

IMPLEMENTASI *CONVOLUTIONAL NEURAL NETWORK* DALAM APLIKASI ANDROID UNTUK KLASIFIKASI KEMATANGAN BUAH PISANG

Sabdha Putra Laudri

Program Studi Teknik Informatika, Universitas Indraprasta PGRI
Jalan Raya Tengah No 80, Jakarta Timur – DKI Jakarta, Indonesia
Email: sabdhacipaku04@gmail.com

ABSTRAK

Indonesia merupakan negara tropis dengan produksi buah yang melimpah, salah satunya adalah pisang. Berdasarkan data Badan Pusat Statistik (BPS) tahun 2023, pisang menjadi komoditas buah dengan jumlah produksi tertinggi, yaitu sebesar 9,3 juta ton. Maka dari itu, penelitian ini bertujuan untuk mengembangkan dan mengimplementasikan model klasifikasi berbasis *Convolutional Neural Network* (CNN) dalam aplikasi Android untuk mengenali tingkat kematangan buah pisang, seperti mentah, setengah matang, matang, dan terlalu matang secara otomatis, cepat, dan objektif. Model *Convolutional Neural Network* (CNN) dilatih menggunakan 4112 citra yang dibagi menjadi data pelatihan (3288 citra), validasi (412 citra), dan pengujian (412 citra) dengan rasio 80:10:10. Untuk mendukung kemudahan penggunaan, aplikasi Android dirancang dengan antarmuka yang sederhana, interaktif, dan ramah pengguna. Selain itu, model dioptimalkan agar dapat berjalan secara efisien pada perangkat dengan keterbatasan sumber daya. Hasil akhir dari penelitian menunjukkan bahwa model *Convolutional Neural Network* (CNN) mampu mencapai akurasi sebesar 95,87% dalam mengklasifikasikan tingkat kematangan pisang.

Kata kunci: Android, Buah Pisang, *Convolutional Neural Network*, Implementasi

ABSTRACT

Indonesia is a tropical country with abundant fruit production, one of which is bananas. Based on data from the Central Statistics Agency (BPS) in 2023, bananas are the fruit commodity with the highest production amount, which is 9.3 million tonnes. Therefore, this research aims to develop and implement a classification model based on *Convolutional Neural Network* (CNN) in an Android application to recognise the ripeness level of bananas, such as unripe, half-ripe, ripe, and overripe automatically, quickly, and objectively. The *Convolutional Neural Network* (CNN) model was trained using 4112 images divided into training data (3288 images), validation (412 images), and testing (412 images) with a ratio of 80:10:10. To support ease of use, the Android application was designed with a simple, interactive, and user-friendly interface. In addition, the model was optimised to run efficiently on resource-constrained devices. The final results of the study show that the *Convolutional Neural Network* (CNN) model is able to achieve an accuracy of 95.87% in classifying the ripeness level of bananas.

Keywords: Android, Banana Fruit, *Convolutional Neural Network*, Implementation

1. PENDAHULUAN

Indonesia adalah negara tropis yang kaya akan sumber daya alam, termasuk hasil pertanian seperti buah-buahan yang menjadi komoditas ekspor ke berbagai negara. Salah satu upaya pemerintah untuk meningkatkan produktivitas dan daya saing ekspor buah dilakukan melalui program Kampung Hortikultura yang fokus pada pengembangan komoditas buah di tingkat desa. Berdasarkan data Badan Pusat Statistik (BPS) tahun 2023, total produksi buah di Indonesia mencapai 28,24 juta ton, dengan pisang sebagai komoditas tertinggi sebesar 9,3 juta ton [1]. Meskipun memiliki nilai ekonomi yang tinggi, potensi tersebut belum dimanfaatkan secara optimal. Karena pisang hanya bertahan 5-7 hari setelah panen, diperlukan proses klasifikasi yang cepat sebelum distribusi untuk mencegah kerugian akibat pembusukan.

Meskipun teknologi sudah berkembang pesat, klasifikasi tingkat kematangan buah pisang umumnya masih dilakukan secara visual oleh kebanyakan orang. Proses ini bersifat subjektif dan rentan terhadap inkonsistensi, terutama jika dilakukan oleh orang dengan tingkat pengalaman yang berbeda-beda. Tidak adanya standar penilaian yang konsisten dapat mengakibatkan kesalahan klasifikasi, yang pada akhirnya memengaruhi pengambilan keputusan lebih lanjut, seperti proses distribusi, pengolahan, atau konsumsi. Masalah ini menjadi semakin relevan ketika proses klasifikasi dilakukan dalam skala besar atau dengan keterbatasan waktu.

