

**KLASIFIKASI KEMATANGAN
BUAH PISANG MENGGUNAKAN CNN**



Oleh:

Annisa Reida Raheima (09021282227072)

Nabila Kurnia Aprianti (09021182227003)

Adityo Pangestu (09021282227061)

Mata Kuliah :

Computer Vision

**JURUSAN TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS SRIWIJAYA**

DAFTAR ISI

DAFTAR ISI	2
BAB I PENDAHULUAN.....	4
1.1. Pendahuluan.....	4
1.2. Latar Belakang.....	4
1.3. Rumusan Masalah.....	5
1.4. Tujuan Penelitian	5
1.5. Manfaat Penelitian	5
1.6. Batasan Masalah	6
1.7. Sistematika Penulisan	6
1.8. Kesimpulan	7
2. BAB II KAJIAN LITERATUR	8
2.1. Pendahuluan.....	8
2.2. Tinjauan Pustaka	8
2.2.1. Pisang.....	8
2.2.2. Konvlousi.....	8
2.2.3. <i>Convolutional Neural Network</i>	9
2.2.4. <i>Deep Learning</i>	10
2.3. Penelitian Lain yang Relevan	11
2.4. Kesimpulan	12
3. BAB III METODOLOGI PENELITIAN	13
3.1. Pendahuluan.....	13
3.2. Pengumpulan Data	13
3.3. Tahapan Penelitian	13
3.3.1. Menentukan Kerangka Kerja Penelitian	13
3.3.2. Menentukan Kriteria Pengujian	16
3.3.3. Menentukan Kriteria Pengujian	17
3.3.4. Menentukan Alat Bantu Penelitian	17
3.3.5. Menentukan Pengujian Penelitian.....	17
3.3.6. Melakukan Analisis dan Membuat Kesimpulan	18
3.4. Manajemen Proyek Penelitian	18

4. BAB IV IMPLEMENTASI	19
4.1. Pendahuluan.....	19
4.2. Pengembangan Model Pelatihan Data	19
4.2.1. Import Package	19
4.2.2. Pra-pemrosesan Dataset.....	20
4.2.3. Augmentasi Dataset	21
4.2.4. Struktur Direktori Dataset.....	22
4.2.5. Inisialisasi Model.....	23
4.2.6. Pelatihan Model	24
4.2.7. Evaluasi Model	25
4.3. Kesimpulan	26
5. BAB V HASIL DAN ANALISIS PENELITIAN.....	27
5.1. Data Hasil Percobaan.....	27
5.1.1. Pelatihan Model	27
5.1.2. Hasil Training Model	27
5.1.3. Evaluasi.....	28
5.1.4. Analisis Gambar.....	30
5.1.5. Testing.....	31
6. BAB IV KESIMPULAN DAN SARAN.....	33
6.1. Kesimpulan	33
6.2. Saran	33
DAFTAR PUSTAKA.....	34

BAB I

PENDAHULUAN

1.1. Pendahuluan

Pada bab ini menguraikan latar belakang dilakukannya penelitian ini, menentukan perumusan masalah, tujuan penelitian, manfaat penelitian, dan batasan masalah yang ada berdasarkan latar belakang yang telah dibuat, serta sistematika penulisan.

1.2. Latar Belakang

Buah pisang merupakan salah satu komoditas pertanian yang memiliki peran penting dalam industri pangan global (Wiharja, Y. P., & Harjoko, A. 2014). Tingkat kematangan pisang sangat menentukan nilai jual dan penerimaannya di pasar, baik untuk konsumsi langsung maupun untuk keperluan industri pengolahan (Setiawan, H, H, & Adi, C, K. 2024). Saat ini, klasifikasi kematangan pisang masih banyak dilakukan secara manual berdasarkan pengamatan visual oleh petani atau pedagang (Jusrawati, et al, 2021). Metode manual ini memiliki berbagai kelemahan, seperti subjektivitas, ketidakkonsistenan, dan keterbatasan dalam skala besar (Yanti, R, et al, 2024). Oleh karena itu, diperlukan pendekatan otomatis berbasis kecerdasan buatan yang lebih akurat dan efisien untuk mengklasifikasikan kematangan pisang (Afriansyah, M. et al 2024).

Beberapa penelitian sebelumnya telah mengembangkan sistem klasifikasi kematangan pisang menggunakan metode berbasis pemrosesan citra dan machine learning. Yulianto et al. (2017) menerapkan metode Naïve Bayes dalam klasifikasi kematangan Pisang Ambon dengan menggunakan fitur warna dan tekstur. Hasil penelitian menunjukkan bahwa model dengan kombinasi fitur RGB, HSV, LAB, entropi, energi, dan homogenitas menghasilkan akurasi 90,48%. Bere et al. (2016) menggunakan algoritma K-Nearest Neighbors (K-NN) untuk klasifikasi kematangan Pisang Sunpride berbasis warna, dengan tingkat akurasi 60%-100% tergantung pada kelas kematangan yang diuji. Sementara itu, Huda dan Putra (2023) menerapkan Convolutional Neural Network (CNN) dengan arsitektur VGG16 untuk klasifikasi jenis pisang, menghasilkan akurasi 78% dan presisi 81%, menunjukkan efektivitas CNN dalam klasifikasi berbasis citra.

Meskipun beberapa metode telah diterapkan dalam klasifikasi kematangan pisang, masih terdapat keterbatasan dalam penelitian sebelumnya (Putro, A. D., & Hermawan, A. 2021). Metode Naïve Bayes dan K-NN lebih bergantung pada fitur yang diekstraksi secara manual, sehingga akurasi bergantung pada pemilihan fitur yang tepat (Kosasih, R. (2021). Sementara itu, CNN memiliki keunggulan dalam mengekstraksi fitur secara otomatis dari citra, mengurangi ketergantungan pada pemrosesan fitur manual, dan meningkatkan akurasi dalam klasifikasi berbasis gambar (Pratama, M. D., et al 2024).

Berdasarkan latar belakang tersebut, penelitian ini akan mengembangkan sistem klasifikasi kematangan buah pisang menggunakan Convolutional Neural Network (CNN). Model CNN akan dilatih menggunakan dataset citra pisang dengan berbagai tingkat kematangan untuk mengidentifikasi dan mengklasifikasikan kematangan secara otomatis (Yana, Y. E., & Nafi'iyah, N. 2021). Evaluasi model akan dilakukan menggunakan metrik akurasi, presisi, recall, dan F1-score untuk mengukur performa klasifikasi. Dengan adanya penelitian ini, diharapkan dapat memberikan solusi yang lebih akurat dan efisien dalam klasifikasi kematangan pisang serta mendukung industri pertanian dalam meningkatkan kualitas produk dan distribusi.

1.3. Rumusan Masalah

Berdasarkan latar belakang di atas, maka dapat dirumuskan masalah yang dihadapi sebagai berikut :

1. Bagaimana melakukan identifikasi kematangan pada buah pisang menggunakan metode CNN?
2. Bagaimana cara mendapatkan hasil identifikasi yang akurat sesuai yang diharapkan?

1.4. Tujuan Penelitian

Tujuan dari penelitian ini adalah melakukan analisis dan identifikasi kematangan pada buah pisang dengan metode CNN untuk mendapatkan hasil identifikasi yang akurat buah pisang.

1.5. Manfaat Penelitian

Manfaat dari penelitian ini adalah sebagai berikut:

1. Dapat membantu dalam mengklasifikasikan kualitas buah pisang
2. Memberikan pengetahuan mengenai metode Deep Learning menggunakan metode Convolutional Neural Network untuk klasifikasi.

1.6. Batasan Masalah

Batasan masalah berdasarkan uraian permasalahan di dalam mengimplementasikan Klasifikasi kematangan buah pisang:

1. Data diperoleh dari situs Kaggle. Di dalam data terdapat 2 buah dataset yaitu data train dan data test. Masing-masing data terdapat 3 macam buah yaitu apel, jeruk dan pisang. Pada penelitian kali ini kami hanya menggunakan buah pisang.
2. Implementasi Convolution Neural Network Algorithm menggunakan bahasa pemrograman Python.

1.7. Sistematika Penulisan

Penulis menyusun laporan ini dengan sistematika yang mencakup beberapa sub bagian, yang memberikan penjelasan secara detail, jelas, dan tepat. Sistematika penulisan tugas akhir ini adalah sebagai berikut :

BAB I. PENDAHULUAN

Bab ini membahas mengenai latar belakang, rumusan masalah, tujuan, manfaat, batasan masalah, dan sistematika penulisan.

BAB II. KAJIAN LITERATUR

Bab ini membahas dasar-dasar teori yang mendasari penelitian ini serta mengevaluasi literatur terkait yang relevan. Teori-teori yang dijadikan landasan berasal dari jurnal dan literatur.

