same”.
 - b. *MaxPooling Layer* (MaxPooling2D): *MaxPooling2D* dengan ukuran (2x2) dan *strides* (2x2).
 - c. Jumlah Layer dalam Block 4: 4.
- 5. Block 5
 - a. *Convolutional Layer* (Conv2D): Blok kelima adalah blok terakhir yang memiliki tiga lapisan Conv2D dengan 512 filter, aktivasi ReLu, dan *padding* “same”.
 - b. *MaxPooling Layer* (MaxPooling2D): *MaxPooling2D* dengan ukuran (2x2) dan *strides* (2x2).
 - c. Jumlah Layer dalam Block 5: 4.

6. Fully Connedted Layers

- a. Setelah lima blok konvolusi, selanjutnya pada lapisan *Dense* (fully connected) memiliki tiga lapisan dengan masing-masing 128 neuron, aktivasi ReLu, dan dropout dnegan tingkat 0,5.
- b. Jumlah Layer Fully Connected: 3

7. Output Layer

- a. Lapisan output terakhir adalah lapisan *Dense* dengan jumlah neuron yang sesuai dengan jumlah kelas yang diinginkan untuk melakukan prediksi (*num_classes*) dan aktivasi *softmax* untuk menghasilkan probabilitas kelas-kelas prediksi, dan penggunaan *softmax* sesuai dengan jumlah kategori kelas pada dataset yang terdiri dari 3 kelas.
- b. Jumlah Layer Output: 1.

Jumlah total layer pada model ini adalah 4 (Block 1) + 3 (Block 2) + 4 (Block 3) + 4 (Block 4) + 4 (Block 5) + 3 (Fully Connected) + 1 (Output) = 19 layer. Ini adalah implementasi yang cukup terhadap arsitektur VGG16 dengan 16 lapisan konvolusi dan 3 lapisan *fully connected*, dengan tambahan dropout untuk mengurangi *overfitting*.

4.7 Penentuan Parameter Model

Dalam upaya mencari model terbaik, penting untuk mengidentifikasi nilai-nilai parameter yang paling optimal dalam konteks model CNN. Parameter-parameter ini mencakup jumlah epoch, ukuran gambar input, jumlah data latihan (*training*), dan *learning rate*. Tujuan dari penentuan parameter model yang berbeda dan menentukan model yang paling optimal dengan mempertimbangkan nilai-nilai parameter yang relevan.

4.6.1 Pengaruh Banyaknya Epoch

Epoch merupakan fase di mana seluruh dataset telah melewati proses pelatihan dalam *neural network* hingga satu putaran siklus pelatihan (*training*) selesai. Dalam konteks jaringan saraf (*neural network*), satu *epoch* yang mencakup seluruh dataset dianggap terlalu besar dan dapat

memekan waktu yang lama. Oleh karena itu, untuk mempercepat dan mempermudah proses pelatihan, biasanya dataset dibagi menjadi ukuran batch yang lebih kecil. Penentuan ukuran batch ini sering kali bergantung pada preferensi peneliti dan jumlah sampel yang ada. Berikut adalah hasil perbandingan *epoch* yang diperoleh dari pelatihan (*training*) model.

Tabel 4.1 Tabel Banyaknya Epoch

Epoch	Accuracy Validation	Loss Validation	Time Epoch	Time
10	59%	76%	17m	22m
20	93%	25%	36m	37m
30	95%	18%	51m	53m
40	98%	5%	55m	58m

Dari tabel diatas, dengan penggunaan *learning rate* sebesar 0,0001, terdapat pencapaian akurasi yang tinggi, mencapai 98%. Hasil dari tabel ini mengindikasikan bahwa semakin mendekati nilai 40 *epoch* yang digunakan, akurasi dari pengujian semakin meningkat, nilai *loss* cenderung menurun.