Untuk menjawab tantangan tersebut, *Convolutional Neural Network* (CNN) dapat dimanfaatkan untuk mengotomatisasi proses klasifikasi berdasarkan citra buah pisang. *Convolutional Neural Network* (CNN) adalah salah satu metode machine learning dari pengembangan *Multi-Layer Perceptron* (MLP) yang didesain untuk mengolah data dua dimensi [2]. Dengan melatih model *Convolutional Neural Network* (CNN) menggunakan dataset gambar pisang dengan variasi tingkat kematangan tertentu, sistem dapat mempelajari karakteristik visual dari masing-masing kelas [3]. Implementasi model ini ke dalam aplikasi berbasis Android yang dapat menjadi nilai tambah, Android adalah sistem operasi berbasis Linux yang dirancang untuk perangkat bergerak layar sentuh seperti telepon pintar. Sistem operasi Android bersifat *open-source* yang membebaskan penggunaannya untuk menciptakan aplikasi sendiri secara bebas dan mandiri [4], mengingat fleksibilitas dan aksesibilitas tinggi dari perangkat *mobile*. Pengguna cukup menggunakan kamera *smartphone* untuk mengambil gambar dan mengupload-nya ke dalam aplikasi lalu mendapatkan hasil klasifikasi secara langsung.

Penelitian ini bertujuan untuk merancang dan mengimplementasikan sistem klasifikasi citra berbasis *Convolutional Neural Network* (CNN) dalam *platform* Android, guna memberikan solusi otomatisasi yang efisien dan akurat untuk identifikasi tingkat kematangan buah pisang. Diharapkan sistem ini dapat menjadi prototipe awal dalam pengembangan teknologi klasifikasi visual di perangkat mobile.

2. METODE PENELITIAN

Penelitian ini menggunakan pendekatan rekayasa perangkat lunak yang berfokus pada tahapan pengembangan sistem klasifikasi citra berbasis CNN serta implementasinya ke dalam aplikasi Android. Metode ini mencakup lima tahap utama, yaitu:

a. Pengumpulan *Dataset*

Tahap pertama adalah pengumpulan data. *Dataset* yang digunakan dalam penelitian ini diperoleh dari berbagai sumber daring seperti Kaggle dan Google *Dataset Search*. *Dataset* terdiri dari 4112 gambar buah pisang yang telah diklasifikasikan secara manual ke dalam empat kategori berdasarkan tingkat kematangan: mentah, setengah matang, matang, dan terlalu matang. Rincian distribusi datanya adalah sebagai berikut: 857 gambar untuk kategori mentah, 489 gambar untuk setengah matang, 1916 gambar untuk matang, dan 850 gambar untuk terlalu matang.

b. Pra-pemrosesan Data

Tahap kedua adalah pra-pemrosesan data. Seluruh gambar diubah ukurannya menjadi 150x150 piksel agar sesuai dengan arsitektur input model CNN. Data kemudian dinormalisasi ke dalam skala 0–1 agar proses pelatihan model lebih optimal. Label kategori diberikan pada masing-masing gambar, dan pembagian data dilakukan dengan rasio 80% untuk pelatihan, 10% validasi, dan 10% untuk pengujian.

c. Pembangunan dan Pelatihan model CNN

Tahap ketiga adalah pembangunan dan pelatihan model CNN. Model CNN dirancang menggunakan *framework* TensorFlow dan Keras. TensorFlow adalah *library* perangkat lunak yang dikembangkan oleh tim Google untuk mendukung pembelajaran mesin dan *deep learning*. TensorFlow menggabungkan aljabar komputasi dengan teknik optimasi kompilasi, memfasilitasi perhitungan banyak ekspresi matematika, terutama dalam mendefinisikan, mengoptimalkan, dan menghitung operasi matematika yang melibatkan array multidimensi (*tensors*). Salah satu keunggulan utama TensorFlow adalah kemampuannya menjalankan kode yang sama di CPU maupun GPU, serta skalabilitas komputasi yang tinggi untuk menangani kumpulan data besar, sehingga sangat efektif untuk aplikasi *machine learning* dan *deep learning* [5]. Sedangkan Keras adalah *framework deep learning* tingkat tinggi yang ditulis dengan bahasa Python dan berjalan di atas backend seperti TensorFlow, CNTK, atau Theano. Keras dirancang untuk mempermudah pengembangan model neural network, khususnya CNN yang sering digunakan dalam pengolahan citra. Framework ini menyediakan API yang intuitif sehingga memudahkan proses pelatihan dan pengujian model *deep learning* [6] dengan hasil yang akurat [7]. Arsitektur model terdiri dari beberapa *convolutional layer* untuk ekstraksi fitur. *Convolution Layer* adalah bagian utama dari CNN yang digunakan dalam pemrosesan gambar dan analisis video. Tujuan utama dari lapisan konvolusi adalah untuk mengekstraksi fitur dari input gambar, seperti tepi, tekstur, dan pola. Diikuti dengan *max pooling layer* untuk mereduksi dimensi, dan *dense layer (fully connected)* untuk klasifikasi akhir. Fungsi aktivasi *ReLU* digunakan untuk *convolutional layer* dan *softmax* untuk *output layer*. Model dilatih selama 10 *epoch* dengan ukuran *batch* 32, *optimizer Adam*, dan *loss function categorical crossentropy*.