BAB III. METODOLOGI PENELITIAN

Bab ini merinci langkah-langkah penelitian yang mengikuti kerangka kerja yang telah ditentukan. Bagian akhir bab ini mencakup perencanaan manajemen proyek yang diperlukan untuk menjalankan penelitian.

BAB III. PENGEMBANGAN PERANGKAT LUNAK

Bab ini membahas mengenai proses pengembangan perangkat lunak yang dikembangkan saat penelitian pembuatan rencana perjalanan wisata menggunakan simulated annealing ini. Setiap tahapan dijelaskan secara rinci untuk memberikan pemahaman yang komprehensif tentang proses pengembangan.

BAB IV. HASIL DAN ANALISIS PENELITIAN

Bab ini menyajikan hasil penelitian yang diperoleh dan melakukan analisis mendalam terhadapnya. Analisis ini menjadi dasar dalam penarikan kesimpulan penelitian.

BAB V. KESIMPULAN DAN SARAN

Bab ini membahas kesimpulan yang telah didapatkan dan dianalisis berdasarkan hasil dari penelitian serta saran yang diberikan berdasarkan penelitian yang telah dilakukan.

1.8. Kesimpulan

Bab ini telah menjelaskan latar belakang penelitian, rumusan masalah, tujuan penelitian, manfaat penelitian, batasan masalah, dan sistematika penulisan. Penelitian mengenai Klasifikasi kualitas buah pisang diharapkan dapat memberikan hasil yang optimal.

BAB II

KAJIAN LITERATUR

2.1. Pendahuluan

Pada bab ini diuraikan tentang landasan teori dan penelitian lain yang relevan dengan Klasifikasi kualitas buah pisang Menggunakan Metode CNN.

2.2. Tinjauan Pustaka

2.2.1. Pisang

Tanaman buah pisang berasal dari Asia Tenggara, termasuk Indonesia, dan kemudian menyebar ke Afrika, Amerika Selatan, dan Amerika Tengah (Amrozi, Y., et al 2022). Pisang adalah salah satu jenis buah yang paling banyak dikonsumsi orang di segala usia, dari bayi hingga orang tua. Itu juga kaya akan vitamin, mineral, dan karbohidrat (Hakim, Z. et al., 2022). Setiap buah memiliki karakteristik yang menunjukkan tingkat kematangannya. Analisis warna kulit buah dengan mata manusia digunakan untuk mengidentifikasi buah pisang oleh petani atau masyarakat umum (Ajizi et al., 2019). Kulit hijau dan kandungan gula yang rendah biasanya ditemukan pada pisang yang belum matang. Pisang yang belum matang biasanya memiliki kulit hijau muda dengan kandungan gula yang rendah, sementara pisang yang sudah matang memiliki kulit kuning dengan kandungan gula yang lebih tinggi (Arya P et al., 2024). Pisang yang terlalu matang memiliki kulit coklat kehitaman dengan kandungan gula yang lebih tinggi (Mulia, M. R., et al., 2024).

2.2.2. Konvolusi

Konvolution dalam matematika adalah operasi matematika di mana dua fungsi menghasilkan fungsi ketiga sebagai akibat dari perubahan fungsi satu oleh fungsi lainnya. Untuk mendapatkan fitur penting dari suatu gambar, konversi dilakukan. (Halim, J., & Fajar, A. N. 2023).

Suatu kernel konvolusi digunakan untuk menumpang tindihkan suatu gambar input ke dalam citra untuk mengubahnya menjadi citra (Windiawan, R. & Suharso, A. 2021). Berikutnya, kita dapat mengalikan setiap angka ke tempatnya.

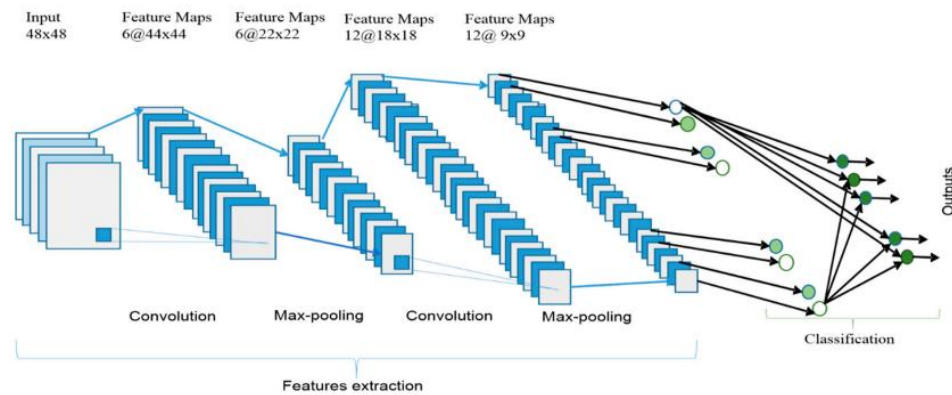
Suatu angka akan dihasilkan jika seluruh hasil perkalian dijumlahkan. Kemudian geser kernel ke bawah, lalu ulangi langkah yang sama sampai kernel berada di ujung bawah gambar input. Lakukan hal yang sama sekali lagi sampai kernel berada di ujung kanan bawah gambar. Proses konvolusi menghasilkan semua jumlah yang dihasilkan dari proses tersebut (Rikendry & Maharil, A. 2022).

2.2.3. *Convolutional Neural Network*

Struktur jaringan ini pertama kali diusulkan oleh Fukushima pada tahun 1988. Pada tahun 1990-an, mengembangkan algoritma pembelajaran berbasis penilaian untuk CNN, yang berhasil memecahkan masalah klasifikasi digit. Hal ini menyebabkan penelitian lebih lanjut tentang CNN dan aplikasinya di berbagai bidang. CNN memiliki beberapa keunggulan dibandingkan DNN, termasuk lebih mirip dengan sistem visual manusia, dioptimalkan untuk visualisasi 2D dan 3D, dan efektif dalam pelatihan dan abstraksi bentuk 2D (Irfansyah, D., et al 2021). CNN yang optimal untuk mencapai variasi bentuk dicapai melalui koneksi dengan bot terdekat dengan jarak parameter yang lebih tinggi dari wilayah yang sama. Dibandingkan dengan algoritma pembelajaran berbasis penilaian, CNN dapat meminimalkan kriteria klasifikasi secara seragam, menjadikannya pilihan yang optimal untuk berbagai aplikasi (Rismiyati, & Luthfiarta, A. 2021).

Arsitektur CNN terdiri dari dua komponen utama: ekstraktor fitur dan pengklasifikasi, seperti yang ditunjukkan pada **Gambar II-1**. Setiap lapisan jaringan dalam lapisan ekstraksi fitur menerima output dari lapisan sebelumnya sebagai input dan meneruskan output tersebut ke lapisan berikutnya. Arsitektur CNN terdiri dari kombinasi dari tiga jenis lapisan: konvolusi, pengumpulan maksimum, dan klasifikasi (Nazhirin, A. F. Z., et al 2023). Lapisan tingkat rendah dan menengah jaringan terdiri dari dua jenis lapisan: lapisan konvolusi dan lapisan pengumpulan maksimum. Lapisan konvolusi bernomor genap untuk konvolusi dan lapisan pengumpulan maksimum bernomor ganjil untuk operabilitas. Pemetaan fitur adalah bidang dua dimensi di mana node output dari lapisan konvolusi dan pengumpulan maksimum dikumpulkan (Gampur, G., et a 2023). Node bidang terhubung ke area kecil dari setiap bidang yang terhubung dari lapisan sebelumnya.

Setiap bidang lapisan biasanya berasal dari kombinasi satu atau lebih bidang dari lapisan sebelumnya (Rendra S, et al 2024).



Gambar II- 1 Arsitektur keseluruhan Jaringan Saraf Konvolusional (CNN)
(Alom, M. Z., 2019)

2.2.4. *Deep Learning*

Pembelajaran Mendalam (DL) dilahirkan dari Jaringan Syaraf Tiruan (NN), yang merupakan bagian dari ML. Sejak awal, DL telah menciptakan gangguan yang semakin besar, dan mencapai tingkat keberhasilan yang luar biasa di hampir setiap domain aplikasi (Gramandha W, I, 2024). Sejak tahun 2006, kelas pembelajaran mesin yang sebagian besar dikembangkan adalah DL yang menggunakan arsitektur pembelajaran mendalam (juga dikenal sebagai pendekatan pembelajaran hierarkis) (Dedi T, H, et al, 2021). Pembelajaran adalah proses yang terdiri dari menaksir parameter model sehingga model yang dipelajari (algoritme) dapat melakukan fungsi tertentu (Hastungkoro, A W, et al 2024). Misalnya, matriks bobot adalah parameter Jaringan Syaraf Tiruan (JST). Sebaliknya, di antara lapisan masukan dan keluaran, DL terdiri dari berbagai lapisan, yang memungkinkan berbagai tahap unit pemrosesan informasi non-linier dengan arsitektur hierarkis, yang dapat digunakan untuk pembelajaran fitur dan klasifikasi pola (Didit I, & Dewi H. (2022). Pembelajaran representasi adalah istilah lain untuk pendekatan pembelajaran berdasarkan representasi data (Herdianto, & Darmeli N 2023). Penelitian terbaru menunjukkan bahwa pembelajaran representasi berbasis DL melibatkan hierarki konsep atau fitur, di mana konsep tingkat tinggi dapat berasal dari konsep tingkat rendah dan konsep tingkat rendah dapat berasal dari konsep tingkat tinggi. DL telah

dijelaskan dalam beberapa artikel sebagai metode pembelajaran universal 2.2.3. Convolutional Neural Network yang mampu memecahkan.

