

BAB I

PENDAHULUAN

A. Latar Belakang

Kemajuan teknologi saat ini berkembang pesat berkat banyaknya penemuan-penemuan baru di berbagai sektor yang telah membawa banyak perubahan yang secara signifikan meningkatkan kemampuan masyarakat dalam menjalani kehidupan sehari-hari. Kemajuan teknis saat ini sangatlah banyak, dan kecerdasan buatan (AI) hanyalah salah satunya. Menggunakan AI sebagai kamera pintar untuk mengenali berbagai jenis objek, termasuk wajah orang, hewan, mobil, dan benda lainnya, merupakan salah satu cara untuk memanfaatkan teknologi tersebut. Saat ini, kemajuan teknologi pengolahan gambar membawa dampak yang signifikan.

Aplikasi pendeteksi kematangan buah adalah suatu teknologi yang dikembangkan untuk membantu dalam mengidentifikasi tingkat kematangan buah-buahan. Penelitian ini sangat relevan dalam industri pertanian dan distribusi makanan, karena kematangan buah dapat memengaruhi rasa, tekstur, dan daya tahan buah tersebut.

Aplikasi pendeteksi kematangan buah sebenarnya sudah tercipta sebelum adanya proposal penelitian ini, namun diantaranya masih berfokus pada satu objek buah. Sedangkan dalam proposal penelitian ini, penulis akan mengambil sampel dari beberapa jenis buah dengan menggunakan algoritma *CNN*.

Penelitian ini didorong oleh kemajuan pesat dalam bidang kecerdasan buatan, terutama dalam penggunaan algoritma *CNN (Convolutional neural network)* yang telah terbukti sangat efektif dalam pengenalan pola dalam data gambar. Dengan mengaplikasikan algoritma ini dalam penelitian, kita dapat menciptakan solusi yang akurat dan efisien untuk pendeteksian kematangan buah yang dapat digunakan secara luas dalam industri pertanian, distribusi makanan, serta oleh konsumen akhir.

Selama periode 2000-2019, Indonesia menghasilkan sampah makanan di kisaran 23-48 juta ton per tahun. Angka itu merupakan gabungan dari *food loss* (pangan yang terbuang pada tahap produksi, pascapanen/penyimpanan, dan pemrosesan/pengemasan), serta *food waste* (pangan yang terbuang pada tahap distribusi/pemasaran dan sisa konsumsi). Banyaknya sampah makanan Indonesia itu tercatat dalam Laporan Kajian *Food Loss and Waste* di Indonesia (2021), hasil riset kolaborasi Kementerian PPN/Bappenas dengan *Waste4Change* dan *World Resource Institute*. Adapun jika dirinci per kategori, secara rata-rata selama periode 2000-2019 mayoritas atau 44% sampah makanan Indonesia berupa padi-padian. "Timbulan FLW (*food loss and waste*) terbesar dikontribusikan oleh sektor tanaman pangan, tepatnya dari padi-padian, yaitu sebesar 12-21 juta ton per tahun," kata tim Bappenas dalam laporannya. Setelah padi, sampah makanan terbanyak Indonesia adalah buah-buahan dan sayur-sayuran.

Pilihan yang buruk saat membeli buah dan sayur berkontribusi terhadap sampah makanan. Untuk mengatasi hal ini, kami bertujuan untuk mengembangkan aplikasi yang menggunakan kamera ponsel pintar untuk mengetahui kematangan produk.

konsumen seringkali kesulitan dalam menentukan tingkat kematangan buah yang tepat saat melakukan pembelian di pasar maupun di supermarket. Hal ini seringkali menyebabkan mereka salah membeli buah-buahan yang tidak sesuai dengan keinginannya sehingga mengakibatkan makanan terbuang sia-sia.

Penggunaan teknologi pengolahan citra untuk mendeteksi kematangan buah juga berpotensi digunakan untuk membantu penyandang disabilitas (penderita buta warna). Aplikasi ini memberikan manfaat kepada pengguna dengan memudahkan pengguna dalam menentukan kematangan buah yang diinginkan. Sistem dilakukan secara real time dan menggunakan metode kotak pembatas, pengguna dapat memasukkan lebih dari 1 buah untuk dideteksi. Berikut adalah beberapa cara di mana aplikasi semacam itu dapat bermanfaat:

1. **Kemandirian:** Aplikasi ini dapat membantu konsumen mengambil keputusan yang tepat saat berbelanja dan memastikan mereka memilih buah pada tingkat kematangan yang diinginkan untuk dikonsumsi segera atau untuk umur simpan yang lebih lama. Penyandang disabilitas (buta warna) mungkin mengalami kesulitan memeriksa buah berdasarkan warna untuk menentukan tingkat kematangannya. Aplikasi yang dapat mendeteksi kematangan buah dapat membantu mereka menjadi lebih mandiri dalam menjalani kehidupan sehari-hari.
2. **Menghemat waktu:** Aplikasi yang dapat mendeteksi kematangan buah dengan cepat dan akurat dapat menghemat waktu bagi penyandang disabilitas yang mungkin memiliki mobilitas atau energi terbatas.

3. Hemat biaya: Penyandang disabilitas mungkin memiliki sumber daya keuangan yang terbatas, dan aplikasi yang dapat membantu mereka memilih buah yang matang dapat menghemat biaya dalam jangka panjang. Secara keseluruhan, aplikasi yang dapat mendeteksi kematangan buah berpotensi menjadi alat yang berguna bagi penyandang disabilitas, membantu mereka menjadi lebih mandiri, menghemat waktu, meningkatkan gizi, dan mengurangi sampah makanan.

Dari penjelasan diatas, guna pengimplementasian ilmu dan teknologi pengolahan citra yang berkembang di era digital saat ini, maka penulis ingin membuat proposal penelitian dengan judul “**Aplikasi Pendeteksi Kematangan Pada Buah Menggunakan Algoritma CNN**” yang sekaligus akan menjadi judul skripsi penulis untuk memenuhi syarat kelulusan.

B. Rumusan Masalah

Berdasarkan latar belakang diatas, Adapun rumusan masalah yang akan dibahas penulis yaitu:

1. Bagaimana cara merancang dan membuat aplikasi pendeteksi kematangan pada buah menggunakan algoritma *CNN (Convolutional neural network)*?

C. Tujuan Penelitian

Tujuan dari penelitian ini adalah untuk mendeteksi kematangan pada buah dengan menggunakan algoritma *CNN (Convolutional neural network)*.

D. Batasan Masalah

Dalam penelitian ini, terdapat beberapa batasan masalah yang diberikan oleh penulis sebagai berikut:

1. Penelitian ini menggunakan algoritma *CNN (Convolutional neural network)*.
2. Pendeteksian buah hanya dapat dilakukan dengan pencahayaan yang cukup.
3. Aplikasi yang dihasilkan hanya akan berfokus pada deteksi warna pada objek.
4. Jenis buah yang di deteksi yaitu pepaya, tomat, buah naga dan semangka.

E. Manfaat Penelitian

Manfaat yang dari penelitian ini menciptakan sebuah aplikasi pendeteksi kematangan pada buah yang diharapkan dapat membantu petani buah dan konsumen dalam mempermudah perbandingan kematangan buah. Dan juga memperoleh cara untuk mengklasifikasi kematangan buah selain cara manual, yaitu dengan bantuan teknologi seperti yang dimaksud dalam penelitian ini.

Aplikasi ini dapat membantu petani buah untuk mengoptimalkan waktu panen, mengurangi pemborosan dengan mengidentifikasi buah matang untuk dipetik, dan memantau proses kematangan untuk memilih buah yang baik untuk diperjual belikan.

Aplikasi ini juga bermanfaat bagi Supermarket dan Pedagang di pasar. Pengecer dapat menggunakan aplikasi ini untuk menilai kematangan buah-buahan yang mereka simpan di rak buah mereka, sehingga memungkinkan mereka menawarkan produk dengan kualitas lebih baik kepada pelanggan. Hal ini dapat membantu pedagang dalam mengidentifikasi buah-buahan yang siap dijual atau buah-buahan yang memerlukan waktu lebih lama untuk matang.

Selain itu, penelitian ini juga dapat bermanfaat bagi penderita disabilitas (buta warna) agar dapat membedakan buah yang matang dan belum matang. Penelitian ini juga diharapkan dapat menjadi referensi kepada akademisi mengenai aplikasi poendeteksi kematangan pada buah menggunakan algoritma *CNN (Convolutional neural network)*.

BAB II

TINJAUAN PUSTAKA

A. Kajian Hasil Penelitian Terdahulu

Peneliti dengan kasus yang sama namun objek dan metode yang berbeda yaitu **Aplikasi Pendeteksi Kematangan Buah Menggunakan Algoritma CNN** (*Convolutional Neural Network*) pada penelitian ini digunakan tiga penelitian terdahulu yang menjadi rujukan ilmiah yaitu:

1. Penelitian yang dilakukan oleh (Huda et al., 2023) yang berjudul “*Klasifikasi Jenis Buah Pisang Menggunakan Algoritma Convolutional Neural Network*”. Dalam penelitiannya, peneliti bertujuan untuk mengklasifikasikan jenis buah pisang menggunakan algoritma CNN (*Convolutional Neural Network*). Hasil dari penelitian ini memberikan kontribusi signifikan dalam pengembangan teknologi untuk mendukung industri pertanian dan distribusi buah pisang. Model CNN yang diimplementasikan mampu memberikan hasil klasifikasi dan deteksi kematangan yang memadai. Penggunaan *K-fold cross-validation* dan *StratifiedKFold* meningkatkan generalisasi model, dan hasil evaluasi memberikan landasan yang solid untuk pengembangan lebih lanjut. Model ini dapat diterapkan dalam otomatisasi proses identifikasi jenis dan kematangan buah pisang pada skala industri.

2. Penelitian yang dilakukan oleh (Damayanti et al., 2021) yang berjudul ***“Klasifikasi Buah Mangga Badami Untuk Menentukan Tingkat Kematangan dengan Metode CNN”***. Pada penelitiannya, peneliti ingin mengimplementasikan algoritma CNN untuk mengklasifikasikan jenis buah mangga badami. Pada riset ini dilakukan pemrograman sistem yang mendapatkan pendeteksi kematangan pada warna mangga Badami dengan menerapkan metode *Convolutional Neural Networks*(CNN) pada aplikasi pengolahan citra digital, sehingga dapat dipastikan buah mangga yang telah melewati masa kematangan untuk dikonsumsi. Pengujian ini dengan memakai citra sebanyak 25 citra sebagai citra uji dan 179 citra sebagai citra latih dari 204 total citra. Dengan akurasi pengujian model sebesar 97,2% .
3. Penelitian yang dilakukan oleh (Yusman et al., 2023) yang berjudul ***“Klasifikasi Kematangan Buah Tin Menggunakan Convolutional Neural Network Berbasis Android”***. Penelitian ini dilakukan untuk mendeteksi kematangan pada buah tin. Pada penelitian ini menggunakan algoritma CNN dalam melakukan klasifikasi kematangan buah Tin berjalan dengan baik pada sistem melalui kamera maupun dari memori penyimpanan. Berdasarkan hasil pengujian, penggunaan CNN dalam klasifikasi kematangan buah Tin mendapatkan akurasi sebesar 94% dengan *presisi*, *recall* dan *f1-score* yang baik. Berdasarkan hasil yang telah diperoleh dapat disimpulkan bahwa akurasi tersebut sudah baik, sehingga aplikasi dapat melakukan klasifikasi kematangan buah Tin dengan baik.

4. Penelitian yang dilakukan oleh (Darwis Gusni., 2022) yang berjudul ***“Implementasi Metode Logika Fuzzy Pada Deteksi Tingkat Kematangan Buah Durian Menggunakan Bahasa Pemrograman Python”***. Tujuan penelitian adalah untuk menerapkan logika *fuzzy tsukamoto* pada kematangan buah durian agar dapat diketahui apakah buah durian mentah atau masak. Berdasarkan hasil penelitian tersebut, dihasilkan sebuah aplikasi menilai kematangan buah durian menggunakan metode logika *fuzzy* berbasis web. Dari hasil pengujian akurasi dilakukan dengan membandingkan hasil kematangan buah durian sebanyak 10 kali percobaan, maka didapatkan data kematangan buah durian 80% dengan MSE sebanyak 20%.