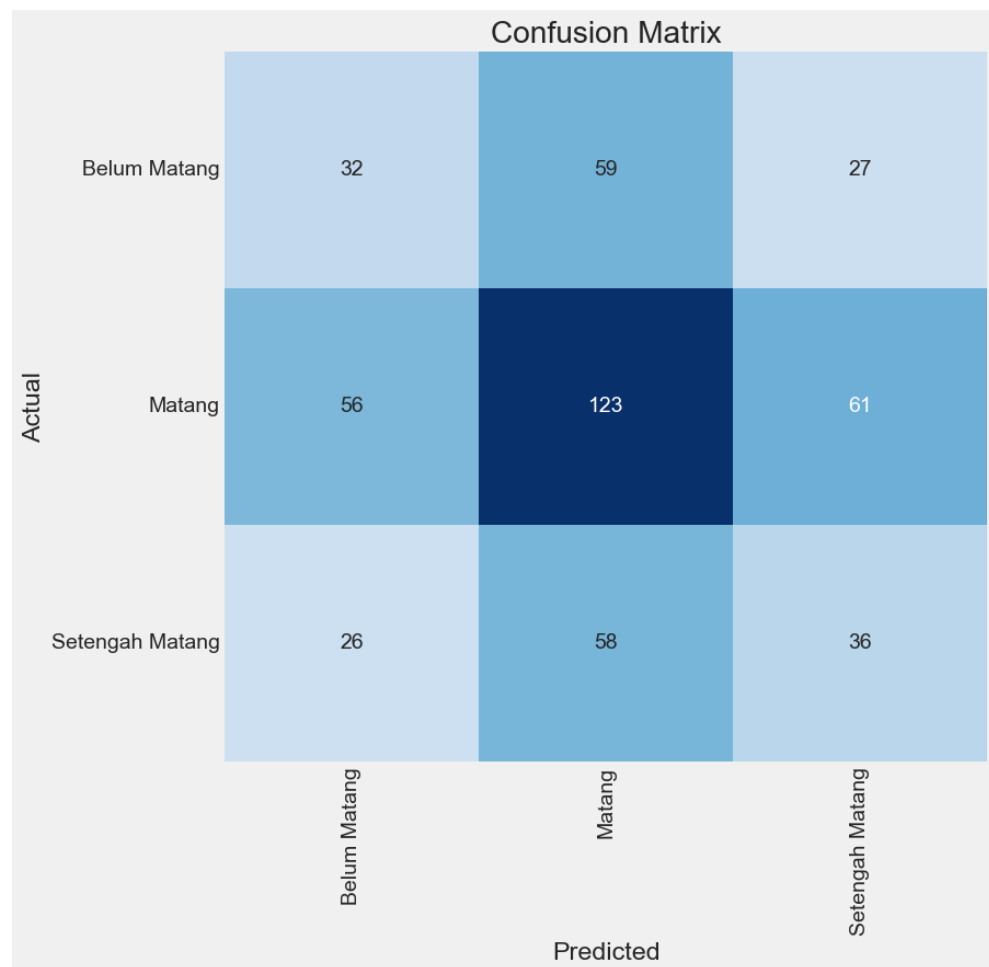
4.8 Pengujian Sistem

Hasil pengujian pada sistem merujuk pada evaluasi kinerja dan performa sistem yang telah diuji. Pengujian sistem bertujuan untuk mengetahui sejauh mana sistem dapat memenuhi persyaratan, mengidentifikasi masalah atau kegagalan dalam sistem, serta memastikan bahwa sistem dapat beroperasi dengan yang telah direncanakan atau diharapkan. Beberapa hasil yang mungkin dihasilkan dari pengujian sistem adalah sebagai berikut:

4.7.1 Hasil Proses Training

Salah satu aspek kunci dalam mengidentifikasi kematangan buah nanas berdasarkan warna terletak pada hasil dari pelatihan. Pelatihan ini memiliki dampak yang sangat signifikan terhadap hasil yang akan diperoleh saat pengujian. Parameter iterasi pada pelatihan dilakukan sebanyak 40 *epoch*

dengan *batch size* sebesar 32. Maka dari itu, proses pelatihan akan dijalankan dan diulang sebanyak 40 iterasi untuk mendapatkan ekstraksi fitur yang diperlukan. Selain itu, *learning rate* yang diterapkan dalam pelatihan ini adalah sebesar 0,0001. Nilai *learning rate* ini digunakan untuk mengupdate bobot pada setiap tahap proses *backward-pass* algoritma. Pada **Gambar 4.8** adalah hasil dari *Confusion Matrix* hasil training.

