

Aplikasi Pendeteksi Kematangan Buah Menggunakan Algoritma CNN

Nadia Eka Patimah¹, Muhammad Basri², Marlina³

Program Studi Teknik Informatika, Universitas Muhammadiyah Parepare, Indonesia
Email : nadiaekapatimah2802@gmail.com

Abstract: One of the main impacts of errors in selecting fruit ripeness is food waste. Fruits that do not meet the expected quality or ripeness standards end up as waste. The purpose of this study is to produce an application that is expected to help consumers choose fruit with the appropriate level of ripeness, which can reduce food waste. The research method uses a quantitative method, using the Android Studio and Visual Studio Code text editors with the Kotlin and Python programming languages. This application works by detecting in real-time, utilizing the bounding box method to display labels that indicate the level of fruit ripeness. This study produces an application that has a camera feature that displays the success of detecting ripeness and can make it easier for consumers to choose between ripe and unripe fruit.

Keywords: CNN; Android; Python; Yolo

1. PENDAHULUAN

Convolutional Neural Network (CNN) merupakan salah satu metode dari *Deep Learning* yang memanfaatkan jaringan syaraf tiruan untuk implementasi permasalahan dengan mempelajari data data yang ada. Metode *Convolutional Neural Network* memiliki hasil yang paling signifikan dalam pengenalan citra digital. (Febrywinata, 2024). 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. Jaringan ini dirancang khusus untuk memproses citra sebagai masukan, dengan lapisan konvolusi sebagai komponen utamanya. Pendekatan ini meningkatkan efisiensi pembelajaran citra dan mempermudah aplikasinya (Sayyidin et al., 2024) (Yunus and Riyadi 2024)

Android adalah sebuah sistem operasi perangkat *mobile* berbasis *linux* yang mencakup sistem operasi, *middleware* dan aplikasi. *Android* menyediakan platform terbuka bagi para pengembang untuk menciptakan aplikasi mereka. Awalnya, *Google inc* memberi *android inc* merupakan pendatang baru yang membuat perangkat lunak untuk ponsel atau *smartphone*. (Dhita R. L. et al., 2023)

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. *Python* merupakan sebuah bahasa pemrograman tingkat tinggi yang dibuat oleh Guido Van Rossum dan dirilis pada tahun

1991 *Python* juga merupakan bahasa yang sangat populer belakangan ini. Selain itu *python* juga merupakan bahasa pemrograman yang multi fungsi salah satunya pada bidang *Machine Learning* dan *Deep Learning*. (Alfarizi et al., 2023).

Model pertama dari *YOLO* dijelaskan pertama kali oleh Joseph Redmon, pada tahun 2015 jurnal yang berjudul *You Only Look Once: Unified, Real-Time Object Detection*. Tercatat bahwa *Ross Girshick*, pengembang dari *R-CNN*, merupakan penulis dan *contributor* dalam jurnal tersebut. Pendekatannya melibatkan jaringan saraf tunggal yang dilatih *end-to-end* menggunakan gambar sebagai input dan memprediksi bounding box dan label kelas untuk setiap *bounding box*. Teknik ini menawarkan akurasi prediksi yang lebih rendah, meskipun beroperasi pada 45 fps hingga 155 fps dengan optimasi kecepatan dari versi model ini. (Faruqi, 2021)

Penelitian yang 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. (Yusman et al., 2023)

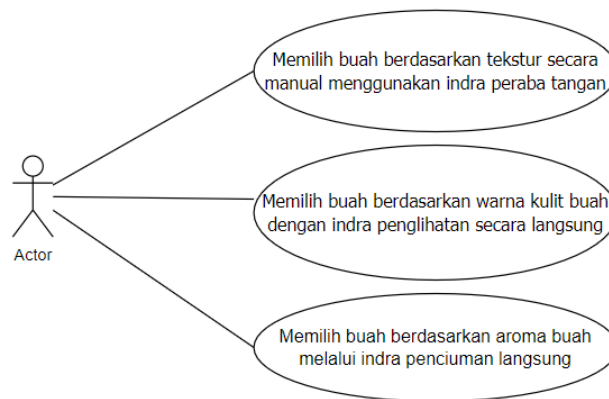
Dari beberapa penelitian diatas maka fokus penelitian penulis adalah merancang aplikasi pendeteksi kematangan buah untuk membantu konsumen dalam pemilihan buah. Adapun perangkat yang digunakan terbagi dua, yaitu perangkat keras dan perangkat lunak. Perangkat keras yang digunakan yaitu Laptop ACER dengan spesifikasi *Processor* AMD A4, RAM 4 GB, *Hardisk* 1 TB, *LCD Monitor* 14 inci dan perangkat *Android Mobile* dengan *Processor* OPPO A71, RAM 8 GB dan LCD 6,5". Dan perangkat lunak yang digunakan yaitu Windows 11, *Android Studio*, *Visual Sstudio Code*.