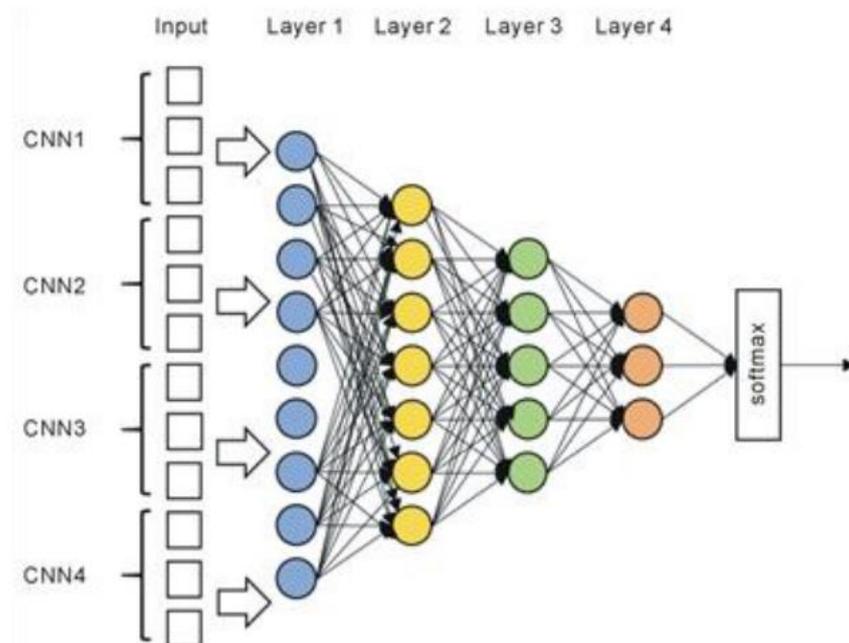
B. Kajian Teori

1. CNN (*Convolutional Neural Network*)

Hubel dan Wiesel mengembangkan *Convolutional Neural Network (CNN)* setelah mempelajari neuron dalam sistem visual kucing yang bertanggung jawab atas orientasi sensitif selektif lokal pada tahun 1960an. Mereka menemukan bahwa struktur jaringan yang unik berhasil mengurangi kompleksitas jaringan umpan balik 15 saraf. Pengenalan pola dan pemrosesan gambar memanfaatkan Jaringan Syaraf Konvolusional secara ekstensif, yang merupakan algoritma pengenalan yang efektif. Jaringan saraf konvolusional bersifat serbaguna, memiliki parameter pelatihan yang lebih sedikit, dan memiliki struktur yang mudah dipahami. Jaringan Neural Konvolusional telah mendapatkan popularitas di bidang pengenalan gambar dan analisis ucapan. Struktur jaringan bersama lebih mirip dengan jaringan saraf biologis karena

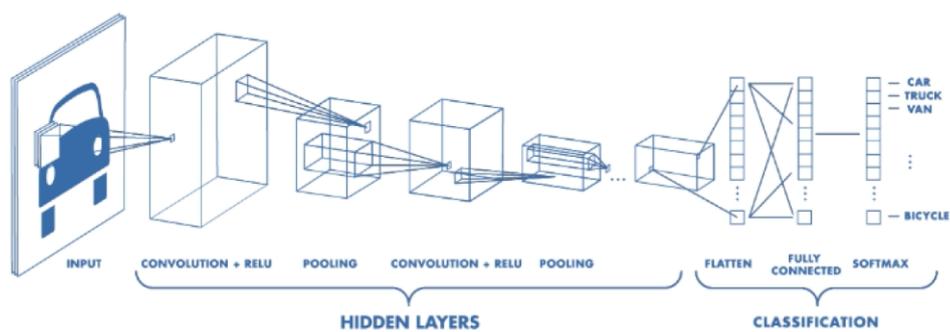
bobotnya. Hasilnya, kompleksitas dan jumlah bobot model jaringan berkurang. (T. Liu et al., 2015).

Biasanya, ada dua lapisan dalam struktur CNN. Lapisan pertama disebut lapisan ekstraksi fitur; itu mengekstrak fitur lokal dengan menghubungkan input setiap neuron ke bidang reseptif lokal lapisan sebelumnya. Hubungan posisi antara fitur lokal dan fitur lainnya juga akan dipastikan setelah fitur tersebut diekstraksi. Yang lainnya adalah lapisan peta fitur; beberapa peta fitur membentuk setiap lapisan komputasi jaringan. Setiap peta fitur adalah sebuah bidang, dan setiap neuron dalam sebuah bidang memiliki bobot yang sama. Struktur peta fitur mencapai invarian pergeseran dengan menggunakan fungsi sigmoid sebagai fungsi aktivasi jaringan konvolusi. Selain itu, jaringan memiliki lebih sedikit parameter bebas karena neuron dalam bidang pemetaan yang sama memiliki bobot yang sama. Lapisan komputasi yang menghitung rata-rata lokal dan mengekstrak lapisan kedua, yang merupakan dua lapisan ekstraksi fitur berbeda yang menurunkan resolusi, muncul setelah setiap lapisan konvolusional dalam jaringan saraf konvolusional. (T. Liu et al., 2015).



Gambar 2.1 Struktur CNN

Gambar dibawah merupakan jaringan arsitektur *Convolutional Neural Network*.



Gambar 2.2 jaringan arsitektur *Convolutional Neural Network*

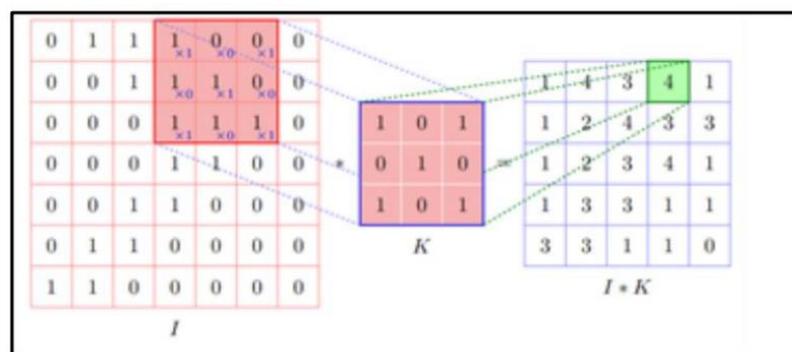
Tahap konvolusi merupakan tahap awal dari arsitektur CNN, seperti terlihat pada Gambar 2.2 di atas. Ukuran kernel tertentu digunakan untuk

langkah ini. Kuantitas fungsi yang dihasilkan menentukan berapa banyak kernel yang digunakan. Selanjutnya, nonaktifkan fungsinya. ReLU (Rectifier Linear Unit) biasanya digunakan untuk mengaktifkan fungsi tersebut. Setelah proses aktivasi fungsi selesai, pengelompokan (atau penggabungan) adalah langkah berikutnya. Hingga peta fitur yang cukup besar untuk menavigasi ke jaringan saraf yang terhubung sepenuhnya diperoleh dan kelas keluaran dari jaringan saraf yang terhubung sepenuhnya ditemukan, proses ini akan diulangi beberapa kali. (Liu et al., n.d.).

Arsitektur CNN terdiri dari beberapa lapisan (*layer*) yang berbeda, yaitu:

1. Lapisan konvolusi (*convolutional layer*)

Tugas mengekstraksi fitur dari gambar berada pada lapisan ini. Operasi konvolusi diterapkan pada gambar masukan di lapisan ini menggunakan filter konvolusi. Fitur-fitur pada gambar akan diwakili oleh peta fitur yang dibuat oleh filter konvolusi ini.



Gambar 2.3 Ilustrasi Proses Konvolusi

Dari gambar tersebut sehingga didapat perhitungan sebagai berikut:

$$FM(0,3) = 1 \times 1 + 0 \times 0 + 0 \times 1 + 1 \times 0 + 1 \times 1 + 0 \times 0 + 1 \times 1 + 1 \times 0 + 1 \times 1 = 4$$

2. Lapisan *pooling* (*pooling layer*): Tujuan dari lapisan ini adalah untuk meminimalkan ukuran peta fitur yang dihasilkan lapisan konvolusi. Hal ini dilakukan untuk meningkatkan invarian sehubungan dengan terjemahan gambar dan mengurangi jumlah parameter yang perlu dipelajari jaringan. Pada lapisan ini rumusnya adalah:

$$\text{Ukuran output} = ((\text{ukuran input} - \text{ukuran filter}) / \text{stride}) + 1$$

3. Lapisan *fully connected* (*fully connected layer*): Klasifikasi gambar masukan adalah tanggung jawab lapisan ini. Setiap neuron pada lapisan ini digabungkan dengan setiap neuron lain pada lapisan di atasnya. Untuk lapisan tersebut rumus perhitungannya adalah:

$$\text{Jumlah parameter} = (\text{jumlah neuron pada layer sebelumnya} + 1) \times \text{jumlah neuron pada layer saat ini}$$

Pengambilan citra buah yang akan ditentukan kematangannya dan mengolahnya melalui arsitektur CNN yang telah dilatih sebelumnya merupakan cara penerapan algoritma CNN untuk identifikasi kematangan pada buah.

Gambar buah dan label yang menunjukkan tingkat kematangannya dimasukkan ke dalam arsitektur CNN untuk melatihnya. Jaringan saraf akan memperoleh kemampuan untuk mengidentifikasi label kematangan yang relevan dengan mengekstraksi fitur dari foto buah. (T. Liu et al., 2015).

Foto buah baru dapat diuji dengan memasukkannya ke dalam jaringan saraf dan mengamati keluarannya setelah arsitektur CNN dilatih. Informasi mengenai kematangan buah yang diuji akan diberikan pada keluaran ini.

2. *Tensor Flow*

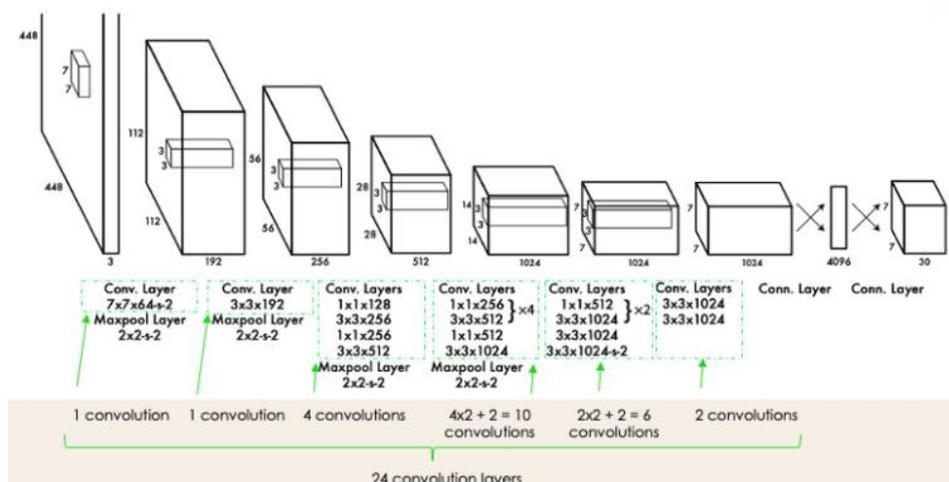
Grafik aliran data digunakan di *Tensorflow*, kumpulan perangkat lunak sumber terbuka untuk komputasi numerik. Tepi jaringan mencerminkan susunan data multidimensi (Tensor) yang ditransfer antar tepinya, sedangkan simpul pada grafik mewakili operasi matematika. Hanya dengan satu API, arsitekturnya yang dapat disesuaikan dapat digunakan untuk mendistribusikan pemrosesan melalui satu atau lebih CPU atau GPU di desktop, server, atau perangkat seluler. Untuk tujuan melakukan penelitian pembelajaran mesin dan jaringan saraf dalam, para peneliti dan insinyur di Tim Otak Google dari kelompok riset Kecerdasan Mesin Google awalnya mengembangkan *TensorFlow*. Namun, sistem ini juga cukup umum untuk diterapkan pada aneka domain lainnya. *TensorFlow* ini digunakan sebagai *backend* dari Keras, yang artinya adalah *TensorFlow* sebagai bagian belakang layar dari sebuah aplikasi atau website dengan menggunakan bahasa pemrograman Python dan dikombinasikan dengan library Keras dari *Deep Learning*.