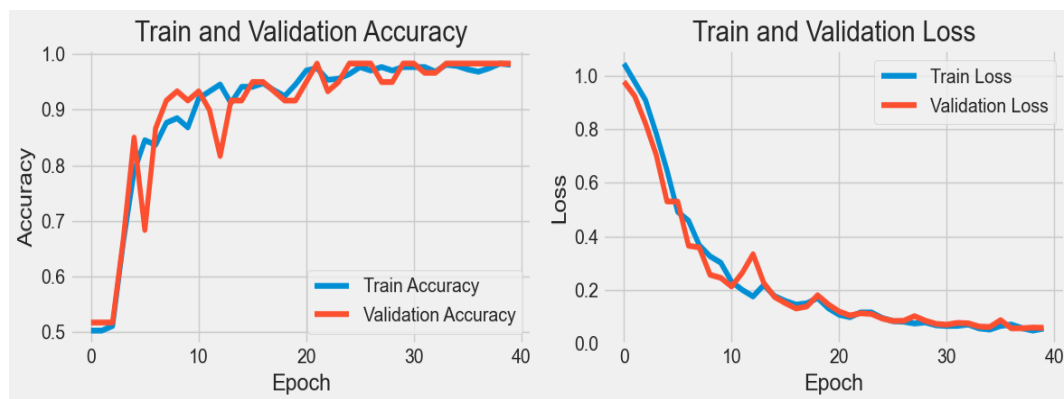


Gambar 4.8 Hasil Prediksi proses training

```
2/2 [=====] - 12s 10s/step - loss: 0.0586 - accuracy: 0.9833
Validation Loss: 0.05858598276972771
Validation Accuracy: 98.33%
```

Gambar 4.9 Hasil akurasi proses training

Dari **Gambar 4.9** yang disajikan, terlihat bahwa akurasi dari proses pelatihan mencapai *accuracy* 0,98 (**98%**), dengan nilai *loss* sebesar 0,05 (**5%**). Grafik akurasi ini dapat ditemukan dalam hasil proses pelatihan setelah 40 kali iterasi (*epoch*) telah dilakukan dan hasil grafik *accaccuracy* dan *loss* dapat dilihat pada **Gambar 4.10**.












Gambar 4.10 Grafik akurasi dan loss






4.7.2 Hasil Proses Testing





Hasil dari proses testing ditampilkan dalam **Tabel 4.2** berikut:

Tabel 4.2 Tabel Pengujian Model





No	Nanas	Aktual	Prediksi
1		Buah Nanas Sudah Matang	Buah Nanas Sudah Matang
2		Buah Nanas Sudah Matang	Buah Nanas Sudah Matang
3		Buah Nanas Sudah Matang	Buah Nanas Sudah Matang
4		Buah Nanas Sudah Matang	Buah Nanas Sudah Matang





5		Buah Nanas Sudah Matang	Buah Nanas Sudah Matang
6		Buah Nanas Sudah Matang	Buah Nanas Sudah Matang
7		Buah Nanas Sudah Matang	Buah Nanas Sudah Matang
8		Buah Nanas Sudah Matang	Buah Nanas Sudah Matang
9		Buah Nanas Sudah Matang	Buah Nanas Sudah Matang





10		Buah Nanas Sudah Matang	Buah Nanas Sudah Matang
11		Buah Nanas Sudah Matang	Buah Nanas Sudah Matang
12		Buah Nanas Sudah Matang	Buah Nanas Sudah Matang
13		Buah Nanas Sudah Matang	Buah Nanas Sudah Matang
14		Buah Nanas Sudah Matang	Buah Nanas Sudah Matang



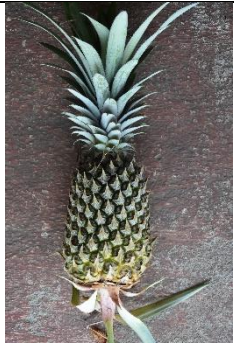

15		Buah Nanas Sudah Matang	Buah Nanas Sudah Matang
16		Buah Nanas Sudah Matang	Buah Nanas Sudah Matang
17		Buah Nanas Sudah Matang	Buah Nanas Sudah Matang
18		Buah Nanas Sudah Matang	Buah Nanas Sudah Matang



19		Buah Nanas Sudah Matang	Buah Nanas Sudah Matang
20		Buah Nanas Sudah Matang	Buah Nanas Sudah Matang
21		Buah Nanas Sudah Setengah Matang	Buah Nanas Sudah Setengah Matang
22		Buah Nanas Sudah Setengah Matang	Buah Nanas Sudah Setengah Matang

23		Buah Nanas Sudah Setengah Matang	Buah Nanas Sudah Setengah Matang
24		Buah Nanas Sudah Setengah Matang	Buah Nanas Sudah Setengah Matang
25		Buah Nanas Sudah Setengah Matang	Buah Nanas Sudah Setengah Matang
26		Buah Nanas Sudah Setengah Matang	Buah Nanas Sudah Setengah Matang

27		Buah Nanas Sudah Setengah Matang	Buah Nanas Sudah Setengah Matang
28		Buah Nanas Sudah Setengah Matang	Buah Nanas Sudah Setengah Matang
29		Buah Nanas Sudah Setengah Matang	Buah Nanas Sudah Matang
30		Buah Nanas Sudah Setengah Matang	Buah Nanas Sudah Setengah Matang