2.3. Penelitian Lain yang Relevan

Penelitian yang dilakukan oleh (Yulianto et al. 2017) menggunakan metode Naïve Bayes untuk klasifikasi tingkat kematangan Pisang Ambon berdasarkan fitur warna dan tekstur. Dataset yang digunakan terdiri dari 105 citra Pisang Ambon, dengan tiga kategori tingkat kematangan (tahap 2, 3, dan 4). Citra diproses dengan mengonversi ruang warna dari RGB ke HSV dan LAB, serta mengekstraksi fitur tekstur menggunakan metode Gray Level Co-occurrence Matrix (GLCM) untuk mendapatkan nilai entropi, energi, homogenitas, dan kontras. Model Naïve Bayes yang diuji terdiri dari tiga variasi fitur, dan model terbaik dengan 9 variabel fitur (RGB, LAB, entropi, energi, homogenitas) mencapai akurasi 90,48%.

Penelitian yang dilakukan oleh (Bere et al. 2016) menerapkan algoritma K-Nearest Neighbors (K-NN) untuk klasifikasi kematangan Pisang Sunpride berdasarkan warna. Dataset penelitian ini terdiri dari 240 gambar Pisang Sunpride, yang dikategorikan dalam empat kelas: Mengkal, Matang, Sangat Matang, dan Busuk. Data warna citra dikonversi dari RGB ke YIQ untuk meningkatkan stabilitas deteksi warna, kemudian diklasifikasikan menggunakan algoritma K-NN dengan nilai $k=1$ dan $k=3$. Hasil penelitian menunjukkan bahwa akurasi tertinggi diperoleh pada kelas "Sangat Matang" sebesar 100%, sedangkan kelas lainnya seperti "Busuk" dan "Mengkal" memiliki akurasi yang lebih rendah, masing-masing 66,67% dan 60%.

Sementara itu, penelitian oleh Huda dan Putra (2023) menggunakan Convolutional Neural Network (CNN) untuk klasifikasi jenis pisang dengan arsitektur VGG16. Dataset yang digunakan terdiri dari 3000 gambar pisang, yang terbagi menjadi 1000 Pisang Jantan, 1000 Pisang Kepok, dan 1000 Pisang Muli. Model CNN yang diterapkan menggunakan fungsi aktivasi ReLU, optimizer Adam, dan loss function categorical_crossentropy. Data dibagi menjadi 80% untuk pelatihan dan 20% untuk pengujian, serta dilakukan teknik augmentasi untuk meningkatkan variasi gambar. Hasil evaluasi menunjukkan bahwa model memiliki akurasi 78%, dengan precision sebesar 81% dan recall sebesar 78%,

menunjukkan bahwa CNN dapat digunakan sebagai metode klasifikasi pisang dengan tingkat akurasi yang cukup baik.

Selain itu, penelitian oleh Adelaida Bere et al. (2016) juga meneliti penggunaan K-NN untuk klasifikasi tingkat kematangan Pisang Sunpride, tetapi menggunakan pendekatan berbasis pemrosesan citra. Dataset terdiri dari 120 gambar pisang yang diambil menggunakan kamera Asus Zenfone 5. Gambar diklasifikasikan berdasarkan fitur warna dengan konversi RGB ke YIQ, lalu dibandingkan menggunakan K-NN dengan $k=3$. Model ini menghasilkan akurasi tertinggi sebesar 66,67% untuk kelas Busuk, dan 100% untuk kelas Sangat Matang, tetapi mengalami kesulitan dalam membedakan kelas Mengkal dan Matang yang hanya mencapai 60%.

Terakhir, penelitian oleh Yulianto et al. (2017) menunjukkan bahwa model Naïve Bayes yang digunakan dalam klasifikasi Pisang Ambon cukup efektif dengan kombinasi fitur warna dan tekstur. Dataset yang digunakan terdiri dari 105 citra Pisang Ambon, yang dikategorikan dalam tiga tahap kematangan. Model terbaik yang diuji menggunakan kombinasi fitur RGB, LAB, entropi, energi, dan homogenitas, menghasilkan akurasi sebesar 90,48%. Model ini menunjukkan bahwa dengan pemrosesan citra yang tepat, metode berbasis probabilistik seperti Naïve Bayes dapat memberikan hasil yang cukup akurat dalam klasifikasi kematangan pisang.

2.4. Kesimpulan

Ada banyak cara untuk menemukan kematangan pisang, menurut penjelasan. Proses preprocessing, pelatihan model, pengklasifikasian, dan pendeteksian adalah beberapa metode. Sebagian besar penelitian yang dipaparkan menggunakan metode CNN, hanya sedikit yang menggunakan metode lain. Maka, penelitian ini akan dilakukan dengan metode CNN.

BAB III

METODOLOGI PENELITIAN

3.1. Pendahuluan

Pada bab ini diuraikan tentang unit penelitian, pengumpulan data, tahapan penelitian, metode pengembangan perangkat lunak dan manajemen proyek penelitian.

3.2. Pengumpulan Data

Data yang digunakan pada penelitian ini adalah dataset Fruits fresh and rotten for classification yang diambil dari kaggle, yang berupa data pisang yang matang dan pisang yang terlalu matang (busuk).



Gambar III- 1 Contoh Dataset Pisang Matang



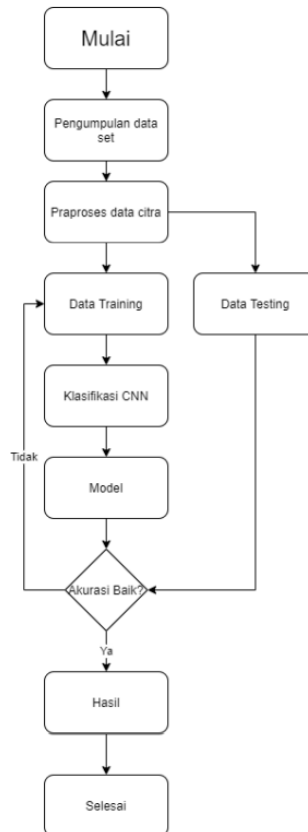
Gambar III- 2 Contoh Dataset Pisang Terlalu Matang (Busuk)

3.3. Tahapan Penelitian

Penelitian dilakukan dalam empat tahap, yakni tahap pengumpulan data, tahap pembagian citra latih dan citra uji, tahap klasifikasi dengan CNN, serta tahap evaluasi.

3.3.1. Menentukan Kerangka Kerja Penelitian

Tahapan penelitian adalah urutan kegiatan yang dikerjakan selama proses penelitian. Dalam melakukan klasifikasi kematangan pisang, terdapat delapan tahapan penelitian yang dilakukan. Berikut adalah diagram yang menggambarkan urutan tahapan penelitian dalam kerangka kerja penelitian tersebut.



Gambar III- 3 Diagram Tahapan Penelitian

Pada Gambar III-3, ditunjukkan beberapa tahapan penelitian yang dilakukan pada penelitian ini. Berikut adalah penjelasan untuk setiap tahapan tersebut.

1. Pengumpulan Data

Data yang digunakan untuk menyelesaikan tugas akhir ini berasal dari dataset Fruits Fresh and Rotten for Classification, yang terdiri dari tiga jenis buah yang dapat diklasifikasikan berdasarkan tingkat kematangannya: pisang, apel, dan jeruk. Dalam penelitian ini, hanya pisang yang matang dan pisang yang terlalu matang (busuk) yang digunakan.