Menggunakan teknik CNN, *TensorFlow* adalah *framework Deep Learning* yang dapat digunakan untuk membuat aplikasi untuk mendeteksi kematangan buah. *TensorFlow* memiliki sejumlah fitur dan kemampuan yang mempermudah pembuatan model CNN, model pelatihan, dan model pengujian..

3. *YoloV8*

Model terbaru dalam keluarga model YOLO (You Only Look Once), yang digunakan untuk mendeteksi objek dalam foto, disebut YOLOv8. Komponen

Convolutional Neural Network (CNN) yang digunakan untuk tugas deteksi objek real-time adalah metode YOLOv8. Dengan demikian, YOLOv8 memanfaatkan CNN untuk mendeteksi objek dengan cepat dan tepat. Metode YOLO menggabungkan pelacakan dan pengenalan objek dalam satu operasi yang efektif. Pendekatan YOLO terhadap identifikasi objek berbeda dari pendekatan standar karena menggabungkan implementasi objek dan segmentasi menjadi satu tahap. YOLO adalah pendekatan "deteksi bidikan tunggal" yang menghilangkan kebutuhan untuk terus memindai setiap area gambar dengan menjalankan proses implementasi gambar penuh sekaligus. Untuk mencapai objek klasifikasi, teknik pembelajaran mesin YOLO memisahkan kotak pembatas objek tertentu dan kelas terkait probabilitas dengan mengadopsi masalah regresi secara spasial. Secara real time, YOLO dapat mengidentifikasi objek dengan kecepatan hingga 155 frame per detik (FPS) dengan akurasi rata-rata (mAP) yang dua kali lipat dari pengklasifikasi objek konvensional. (Venkata, C. 2024)



Gambar 2.4 Algoritma Arsitektur YOLO

4. *Ultralytics*

Platform yang disebut Ultralytics ditujukan untuk pengembangan, pelatihan, dan penerapan model pembelajaran mesin (ML) menggunakan antarmuka no-code. Perpustakaan Satu pustaka perangkat lunak yang sering digunakan dalam visi komputer, terutama untuk pengenalan objek, disebut Ultralytics. Metodologi YOLO (You Only Look Once), yang terkenal dengan kecepatan dan keakuratannya dalam mengidentifikasi hal-hal dalam gambar atau video, berfungsi sebagai dasar untuk perpustakaan. Ultralytics membuat implementasi model YOLO lebih mudah dengan menawarkan antarmuka yang ramah pengguna dan fungsionalitas yang siap digunakan. (Maulana et al., 2023).

5. *Machine Learning*

Bidang kecerdasan buatan (AI) yang berfokus pada pembuatan model dan algoritma komputer yang dapat belajar dari data disebut pembelajaran mesin (*Machine Learning*). Tujuannya adalah agar komputer dapat mengidentifikasi pola dalam data, menarik kesimpulan atau perkiraan dari pola tersebut, dan secara otomatis meningkatkan kinerja tanpa memerlukan pemrograman eksplisit. (T. Liu et al., 2015).

Machine Learning adalah teknologi yang digunakan dalam kecerdasan buatan yang mengajarkan komputer untuk belajar dari data dan pengalaman sehingga mereka dapat mengantisipasi atau membuat penilaian tanpa pemrograman eksplisit. Data dapat dianalisis dengan algoritma pembelajaran

mesin untuk mengidentifikasi pola, yang kemudian dapat digunakan untuk memperkirakan atau menarik kesimpulan tentang data baru yang belum teramati. (T. Liu et al., 2015).

Proses pembelajaran mesin sering kali dapat dipecah menjadi beberapa langkah berikut:

1. Pengumpulan data: Data adalah bahan bakar yang digunakan untuk melatih model. Data ini dapat terdiri dari informasi yang dikumpulkan dari berbagai sumber mengenai permasalahan yang harus diselesaikan.
2. Pemrosesan awal data: Data harus diproses secara rutin sebelum dapat digunakan untuk melatih model. Proses ini meliputi pembersihan, standarisasi, dan pengelompokan untuk menjamin kualitas dan keseragaman data.
3. Pemilihan model: Pemilihan model adalah proses memutuskan arsitektur atau algoritma mana yang akan digunakan untuk tugas tertentu. Jenis masalah yang ada klasifikasi, regresi, atau pengelompokan, misalnya menentukan pilihan ini.
4. Pelatihan model: Model dilatih menggunakan data yang telah diproses. Saat model mendeteksi pola dalam data pelatihan, model menyesuaikan parameter internalnya untuk meningkatkan kemampuannya dalam memprediksi atau mengambil keputusan.
5. Validasi dan evaluasi: Setelah pelatihan, model dinilai dengan data yang belum pernah dilihat sebelumnya (data validasi atau pengujian) untuk memastikan model dapat berhasil digeneralisasi ke data baru.

6. Pengoptimalan dan penyetelan: Jika pengoptimalan tambahan diperlukan, model dapat dibuat dengan menyesuaikan parameter atau menggunakan teknik seperti regularisasi untuk menghentikan overfitting.

6. Pengolahan Citra Digital

Bidang studi untuk metode pengolahan citra disebut pengolahan citra digital. Gambar yang dimaksud dapat berupa stasioner (foto) atau bergerak (video). Sementara itu, pemrosesan citra yang dilakukan melalui komputer adalah apa yang dimaksudkan untuk menjadi "digital" dalam konteks ini. Pemrosesan data dua dimensi lebih umum disebut sebagai pemrosesan citra digital. Komputer digital tidak dapat memproses gambar kecuali disediakan secara kuantitatif dengan angka diskrit. Proses mengubah fungsi kontinu menjadi nilai diskrit disebut digitalisasi gambar. Tujuan pemrosesan gambar adalah untuk meningkatkan kualitas gambar sehingga komputer atau manusia dapat memahaminya dengan lebih mudah.

Digitalisasi adalah proses mengubah gambar menjadi gambar digital. Proses mengubah objek yang dapat diamati seperti gambar, teks, atau suara menjadi data elektronik yang dapat diproses, disimpan, dan digunakan untuk berbagai keperluan dikenal sebagai digitalisasi. Pemrosesan citra digital cenderung menangani kedua dimensi data dengan lebih baik dalam konteks yang lebih besar. Studi tentang pemrosesan gambar dikenal sebagai pemrosesan gambar digital. Gambar ini dapat berupa gambar bergerak (video) atau gambar diam (foto)..

Teknik pengolahan foto, terutama menggunakan komputer, untuk membuat foto berkualitas lebih tinggi dikenal sebagai pengolahan gambar digital. Citra digital dapat diartikan menggunakan fungsi dua dimensi $f(x, y)$, di mana x dan y adalah koordinat koordinat dan f adalah amplitudo di tempat (x, y) . Gambar digital terdiri dari kumpulan nilai skala abu-abu yang disebut piksel. Proses dalam proses digitalisasi gambar adalah kuantisasi dan pengambilan sampel, yang merupakan proses pengambilan nilai koordinat diskrit (x, y) dengan melewati gambar di atas grid.

Pemrosesan gambar digital dimaksudkan untuk membantu orang dalam meningkatkan kualitas gambar atau mengkarakterisasi fitur mereka. Kami memiliki kemampuan visual luar biasa yang memungkinkan kami dengan cepat dan mudah membedakan item dari kebisingan latar belakang selain dapat melihatnya. Di sisi lain, definisi tidak dapat dilakukan oleh komputer atau sensor tanpa menggunakan algoritma intelijen. Diantisipasi bahwa buku tentang pemrosesan gambar digital ini akan memberi pembaca gambaran umum tentang subjek (Pengantar (2020). (n.p.)).

7. *Android*



Gambar 2.5 Logo Android

Untuk perangkat seluler yang menjalankan Linux, Android adalah sistem operasi yang dilengkapi dengan middleware, aplikasi, dan sistem operasi itu sendiri. Android Inc. adalah perusahaan Silicon Valley kecil yang mengembangkan Android. Google kemudian mengumumkan sistem operasi menjadi "Open Source" pada tahun 2005 setelah mengadopsinya. Oleh karena itu tersedia secara bebas untuk semua orang, bahkan kode sumber yang digunakan untuk mengkompilasi sistem operasi. Karena Android memungkinkan pengembang untuk menambahkan lebih banyak fungsi, itu adalah sistem operasi dengan pengguna terbanyak di seluruh dunia.

Lingkungan pengembangan alternatif disediakan oleh *Android*. Setiap aplikasi memiliki level yang sama. Android tidak membedakan antara aplikasi pihak ketiga dan inti. Data sistem serta data smartphone dapat diakses melalui *Application Programming Interface (API)* yang disertakan. Pengguna memiliki opsi untuk menghapus aplikasi utama dan menukarnya dengan aplikasi pihak ketiga.

Adapun Kelebihan dari *android* yaitu:

1. Platform lengkap. Memberi pengembang akses ke alat praktis untuk membangun aplikasi, yang selanjutnya dapat mereka kembangkan.
2. Platform *Open Source*. Pengembang mudah mengembangkannya karena bersifat terbuka
3. Platform gratis. Sistem operasi Android gratis bagi pengembang untuk membuat, berbagi, dan berdagang tanpa mengharuskan mereka membayar royalti untuk mendapatkan lisensi.

8. Android Studio



Gambar 2.6 Logo Android Studio

IDE (Integrated Development Environment) untuk membuat aplikasi *Android* disebut *Android Studio*. *Google* merilis aplikasi pada 16 Mei 2013, dan dapat diunduh secara gratis menggunakan lisensi *Apache 2.0*. *Eclipse*, mantan alat pemrograman *Android*, telah digantikan oleh *Android Studio*. Untuk pengembang perangkat lunak, *IDE (Integrated Development Environment)* adalah program yang mencakup semua alat terintegrasi yang diperlukan untuk membuat perangkat lunak, termasuk editor kode, debugger, kompiler, dan banyak lagi.

Mirip dengan *Eclipse*, *Android Studio* dibuat menggunakan *IntelliJ IDEA* dan *plugin ADT (Android Development Tools)*. Fitur *Android Studio*:

1. Proyek berdasarkan *Gradle Build*.
2. Refactoring cepat dan perbaikan *bug*.
3. Tools baru bernama "Lint" mengklaim dapat dengan cepat memantau kecepatan, kegunaan, dan kompatibilitas aplikasi.
4. Mendukung *Proguard* dan penandatanganan aplikasi untuk keamanan
5. Memiliki GUI aplikasi *Android* lebih mudah

6. Didukung oleh Google Cloud Platform untuk setiap aplikasi yang dikembangkan.

9. Kotlin

Bahasa pemrograman Kotlin dan tipe statis adalah bahasa yang menargetkan Java virtual machine (JVM), Android, JavaScript, dan Native. Bahasa pemrograman modern Kotlin dikembangkan oleh JetBrains pada tahun 2011. Bahasa ini dirancang untuk beroperasi pada platform Java Virtual Machine (JVM) dan dapat digunakan untuk membuat aplikasi Android.

10. Python

Python merupakan bahasa pemrograman populer untuk membangun berbagai aplikasi, baik yang berbasis desktop, website, maupun mobile. Dibuat oleh Guido van Rossum di Belanda pada tahun 1990, Python saat ini meraih popularitas yang besar dalam industri dan dunia pendidikan karena kesederhanaan, kemudahan, sintaksis yang intuitif, serta memiliki beragam pustaka. (Romzi & Kurniawan 2020)

11. UML (*Unified Modeling Language*)

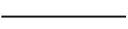
Salah satu alat untuk merancang sistem berorientasi objek adalah UML (*Unified Modeling Language*). Dari sudut pandang filosofis, gagasan pemodelan Berorientasi Objek (OO), yang analog dengan sistem kehidupan nyata yang didominasi oleh objek dan diwakili atau direkam dalam simbol yang sangat rinci, adalah apa yang memunculkan UML. prosedur independen dan seragam.