2. METODOLOGI PENELITIAN

Metode penelitian menggunakan metode penelitian kuantitatif dengan *Python* sebagai pelatihan *Machine Learning* dan *Visual Studio Code* sebagai teks editor. Kemudian dibangun dalam bentuk *Android* menggunakan bahasa pemrograman *Kotlin* dan *Android Studio* sebagai teks editor.

2.1. Use Case Diagram

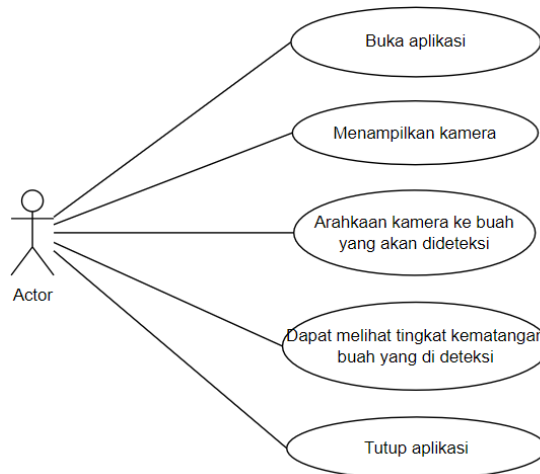
1. Desain sistem yang sedang berjalan



Gambar 1. *Use case* dari sistem yang sedang berjalan

Pada gambar 1. *Use case* 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. Sistem yang berjalan saat ini masih menggunakan cara manual sehingga kurang efisien untuk pemilihan buah dengan cepat.

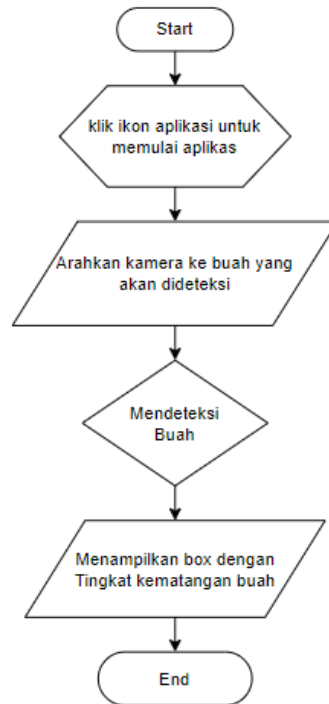
2. Perancangan sistem yang diusulkan



Gambar 2. *Use case* sistem yang diusulkan

Pada gambar 2. *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. System usulan ini lebih efektif dibanding system yang berjalan karena mempermudah pengguna dalam pemilihan buah yang lebih efisien.

3. Flowchart



Gambar 3. Flowchart

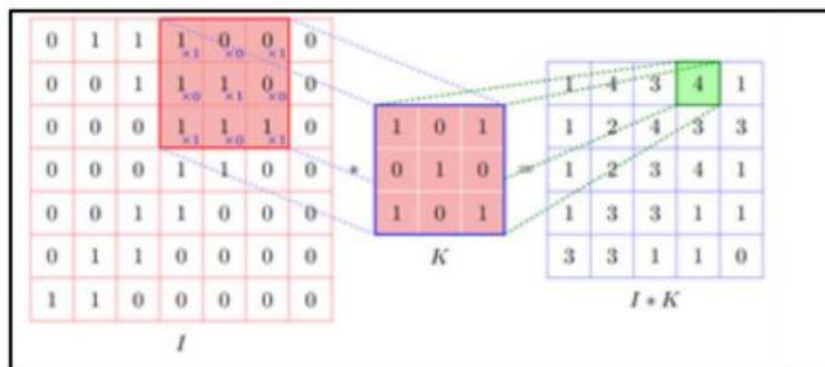
Pada Gambar 3. Flowchart tersebut menjelaskan secara singkat tentang alur Aplikasi Pendeteksi Kematangan Buah Menggunakan Algoritma CNN. Saat pengguna membuka aplikasi, maka pengguna akan langsung melihat tampilan kamera. Pengguna dapat mengarahkan kamera ke buah, maka akan ditampilkan box deteksi yang menampilkan tingkat kematangan buah