d. Integrasi model ke dalam aplikasi Android

Tahap keempat adalah integrasi model ke dalam aplikasi Android. Model CNN yang telah dilatih disimpan ke dalam format Keras dan dijalankan pada server menggunakan *framework* Flask untuk membangun *REST API* agar dapat digunakan pada perangkat Android secara efisien. Aplikasi Android dikembangkan menggunakan *framework* Flutter karena sifatnya yang lintas *platform* dan performa yang ringan. Flutter adalah sebuah software development kit (SDK) *open-source* yang dikembangkan oleh Google, digunakan untuk membangun aplikasi yang dapat dijalankan secara *native* di berbagai *platform* seperti Android, iOS, web, dan

desktop hanya dari satu basis kode. Flutter menggunakan bahasa pemrograman Dart dan menyediakan berbagai widget siap pakai untuk menciptakan antarmuka pengguna yang menarik dan interaktif. Salah satu keunggulan utamanya adalah fitur "*hot-reload*" yang memungkinkan pengembang melihat perubahan kode secara langsung tanpa harus memulai ulang aplikasi, sehingga mempercepat proses pengembangan dan eksperimen [8]. Aplikasi memiliki antarmuka pengguna yang dirancang dengan prinsip kesederhanaan dan kemudahan penggunaan. Pengguna dapat mengunggah gambar pisang dari galeri atau kamera, dan sistem akan menampilkan hasil klasifikasi berdasarkan tingkat kematangan. Firebase digunakan sebagai database untuk menyimpan data riwayat klasifikasi. Firebase adalah sebuah platform database yang mampu memberikan informasi secara *real-time* kepada penggunanya. Firebase menjadi alternatif database yang handal untuk memberikan informasi secara cepat, dengan arsitektur yang berbeda dari *database* pada umumnya, baik dari sisi penamaan data maupun atributnya. Teknologi database ini dapat diimplementasikan ke dalam aplikasi yang membutuhkan penyampaian informasi secara cepat dan *real-time* [9].

e. Evaluasi performa sistem yang dibangun

Tahap terakhir adalah evaluasi performa sistem. Evaluasi dilakukan terhadap akurasi model, kecepatan klasifikasi pada perangkat *mobile*, dan pengalaman pengguna. Pengujian akurasi dilakukan menggunakan data uji sebanyak 412 gambar dari keseluruhan data pada *dataset*.