Diagram UML terutama digunakan oleh tim pengembangan proyek untuk memfasilitasi komunikasi, menyelidiki kemungkinan desain, dan memvalidasi desain arsitek atau pemrogram perangkat lunak. Komponen notasi UML berasal dari tiga notasi yang sudah ada sebelumnya: Ivar Jacobson *OOSE (Object-Oriented Software Engineering)*, Grady Booch *OOD (Object-Oriented Design)*, dan Jun Rumbaugh *OMT (Object Modeling Technique)*.

Diagram struktur, diagram perilaku, dan diagram interaksi adalah tiga subkategori utama UML.

Adapun daftar simbol UML yaitu :

Tabel 2. 1 Symbol Use Case Diagram

No	GAMBAR	NAMA	KETERANGAN
1		<i>Actor</i>	Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan <i>Use Case</i> .
2		<i>Dependency</i>	Hubungan di mana perubahan yang terjadi pada suatu elemen mandiri (<i>Independent</i>) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri.
3		<i>Generalization</i>	Hubungan di mana objek anak (<i>Descendent</i>) berbagai perilaku dan struktur data dari objek yang ada di atasnya objek induk (<i>Ancestor</i>).
4		<i>Include</i>	Menyesifikasikan bahwa Use Case sumber secara Eksplisit.
5		<i>Extend</i>	Menyesifikasikan bahwa <i>use case</i> target memperluas perilaku dari <i>use case</i> sumber pada suatu titik yang diberikan.
6		<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya.
7		<i>System</i>	Menyesifikasikan paket yang menampilkan sistem secara terbatas.

8		<i>Use Case</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu <i>actor</i>
9		<i>Collaboration</i>	Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan perilaku yang lebih besar dari jumlah dan elemennya (sinergi).
10		<i>Note</i>	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi.

Tabel 2. 2 Symbol Class Diagram

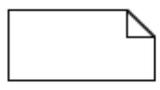
No.	GAMBAR	NAMA	KETERANGAN
1		<i>Generalization</i>	Hubungan di mana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (<i>ancestor</i>).
2		<i>Nary Association</i>	Upaya untuk menghindari asosiasi dengan lebih dari 2 objek.
3		<i>Class</i>	Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama.
4		<i>Collaboration</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu <i>actor</i>
5		<i>Realization</i>	Operasi yang benar-benar dilakukan oleh suatu objek.
6		<i>Dependency</i>	Hubungan di mana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri
7		<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya

Tabel 2. 3 Symbol Sequence Diagram

No	GAMBAR	NAMA	KETERANGAN
1		<i>LifeLine</i>	Objek <i>entity</i> , antarmuka yang saling berinteraksi.

2		<i>Message</i>	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktivitas yang terjadi
3		<i>Message</i>	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktivitas yang terjadi

Tabel 2. 4 *Symbol State Chart Diagram*

No	GAMBAR	NAMA	KETERANGAN
1		<i>State</i>	Nilai atribut dan nilai Link pada suatu waktu tertentu, yang dimiliki oleh suatu objek.
2		<i>Initial Pseudo State</i>	Bagaimana objek dibentuk atau diawali
3		<i>Final State</i>	Bagaimana objek dibentuk dan dihancurkan
4		<i>Transition</i>	Sebuah kejadian yang memicu sebuah state objek dengan cara memperbaharui satu atau lebih nilai atributnya
5		<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya.
6		<i>Node</i>	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi.

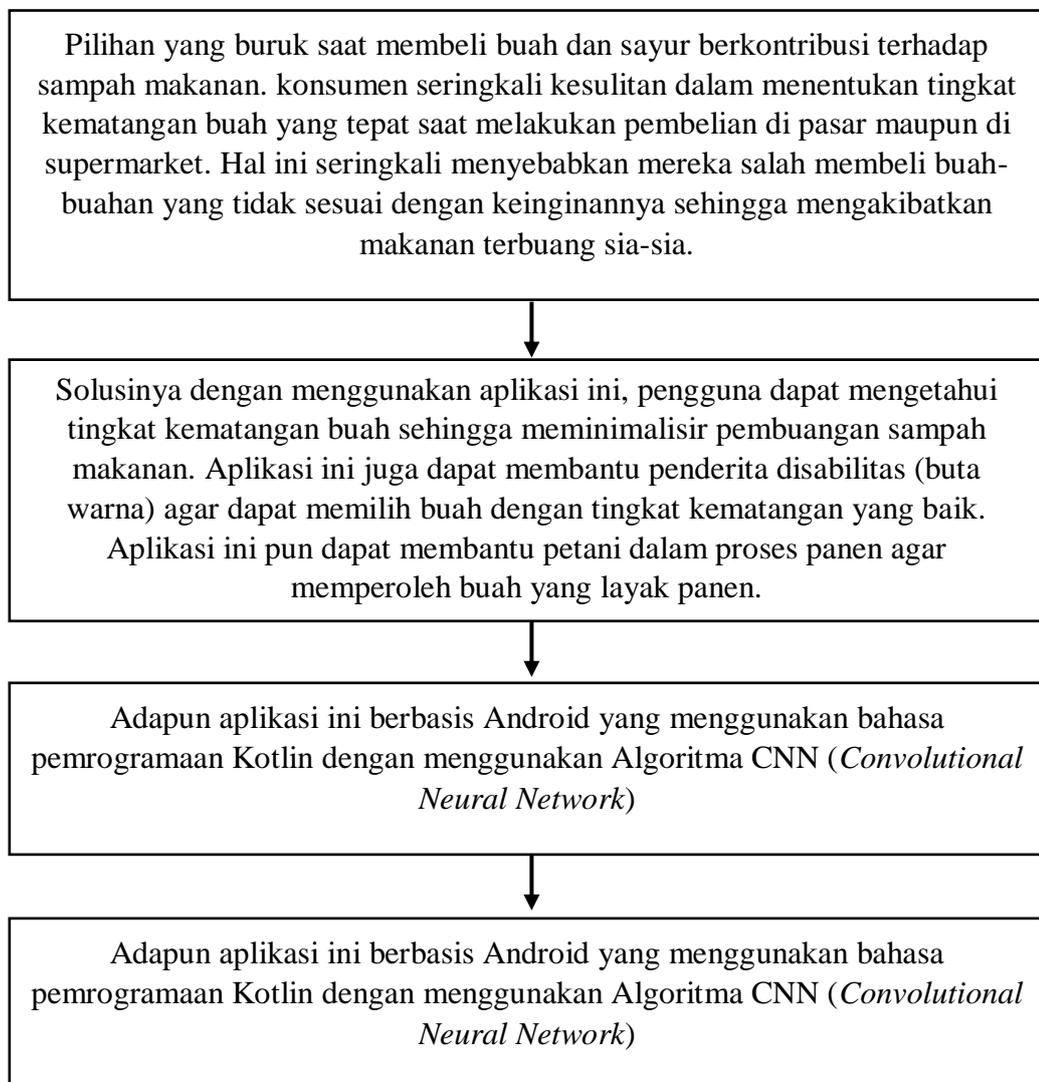
Tabel 2. 5 *Symbol Activity Diagram*

No.	GAMBAR	NAMA	KETERANGAN
1		<i>Actifity</i>	Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain
2		<i>Action</i>	State dari sistem yang mencerminkan eksekusi dari suatu aksi
3		<i>Initial Node</i>	Bagaimana objek dibentuk atau diawali.
4		<i>Actifity Final Node</i>	Bagaimana objek dibentuk dan dihancurkan

5	██████████	<i>Fork Node</i>	Satu aliran yang pada tahap tertentu berubah menjadi beberapa aliran
---	------------	------------------	--

C. Kerangka Pikir

Untuk Memahami alur penelitian diatas, diuraikan ke dalam kerangka berpikir yang akan disajikan dalam bentuk diagram ini:



BAB III

METODE PENELITIAN

A. Jenis Penelitian

Pendekatan penelitian dan pengembangan (R&D) adalah jenis penelitian yang akan diterapkan dalam penelitian ini. Penelitian dan pengembangan (R&D) adalah proses melakukan penelitian yang menghasilkan produk tertentu dan menilai efektivitas metode. Metodologi penelitian yang digunakan dalam bidang pendidikan untuk menghasilkan atau memvalidasi produk yang digunakan dalam proses belajar mengajar disebut penelitian dan pengembangan, atau disingkat R&D. (Nabil et al., 2020).

B. Lokasi dan Waktu Penelitian

Lokasi penelitian dilakukan di Perpustakaan Universitas Muhammadiyah Parepare. Waktu penelitian antara 1 sampai 2 bulan.

Tabel 3. 1 Uraian Kegiatan Penelitian

No.	Uraian Kegiatan
1.	Studi Literatur
2.	Perancangan Aplikasi
3.	Pembuatan Aplikasi
4.	Pengujian Aplikasi
5.	Hasil Pengujian

C. Alat dan Bahan

1. Alat

a. Perangkat Keras (*Hardware*)

1) Laptop

ACER Aspire 3 A314-41-42PE, AMD Dual-Core Processor A4-9120E, 4 GB DDR4 Memory, 1000 GB HDD

2) *Android Mobile*

OPPO A31, RAM 8 GB, ROM 128 GB

b. Perangkat Lunak

1) *Android Studio*

2) *Kotlin*

2. Bahan

Data berupa gambar dari beberapa jenis buah yang akan di deteksi.

D. Teknik Pengumpulan data

Adapun metode yang digunakan untuk menyelesaikan penelitian ini, yaitu sebagai berikut:

1. Metode Pustaka

Metode pustaka adalah strategi penelitian yang menyerukan pencarian dan penilaian literatur yang relevan dengan masalah penelitian. Hal ini juga sering disebut sebagai studi atau tinjauan literatur. Alih-alih mengumpulkan data primer,

metode ini berfokus pada penggunaan sumber sekunder, seperti buku, artikel jurnal, makalah konferensi, dan sumber literatur lainnya.

2. Metode Konsultasi

Melakukan konsultasi dengan orang lain yang lebih berpengetahuan tentang topik yang berhubungan dengan penelitian, seperti dosen pembimbing.

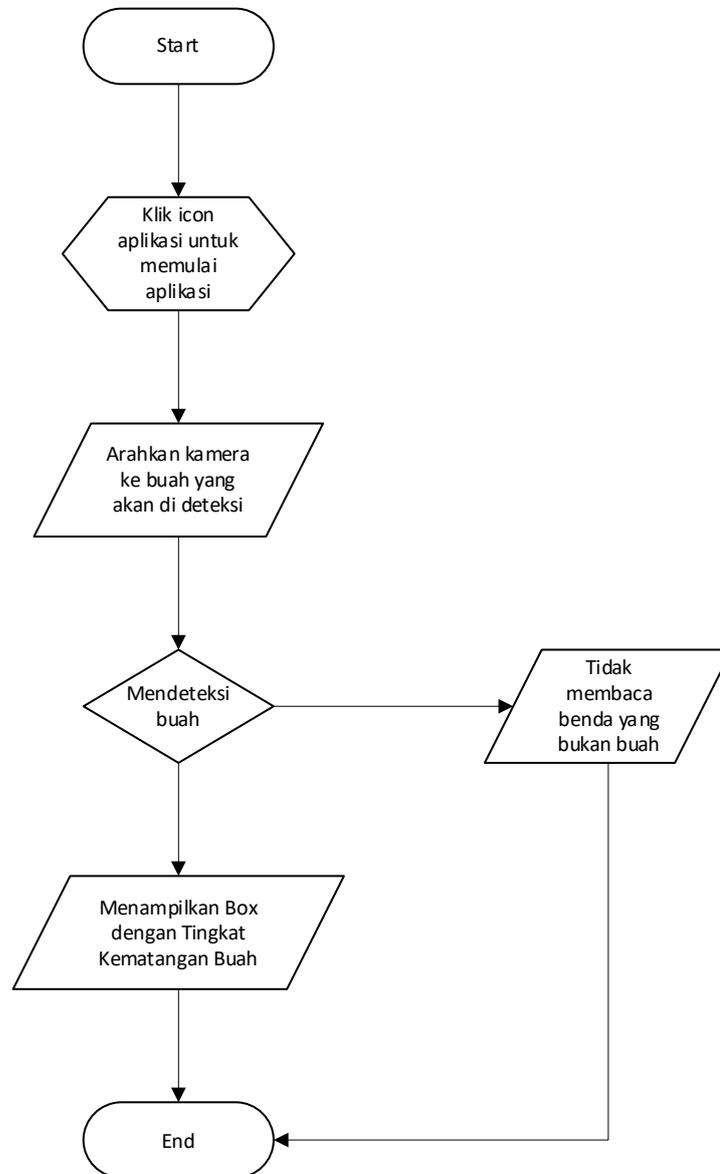
3. Metode Perancangan

Pada metode ini, tahap yang akan dibuat meliputi perencanaan, pembuatan rangkaian, layout komponen dan tata letak komponen.