2.2. CNN (*Convolutional Neural Network*)

Berikut merupakan tahapan arsitektur *CNN*:

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 4. 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:

Ukuran output = ((ukuran input - ukuran filter) / stride) + 1

3. Lapisan *fully connected*

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:

Jumlah parameter = (jumlah neuron pada layer sebelumnya + 1) x jumlah neuron pada layer saat ini

3. HASIL DAN PEMBAHASAN

System pendeteksian tingkat kematangan buah ini dibangun menggunakan metode *YOLO*. Untuk mengembangkan sistem ini, langkah awal melibatkan pengumpulan dataset gambar, yang kemudian dibagi menjadi *train set*, *validation set*, dan *test set*. 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. Dalam tahap ini, pelatihan data dilakukan melalui penerapan algoritma *YOLOv8* dengan menggunakan perangkat *Visual Studio Code* dan *Python* sebagai lingkungannya.

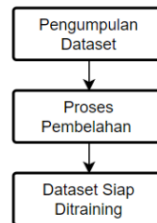
Tabel 1. Kelas Dataset

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

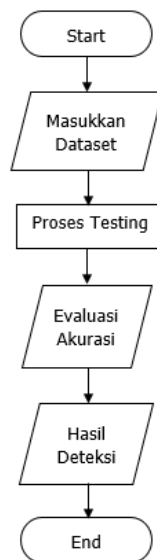
Setelah proses pengumpulan data selesai, setiap gambar dalam dataset diberikan label yang sesuai dengan nama gambar. Proses pelabelan melibatkan pembuatan nama kelas dan penandaan kotak batas pada setiap objek yang terdapat dalam gambar. Untuk meningkatkan akurasi proses pelatihan menggunakan metode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

Kemudian dataset dimasukkan dalam folder pelatihan di *Visual Studio Code* untuk dilatih dengan metode *YOLO* dengan *Library Ultralytics* dengan pelatihan dijalankan menggunakan Bahasa pemrograman *Python*.



Gambar 5. *Pre-Processing data*

Setelah pelabelan selesai, objek dan labelnya dipisahkan ke dalam folder terpisah untuk data pelatihan dan data validasi. Konfigurasi data yang diperlukan juga dilakukan. Gambar 5 di bawa ini menggambarkan langkah-langkah pra-pemrosesan data, sementara Gambar 6 menggambarkan proses pelatihan dan pengujian dalam pendeteksian tingkat kematangan buah tomat dengan menggunakan metode *YOLOv8*.

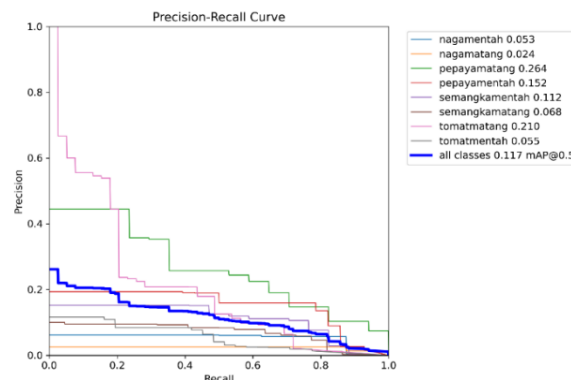


Gambar 6. *Training Process dan Testing Process*

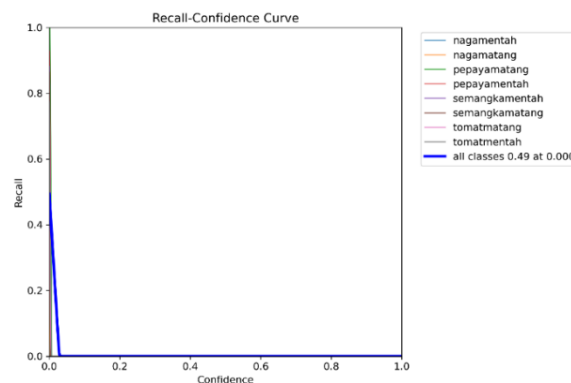
Proses input dataset melibatkan langkah menyediakan kumpulan citra buah tomat, semangka, pepaya, buah naga dalam kematangan, termasuk buah mentah, dan matang, yang dimuat ke dalam *Visual Studio Code*. Hal ini menjadi penting karena dataset harus lengkap dan terperinci agar pendeteksian objek menjadi stabil dan menghasilkan tingkat akurasi yang tinggi. Dataset yang digunakan telah diberi label pada setiap gambar untuk mengindikasikan tingkat kematangan buah yang sesuai.