2. Pembagian Citra Latih dan Citra Uji

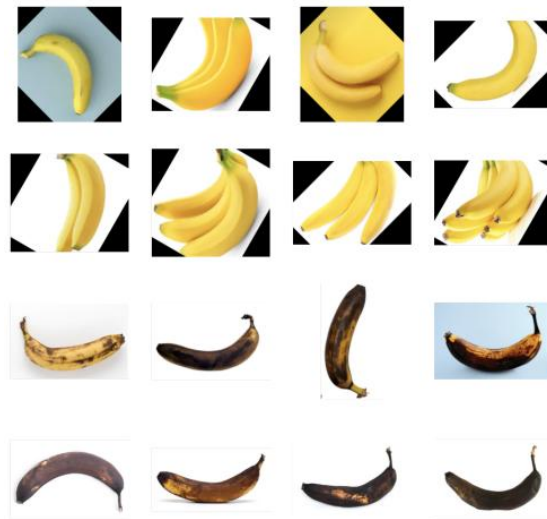
Data dibagi menjadi dua bagian: data pelatihan (latihan) dan data uji. Data pelatihan mengandung 1581 potongan gambar pisang segar dan 2224 potongan gambar pisang busuk, sedangkan data uji mengandung

381 potongan gambar pisang segar dan 530 potongan gambar pisang busuk. Banyak gambar yang dilatih, atau gambar latih, akan mempengaruhi hasil akurasi uji coba. Gambar latih melatih model untuk mencapai tingkat akurasi yang tinggi, sedangkan gambar uji digunakan untuk menguji ketepatan pengklasifikasian kematangan buah pisang sesuai dengan metode yang akan diterapkan.

3. Klasifikasi dengan CNN

Pra Proses Data

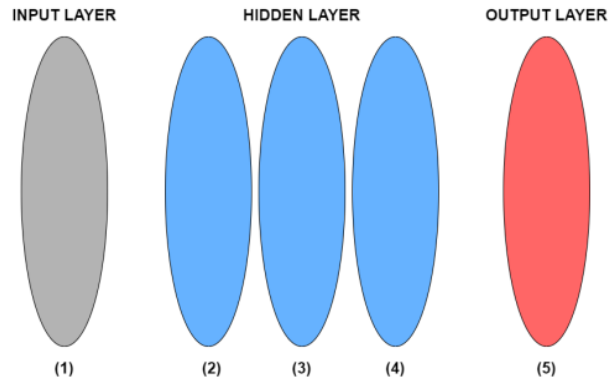
Sebelum memasukkan citra pisang ke dalam Convolution Neural Network, ukuran masing-masing citra harus diubah melalui penskalaan.



Gambar III- 4 Gambar data sebelum dilakukan penskalaan

Pembentukan Model

Untuk melakukan klasifikasi kematangan pisang, model convolutional neural network akan dibuat. Model yang akan dikembangkan adalah Convolution Neural Network dengan satu input layer, tiga hidden layer, dan satu output layer. Proses pengidentifikasian klasifikasi kematangan pisang dilakukan dalam dua tahap. Langkah pertama melibatkan pengajaran data, dan langkah kedua melibatkan pengujian data.



Gambar III- 5 Ilustrasi Model

Pelatihan Model

Pada tahap pelatihan model, gambar pisang matang dan pisang busuk dari data latih disesuaikan dengan jumlah gambar pisang latih dalam dataset. Diharapkan bahwa pengklasifikasian gambar ini dapat mencapai tingkat akurasi yang tinggi, karena jumlah gambar latih yang lebih besar pada dataset menunjukkan kemungkinan tingkat akurasi yang lebih tinggi.

Evaluasi dan pengujian

Setiap proses pengujian menggunakan gambar pisang yang telah melewati tahap praproses. Tahap pengujian dilakukan berulang kali sesuai dengan jumlah gambar pisang matang dan pisang busuk yang ditemukan dalam data uji.

4. Evaluasi

Mulai dari tahap pra proses hingga hasil uji coba, tahap evaluasi diperlukan untuk menemukan hambatan dan kesalahan dalam proses pengklasifikasian gambar.

3.3.2. Menentukan Kriteria Pengujian

Penelitian ini dilakukan dengan mempertimbangkan tingkat akurasi. Salah satu kriteria yang sangat penting adalah tingkat akurasi, yang menunjukkan seberapa baik model dapat membedakan gambar wajah dengan tingkat paralisis wajah. Untuk mendapatkan nilai akurasi, ketepatan, ingatan, dan skor F1, hasil penelitian akan dirangkum dalam matriks confusion untuk mendapatkan nilai

akurasi, ketepatan, ingatan, dan skor F1. Nilai-nilai ini diperoleh dengan menggunakan jumlah prediksi positif benar, negatif benar, dan negatif salah dari matriks confusion.

3.3.3. Menentukan Kriteria Pengujian

Penelitian ini menggunakan confusion matrix sebagai format data pengujian. Tabel III-1 adalah rancangan tabel confusion matrix dan Tabel III-2 adalah rancangan tabel laporan hasil pengujian klasifikasi.

Tabel III- 1 Rancangan Tabel Confusion Matrix

		PREDIKSI	
		Matang	Busuk
A K T U A L	Matang		
	Busuk		

Tabel III- 2 Rancangan Tabel Laporan Hasil Pengujian Klasifikasi

Akurasi	Precision	Recall	F1-Score

3.3.4. Menentukan Alat Bantu Penelitian

Pada penelitian ini dibutuhkan alat untuk membantu proses penelitian. Perangkat lunak yang akan digunakan adalah Google. Google Colab menggunakan bahasa pemrograman Python dan library Tensorflow untuk melakukan training dan evaluasi model.

3.3.5. Menentukan Pengujian Penelitian

Penelitian ini akan menguji metode Convolutional Neural Network untuk mengklasifikasi kematangan pisang. Pengujian ini akan menggunakan data pengujian yang belum pernah dilihat oleh model sebelumnya. Untuk mengukur performa, Confusion Matrix digunakan untuk mengumpulkan nilai Akurasi, Precision, Recall, dan F1-Score. Metode black box akan digunakan untuk menguji perangkat lunak, menekankan pada aspek fungsional.

3.3.6. Melakukan Analisis dan Membuat Kesimpulan

Analisis hasil pengujian dilakukan dengan menggunakan hasil pengujian model yang dapat mengklasifikasikan kematangan pisang. Kesimpulan penelitian ini diperoleh dari analisis ini, dan perbandingan dapat dilihat menggunakan tabel perbandingan.

3.4. Manajemen Proyek Penelitian

Manajemen proyek perangkat lunak merupakan perencanaan kegiatan penelitian. Penelitian ini diawali dengan menyusun rencana penelitian yang mencakup tujuan dan metode penelitian. Tahap selanjutnya adalah pengumpulan data dan pelatihan model deep learning. Setelah itu, dilakukan pengujian model. Akhirnya, mengevaluasi akurasi kinerja sistem klasifikasi kematangan pisang dan membuat kesimpulan dari hasil pengujian.

BAB IV IMPLEMENTASI

4.1. Pendahuluan

Bab ini menjelaskan implementasi sistem klasifikasi kematangan buah pisang menggunakan model Convolutional Neural Network (CNN). Implementasi dilakukan dengan beberapa tahap, mulai dari persiapan lingkungan kerja, pemrosesan data, pelatihan model, evaluasi, hingga pengujian model dengan gambar baru.

4.2. Pengembangan Model Pelatihan Data

Pada pengembangan ini dijelaskan tahapan proses ekstraksi fitur dari dataset training. Hasilnya adalah database berbentuk model yang siap digunakan sebagai acuan dalam melakukan pengujian. Pengembangan ini dilakukan menggunakan Google Colab. Berikut adalah tahapan-tahapan pengembangan model pelatihan data.

4.2.1. Import Package

Tahapan pertama yang dilakukan adalah melakukan import seluruh package yang akan digunakan. Package yang digunakan os, numpy, cv2, matplotlib, tensorflow keras, dan google colab drive.

```
# import library
import os
import numpy as np
import cv2
import matplotlib.pyplot as plt
import pywt.data
import cv2
import matplotlib.pyplot as plt
import matplotlib.image as mpimg
from tensorflow.keras import layers
from tensorflow.keras import Model
from tensorflow.keras.optimizers import RMSprop
from tensorflow.keras.preprocessing.image import ImageDataGenerator
import numpy as np
import random
from tensorflow.keras.preprocessing.image import img_to_array, load_img
from keras.preprocessing import image
import numpy as np
```

Gambar IV- 1 Kode Import Package

4.2.2. Pra-pemrosesan Dataset

Pada awal tahap praproses data penentuan ukuran penskalaan diperlukan untuk menyamakan ukuran citra pisang yang beragam dan mengurangi redundansi pada data citra. Dilakukan proses iterasi untuk mencari ukuran pixel yang cocok. Didapatkan ukuran yang cocok untuk penskalaan adalah 150 x 150, seperti pada Gambar IV-2.