E. Implementasi

Anotasi data dilakukan setelah pengumpulan data. Tujuan dari anotasi data adalah untuk membantu dalam pelabelan foto berdasarkan keadaan dunia nyata atau kondisi data pelatihan. Selanjutnya, lakukan pelatihan data, yaitu mengajarkan komputer dengan menganalisis data beranotasi untuk membuat fitur yang akan diperhitungkan saat membuat prediksi. Algoritma CNN kemudian harus dipraktekkan untuk mengidentifikasi dan mengkategorikan buah. Proses implementasi melibatkan menggabungkan hasil desain ke dalam perangkat lunak dalam bahasa yang dapat dibaca komputer.

F. Diagram Alir



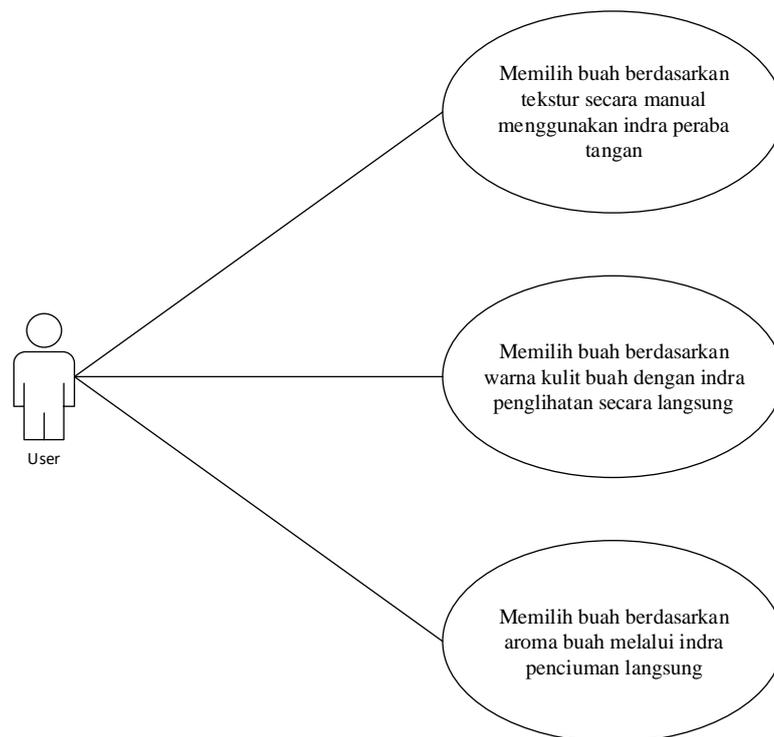
Gambar 3.1 Flowchart

Pada Gambar 3.1, *Flowchart* tersebut menjelaskan secara singkat tentang alur Aplikasi Pendeteksi Kematangan Buah Menggunakan Algoritma CNN. Saat pengguna membuka aplikasi, maka pengguna akan langsung melihat tampilan

kamera. Pengguna dapat mengarahkan kamera ke buah, maka akan ditampilkan box deteksi yang menampilkan tingkat kematangan buah.

G. Desain Sistem

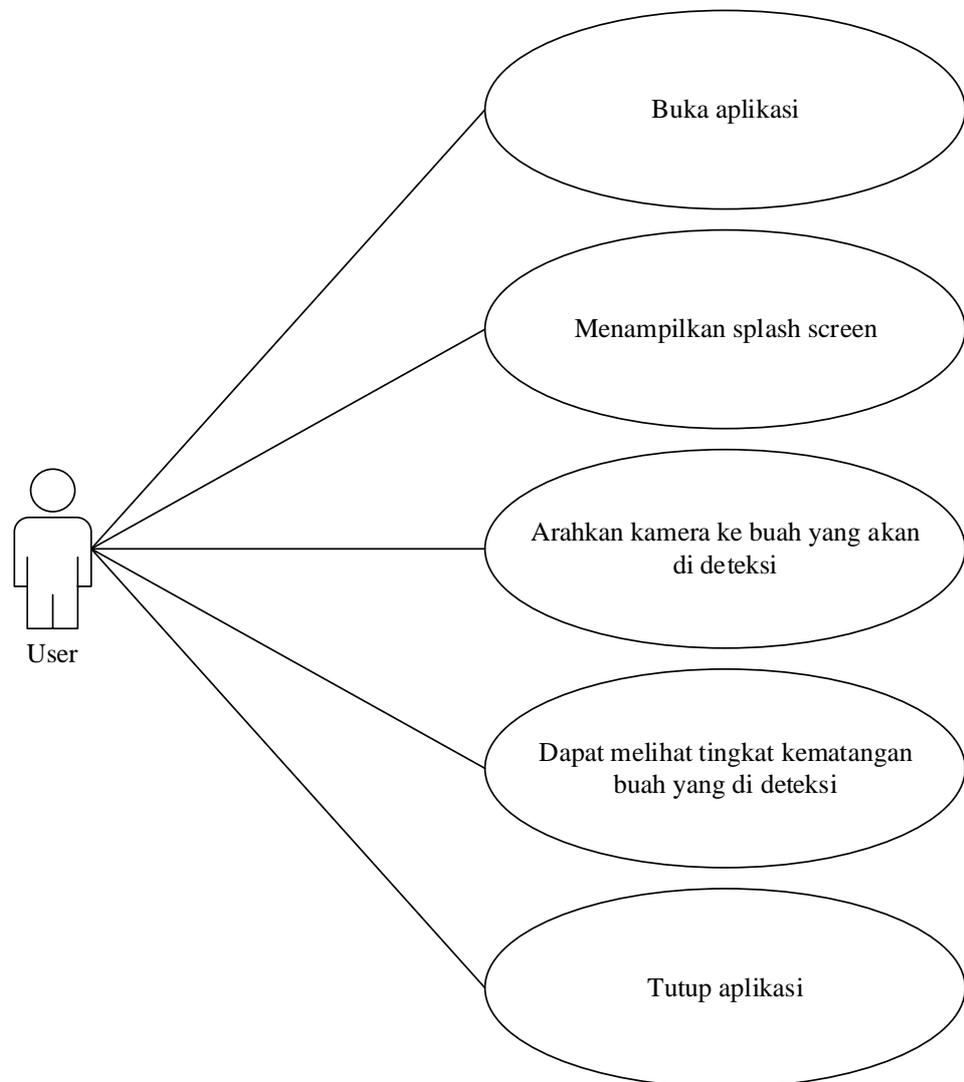
1. Desain sistem yang berjalan



Gambar 3.2 Usecase sistem yang berjalan

Pada gambar *usecase* sistem yang berjalan pada saat ini, konsumen dapat melihat kematangan buah berdasarkan tekstur, warna, dan aroma dengan memegang buah secara langsung dan kemungkinann besar memilih buah yang salah karena salah perkiraan yang dilakukan secara manual tanpa bantuan aplikasi.

2. Desain sistem yang diusulkan



Gambar 3.3 Usecase sistem yang diusulkan

Pada *usecase* sistem yang diusulkan, konsumen dapat mengetahui tingkat kematangan buah berdasarkan persentase kematangan yang ditampilkan aplikasi. Konsumen dapat menggunakan aplikasi dengan membuka aplikasi kemudian mengarahkan kamera ke buah yang akan di deteksi tingkat kematangannya dan kemudian melihat hasil yang ditampilkan oleh aplikasi.

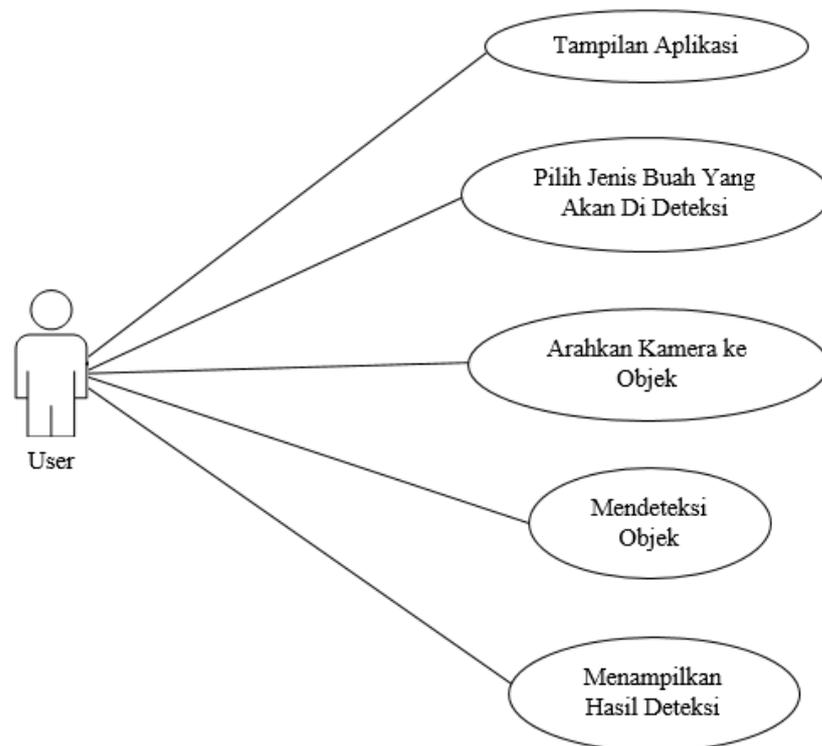
BAB IV

HASIL DAN PEMBAHASAN

A. Analisis Aliran Data dengan UML

1. Use Case Diagram

Jika dilihat dari sudut pandang individu di luar sistem (User), *Use Case Diagram* berfungsi untuk mengimplementasikan manfaat sistem.



Gambar 4.1 Use Case Diagram

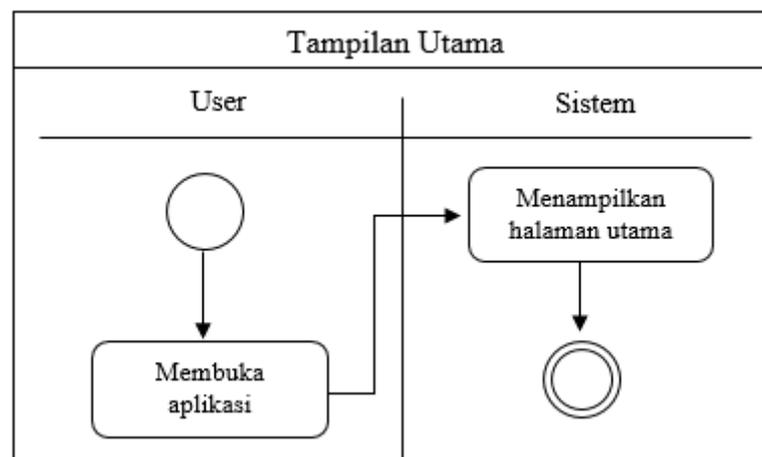
Tabel 4.1 Tabel Keterangan *Use Case*

Nama <i>Use Case</i>	Deskripsi <i>Use Case</i>
Tampilan Aplikasi	Merupakan proses masuk aplikasi dan menampilkan tampilan awal aplikasi
Tampilan Splash Screen	Merupakan tampilan awal yang menampilkan logo dan nama aplikasi
Arahkan Kamera ke Objek	Merupakan proses mulai memindai dan mendeteksi objek
Mendeteksi Objek	Merupakan proses mendeteksi objek yang disorot
Menampilkan Hasil Deteksi	Merupakan proses untuk melihat hasil deteksi yang disorot

2. Activity Diagram

Activity Diagram ini menjelaskan tentang aktifitas-aktifitas yang terjadi dalam sebuah aliran proses pada sistem.