Proses pelatihan dapat dikatakan berhasil jika objek telah terdeteksi dengan benar dan tingkat kematangannya sesuai dengan klasifikasi atau klasnya. Tahap evaluasi akurasi berperan penting dalam mendeteksi objek karena stabilitas pendeteksian objek memerlukan tingkat akurasi yang tinggi. Oleh karena itu, nilai akurasi pendeteksian objek harus dievaluasi agar pendeteksian objek lebih stabil pada gambar ataupun video.

Tahap input dalam penelitian ini melibatkan memasukkan gambar atau video yang akan diuji untuk mendeteksi objek dalamnya. Pada tahap hasil deteksi, gambar atau video tersebut telah melalui proses pendeteksian objek menggunakan metode *YOLOv8*. Dalam hasil akhir gambar atau video tersebut, tingkat kematangan objek telah terdeteksi dan disertai dengan nilai akurasi yang menggambarkan sejauh mana hasil deteksi dapat diandalkan.

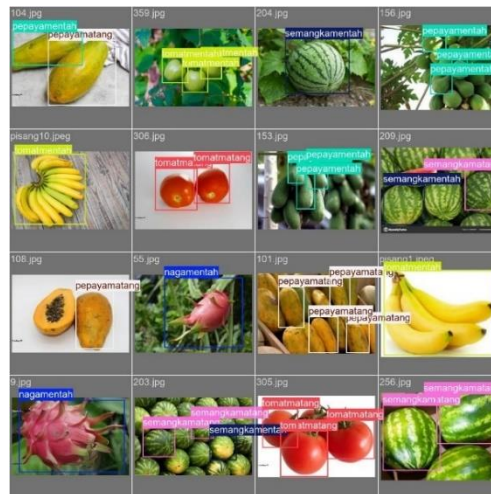


Gambar 5. Kurva hubungan antara *Precision* dengan *Recall*



Gambar 6. Kurva hubungan antara *Recall* dengan *Confidence*

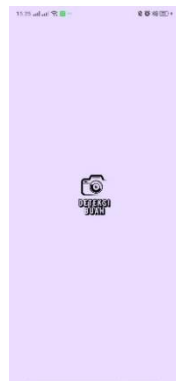
Semakin tinggi nilai recall yang dihasilkan dari proses pelatihan dan validasi data, semakin tinggi pula akurasi sistem yang telah dibuat. Hasil visualisasi pada gambar 5 dan gambar 6 memberikan penjelasan lebih lanjut tentang kinerja sistem. Selain itu, selama proses pelatihan dan pengujian dalam tiap *epoch*-nya, sistem mengalami perkembangan dan peningkatan kinerja. 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. Kekurangan ini menyebabkan pendeteksian pada buah kurang akurat dan kemungkinan kesalahan deteksi juga besar. Kurangnya keakuratan pada hasil deteksi dapat dipengaruhi oleh kekurangan variasi dalam dataset yang digunakan. Hasil pendeteksian tingkat kematangan buah serta nilai akurasi untuk setiap klas atau klasifikasinya direpresentasikan dalam bentuk gambar yang dapat dilihat dalam Gambar 7.



Gambar 7. Format Gambar Dataset

Pada gambar 7 diatas, dapat dilihat hasil dari pelatihan dataset. Setelah pelatihan model berhasil dan sudah dikonversi menjadi file *tflite*, selanjutnya sistem akan di implementasikan ke dalam *Android Studio*. Kotlin merupakan Bahasa pemrograman yang digunakan untuk membuat aplikasi, *Android Studio* digunakan sebagai *text editor*. Berikut tampilan aplikasinya:

a. Tampilan aplikasi



Gambar 8. Tampilan Awal



Gambar 9. Tampilan Kamera

Pada tampilan layar yang mirip dengan yang ada pada gambar 8 di atas 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.

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. Pengguna dapat menggunakan kamera dengan mengarahkan kamera pada buah yang ingin diketahui tingkat kematangannya.