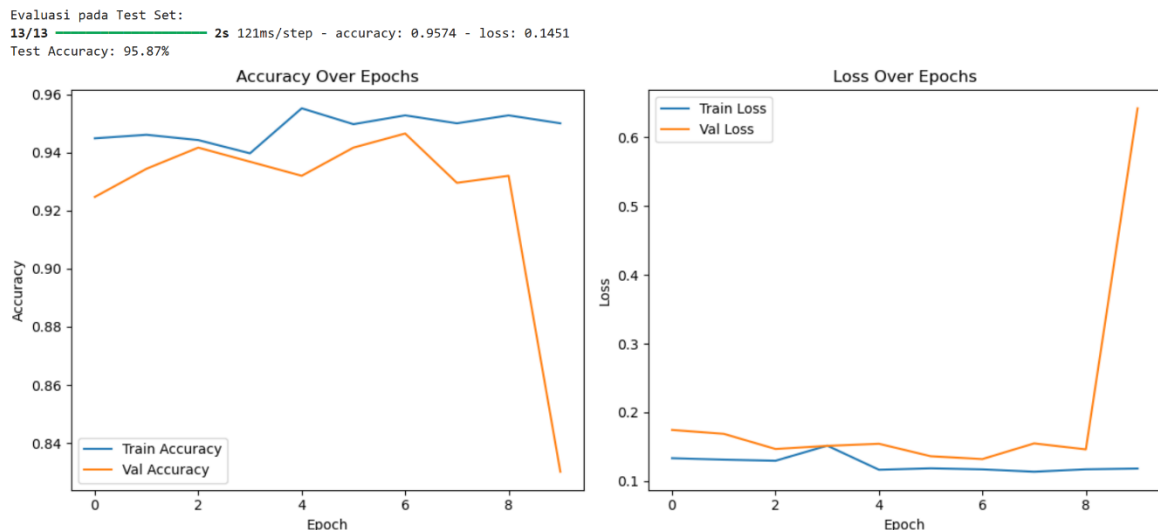
3. HASIL DAN PEMBAHASAN

Pada penelitian ini model CNN yang dikembangkan melalui proses pelatihan pada *dataset*. Hasil pelatihan *dataset* yang diperoleh menunjukkan peningkatan performa model seiring dengan bertambahnya *epoch*.

```
Epoch 1/10
103/103 ————— 40s 382ms/step - accuracy: 0.9485 - loss: 0.1287 - val_accuracy: 0.9248 - val_loss: 0.1745
Epoch 2/10
103/103 ————— 39s 374ms/step - accuracy: 0.9517 - loss: 0.1191 - val_accuracy: 0.9345 - val_loss: 0.1689
Epoch 3/10
103/103 ————— 40s 390ms/step - accuracy: 0.9331 - loss: 0.1421 - val_accuracy: 0.9417 - val_loss: 0.1468
Epoch 4/10
103/103 ————— 38s 372ms/step - accuracy: 0.9520 - loss: 0.1257 - val_accuracy: 0.9369 - val_loss: 0.1514
Epoch 5/10
103/103 ————— 40s 385ms/step - accuracy: 0.9528 - loss: 0.1207 - val_accuracy: 0.9320 - val_loss: 0.1544
Epoch 6/10
103/103 ————— 39s 380ms/step - accuracy: 0.9529 - loss: 0.1144 - val_accuracy: 0.9417 - val_loss: 0.1363
Epoch 7/10
103/103 ————— 38s 371ms/step - accuracy: 0.9496 - loss: 0.1294 - val_accuracy: 0.9466 - val_loss: 0.1322
Epoch 8/10
103/103 ————— 40s 384ms/step - accuracy: 0.9569 - loss: 0.1067 - val_accuracy: 0.9296 - val_loss: 0.1550
Epoch 9/10
103/103 ————— 40s 384ms/step - accuracy: 0.9537 - loss: 0.1073 - val_accuracy: 0.9320 - val_loss: 0.1463
Epoch 10/10
103/103 ————— 39s 382ms/step - accuracy: 0.9543 - loss: 0.1158 - val_accuracy: 0.8301 - val_loss: 0.6420
```

Gambar 1. Pelatihan *Dataset*

Pada *epoch* pertama, model berhasil mencapai akurasi pelatihan sebesar 94,85% dan akurasi validasi sebesar 92,48%, Nilai ini terus meningkat dan menunjukkan kestabilan pada *epoch-epoch* berikutnya. Selanjutnya melakukan test akurasi pada hasil pelatihan, adapun hasil visualisasi proses pelatihan dapat dilihat dalam gambar berikut:



Gambar 2. Plot akurasi dan *loss*

Selanjutnya setelah melakukan *test* akurasi pada hasil pelatihan. Hasil yang didapatkan akan dilakukan metode evaluasi. Metode evaluasi bertujuan untuk mengetahui kinerja dari Algoritma CNN. Metode evaluasi yang digunakan pada penelitian ini adalah *Confusion Matrix*. *Confusion Matrix* digunakan sebagai metode evaluasi yang digunakan untuk menghitung nilai kinerja algoritma CNN setelah didapatkan hasil pada perhitungan atau model dari CNN. Berikut adalah tabel dari *Confusion Matrix* yang ditampilkan pada tabel di bawah ini.