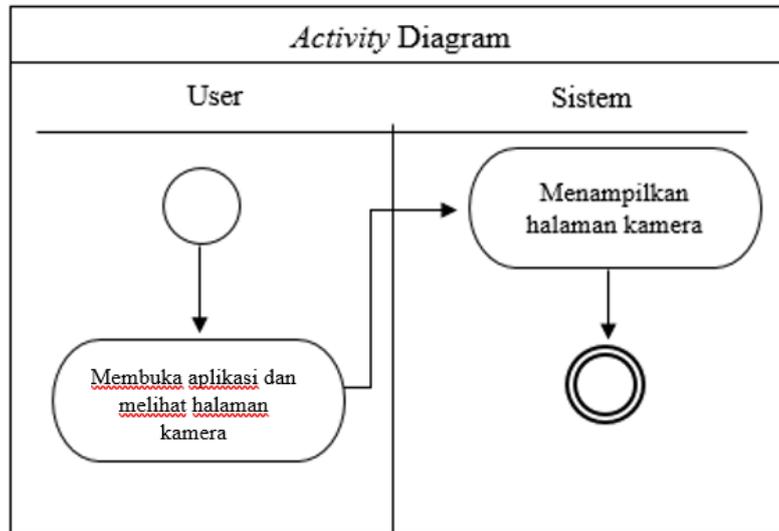
a. *Activity Diagram* Membuka Aplikasi



Gambar 4.2 Activity Diagram Membuka Aplikasi

Tujuan dari Activity Diagram membuka aplikasi adalah untuk menunjukkan bagaimana pengguna meluncurkan program dan bagaimana sistem menampilkan halaman beranda.

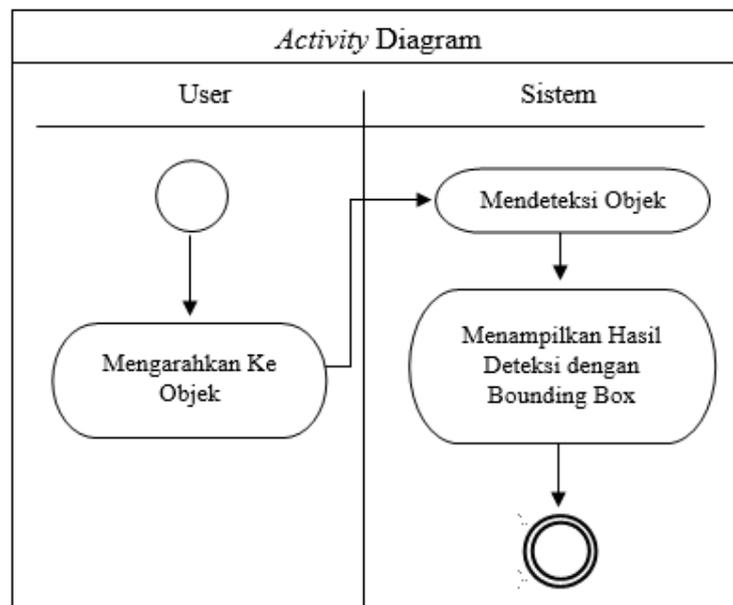
b. *Activity Diagram Menampilkan Kamera*



Gambar 4.3 Activity Diagram Memilih Jenis Buah

Tujuan dari Diagram Aktivitas untuk Membuka kamera untuk mendeteksi buah yang akan diidentifikasi.

c. *Activity Diagram Mengarahkan ke Objek*



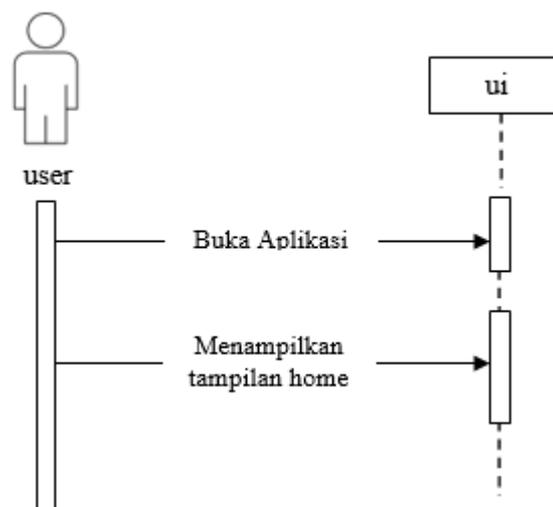
Gambar 4.4 Activity Diagram Mengarahkan ke Objek

Tujuan dari Activity Diagram Mengarahkan ke Objek adalah untuk menunjukkan bagaimana pengguna dapat memulai deteksi suatu objek dengan menunjuknya, dan sistem kemudian akan menampilkan hasil deteksi bersama dengan *bounding box* dan persentase kematangan buah..

3. *Sequence Diagram*

Sequence Diagram merupakan aliran antara objek yang membentuk proses, berikut adalah diagram *sequencenya*.

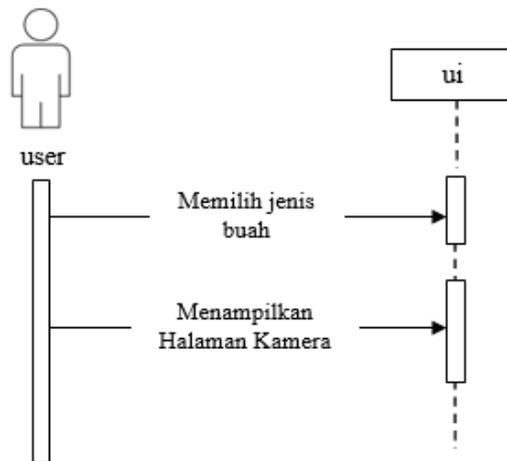
a. *Sequence Diagram* Membuka Aplikasi



Gambar 4.5 *Sequence Diagram* Membuka Aplikasi

Tujuan dari *Sequence Diagram* untuk Membuka Aplikasi adalah untuk menunjukkan bagaimana pengguna meluncurkan program dan bagaimana antarmuka pengguna menampilkan halaman beranda.

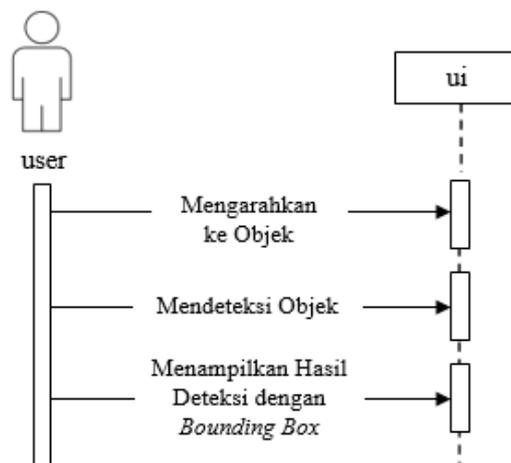
b. *Sequence* Diagram Memilih Jenis Buah



Gambar 4.6 *Sequence* Diagram Memilih Jenis Buah

Sequence Diagram Memilih Jenis Buah bertujuan untuk menunjukkan bagaimana antarmuka pengguna (UI) menampilkan halaman kamera ketika mereka menekan salah satu varietas buah.

c. *Sequence* Diagram Mengarahkan ke Objek



Gambar 4.7 *Sequence* Diagram Mengarahkan ke Objek

Sequence Diagram Mengarahkan ke Objek bertujuan untuk menunjukkan kepada pengguna bagaimana sistem akan mulai mendeteksi objek yang terbuat dari

buah jika dia mengarahkan kamera ke sana. Hasil deteksi akan ditampilkan dengan *Bounding Box* dan persentase kematangan..

B. Perancangan *Input-Output*

1. Tampilan Awal

Pada tampilan layar yang mirip dengan yang ada pada gambar 4.8 di bawah ini akan muncul pada tampilan awal aplikasi pendeteksi kematangan buah. Ada beberapa elemen yang terlihat, termasuk latar belakang ungu dengan kotak dengan karya seni daun yang berfungsi sebagai simbol untuk aplikasi pendeteksi kematangan buah yang diinstal di Android, dan kata "Maurea" sebagai keterangan di bawahnya.



Gambar 4.8 Tampilan Awal

2. Halaman Kamera

Setelah pengguna membuka aplikasi dan tampilan awal, maka pengguna akan di arahkan ke halaman kamera. Pengguna dapat mengarahkan kamera ponsel ke

objek menggunakan halaman kamera ini, dan aplikasi akan mulai menentukan tahap kematangan buah berdasarkan langkah-langkah yang dideteksi.:

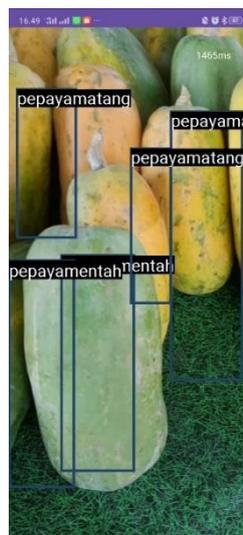
- a. Kumpulkan kumpulan data gambar dari beberapa varietas buah yang akan diidentifikasi, pada berbagai tahap kematangan dan dalam keadaan pencahayaan yang berbeda. Gambar akan diubah menjadi tensor, yang akan digunakan model sebagai input.
- b. Membuat model deteksi objek dengan *TensorFlow* berupa *CNN* dengan banyak layer konvolusi dan *pooling*, yang telah dipersiapkan sebelumnya melalui pelatihan pada dataset yang berisi gambar buah pepaya, buah naga, tomat dan semangka.
- c. Setelah model dibuat, praproses gambar untuk diuji sesuai dengan spesifikasi model. Persiapan dapat mencakup menyesuaikan ukuran gambar, menormalkan nilai piksel, atau melakukan persiapan tambahan yang ditentukan model.
- d. Selanjutnya, gunakan prosedur serupa untuk menjalankan gambar yang diproses melalui model. *Bounding Box* yang berisi objek dan persentase kematangan pada gambar akan diprediksi oleh model.
- e. Langkah selanjutnya adalah mengidentifikasi buah dan menunjukkan visualisasi hasil prediksi pada gambar asli dengan kotak pembatas dan label yang menunjukkan persentase kematangan buah. Tingkat kematangan buah dalam gambar akan terungkap oleh temuan ini.



Gambar 4.9 Gambar halaman Kamera

3. Pengujian Akurasi Deteksi Kematangan

Bounding Box dengan deskripsi yang disediakan di titik sebelumnya ditampilkan di halaman ini. Perangkat lunak ini dapat mengidentifikasi empat varietas buah yang berbeda: semangka, pepaya, buah naga, dan tomat. *Bounding Box* yang menggambarkan tingkat kematangan buah akan muncul jika kamera diarahkan ke salah satu buah.



Gambar 4.10 Deteksi Matang dan Mentah

Ini adalah hasil deteksi pada buah pepaya dengan kondisi matang dan mentah dengan *bounding box* berlabel matang 0,95 dan mentah 0,99.

C. Pengujian Sistem

1. Pengujian *Black Box*

a. Pengujian *Black Box* Tampilan Awal

Tabel 4.2 Pengujian *Black Box* Halaman Utama

Tes Faktor	Hasil	Keterangan
Menu Home	✓	Aplikasi sukses terbuka, dengan menampilkan tampilan seperti gambar dibawah.
		

b. Pengujian *Black Box* Pepaya**Tabel 4.3** Pengujian *Black Box* Pepaya Matang

Tes Faktor	Hasil	Keterangan
Pepaya	✓	Aplikasi berhasil mendeteksi kematangan buah pepaya menampilkan <i>bounding box</i> dengan label pepaya matang
		

Tabel 4.4 Pengujian *Black Box* Pepaya Mentah

Tes Faktor	Hasil	Keterangan
Pepaya	✓	Aplikasi berhasil mendeteksi kematangan buah pepaya menampilkan <i>bounding box</i> dengan label papaya mentah
 A screenshot from a mobile application showing a green papaya being held by a hand against a grassy background. A blue bounding box is drawn around the papaya, and the text 'pepayamentah' is overlaid on the box. The top of the screen shows a status bar with the time 16:50, signal strength, Wi-Fi, and battery icons. In the top right corner of the image area, there is a small text '1451ms'.		

c. Pengujian *Black Box* Buah Naga

Tabel 4.5 Pengujian *Black Box* Buah Naga Matang

Tes Faktor	Hasil	Keterangan
Buah Naga	✓	Aplikasi berhasil mendeteksi kematangan buah naga menampilkan <i>bounding box</i> dengan label naga matang
 A screenshot of a mobile application interface. At the top, there is a status bar with the time 17:17, signal strength, Wi-Fi, and battery icons. Below the status bar, a green dragon fruit is visible in a glass display case. A blue bounding box is drawn around a red dragon fruit held in a hand in the foreground. The text 'nagamatang' is overlaid on the bounding box. In the top right corner of the application area, the text '1416ms' is displayed.		