Tabel 1. Pengujian *Black Box* Buah Pepaya

Tes Faktor	Hasil	Keterangan
Pepaya	✓	Aplikasi berhasil mendeteksi kematangan buah pepaya yang menampilkan <i>bounding box</i> dengan label pepaya matang dan mentah seperti pada gambar di bawah
<i>Screenshot aplikasi</i>		
		

Tabel 2. Pengujian *Black Box* Buah Naga

Tes Faktor	Hasil	Keterangan
Buah Naga	✓	Aplikasi berhasil mendeteksi kematangan buah naga yang menampilkan <i>bounding box</i> dengan label naga matang dan mentah seperti pada gambar di bawah
<i>Screenshot aplikasi</i>		
		

Tabel 3. Pengujian *Black Box* Buah Tomat

Tes Faktor	Hasil	Keterangan
Tomat	✓	Aplikasi berhasil mendeteksi kematangan tomat yang menampilkan <i>bounding box</i> dengan label tomat matang dan mentah seperti pada gambar di bawah
Screenshoot aplikasi		
 		

Tabel 4. Pengujian *Black Box* Buah Semangka

Tes Faktor	Hasil	Keterangan
Semangka	✓	Aplikasi berhasil mendeteksi kematangan semangka yang menampilkan <i>bounding box</i> dengan label semangka matang dan mentah seperti pada gambar di bawah
Screenshoot aplikasi		
 		

b. Pendeteksian Tidak Berhasil



Gambar 10. Pendeteksian Tidak berhasil

pada gambar 10. merupakan contoh hasil pendeteksian tidak berhasil karena mendeteksi objek selain buah namun membacanya sebagai buah papaya 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. Hasil deteksi juga dipengaruhi oleh intensitas cahaya, jadi pendeteksian hanya bias dilakukan dengan pencahayaan yang memadai.

4. KESIMPULAN

Penelitian ini menghasilkan sebuah aplikasi pendeteksi kematangan yang dapat mempermudah petani dan konsumen dalam pemilihan buah. Hasil penelitian menunjukkan bahwa akurasi pendeteksian tingkat kematangan pada saat pengujian mencapai tingkat yang kurang memuaskan, yaitu sebesar 41%. Nilai akurasi ini dipengaruhi oleh berbagai faktor, termasuk kualitas dataset, pengambilan gambar dari berbagai sudut, kualitas gambar, dan video. Penggunaan dataset yang mencakup beragam sudut pengambilan gambar dapat meningkatkan nilai akurasi secara signifikan. Hasil penelitian ini menunjukkan bahwa aplikasi ini masih membutuhkan perbaikan, oleh karena itu untuk penelitian selanjutnya disarankan untuk memperoleh nilai akurasi dan *confidence* yang lebih tinggi, dengan penggunaan gambar yang jelas, dataset yang berkualitas, dan mencakup banyak sudut. Dengan demikian, hasil pendeteksian kematangan dapat ditingkatkan untuk mendapatkan performa yang optimal.

REFERENSI

- Abiyyu, I. F., & Tawakal, H. A. (2021). Pengembangan Aplikasi Pendeteksi Kematangan Buah Melon: Studi Kasus Aplikasi Melonku. *Jurnal Informatika Terpadu*, 7(1), 27-32. <https://journal.nurulfikri.ac.id/index.php/JIT>
- Alfarizi, M. R. S., Al-farish, M. Z., Taufiqurrahman, M., Ardiansah, G., & Elgar, M. (2023). Penggunaan Python Sebagai Bahasa Pemrograman untuk Machine Learning dan Deep Learning. *Karya Ilmiah Mahasiswa Bertauhid (KARIMAH TAUHID)*, 2(1), 1–6. Karimah Tauhid, Volume 2 Nomor 1 (2023), e-ISSN 2963-590X | Alfarizi et al. <http://ejournal.unp.ac.id/index.php/voteknika/index>
- Andrekhya, M. Z., & Huda, Y. (2021). Deteksi Warna Manggis Menggunakan Pengolahan Citra dengan Opencv Python. *Voteteknika (Vocational Teknik Elektronika dan Informatika)*, 9(4), 27-33. <https://doi.org/10.24036/voteteknika.v9i4.114251>
- Aras, S., Tanra, P., & Bazhar, M. (2024). Deteksi Tingkat Kematangan Buah Tomat Menggunakan YOLOv5: Detection of Tomato Fruit Ripeness Level Using YOLOv5. *MALCOM: Indonesian Journal of Machine Learning and Computer Science*, 4(2), 623-628. <https://doi.org/10.57152/malcom.v4i2.1270>
- Basri, M., & Suwardoyo, U. (n.d.). *Menggunakan Library Tensorflow*. X(X), 1–8. <https://jurnal.umpar.ac.id/index.php/sylog>
- Dalimunthe, A. (2021). Deteksi Kematangan Buah Manggis Berdasarkan Fitur Warna Citra Kulit Menggunakan Metode Transformasi Ruang Warna Hsv (Doctoral