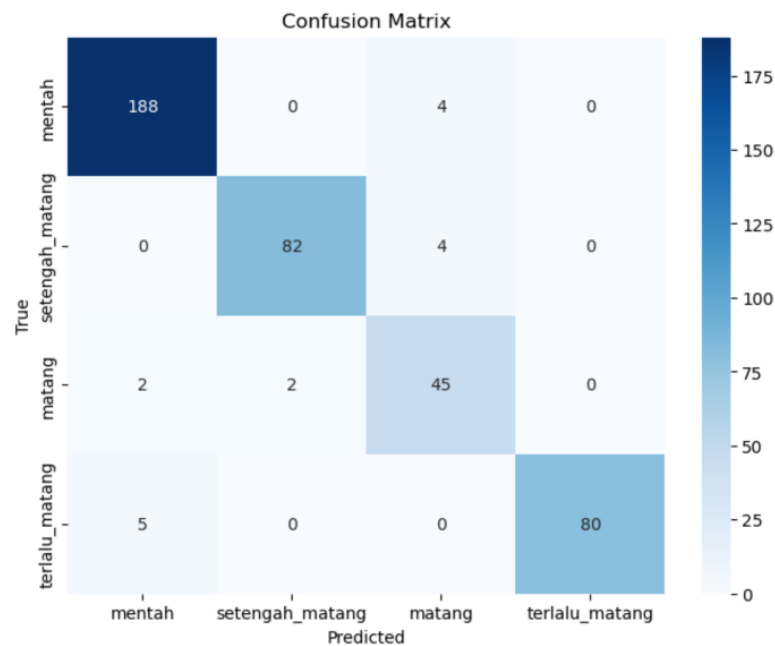
Tabel 1. *Confusion Matrix*

Aktual \ Prediksi	Prediksi	
	Negative (-)	Positif (+)
Negative (-)	True Negative (TN)	False Positive (FP)
Positif (+)	False Negative (FN)	True Positive (TP)

Keterangan:

- True Positive* (TP), adalah kondisi dimana model mengklasifikasikan data sebagai *True* dan jawaban aktualnya adalah *True*.
- True Negative* (TN), adalah kondisi dimana model mengklasifikasikan data sebagai *False* dan jawaban aktualnya adalah *False*.
- False Positive* (FP), adalah kondisi dimana model mengklasifikasikan data sebagai *True* dan jawaban aktualnya adalah *False*.
- False Negative* (FN), adalah kondisi dimana model mengklasifikasikan data sebagai *False* dan jawaban aktualnya adalah *True*.

Berikut adalah gambar dari *Confusion Matrix* setelah pelatihan *dataset*:

Gambar 3. *Confusion Matrix*

Tahap selanjutnya melakukan pengukuran kinerja. Pengukuran kinerja merupakan langkah yang digunakan untuk menghitung kinerja algoritma. Dimana pengukuran kinerja ini dilakukan setelah mendapatkan hasil dari *confusion matrix* yang berupa nilai TP, TN, FP, dan FN. Pengukuran kinerja ini berisi *precision*, *recall*, *f1-score*, *accuracy*, dan *overall accuracy*. Berikut rumus yang digunakan pada pengukuran kinerja:

Precision

$$Precision = \frac{TP}{TP + FP}$$

Precision untuk kelas mentah

$$Precision = \frac{188}{188 + 7} = 0.9641$$

Precision untuk kelas setengah matang

$$Precision = \frac{82}{82 + 2} = 0.9762$$

Precision untuk kelas matang

$$Precision = \frac{45}{45 + 8} = 0.8491$$

Precision untuk kelas terlalu matang

$$Precision = \frac{80}{80 + 0} = 1$$

Recall

$$Recall = \frac{TP}{TP + FN}$$

Recall untuk kelas mentah

$$Recall = \frac{188}{188 + 4} = 0.9792$$

Recall untuk kelas setengah matang

$$Recall = \frac{82}{82 + 4} = 0.9535$$

Recall untuk kelas matang

$$Recall = \frac{45}{45 + 4} = 0.9184$$

Recall untuk kelas terlalu matang

$$Recall = \frac{80}{80 + 5} = 0.9412$$

F1-score

$$F1-score = 2 \times \frac{Recall \times Precision}{Recall + Precision}$$

F1-score untuk kelas mentah

$$F1-score = 2 \times \frac{0.9792 \times 0.9641}{0.9792 + 0.9641} = 0.9716$$

F1-score untuk kelas setengah matang

$$F1-score = 2 \times \frac{0.9535 \times 0.9762}{0.9535 + 0.9762} = 0.9647$$

F1-score untuk kelas matang

$$F1-score = 2 \times \frac{0.9184 \times 0.8491}{0.9184 + 0.8491} = 0.8824$$

F1-score untuk kelas terlalu matang

$$F1-score = 2 \times \frac{0.9412 \times 1}{0.9412 + 1} = 0.9697$$

Accuracy

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

Accuracy untuk kelas mentah

$$Accuracy = \frac{188 + 213}{188 + 7 + 4 + 213} = 0.973301$$

Accuracy untuk kelas setengah matang

$$Accuracy = \frac{82 + 324}{82 + 2 + 4 + 324} = 0.9854$$

Accuracy untuk kelas matang

$$Accuracy = \frac{45 + 355}{45 + 8 + 4 + 355} = 0.9709$$

Accuracy untuk kelas terlalu matang

$$Accuracy = \frac{80 + 327}{80 + 0 + 5 + 327} = 0.9879$$

Overall Accuracy

$$Overall Accuracy = \frac{Jumlah\ Prediksi\ Benar (\sum TP)}{Jumlah\ Total\ Data\ (Per\ Baris)}$$

$$Overall Accuracy = \frac{188 + 82 + 45 + 80}{192 + 86 + 49 + 85} = \frac{395}{412} = 0,9587$$

Hasil *Overall Accuracy* adalah 0,9587 atau 95,87%.