Gambar IV- 2 Hasil prapemrosesan data penskalaan

```
train_datagen = ImageDataGenerator(rescale=1./255)
val_datagen = ImageDataGenerator(rescale=1./255)

train_generator = train_datagen.flow_from_directory(
    train_dir,
    target_size=(150, 150),
    batch_size=32,
    class_mode='binary')

validation_generator = val_datagen.flow_from_directory(
    test_dir,
    target_size=(150, 150),
    batch_size=32,
    class_mode='binary',
    shuffle=False)
```

Gambar IV- 3 Kode prapemrosesan dataset

4.2.3. Augmentasi Dataset

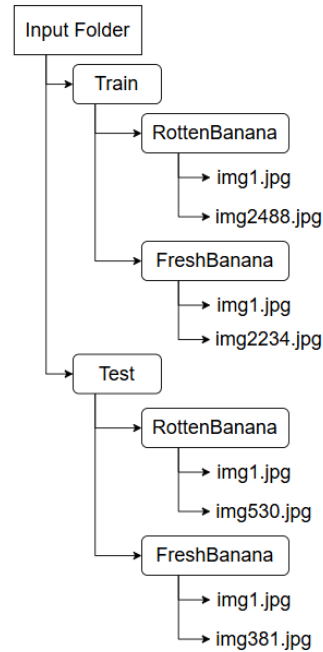
Tahap augmentasi ini dilakukan pada dataset training untuk memperluas dataset dengan menciptakan variasi pada data yang ada. Dalam proses ini, dilakukan augmentasi data pada gambar kelas *fresh banana* untuk menyeimbangkan jumlah gambar dengan kelas *rotten banana*. Augmentasi dilakukan menggunakan ImageDataGenerator dari Keras dengan berbagai transformasi seperti rotasi hingga 20 derajat, pergeseran horizontal dan vertikal hingga 20% dari ukuran gambar, *shear* sebesar 20%, *zoom* hingga 20%, serta pembalikan horizontal. Gambar hasil augmentasi disimpan di folder khusus dengan format JPG dan nama awalan "aug". Target jumlah gambar augmentasi adalah selisih antara jumlah gambar kelas *rotten banana* (2234) dan jumlah gambar awal pada kelas *fresh banana*.

```
# Path folder kelas fresh banana
fresh_banana_dir = '/content/drive/MyDrive/dataset/dataset_tubes_comvis/train/freshbanana'
augmented_dir = '/content/drive/MyDrive/dataset/dataset_tubes_comvis/train/augmentedfreshbanana'
# Buat folder output jika belum ada
os.makedirs(augmented_dir, exist_ok=True)
# Buat objek augmentasi
datagen = ImageDataGenerator(
    rotation_range=20,
    width_shift_range=0.2,
    height_shift_range=0.2,
    shear_range=0.2,
    zoom_range=0.2,
    horizontal_flip=True,
    fill_mode='nearest'
)
# Target augmentasi (misal ingin total mendekati jumlah rotten banana: 2234)
target_augmented_images = 2234 - len(os.listdir(fresh_banana_dir))
image_files = os.listdir(fresh_banana_dir)
generated = 0
for img_name in image_files:
    img_path = os.path.join(fresh_banana_dir, img_name)
    img = load_img(img_path, target_size=(150, 150))
    x = img_to_array(img)
    x = x.reshape((1,) + x.shape)
    i = 0
    for batch in datagen.flow(x, batch_size=1, save_to_dir=augmented_dir, save_prefix='aug', save_format='jpg'):
        i += 1
        generated += 1
        if generated >= target_augmented_images:
            break
    if generated >= target_augmented_images:
        break
print(f"Augmentasi selesai: {generated} gambar disimpan ke {augmented_dir}")
```

Gambar IV- 4 Kode augmentasi dataset

4.2.4. Struktur Direktori Dataset

Untuk mengatur dataset dalam proses pelatihan dan pengujian model, dataset yang telah di augmentasi dibagi ke dalam dua folder utama: train dan test. Masing-masing folder berisi dua subfolder yang merepresentasikan dua kelas, yaitu freshbanana dan rottenbanana, seperti pada Gambar IV-4.



Gambar IV- 5 Struktur Direktori Dataset

Kode yang digunakan untuk mendefinisikan path atau direktori masing-masing kategori dapat dilihat pada gambar.

```
base_path = '/content/drive/MyDrive/dataset/dataset_tubes_comvis'

# Path folder train dan test
train_dir = os.path.join(base_path, 'train')
test_dir = os.path.join(base_path, 'test')

# Path folder training
train_fresh_dir = os.path.join(train_dir, 'freshbanana_combined')
train_rot_dir = os.path.join(train_dir, 'rottenbanana')

# Path folder testing
test_fresh_dir = os.path.join(test_dir, 'freshbanana')
test_rot_dir = os.path.join(test_dir, 'rottenbanana')

# Cek apakah semua direktori valid (opsional)
print("Train Fresh Dir:", os.path.exists(train_fresh_dir))
print("Train Rotten Dir:", os.path.exists(train_rot_dir))
print("Test Fresh Dir:", os.path.exists(test_fresh_dir))
print("Test Rotten Dir:", os.path.exists(test_rot_dir))
```

Gambar IV- 6 Kode Pembagian Dataset

4.2.5. Inisialisasi Model

Pada percobaan ini, digunakan model Convolutional Neural Network (CNN) dengan arsitektur yang terdiri dari dua bagian utama, yaitu **feature extraction layer** dan **classification layer**. Pada bagian feature extraction, dilakukan proses ekstraksi fitur melalui tiga tahapan konvolusi dan max-pooling. Tahap pertama menggunakan konvolusi dengan 32 filter berukuran 3x3 dan fungsi aktivasi ReLU, diikuti dengan max-pooling menggunakan window berukuran 2x2. Tahap kedua melibatkan konvolusi dengan 32 filter berukuran 3x3 dan fungsi aktivasi ReLU, dilanjutkan dengan max-pooling berukuran 2x2. Tahap ketiga menggunakan 64 filter dengan ukuran 3x3 dan fungsi aktivasi ReLU, yang juga diikuti oleh max-pooling dengan window 2x2.

Selanjutnya, pada bagian classification, dilakukan proses **flattening** untuk mengubah feature map menjadi bentuk satu dimensi. Kemudian, data hasil flattening diproses oleh fully connected layer (FC layer) yang terdiri dari 512 node dan menggunakan fungsi aktivasi ReLU. Akhirnya, output layer memiliki 1 node dengan fungsi aktivasi sigmoid, yang berfungsi untuk menghasilkan prediksi klasifikasi, yaitu angka 0 jika gambar yang diuji merupakan pisang matang, dan angka 1 jika gambar tersebut adalah pisang busuk.

Model: "functional_2"

Layer (type)	Output Shape	Param #
input_layer_1 (InputLayer)	(None, 150, 150, 3)	0
conv2d_3 (Conv2D)	(None, 148, 148, 32)	896
max_pooling2d_3 (MaxPooling2D)	(None, 74, 74, 32)	0
conv2d_4 (Conv2D)	(None, 72, 72, 32)	9,248
max_pooling2d_4 (MaxPooling2D)	(None, 36, 36, 32)	0
conv2d_5 (Conv2D)	(None, 34, 34, 64)	18,496
max_pooling2d_5 (MaxPooling2D)	(None, 17, 17, 64)	0
flatten_1 (Flatten)	(None, 18496)	0
dense_2 (Dense)	(None, 512)	9,470,464
dense_3 (Dense)	(None, 1)	513

Total params: 9,499,617 (36.24 MB)
Trainable params: 9,499,617 (36.24 MB)
Non-trainable params: 0 (0.00 B)

Gambar IV- 7 Layer CNN

```

# Input berdimensi 150x150px dengan 3 channel untuk Blue, Green, dan Red
img_input = layers.Input(shape=(150, 150, 3))
# Convolution layer I: 16 filter berukuran 3x3
x = layers.Conv2D(32, 3, activation='relu')(img_input)
# Pooling layer: Max-pooling layer 2x2
x = layers.MaxPooling2D(2)(x)
# Convolution layer II: 32 filter berukuran 3x3
x = layers.Conv2D(32, 3, activation='relu')(x)
# Pooling layer: Max-pooling layer 2x2
x = layers.MaxPooling2D(2)(x)
# Convolution layer III: 64 filter berukuran 3x3
x = layers.Conv2D(64, 3, activation='relu')(x)
# Pooling layer: Max-pooling layer 2x2
x = layers.MaxPooling2D(2)(x)
# Fully connected layers
x = layers.Flatten()(x)
x = layers.Dense(512, activation='relu')(x)
output = layers.Dense(1, activation='sigmoid')(x)
# Menggabungkan layer input dan output
model = Model(img_input, output)
model.summary()

```

Gambar IV- 8 Kode Inisialisasi Model

Model ini dioptimalkan menggunakan RMSprop optimizer dan loss binary crossentropy, yang sesuai untuk klasifikasi dua kelas. Total parameter yang dilatih pada model ini adalah sekitar 9,4 juta dengan ukuran memori sebesar 36,2 MB.