Tabel 4.6 Pengujian *Black Box* Buah Naga Mentah

Tes Faktor	Hasil	Keterangan
Buah Naga	✓	Aplikasi berhasil mendeteksi kematangan buah naga menampilkan <i>bounding box</i> dengan label naga mentah



d. Pengujian *Black Box* Tomat**Tabel 4.7** Pengujian *Black Box* Tomat Matang

Tes Faktor	Hasil	Keterangan
Tomat	✓	Aplikasi berhasil mendeteksi kematangan buah tomat menampilkan <i>bounding box</i> dengan label tomat matang
 A screenshot of a mobile application interface. At the top, there is a status bar with the time 17.06, signal strength, Wi-Fi, and battery icons. Below the status bar, the text '1624ms' is displayed. The main content area shows a photograph of several tomatoes on a dark blue, textured surface. Two bounding boxes are drawn around the tomatoes, each with the label 'tomatmatang' written above it in a black box. The bounding boxes are blue and accurately enclose the tomatoes.		

Tabel 4.8 Pengujian *Black Box* Tomat Matang

Tes Faktor	Hasil	Keterangan
Tomat	✓	Aplikasi berhasil mendeteksi kematangan buah tomat menampilkan <i>bounding box</i> dengan label tomat mentah dan tomat matang
		

e. Pengujian *Black Box* Semangka**Tabel 4.9** Pengujian *Black Box* Semangka Matang

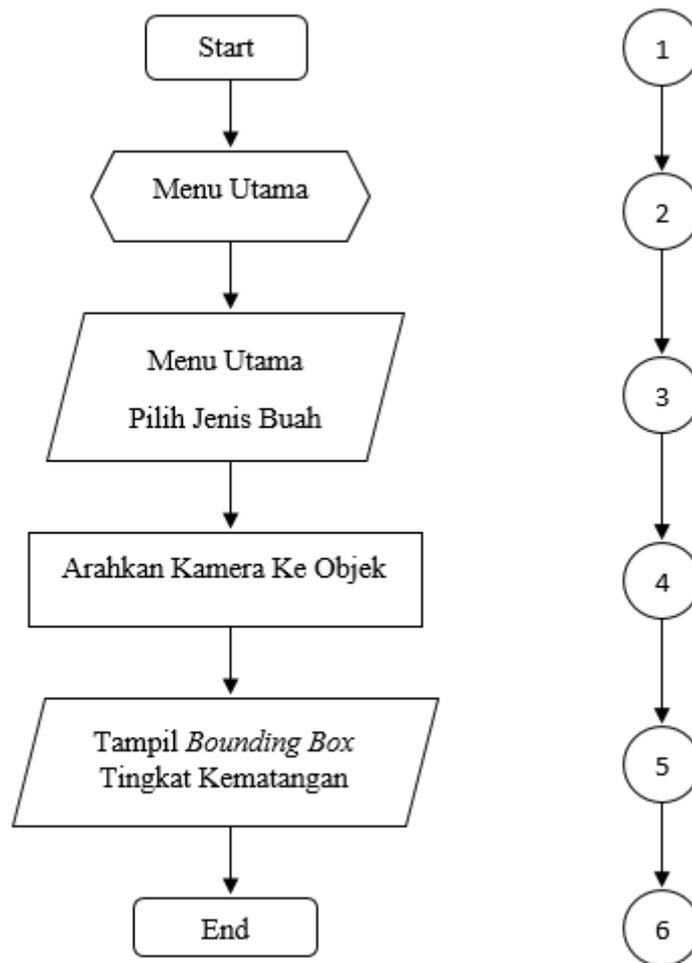
Tes Faktor	Hasil	Keterangan
Semangka	✓	Aplikasi berhasil mendeteksi kematangan buah semangka menampilkan <i>bounding box</i> dengan label semangka matang
 <p>The image is a screenshot of a mobile application interface. It shows a close-up of several watermelons. A blue bounding box is drawn around one of the watermelons. Above the bounding box, the text 'semangkamatar' is displayed. In the top right corner of the screenshot, there is a timestamp '1589ms'. The top status bar of the phone is visible, showing the time '16:59' and various icons.</p>		

Tabel 4.10 Pengujian *Black Box* Semangka Mentah

Tes Faktor	Hasil	Keterangan
Semangka	✓	Aplikasi berhasil mendeteksi kematangan buah semangka menampilkan <i>bounding box</i> dengan label semangka mentah
 A screenshot of a mobile application interface. At the top, there is a status bar with the time 16:56 and various icons. Below the status bar, the text 'semangkament' is displayed in white on a dark background, with '1591ms' underneath it. The main part of the image shows a photograph of several watermelons. A blue bounding box is drawn around one of the watermelons in the foreground. The label 'semangkamentah' is written in white text on a black background, positioned above the bounding box. The watermelons are green with dark green stripes and are resting on a wooden surface.		

2. Pengujian *White Box*

a. *Flowchart* dan *Flowgraph* halaman utama



Gambar 4.11 *Flowchart* dan *Flowgraph* Halaman Utama

Dari *flowgraph* halaman utama di atas, dapat dilakukan proses perhitungan sebagai berikut:

- 1) Menghitung *Cyclomatic Complexity* $V(G)$ dari Edge dan Node

Dengan Rumus : $V(G) = E - N + 2$

$$N(\text{node}) = 6$$

$$E(\text{edge}) = 5$$

$$P (\text{Predikat node}) = 0$$

Penyelesaian ;

$$V(G) = E - N + 2$$

$$= 5 - 6 + 2$$

$$= 1$$

$$\text{Predikat (P)} = P + 1$$

$$= 0 + 1$$

$$= 1$$

2) Berdasarkan perhitungan *Cyclomatic Complexity* dari *Flowgraph* di atas memiliki *Region* = 1

3) Independent path pada *Flowgraph* di atas adalah :

$$\text{Path 1} = 1,2,3,4,5,6$$

4) Grafik matriks halaman utama

Tabel 4.11 Matriks Halaman Utama

	1	2	3	4	5	6	7	E - 1
1		1						1 - 1 = 0
2			1					1 - 1 = 0
3				1				1 - 1 = 0
4					1			1 - 1 = 0
5						1		1 - 1 = 0
SUM (E + 1)								0 + 1 = 1

3. Hasil Pendeteksian Tidak Berhasil



Gambar 4.12 Hasil pendeteksian tidak berhasil

Pada gambar 4.12 merupakan contoh hasil pendeteksian tidak berhasil karena mendeteksi objek selain buah namun membacanya sebagai buah pepaya matang. Hal demikian terjadi apabila warna objek yang ditangkap kamera mengandung unsur warna buah yang sudah di latih pada machine learning. Karena aplikasi ini hanya berfokus pada tingkat kematangan berdasarkan warna, jadi hal ini akan terjadi apabila ada objek lain yang warnanya mirip.

D. Proses Deteksi

Sistem pendeteksian tingkat kematangan buah dibangun menggunakan YoloV8. Untuk mengembangkan sistem ini, langkah awal melibatkan pengumpulan dataset gambar, yang kemudian dibagi menjadi train set, validation

set, dan test set. Setelah berhasil mengumpulkan dataset gambar, dilakukan proses labeling dengan tiga kelas yaitu muda, setengah matang, dan matang. Setelah pelabelan selesai, langkah berikutnya adalah melakukan proses training dataset untuk melatih model sehingga mampu memprediksi dan mendeteksi objek dengan lebih baik.

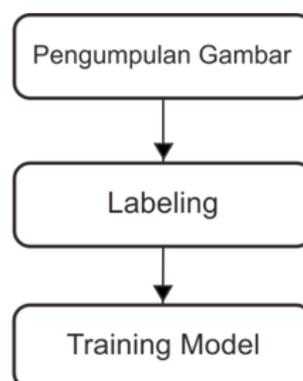
Pembuatan dataset dilakukan untuk menyiapkan data yang akan digunakan dalam proses pelatihan. Data yang digunakan dalam penelitian ini berupa gambar-gambar buah yang telah diklasifikasikan menjadi delapan kategori berbeda, yaitu "pepaya matang", "pepaya mentah", "tomat matang", "tomat mentah", "naga matang", "naga mentah", "semangka matang", "semangka mentah". Proses pengambilan gambar dan dokumentasi dilakukan secara independen. Total jumlah gambar yang berhasil disiapkan mencapai 440, yang terdiri dari 320 gambar untuk data pelatihan, 80 gambar untuk data validasi, dan 40 gambar untuk data pengujian. Pada tujuan akhirnya, pelatihan data ditujukan untuk menghasilkan bobot yang kemudian akan dimanfaatkan dalam pendeteksian tingkat kematangan buah. Dalam tahap ini, pelatihan data dilakukan melalui penerapan algoritma YOLOv8 dengan menggunakan perangkat *Visual Studio Code* dan *Python* sebagai lingkungan kerjanya.

Tabel 4.12 Kelas Dataset

No.	Label Dataset	Train	Validation	Test
1.	Pepaya Matang	40	10	5
2.	Pepaya Mentah	40	10	5
3.	Tomat Matang	40	10	5
4.	Tomat Mentah	40	10	5
5.	Naga matang	40	10	5
6.	Naga Mentah	40	10	5
7.	Semangka Matang	40	10	5
8.	Semangka Mentah	40	10	5

Pengumpulan data dalam penelitian ini melalui langkah-langkah pengambilan gambar secara independen. Setelah proses pengumpulan data selesai, setiap gambar dalam dataset diberikan label yang sesuai dengan nama gambar. Proses pelabelan melibatkan pembuatan nama kelas dan penandaan kotak batas pada setiap objek yang terdapat dalam gambar. Untuk meningkatkan akurasi proses pelatihan menggunakan metode YOLOv8, ukuran gambar untuk setiap objek yang telah dilabeli disesuaikan.

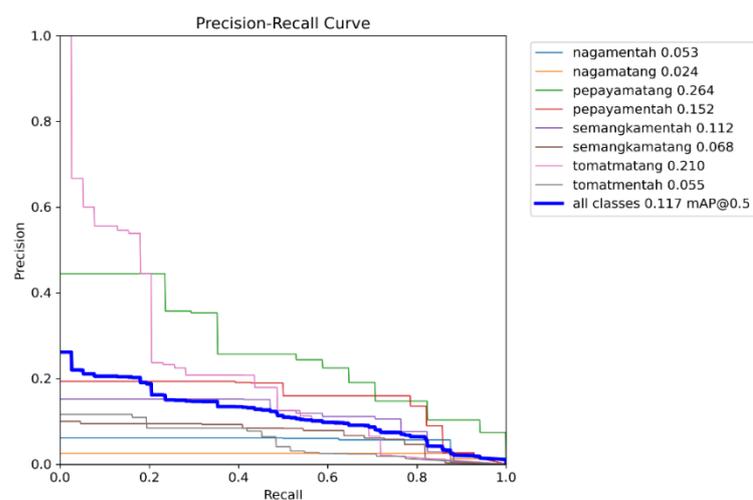
a. Pre-processing Data

**Gambar 4.13** *Preprocessing Data*

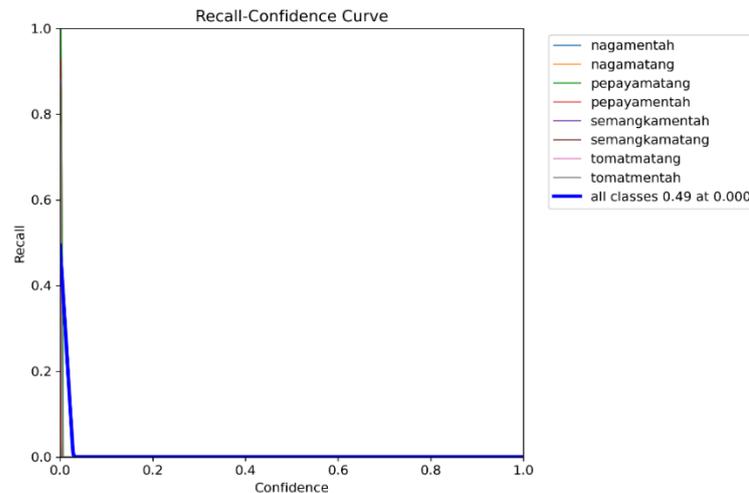
Berikut ini merupakan tahap yang dilakukan dalam pre-processing data untuk membuat model pelatihan yang akan digunakan untuk mendeteksi dengan algoritma YOLO.