Berikut tabel rincian matriks evaluasi untuk setiap kelas:

Tabel 2. Rincian Matriks

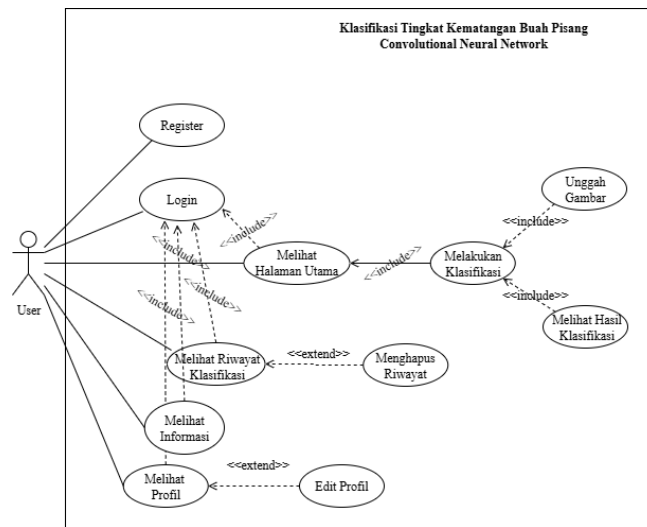
Kelas	Precision	Recall	F1-Score	Accuracy
Mentah	96%	97%	97%	97%
Setengah Matang	97%	95%	96%	99%
Matang	84%	91%	88%	97%
Terlalu Matang	100%	94%	96%	99%

Setelah mendapatkan hasil akhir dari akurasi model yang dilatih, selanjutnya model akan disimpan untuk diintegrasikan model ke dalam aplikasi android. Dalam pembuatan aplikasi android salah satu tahapannya yaitu pembuatan pemodelan perangkat lunak, Pemodelan perangkat lunak adalah proses membuat representasi abstrak

dari sistem perangkat lunak untuk memungkinkan pemahaman, dokumentasi, dan komunikasi tentang struktur dan perilakunya. Proses ini melibatkan penggunaan berbagai diagram dan teknik untuk menangkap kebutuhan sistem, desain, dan implementasi. Untuk membuat perancangan, sistem ini menggunakan UML sebagai bahasa pemodelan yang terdiri dari *Class Diagram*, *Use Case Diagram*, *Activity Diagram*, *Sequence Diagram*. *Unified Modeling Language (UML)* adalah sebuah bahasa pemodelan grafis yang digunakan sebagai standar untuk memodelkan sistem dengan metodologi pemodelan berorientasi objek. UML distandarisasi oleh *Object Management Group (OMG)* dan berfungsi untuk memvisualisasikan, menspesifikasikan, membangun, dan mendokumentasikan sistem perangkat lunak. UML menyediakan berbagai jenis diagram seperti *Use Case*, *Class*, *Object*, *Activity*, *Sequence*, *Collaboration*, *Statechart*, *Component*, dan *Deployment Diagram* untuk mendukung proses pengembangan sistem [10].

Use Case Diagram

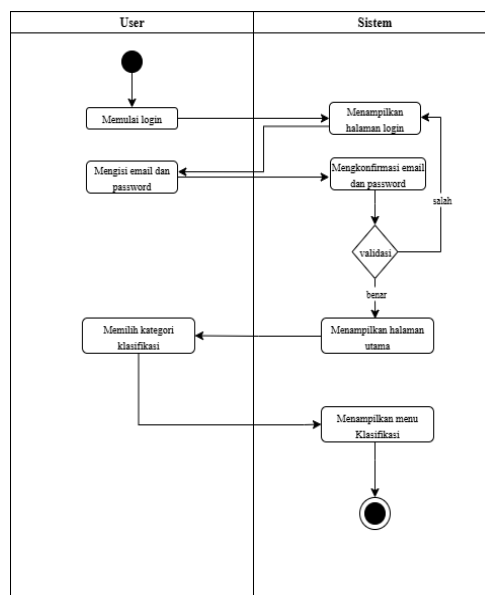
Use case diagram adalah salah satu jenis diagram yang digunakan dalam pemodelan sistem berbasis objek, khususnya dalam *Unified Modeling Language (UML)*. Diagram ini menggambarkan interaksi antara *actor* (pengguna atau sistem eksternal) dengan sistem yang akan dikembangkan.