4.2.6. Pelatihan Model

Pada pembuatan model di tahap ini model sudah bisa dilatih. Pada konfigurasi ini model akan dilatih menggunakan learning rate 0.001 dan epoch 15.

```

from tensorflow.keras.optimizers import RMSprop

model.compile(loss='binary_crossentropy',
              optimizer=RMSprop(learning_rate=0.001),
              metrics=['acc'])

```

```

history = model.fit(
    train_generator,
    steps_per_epoch=120, # 3805 images = batch_size * steps =
    epochs=15,
    validation_data=validation_generator,
    validation_steps=30, # 911 images = batch_size * steps =
    verbose=2)

```

Gambar IV- 9 Kode pelatihan model

4.2.7. Evaluasi Model

Setelah model dilatih, model akan dievaluasi dengan menampilkan grafik accuracy training and validation, loss training and validation pada Gambar IV-10, tabel confusion matrix pada Gambar dan laporan klasifikasi yang akan menampilkan nilai akurasi, Precision, Recall, dan F1-Score pada Gambar IV-11.

```
acc = history.history['acc']
val_acc = history.history['val_acc']

loss = history.history['loss']
val_loss = history.history['val_loss']

epochs = range(len(acc))

plt.plot(epochs, acc)
plt.plot(epochs, val_acc)
plt.title('Training and validation accuracy')

plt.figure()

plt.plot(epochs, loss)
plt.plot(epochs, val_loss)
plt.title('Training and validation loss')
```

Gambar IV- 10 Kode grafik akurasi dan loss

```
# Threshold yang bisa kamu sesuaikan (misalnya 0.5, 0.6, 0.7)
threshold = 0.5
# Prediksi pada validation set
pred_probs = model.predict(validation_generator, verbose=1)
# Ambil true labels
true_labels = validation_generator.classes[:len(pred_probs)]
# Ubah ke prediksi kelas berdasarkan threshold
pred_classes = (pred_probs > threshold).astype("int32").flatten()
# Confusion Matrix
cm = confusion_matrix(true_labels, pred_classes)
class_names = list(validation_generator.class_indices.keys())
# Visualisasi Confusion Matrix
plt.figure(figsize=(6, 4))
sns.heatmap(cm, annot=True, fmt='d', cmap='Blues', xticklabels=class_names, yticklabels=class_names)
plt.xlabel('Predicted')
plt.ylabel('Actual')
plt.title(f'Confusion Matrix (Threshold = {threshold})')
plt.show()
# Metrik Evaluasi
print("Classification Report:")
print(classification_report(true_labels, pred_classes, target_names=class_names))
print(f"Akurasi : {accuracy_score(true_labels, pred_classes):.4f}")
print(f"Presisi : {precision_score(true_labels, pred_classes):.4f}")
print(f"Recall : {recall_score(true_labels, pred_classes):.4f}")
print(f"F1-score : {f1_score(true_labels, pred_classes):.4f}")
```

Gambar IV-11 Kode confusion matrix dan classification report

4.3. Kesimpulan

Berdasarkan pembahasan dalam bab ini, dapat disimpulkan bahwa tahapan tahapan yang diterapkan dalam pengembangan model pelatihan data ini berjalan lancar.

BAB V

HASIL DAN ANALISIS PENELITIAN

5.1. Data Hasil Percobaan

5.1.1. Pelatihan Model

Pada pelatihan model didapatkan 4722 gambar yang terbagi kedalam 2 kelas untuk data latih, dan 911 gambar yang terbagi kedalam 2 kelas untuk data validasi. Konfigurasi yang digunakan adalah membuat model yang dilatih dari awal dengan menggunakan epoch 15.

5.1.2. Hasil Training Model

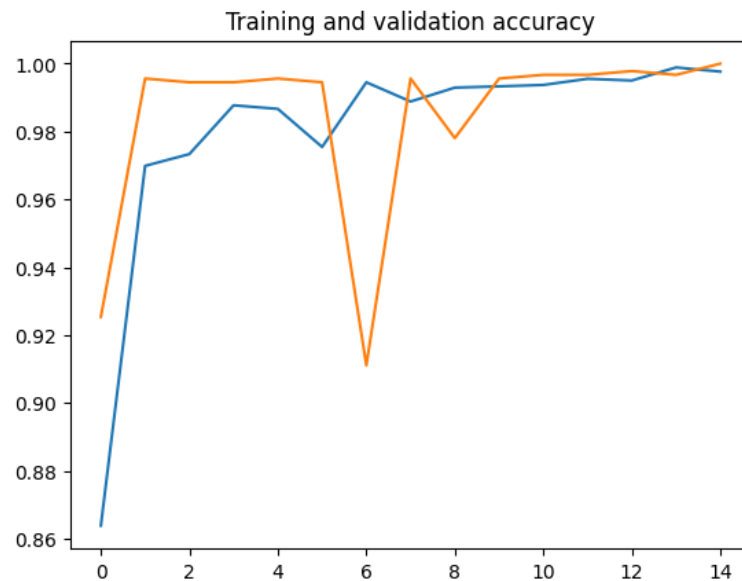
Sebanyak 4722 gambar yang terdapat di pelatihan model selanjutnya dilatih dengan 15 epochs dengan 911 gambar sebagai data validasi. Sebelum melakukan training tentukan terlebih dahulu jumlah langkah yang akan dihasilkan dari generator sebelum mendeklarasikan satu epoch selesai dan memulai epoch berikutnya (step per epoch). Jumlah step biasanya tergantung pada jumlah batch size dan jumlah datasetnya. Disini step per epoch untuk data yang akan dilatih adalah sebanyak 120. Kemudian tentukan juga step untuk data validasinya-nya, yaitu sebanyak 30 step.

```
Epoch 1/15
/usr/local/lib/python3.11/dist-packages/keras/src/trainers/epoch_iterator.py:107: UserWarning:
  self._interrupted_warning()
120/120 - 184s - 2s/step - acc: 0.8638 - loss: 0.3465 - val_acc: 0.9254 - val_loss: 0.1820
Epoch 2/15
120/120 - 50s - 417ms/step - acc: 0.9699 - loss: 0.0912 - val_acc: 0.9956 - val_loss: 0.0351
Epoch 3/15
120/120 - 170s - 1s/step - acc: 0.9733 - loss: 0.0790 - val_acc: 0.9945 - val_loss: 0.0309
Epoch 4/15
120/120 - 50s - 419ms/step - acc: 0.9877 - loss: 0.0516 - val_acc: 0.9945 - val_loss: 0.0207
Epoch 5/15
120/120 - 173s - 1s/step - acc: 0.9867 - loss: 0.0432 - val_acc: 0.9956 - val_loss: 0.0211
Epoch 6/15
120/120 - 57s - 472ms/step - acc: 0.9754 - loss: 0.0636 - val_acc: 0.9945 - val_loss: 0.0241
Epoch 7/15
120/120 - 205s - 2s/step - acc: 0.9945 - loss: 0.0194 - val_acc: 0.9111 - val_loss: 0.2574
Epoch 8/15
120/120 - 53s - 438ms/step - acc: 0.9888 - loss: 0.0423 - val_acc: 0.9956 - val_loss: 0.0169
Epoch 9/15
120/120 - 169s - 1s/step - acc: 0.9929 - loss: 0.0217 - val_acc: 0.9780 - val_loss: 0.0581
Epoch 10/15
120/120 - 57s - 472ms/step - acc: 0.9933 - loss: 0.0109 - val_acc: 0.9956 - val_loss: 0.0102
Epoch 11/15
120/120 - 204s - 2s/step - acc: 0.9937 - loss: 0.0203 - val_acc: 0.9967 - val_loss: 0.0091
Epoch 12/15
120/120 - 52s - 435ms/step - acc: 0.9955 - loss: 0.0175 - val_acc: 0.9967 - val_loss: 0.0096
Epoch 13/15
120/120 - 217s - 2s/step - acc: 0.9950 - loss: 0.0181 - val_acc: 0.9978 - val_loss: 0.0054
Epoch 14/15
120/120 - 50s - 416ms/step - acc: 0.9989 - loss: 0.0047 - val_acc: 0.9967 - val_loss: 0.0084
Epoch 15/15
120/120 - 174s - 1s/step - acc: 0.9976 - loss: 0.0085 - val_acc: 1.0000 - val_loss: 0.0016
```