- dissertation, Universitas Islam Negeri Sumatera Utara Medan).
- Dhita R. L., Faulina, S. T., & Wisnumurti. (2023). Rancang Bangun Aplikasi Layanan Pengaduan Pada Dinas Pendidikan Kabupaten Oku Berbasis Android Menggunakan Android Studio. *Jik, 14Faulina,(2)*, 25–35.
- Febrywinata, E. (2024). Pengenalan Dan Klasifikasi Jenis Buah Menggunakan Metode CNN Secara Sederhana Dengan Menggunakan Google Colab. *Juli, 2(4)*, 185–193. <https://doi.org/10.61132/mercurius.v2i4.162>
- Hawibowo, M. S., & Muhimmah, I. (2024). Aplikasi Pendeteksi Tingkat Kematangan Pepaya menggunakan Metode Convolutional Neural Network (CNN) Berbasis Android. *JEPIN (Jurnal Edukasi dan Penelitian Informatika)*, 10(1), 162-170.
- 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. <https://doi.org/10.58602/jaiti.v1i3.61>
- Lusiana, L., Wibowo, A., & Dewi, T. K. (2023). Implementasi Algoritma Deep Learning (YOLOv5) Untuk Deteksi Buah Segar Dan Busuk. *Paspalum: Jurnal Ilmiah Pertanian*, 11 (1), 123-130. <http://dx.doi.org/10.35138/paspalum.v11i1.489>
- Maneno, R., Baso, B., Manek, P. G., & Fallo, K. (2023). Deteksi Tingkat Kematangan Buah Pinang Menggunakan Metode Support Vector Machine Berdasarkan Warna Dan Tekstur. *Journal of Information and Technology, 3(2)*, 60-66. <https://doi.org/10.32938/jitu.v3i2.5323>
- Raysyah, S., Arinal, V., & Mulyana, D. I. (2021). Klasifikasi tingkat kematangan buah kopi berdasarkan deteksi warna menggunakan metode knn dan pca. *JSiI (Jurnal Sistem Informasi)*, 88-95. *JSiI | Jurnal Sistem Informasi | Vol. 8 | No. 2 | September 2021 | 88-95*
- Saputra, R. A., Puspitasari, D., & Baidawi, T. (2022). Deteksi kematangan buah melon dengan algoritma support vector machine berbasis ekstraksi fitur glcm. *Jurnal Infotech, 4(2)*, 200-206. <https://journal.irpi.or.id/index.php/malcom>
- Sari, W. E., Muslimin, M., Franz, A., & Sugiartawan, P. (2022). Deteksi Tingkat Kematangan Tandan Buah Segar Kelapa Sawit dengan Algoritme K-Means. *SINTECH (Science and Information Technology) Journal, 5(2)*, 154-164. <https://doi.org/10.31598>
- Sayyidin, M., #1, H., & Muhimmah, I. (2024). Aplikasi Pendeteksi Tingkat Kematangan Pepaya menggunakan Metode Convolutional Neural Network (CNN) Berbasis Android. *Jurnal Edukasi Dan Penelitian Informatika, 10(1)*, 162–170. *JEPIN (Jurnal Edukasi dan Penelitian Informatika)*, Vol. 10, No. 1, April 2024
- Yunus, Mughaffir, and Syamsu Riyadi. 2024. "Aplikasi Deteksi Wajah Dan E-Learning Berbasis Pengenalan Untuk Otentikasi Mahasiswa." *Jurnal Sintaks Logika* 4(1): 31–35. doi:10.31850/jsilog.v4i1.2784. <https://jurnal.umpar.ac.id/index.php/sylog>
- 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
- Zainal, M., Selao, A., & Alamsyah, M. I. (n.d.). *Deteksi Objek Pada Gambar Menggunakan Algoritma Yolo (You Only Look Once)*. 1–7. <https://jurnal.umpar.ac.id/index.php/sylog>