31		Buah Nanas Saat ini Belum Matang	Buah Nanas Saat ini Belum Matang
32		Buah Nanas Saat ini Belum Matang	Buah Nanas Saat ini Belum Matang
33		Buah Nanas Saat ini Belum Matang	Buah Nanas Saat ini Belum Matang
34		Buah Nanas Saat ini Belum Matang	Buah Nanas Saat ini Belum Matang

35		Buah Nanas Saat ini Belum Matang	Buah Nanas Saat ini Belum Matang
36		Buah Nanas Saat ini Belum Matang	Buah Nanas Saat ini Belum Matang
37		Buah Nanas Saat ini Belum Matang	Buah Nanas Saat ini Belum Matang
38		Buah Nanas Saat ini Belum Matang	Buah Nanas Saat ini Belum Matang

39		Buah Nanas Saat ini Belum Matang	Buah Nanas Saat ini Setengah Matang
40		Buah Nanas Saat ini Belum Matang	Buah Nanas Saat ini Belum Matang

Dari hasil pengujian tersebut, kita mendapatkan sebuah tabel yang menggambarkan nilai matriks konfusi (*confusion matrix*) sebagai berikut:

Tabel 4.3 Tabel confusion matriks

		Kelas Aktual			Total Prediksi
		Matang	Setengah Matang	Belum Matang	
Kelas Prediksi	Matang	20	0	0	20
	Setengah Matang	1	9	0	10
	Belum Matang	0	1	9	10
Total Prediksi		20	10	10	40

a. *Accuracy*

Dari *confusion matriks diatas* yang tertera diatas, dihitung nilai akurasi menggunakan rumus. Akurasi mengindikasikan sejauh mana model melakukan klasifikasi secara tepat.

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

Rumus yang digunakan untuk menghitung nilai akurasi dalam tabel di atas adalah sebagai berikut:

$$\frac{20 + 9 + 9}{20 + 0 + 0 + 1 + 9 + 0 + 0 + 1 + 9} \times 100\% = 95\%$$

Akurasi yang berhasil dicapai mencapai 100%, dengan rincian bahwa dataset terdiri dari 40 data yang meliputi buah nanas yang sudah matang, setengah matang, dan belum matang.

b. *Precision*

Precision adalah sebuah matrix yang digunakan untuk mengukur sejauh mana model klasifikasi mampu mengidentifikasi dengan akurat kasus-kasus positif.

Tabel 4.4 Tabel precision

	Matang	Setengah Matang	Belum Matang
TP	20	9	9
FP	0	1	1
TP/ (TP+FP)	1	0,9	0,9

Hal yang dapat disimpulkan dari perhitungan berdasarkan tabel diatas adalah:

$$Precision = 2,8 / 3 \times 100\% = 93\%$$

c. *Recall*

Recall, dalam konteks evaluasi klasifikasi, mengukur sejauh mana kemampuan model untuk mendeteksi semua kasus positif yang benar-benar ada.

Tabel 4.5 Tabel recall

	Matang	Setengah Matang	Belum Matang
TP	20	9	9
FP	0	1	1
TP/ (TP+FP)	1	0,9	0,9

Hal yang dapat disimpulkan dari perhitungan berdasarkan tabel diatas adalah:

$$Recall = 2,8 / 3 \times 100\% = 93\%$$

d. *F1-score*

F1-score adalah matrix evaluasi dalam konteks klasifikasi yang menggabungkan *precision* dan *recall* menjadi satu ukuran tunggal.

Hal yang dapat ditarik kesimpulan dari perhitungan F1 Score adalah:

$$F1-score = \frac{2 \times (Precision \times Recall)}{Precision + Recall} \times 100\%$$

$$F1-score = \frac{2 \times (0,93 \times 0,93)}{0,93 + 0,93} \times 100\% = 93\%$$

BAB 5

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Setelah mengevaluasi hasil dari percobaan yang dilakukan dalam usaha mengklasifikasikan tingkat kematangan buah nanas berdasarkan warna menggunakan pendekatan *Convolutional Neural Network* dalam bidang *Deep Learning*, beberapa kesimpulan dapat diambil sebagai berikut:

1. *Convolutional Neural Network (CNN)* adalah pendekatan yang efektif dalam mengklasifikasikan kematangan buah nanas berdasarkan warna. Hasil percobaan menunjukkan tingkat akurasi yang memuaskan dalam klasifikasi.
2. Penggunaan arsitektur model VGG16 dalam *training* dan *testing* CNN terbukti efektif dan menghasilkan hasil yang lebih akurat.
3. Tingkat *accuracy* yang berhasil dicapai adalah 95%, dengan tingkat *precision* sekitar 93%. *Recall* juga mencapai 93%, nilai F1-score mencapai 93%