Gambar 4. Use Case Diagram

Activity Diagram

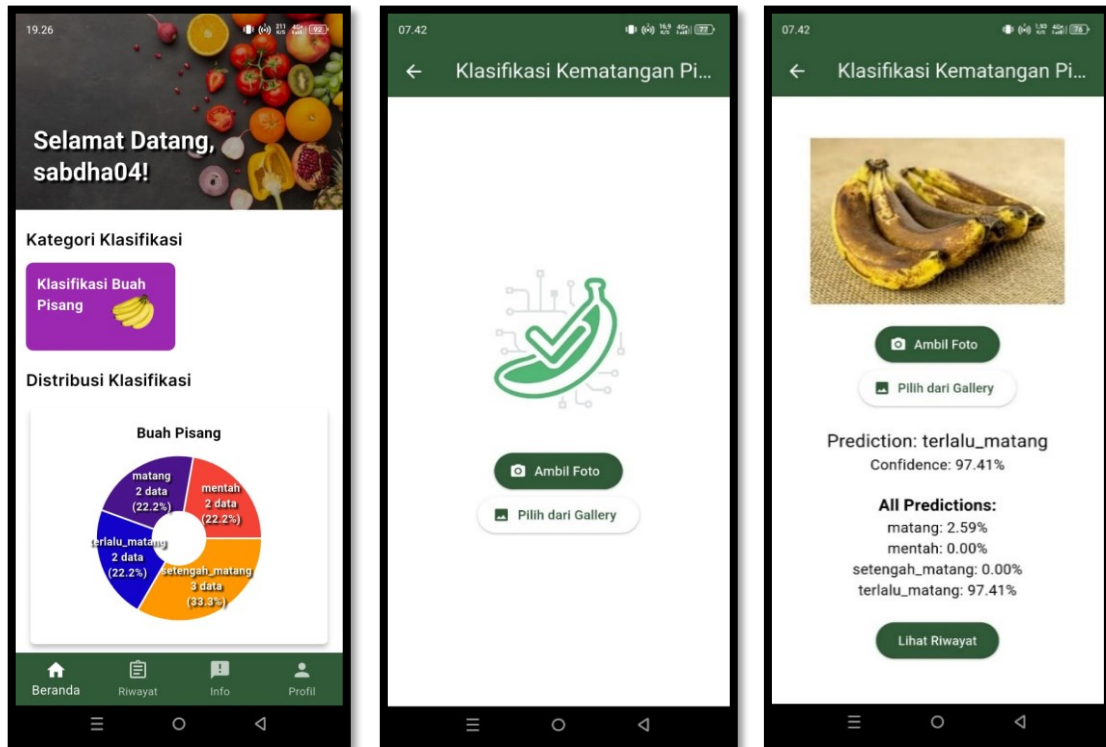
Activity diagram menunjukkan alur kerja sistem. Kita dapat membuat *activity diagram* berdasarkan *use case diagram* yang telah dibuat, yang menunjukkan alur aktivitas dari memilih menu hingga menampilkan hasil klasifikasi gambar. Dan berikut adalah salah satu contohnya yaitu *Activity Diagram Klasifikasi* yang digunakan.



Gambar 5. Activity Diagram Klasifikasi

Tampilan Layar

Hasil penelitian yang telah dilakukan, menghasilkan *output* berupa aplikasi *mobile* untuk klasifikasi tingkat kematangan pisang berbasis CNN. Tampilan pertama menunjukkan halaman utama dengan menu klasifikasi dan distribusi data, tampilan kedua memperlihatkan opsi input citra melalui kamera atau galeri, sedangkan tampilan ketiga menampilkan hasil klasifikasi lengkap dengan tingkat kepercayaan prediksi. Secara ilmiah, gambar ini merepresentasikan penerapan model CNN dalam aplikasi Android yang mampu memproses citra secara langsung dan memberikan hasil klasifikasi tingkat kematangan pisang secara praktis serta mudah dipahami pengguna.