Gambar V- 1 Hasil Training

5.1.3. Evaluasi

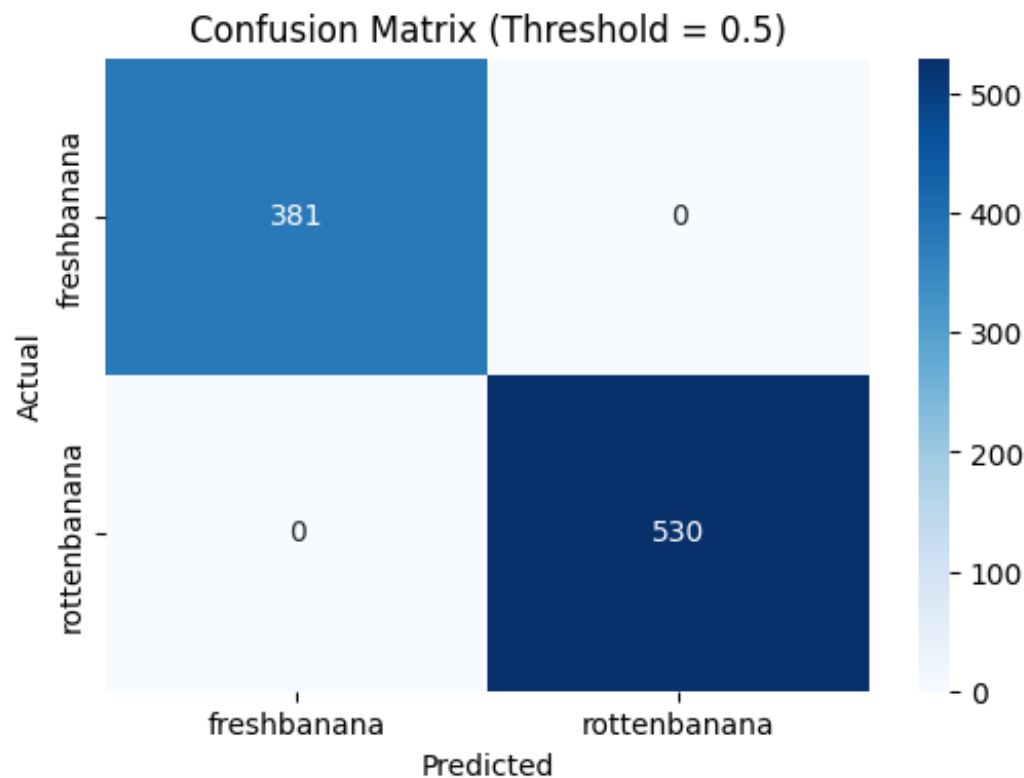
Konfigurasi percobaan dengan learning rate 0,001 dan epoch 15. Gambar V-2 menunjukkan grafik akurasi training dan validation, Gambar V-3 menunjukkan grafik loss training dan validation, Gambar V-4 menunjukkan hasil confusion matrix dan Tabel V-1 menunjukkan hasil classification report dari percobaan konfigurasi I. Training dan validasi akurasi terlihat yang terus naik dan stabil pada nilai 0.99 menunjukkan model yang bagus dan tidak menyebabkan overfitting.



Gambar V- 2 Grafik akurasi training dan validasi



Gambar V- 3 Margin of error training dan validasi



Classification Report:

	precision	recall	f1-score	support
freshbanana	1.00	1.00	1.00	381
rottenbanana	1.00	1.00	1.00	530
accuracy			1.00	911
macro avg	1.00	1.00	1.00	911
weighted avg	1.00	1.00	1.00	911

Gambar V- 4 *Confusion Matrix*

Tabel V- 1 Laporan Hasil Pengujian Klasifikasi

Akurasi	Precision	Recall	F1-Score
1.00	1.00	1.00	1.00

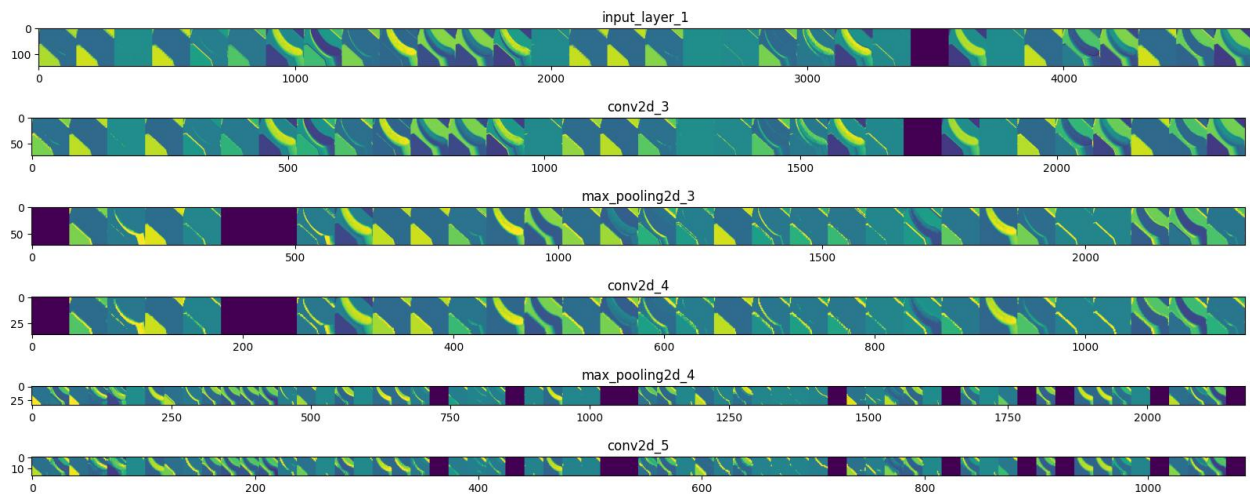
Berdasarkan hasil pengujian model, klasifikasi terhadap gambar buah pisang menunjukkan bahwa sebanyak 381 gambar freshbanana berhasil diklasifikasikan dengan benar sebagai pisang segar. Sementara itu, untuk kategori rottenbanana, terdapat 530 gambar yang juga berhasil dikenali dengan tepat sebagai pisang busuk. Hasil ini menunjukkan bahwa model memiliki performa yang baik dalam

membedakan antara pisang segar dan pisang busuk, dengan jumlah klasifikasi benar yang cukup signifikan pada kedua kelas tersebut.

5.1.4. Analisis Gambar



Gambar V- 5 Pisang yang dilakukan analisis gambar



Gambar V- 6 *Display Feature Map*

Dapat dilihat bahwa bercak hitam pada badan pisang menunjukan kebusukan pada buah pisang. Setelah dilakukan 3 kali proses konvolusi dan pooling didapatkan hasil display feature map seperti gambar di atas. Fragment ini dihasilkan oleh image pooling dan sesuai dengan gambar yang diinginkan karena dapat membedakan bercak kehitaman pada buah pisang.

5.1.5. Testing

Testing dilakukan dengan memasukkan gambar pisang dan didapatkan hasil sebagai berikut:

Testing Pisang A:

```
# Path ke folder yang berisi gambar-gambar
folder_path = '/content/drive/MyDrive/Adityo Pangestu/pisang'
# Ambil semua nama file gambar
image_files = [f for f in os.listdir(folder_path) if f.lower().endswith(('.png', '.jpg', '.jpeg'))]
# Ukuran input yang dibutuhkan model
target_size = (150, 150)
# Proses setiap gambar
for file_name in image_files:
    img_path = os.path.join(folder_path, file_name)
    # Load dan preprocess gambar
    img = image.load_img(img_path, target_size=target_size)
    x = image.img_to_array(img)
    x = np.expand_dims(x, axis=0) / 255. # Normalisasi
    # Prediksi
    preds = model.predict(x)
    label = "Fresh Banana" if preds[0][0] < 0.5 else "Rotten Banana"
    confidence = preds[0][0]
    # Tampilkan hasil
    plt.imshow(img)
    plt.title(f"{file_name}\nPrediction: {label} ({confidence:.2f})")
    plt.axis('off')
    plt.show()
```

Gambar V- 7 Kode Testing



Gambar V- 8 Pisang bagus yang dites

Dapat dilihat hasil testing yang dilakukan untuk gambar fresh banana dapat diprediksi dengan output 0 yang berarti pisang tersebut tidak busuk atau dapat dikatakan pisang tersebut merupakan pisang segar.

Testing Pisang B

pisang4.jpg
Prediction: Rotten Banana (1.00)



Gambar V- 9 Pisang busuk yang dites

Dapat dilihat hasil testing yang dilakukan untuk gambar rotten banana dapat diprediksi dengan output 1 yang berarti pisang tersebut busuk.

BAB IV

KESIMPULAN DAN SARAN

6.1. Kesimpulan

Berdasarkan hasil penelitian dan diskusi yang kami lakukan, dalam pengidentifikasian kematangan buah pisang menggunakan pendekatan CNN, kami menerapkan 3 lapis proses konvolusi dan pooling, lalu 1 kali fully connected layer dan dengan epoch sebanyak 15 kali, didapatkan hasil model yang memiliki akurasi sebesar 99,7% untuk data latih dan 100% untuk data uji. Sehingga dapat dikatakan model pengidentifikasian menggunakan Convolutional Neural Network ini memberikan hasil yang cukup baik dan optimal

6.2. Saran

Saran yang dapat menjadi panduan untuk penelitian selanjutnya antara lain adalah dengan melakukan segmentasi pada area pisang dalam dataset sebelum digunakan untuk melatih model. Hal ini bertujuan agar proses klasifikasi dapat difokuskan hanya pada area objek pisang, sehingga mengurangi gangguan dari latar belakang dan meningkatkan akurasi prediksi. Selain itu, disarankan untuk meningkatkan jumlah data pada setiap kelas agar dataset menjadi lebih seimbang dan representatif. Keseimbangan jumlah data pada masing-masing kelas sangat penting untuk mencegah bias model terhadap kelas tertentu dan memperbaiki performa generalisasi model.