- a. Pengumpulan gambar merupakan tahap untuk memperoleh citra objek yang kemudian dikelompokkan berdasarkan kelasnya.
- b. Labelling proses merupakan proses menggunakan software labelling untuk memberikan label pada citra yang sesuai kelasnya.
- c. Training model merupakan pelatihan data yang telah didapatkan sehingga dapat dipakai untuk deteksi metode YOLO sebagai acuan.

Untuk tahapan ini citra yang telah diperoleh diberikan label. Pada setiap buah pepaya akan diberikan label yang sesuai dengan kelas berdasarkan tingkat kematangannya. Dilakukan proses labelling pada citra buah pepaya dengan bantuan software labeling dengan membuat bounding box dan menentukan kelas sesuai tingkat kematangan.



Gambar 4.14 Kurva hubungan antara *Precision* dengan *Recall*



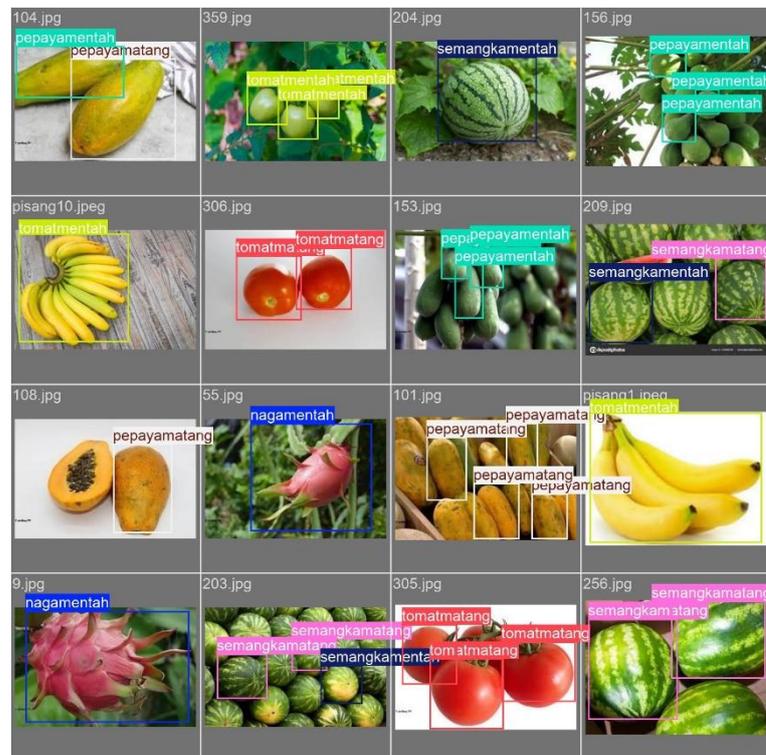
Gambar 4.15 Kurva hubungan antara *Recall* dengan *Confidence*

Semakin tinggi nilai recall yang dihasilkan dari proses pelatihan dan validasi data, semakin tinggi pula akurasi sistem yang telah dibuat. Hasil visualisasi pada gambar 4.14 dan gambar 4.15 memberikan penjelasan lebih lanjut tentang kinerja sistem. Selain itu, selama proses pelatihan dan pengujian dalam tiap epoch-nya, sistem mengalami perkembangan dan peningkatan kinerja.

Secara keseluruhan, penelitian ini mencapai hasil yang kurang memuaskan dalam pendeteksian kematangan buah, dan kinerja sistem terus meningkat melalui proses pelatihan dan pengujian. Gambar-gambar yang disediakan membantu mengilustrasikan peningkatan kinerja dan keakuratan sistem

Dalam penelitian ini, berhasil dicapai nilai akurasi confidence sebesar 0,49 atau setara dengan 49% dalam mendeteksi tingkat kematangan buah. Metode YOLOv8 yang digunakan dalam pendeteksian ini berjalan dengan baik namun menghasilkan nilai akurasi yang kurang memuaskan. Kurangnya keakuratan pada hasil deteksi dapat dipengaruhi oleh kekurangan variasi dalam dataset yang digunakan. Hasil pendeteksian tingkat kematangan tomat serta nilai akurasi untuk

setiap klas atau klasifikasinya direpresentasikan dalam bentuk gambar yang dapat dilihat dalam Gambar 4.16.



Gambar 4.16 Format Gambar Dataset

Tabel 4.13 Sampel

No.	Nama	Jumlah
1.	Pendeteksian Berhasil	14
2.	Pendeteksian Tidak Berhasil	20
3.	Jumlah	34

Dari 34 data sampel diatas dengan pendeteksian berhasil sejumlah 14 sampel dan pendeteksian tidak berhasil berjumlah 20 sampel, maka diperoleh aplikasi pendeteksi kematangan buah dengan hasil akurasi aplikasi sebesar 41%.

BAB V

PENUTUP

A. Kesimpulan

Bab-bab sebelumnya telah memberikan gambaran umum tentang temuan dari pengujian dan diskusi Aplikasi Deteksi Kematangan Buah. Baik pengujian *black box* dan pengujian *white box*, yang telah diverifikasi secara manual untuk akurasi, telah digunakan untuk mengevaluasi program Deteksi Kematangan Buah. Berikut ini adalah beberapa kesimpulan yang dapat dibuat:

1. Aplikasi dibangun menggunakan *Android Studio*, Python dan bahasa pemrograman *Kotlin*.
2. Menggunakan beberapa gambar buah sebagai datasetnya.
3. Sistem dapat mendeteksi kematangan beberapa jenis buah seperti: buah naga, pepaya, tomat dan semangka dengan menampilkan *bounding box* dengan label keterangan tingkat kematangan buah yang terdeteksi.
4. Aplikasi Pendeteksi Kematangan Buah ini telah diuji untuk mendeteksi buah naga, pepaya, tomat dan semangka dari berbagai tingkat kematangan, semuanya berhasil terdeteksi dengan baik.
5. Penelitian ini menunjukkan bahwa aplikasi pendeteksi kematangan buah ini memiliki tingkat keberhasilan sebesar 41% dalam mendeteksi kematangan buah. Hasil ini diperoleh dari pengujian terhadap 34 sampel buah, di mana aplikasi berhasil mendeteksi dengan benar kematangan

pada 35 buah. Faktor-faktor seperti kualitas gambar dan kondisi pencahayaan mempengaruhi hasil deteksi dan perlu diperhatikan untuk meningkatkan akurasi aplikasi. Nilai akurasi ini dipengaruhi oleh berbagai faktor, termasuk kualitas dataset, pengambilan gambar dari berbagai sudut, kualitas gambar, dan kualitas video. Penggunaan dataset yang mencakup beragam sudut pengambilan gambar dapat meningkatkan nilai akurasi secara signifikan. Selain itu, perhitungan area pada gambar juga berperan penting dalam menentukan akurasi, karena saat objek bertumpuk atau terpotong, hal ini dapat mempengaruhi hasil akurasi yang dihasilkan.

B. Saran

Disarankan untuk memperoleh nilai akurasi dan confidence yang lebih tinggi, penggunaan video dengan gambar yang jelas, dataset yang berkualitas, dan mencakup banyak sudut pandang adalah hal yang dianjurkan. Dengan demikian, hasil pendeteksian kematangan buah dapat ditingkatkan untuk mendapatkan performa yang optimal.

Fitur yang lebih menarik harus ditambahkan ke aplikasi ini di masa depan sehingga dapat melakukan lebih dari sekadar menentukan apakah buah sudah matang; Itu juga dapat menentukan jenis buah dan tidak dapat mendeteksi apa pun selain buah. Selain itu, menambahkan fitur jual beli akan menambah minat aplikasi.

DAFTAR PUSTAKA

- Damayanti, S. A., Arkadia, A., & Prasvita, D. S. (2021). Klasifikasi Buah Mangga Badami Untuk Menentukan Tingkat Kematangan dengan Metode CNN. In *Prosiding Seminar Nasional Mahasiswa Bidang Ilmu Komputer dan Aplikasinya* (Vol. 2, No. 2, pp. 158-165).
- 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
- Liu, T., Fang, S., Zhao, Y., Wang, P., & Zhang, J. (n.d.). *Implementation of Training Convolutional*.
- Nabil, M., Andryana, S., & Benrahman. (2020). Implementasi QR Code Menggunakan Aplikasi Android Untuk Melakukan Presensi Mahasiswa. *Seminar Nasional Teknologi Komputer & Sains*, 351–355.
- Maulana, I., Rahaningsih, N., & Suprpti, T. (2023). ANALISIS PENGGUNAAN MODEL YOLOV8 (YOU ONLY LOOK ONCE) TERHADAP DETEKSI CITRA SENJATA BERBAHAYA. *JATI (Jurnal Mahasiswa Teknik Informatika)*, 7(6), 3621-3627.
- Miftahuddin, Y. (2022). Perbandingan Metode EfficientNetB3 dan MobileNetV2 Untuk Identifikasi Jenis Buah-buahan Menggunakan Fitur Daun: Metode EfficientNetB3 dan MobileNetv2. *Jurnal Ilmiah Teknologi Infomasi Terapan*, 9(1).
- Pengolahan Citra Digital : Sebuah Pengantar. (2020). (n.p.): Desanta Publisher.
- Riyandi, A., Widodo, T., & Uyun, S. (2022). Classification of Damaged Road Images Using the Convolutional Neural Network Method. *Telematika: Jurnal Informatika dan Teknologi Informasi*, 19(2), 147-158
- Romzi, M., & Kurniawan, B. (2020). Pembelajaran Pemrograman Python Dengan Pendekatan Logika Algoritma. *JTIM: Jurnal Teknik Informatika Mahakarya*, 3(2), 37-44.
- Sanjaya, S. (2022). Aplikasi Pengenalan Tingkat Kematangan Buah Tomat Menggunakan Fitur Warna Hsv Berbasis Android. *Jurnal Teknoinfo*, 16(1), 26. <https://doi.org/10.33365/jti.v16i1.1489>
- Saputra, A. chandra, & Oktaviyani, E. D. (2023). Rancang Bangun Sistem Deteksi Kematangan Buah Kelapa Sawit Berdasarkan Deteksi Warna Menggunakan Algoritma K-Nn. *Jurnal Teknologi Informasi: Jurnal Keilmuan Dan Aplikasi Bidang Teknik Informatika*, 7(2), 222–229. <https://doi.org/10.47111/jti.v7i2.9232>

- Sohan, M., Sai Ram, T., Reddy, R., & Venkata, C. (2024). A review on yolov8 and its advancements. In *International Conference on Data Intelligence and Cognitive Informatics* (pp. 529-545). Springer, Singapore.
- Tang, D., Qin, B., & Liu, T. (2015, September). Document modeling with gated recurrent neural network for sentiment classification. In *Proceedings of the 2015 conference on empirical methods in natural language processing* (pp. 1422-1432).
- Wibowo, A., Lusiana, L., & Dewi, T. K. (2023). Implementasi Algoritma Deep Learning You Only Look Once (YOLOv5) Untuk Deteksi Buah Segar Dan Busuk. *Paspalum: Jurnal Ilmiah Pertanian*, *11*(1), 123. <https://doi.org/10.35138/paspalum.v11i1.489>
- Yusman, M. A., Evanita, E., & Riadi, A. A. (2023). Klasifikasi Kematangan Buah Tin Menggunakan Convolutional Neural Network Berbasis Android. *Jurnal Teknik Informatika dan Sistem Informasi*, *9*(2), 167-176