Gambar 6. Tampilan aplikasi

4. SIMPULAN

Berdasarkan penelitian yang telah dilakukan, pengembangan model klasifikasi tingkat kematangan buah pisang berbasis *Convolutional Neural Network* (CNN) berhasil diimplementasikan secara efektif ke dalam aplikasi Android. Model dilatih menggunakan 4112 citra dengan rasio data pelatihan, validasi, dan pengujian sebesar 80:10:10. Dengan menggunakan *batch size* = 32, *epochs* = 10, *learning rate* = 0,0001 menghasilkan *precision* tingkat matang sebesar 84%, *precision* tingkat mentah sebesar 96%, *precision* tingkat setengah matang sebesar 97%, *precision* tingkat terlalu matang sebesar 100%, *recall* tingkat matang sebesar 91%, *recall* tingkat mentah sebesar 97%, *recall* tingkat setengah matang sebesar 95%, *recall* tingkat terlalu matang sebesar 94%, *f1-score* tingkat matang sebesar 88%, *f1-score* tingkat mentah sebesar 97%, *f1-score* tingkat setengah matang sebesar 96%, *f1-score* tingkat terlalu matang sebesar 96%, *accuracy* keseluruhan sebesar 95,87%. Hasil evaluasi menunjukkan bahwa model mampu mengklasifikasikan tingkat kematangan buah pisang ke dalam empat kelas, yaitu mentah, setengah matang, matang, dan terlalu matang. Hasil ini membuktikan bahwa CNN efektif dalam mengidentifikasi berbagai tingkat kematangan buah pisang secara otomatis. Selain itu, integrasi model ke dalam aplikasi Android berbasis Flutter juga menunjukkan bahwa sistem dapat berjalan efisien dan ramah pengguna.

DAFTAR PUSTAKA

- [1] Badan Pusat Statistik, "Produksi Tanaman Buah-Buahan dan Sayuran Tahunan Menurut Provinsi dan Jenis Tanaman," 2023, Accessed: Jul. 12, 2025. [Online]. Available: <https://www.bps.go.id/id>.
- [2] R. Mawan, "Klasifikasi motif batik menggunakan Convolutional Neural Network," *JNANALOKA*, vol. 1, no. 1, pp. 45-50, April 2020, doi: 10.36802/jnanaloka.2020.v1-no1-45-50.
- [3] A. Roihan, P. Abas Sunarya, and A. S. Rafika, "Pemanfaatan Machine Learning dalam Berbagai Bidang: Review paper," *IJCIT (Indonesian Journal on Computer and Information Technology)*, vol. 5, no. 1, pp. 75–82, Mar. 2020, doi: 10.31294/ijcit.v5i1.7951.
- [4] A. G. I. Hutabarat and A. C. Padmasari, "Rancang Bangun Game Tradisional 'Tambah Satu' berbasis Platform Android," *Edsence: Jurnal Pendidikan Multimedia*, vol. 2, no. 1, pp.29-44, June 2020, doi: 10.17509/edsence.v2i1.25028.

- [5] M. Ihsan, R. K. Niswatin, and D. Swanjaya, "Deteksi Ekspresi Wajah Menggunakan Tensorflow ," *JOUTICA*, vol. 6, no. 1, pp. 428-433, 2021. doi: 10.30736/jti.v6i1.554.
- [6] A. Raup, W. Ridwan, Y. Khoeriyah, and Q. Y. Zaqiah, "Deep Learning dan Penerapannya dalam Pembelajaran," *JIP (Jurnal Ilmiah Ilmu Pendidikan)*, vol. 5, no. 9, pp. 3742-3754, Sep. 2022, [Online]. doi: 10.54371/jiip.v5i9.805.
- [7] A. Santoso and G. Ariyanto, "Implementasi Deep Learning Berbasis Keras untuk Pengenalan Wajah," *Jurnal Teknik Elektro*, vol. 18, no. 01, pp. 3258-3267, Sep. 2024, doi: 10.54371/jiip.v5i9.805.
- [8] Febri Listiana Harahap, Richi Andrianto, Pebriana Panggabean, Putri Hanafia, Panusunan Panusunan, and Mulia Garang, "Pengembangan Aplikasi Kalkulator Multifungsi Menggunakan Flutter Framework," *Jurnal Penelitian Teknologi Informasi dan Sains*, vol. 1, no. 3, pp. 76–84, Sep. 2023, doi: 10.54066/jptis.v1i3.815.
- [9] Sugiyatno, "Pengiriman Informasi Real Time Menggunakan Teknologi Database Firebase pada Aplikasi Mobile Android," *Jurnal Informatika Komputer, Bisnis dan Manajemen Bisnis dan Manajemen*, vol. 21, no. 2, pp. 46-55, Nov. 2023, doi: 10.61805/fahma.v21i2.17.
- [10] M. Hamas and Z. Imaduddin, "Pengembangan Sistem Jual Beli Bahan Pokok Petani Berbasis Aplikasi Mobile," *Jurnal Informatika Terpadu*, vol. 5, no. 2, pp. 49–55, Sep. 2019, doi: 10.54914/jit.v5i2.198.