DAFTAR PUSTAKA

- Afriansyah, M., Saputra, J., Ardhana, V. Y. P., & Sa'adati, Y. (2024). Algoritma Naive Bayes Yang Efisien Untuk Klasifikasi Buah Pisang Raja Berdasarkan Fitur Warna. *Journal of Information Systems Management and Digital Business*, 1(2), 236-248.
- Ajizi, M. F., Syauqy, D., & Ichsan, M. H. H. (2019). Klasifikasi Kematangan Buah Pisang Berbasis Sensor Warna Dan Sensor Load Cell Menggunakan Metode Naive Bayes. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, 3(3), 2472-2479.
- Amrozi, Y., Yuliati, D., Susilo, A., Novianto, N., & Ramadhan, R. (2022). Klasifikasi Jenis Buah Pisang Berdasarkan Citra Warna dengan Metode SVM. *Jurnal SISFOKOM*, 11(3), 394–399. <https://doi.org/10.32736/sisfokom.v11i3.1502>
- Arya Pangestu, Bedy Purnama, & Risnandar. (2024). Vision Transformer untuk Klasifikasi Kematangan Pisang. *Jurnal Teknologi Informasi dan Ilmu Komputer (JTIK)*, 11(1), 75-84. DOI: 10.25126/jtiik.20241117389
- Arya Widya Hastungkoro, Aditya Dwi Putro Wicaksono, & Yesy Diah Rosita. (2024). Klasifikasi Kualitas dan Kematangan Pisang Cavendish Menggunakan Convolutional Neural Network. *Jurnal Saintekom*, 14(2), 185-194. DOI: 10.33020/saintekom.v14i2.686.
- Bere, G. A., Tamatjita, E. N., & Kusumaningrum, A. (2016, November). Klasifikasi Untuk Menentukan Tingkat Kematangan Buah Pisang Sunpride. In *Conference SENATIK STT Adisutjipto Yogyakarta* (Vol. 2, pp. 109-113).
- Dedi Tri Hermanto, Arief Setyanto, & Emha Taufiq Luthfi. (2021). Algoritma LSTM-CNN untuk Sentimen Klasifikasi dengan Word2vec pada Media Online. *Citec Journal*, 8(1), 64-65. ISSN: 2354-5771.
- Didit Iswantoro, & Dewi Handayani. (2022). Klasifikasi Penyakit Tanaman Jagung Menggunakan Metode Convolutional Neural Network (CNN). *Jurnal Ilmiah Universitas Batanghari Jambi*, 22(2), 900-905. DOI: 10.33087/jiubj.v22i2.2065.
- Gampur, G., Ordiyasa, I. W., & Mulyani, S. H. (2023). Klasifikasi Jenis Pisang Menggunakan Convolutional Neural Network (CNN). Universitas Respati Yogyakarta.
- Gramandha Wega Intyanto. (2024). Klasifikasi Citra Bunga dengan Menggunakan Deep Learning: CNN (Convolution Neural Network). *Jurnal Arus Elektro Indonesia (JAEI)*, 80-81.
- Hakim, Z., Rahayu, S., & Irawati, K. (2022). Klasifikasi Tingkat Kematangan Buah Pisang Kepok Menggunakan Algoritma Naive Bayes. *Academic Journal of Computer Science Research*, 4(1), 8-11.
- Halim, J., & Fajar, A. N. (2023). Klasifikasi Pisang Berbasis Algoritma VGG16 Melalui Metode CNN Deep Learning. *Jurnal INFORMASI*.

- Herdianto, & Darmeli Nasution. (2023). Implementasi Metode CNN untuk Klasifikasi Objek. *METHOMIKA: Jurnal Manajemen Informatika & Komputerisasi Akuntansi*, 7(1), 54-60. DOI: <https://doi.org/10.46880/jmika.Vol7No1.pp54-60>.
- Hironimus Hendra Setiawan, & Cyprianus Kuntoro Adi. (2024). Klasifikasi Jenis Buah Pisang dengan Image Processing Menggunakan Metode Backpropagation. Universitas Sanata Dharma
- Huda, F., & Putra, M. P. K. (2023). Klasifikasi Jenis Buah Pisang Menggunakan Algoritma Convolutional Neural Network. *Journal of Artificial Intelligence and Technology Information*, 1(3), 100-105.
- Irfansyah, D., Mustikasari, M., & Suroso, A. (2021). Implementasi Arsitektur CNN AlexNet untuk Klasifikasi Hama pada Citra Daun Tanaman Kopi. *Jurnal Informatika JPIT*.
- Jusrawati, Ayu Fitri, & Andi Baso Kaswar. (2021). Klasifikasi Tingkat Kematangan Buah Pisang Dalam Ruang Warna RGB Menggunakan Jaringan Syaraf Tiruan (JST). *Jurnal JESSI*, 2(1), 49-50. DOI: <https://ojs.unm.ac.id/JESSI/index>.
- Kosasih, R. (2021). Klasifikasi Tingkat Kematangan Pisang Berdasarkan Ekstraksi Fitur Tekstur dan Algoritme KNN. *Jurnal Nasional Teknik Elektro dan Teknologi Informasi*, 10(4), 383-388.
- Mulia, M. R., Kaswar, A. B., Andayani, D. D., & Agung, A. S. (2024). Klasifikasi Kandungan Nutrisi Buah Pisang Berdasarkan Fitur Tekstur dan Warna LAB menggunakan Jaringan Syaraf Tiruan Berbasis Pengolahan Citra Digital. *Jurnal Teknologi Informasi dan Ilmu Komputer*, 11(3), 507-518.
- Nazhirin, A. F. Z., Muttaqin, M. R., & Hermanto, T. I. (2023). Klasifikasi Kondisi Ban Kendaraan Menggunakan Arsitektur VGG16. *Jurnal INTI NUSA MANDIRI*.
- Pratama, M. D., Gustriansyah, R., & Purnamasari, E. (2024). Klasifikasi Penyakit Daun Pisang menggunakan Convolutional Neural Network (CNN). *Jurnal Teknologi Terpadu*, 10(1), 1-6.
- Putro, A. D., & Hermawan, A. (2021). Pengaruh Cahaya dan Kualitas Citra dalam Klasifikasi Kematangan Pisang Cavendish Berdasarkan Ciri Warna Menggunakan Artificial Neural Network. *MATRIK: Jurnal Manajemen, Teknik Informatika dan Rekayasa Komputer*, 21(1), 215-228.
- Rahma Yanti, Fajri Rinaldi Chan, & Agung Ramadhanu. (2024). Penerapan Image Processing untuk Identifikasi Jenis Pisang Emas dan Pisang Kapas Menggunakan Metode K-Means Clustering. *Journal of Education Research*, 5(4), 4938-4943. ISSN: 2746-0738.
- Rendra Soekarta, Muh. Fadli Hasa, & Endang Stahputri Ode. (2024). Klasifikasi Kematangan Buah Pisang Secara Real-Time Menggunakan Convolutional Neural Network Berbasis Android. *INSECT*, 10(1), 11-20. ISSN: 2988-3229.

- Rikendry, & Maharil, A. (2022). Perbandingan Arsitektur VGG16 dan ResNet50 untuk Rekognisi Tulisan Tangan Aksara Lampung. *Jurnal JATIKA*.
- Rismiyati, & Luthfiarta, A. (2021). Klasifikasi Kualitas Buah Salak dengan Transfer Learning Arsitektur VGG16. *Jurnal Telematika*.
- Wiharja, Y. P., & Harjoko, A. (2014). Pemrosesan Citra Digital untuk Klasifikasi Mutu Buah Pisang Menggunakan Jaringan Saraf Tiruan (JST). *IJEIS (Indonesian Journal of Electronics and Instrumentation Systems)*, 4(1), 57–68. <https://doi.org/10.22146/ijeis.4222>
- Windiawan, R., & Suharso, A. (2021). Identifikasi Penyakit Daun Kopi Menggunakan Metode Deep Learning VGG16. *Jurnal Explore IT*.
- Yana, Y. E., & Nafi'iyah, N. (2021). Klasifikasi Jenis Pisang Berdasarkan Fitur Warna, Tekstur, Bentuk Citra Menggunakan SVM dan KNN. *Research Journal of Computer, Information System & Technology Management*, 4(1). <https://doi.org/10.25273/research.v4i1.6687>
- Yulianto, D., Whidhiasih, R. N., & Maimunah, M. (2017). Klasifikasi Tahap Kematangan Pisang Ambon Berdasarkan Warna Menggunakan Naive Bayes. *PIKSEL: Penelitian Ilmu Komputer Sistem Embedded and Logic*, 5(2), 60-67.