5.2 Saran

Dibawah ini disajikan beberapa rekomendasi yang dapat dipertimbangkan untuk pengembangan sistem ke depan:

1. Disarankan untuk pengambilan *dataset* sangat penting untuk mempertimbangkan variasi jarak pengambilan gambar dan tingkat kecerahan agar hasil prediksi lebih baik.
2. Penelitian ini dapat dikembangkan dengan variasi model CNN dan jenis buah lainnya.

DAFTAR PUSTAKA

- Agustin, S., & Wijaya, R. (2019). Beef Image Classification using K-Nearest Neighbor Algorithm for Identification Quality and Freshness. *Journal of Physics: Conference Series*, 1179(1).
- B. MacEachern, Craig., et al. (2023). Detection of Fruit Maturity Stage and Yield Estimation in Wild Bluberry Using Deep Learning Convolutional Neural Networks. *Journal Homepage: Smart Agricultural Technology*. Canada.
- Danukusumo, K.P. (2017). *Implementasi Deep Learning Menggunakan Convolutional Neural Network untuk Klasifikasi Citra Candi Berbasis GPU*. Fakultas Teknologi Industri Universitas Atma Jaya.
- Hadiati, S., & Indriyani, N. L. (2008). Budidaya Nanas. In Balai Penelitian Tanaman Buah Tropika, Pusat Penelitian dan Pengembangan Holtikultura, Badan Penelitian dan Pengembangan Pertanian.
- Kim, P. (2017). *MATLAB deep learning : with machine learning, neural networks and artificial intelligence*. New York, NY: Apress.
- LeCun, Y., Bengio, Y., et al. (2015). Deep Learning. *Nature*, 521, 436. Nature Publishing Group, a division of Macmillan Publisher Limited. All Ridhts Reserved.
- Lestari, Sri., Irvan, Nauval, Kurniawan.(2022). Implementasi Deteksi Objek Penyakit Daun Kentang Dengan Metode Convolutional Neural Network. *Jurnal Aplikasi Teknologi Informasi dan Manajemen (JATIM)*, Vol.3 No.2 Oktober 2022. STIKOM CKI, Jakarta, Indonesia.
- Liu, T., Fang, S., Zhao, Y., Wang, P., & Zhang, J. (2015). *Implementation Of Training Convolutional Neural Networks*.
- M. AL-Masawabe, Marah,. et al. (2021). Papaya Maturity Classification Using Deep Learning Convolutional Networks. *International Journal Of Engineering and Information Systems (IJEAIS)* Vol.5 Issue 12, Desember - 2021, Pages:60-67.

- N. Abu-Jamie, Tanseem., et al. (2022). *Six Fruits Classification Using Deep Learning*. International Journal of Academic Information System Research (IJAIRS) Vol. 6 Issue 1, January - 2022, Pages: 1-8.
- Sarkar, T., Bhattacharjee, R., Salauddin, M., Giri, A., & Chakraborty, R. (2020). Application of Fuzzy Logic Analysis on Pineapple Rasgulla. *Procedia Computer Science*, 167(2019), 779-787.
- Sarkar, T., Nayak, P., & Chakraborty, R. (2018). Review Pineapple [Ananas Comosus (L.)] Product Processing Techniques and Packaging : a Review. *THE IIOAB JOURNAL*, 9, 6-12.
- Shanmugamani, R. (2018). *Deep Learning For Computer Vision: Expert Technique To Train Advanced Neural Networks Using Tensorflow And Keras* (S. M. Moore (Ed)). Packt Publishing.
- Shukla, N. (2018) *Machine Learning with Tensorflow*. Shelter Island, NY: Manning Publications.
- Vasilev., et al. (2019). *Python Deep Learning Second Edition*. Birmingham: Pakt Publishing Ltd.
- Yanto, Budi., et al. (2021). Klasifikasi Kematangan Buah Nanas Dengan Ruang Warna Hue Saturation Intensity. *Jurnal Inovtel Polbeng - Seri Informatika*, Vol. 6, No.1,2021.
- You, W., Shen, C., et al. (2017). A hybrid technique based on convolutional neural network and support vector regression for intelligent diagnosis of rotating machinery. *Advances in Mechanical Engineering*, 9(6), 1687814917704146. SAGE Publications Sage UK: London, England.