
Klasifikasi Kematangan Pisang Menggunakan Metode *Convolutional Neural Network*

Afifah Inas Hanifah*, Arief Hermawan

Program Studi Informatika, Fakultas Sains dan Teknologi, Universitas Teknologi Yogyakarta
Jl. Siliwangi (Ringroad Utara), Jombor, Sleman, D.I. Yogyakarta 55285

*email : afifah.5200411042@student.uty.ac.id

(Naskah masuk: 15 Juni 2023; diterima untuk diterbitkan: 29 Agustus 2023)

ABSTRAK – Pisang adalah tumbuhan yang berasal dari wilayah Asia Tenggara dan termasuk dalam genus *Musa* serta family *Musaceae*. Tumbuh di wilayah tropis dan subtropis, pisang termasuk ke dalam salah satu komoditas pertanian dengan produksi paling besar dibandingkan buah lainnya. Indonesia adalah salah satu negara yang memproduksi pisang terbesar di dunia. Hasil panen tersebut kemudian dipilah berdasarkan tingkat kematangan dengan melihat dari perubahan warna kulit pisang. Namun, proses pemilahan buah pisang memerlukan waktu dan tenaga cukup besar sebab produksi buah pisang yang banyak. Selain itu, perbedaan penilaian setiap individu terhadap perubahan warna kulit pisang sehingga mengakibatkan pemilahan buah pisang yang tidak stabil atau konsisten. Maka dari itu, penelitian ini bermaksud untuk membuat sistem klasifikasi kematangan buah pisang berdasarkan perubahan warna kulit dengan tujuan proses pemilahan tersebut dapat dilakukan dengan efisien dan akurat. Varian warna yang digunakan mulai dari dominan hijau untuk pisang mentah, dominan kuning untuk pisang matang dan mulai munculnya bercak berwarna coklat kehitaman untuk pisang terlalu matang. Metode yang diterapkan adalah *Convolutional Neural Network* dengan arsitektur yang dirancang sendiri. Hasil penelitian menunjukkan akurasi mencapai 88% dengan pengaturan learning rate sebesar 0,001 dan batasan maksimal epoch sebanyak 15.

Kata Kunci – Klasifikasi; Buah Pisang; *Convolutional Neural Network*; Deep Learning; Computer Vision.

Classification of Banana Fruit Maturity Levels Using the Convolutional Neural Network Method

ABSTRACT – Bananas are plants from the Southeast Asian region and belong to the genus *Musa* and the family *Musaceae*. Grown in tropical and subtropical regions, bananas are one of the agricultural commodities with the largest production compared to other fruits. Indonesia is one of the countries that produce the largest bananas in the world. The yields are then sorted based on the level of ripeness by looking at the color change of the banana skin. However, the process of sorting bananas requires a lot of time and effort due to the large production of bananas. In addition, differences in the assessment of each individual on changes in the color of banana peels result in an unstable or consistent sorting of bananas. Therefore, this study intends to create a ripeness classification system for bananas based on changes in skin color with the aim that the sorting process can be carried out efficiently and accurately. The color variants used range from dominant green for unripe bananas, dominant yellow for ripe bananas and blackish brown spots for overripe bananas. The method used is a *Convolutional Neural Network* with a self-designed architecture. The results showed that the accuracy reached 88% with a learning rate setting of 0.001 and a maximum epoch limit of 15.

Keywords – Classification; Banana Fruit; *Convolutional Neural Network*; Deep Learning; Computer Vision.

1. PENDAHULUAN

Pisang adalah tumbuhan asli Asia Tenggara yang termasuk dalam genus *Musa* dan dapat ditemukan di daerah tropis dan subtropis. Tumbuhan pisang banyak ditemukan di Indonesia seperti negara-negara Asia Tenggara lainnya, terutama di wilayah yang mendapat banyak sinar matahari. Merujuk data BPS diketahui bahwa Indonesia terus mengalami peningkatan dalam produksi buah-buahan. Pada tahun 2021 produksinya meraih angka 25,96 juta ton atau mengalami perkembangan sebanyak 5,4% dibandingkan dengan produksi pada tahun 2020 dengan komoditas produksi terbesar yaitu pisang sebanyak 8,74 ton. Serta masyarakat Indonesia mengkonsumsi buah rata-rata mencapai 81,14 gram per kapita per hari [1]. Dalam hal ini proses pemilahan pascapanen menjadi faktor penting dalam produksi buah pisang karena proses pemilahan masih banyak dilakukan secara manual sehingga memerlukan waktu dan tenaga yang banyak. Terlebih lagi penilaian para petani pisang yang berbeda-beda terhadap tingkat kematangan pisang yang menyebabkan hasil pemilahan yang tidak konsisten. Dengan adanya teknologi dapat menjadi solusi dari masalah ini.

Penelitian tentang klasifikasi buah dengan metode *Convolutional Neural Network* (CNN) sudah cukup banyak dilakukan seperti penelitian [2] tentang klasifikasi buah mangga badami dengan data kaggle sebanyak 204 citra. Hasil penelitian tersebut mendapatkan hasil akurasi sebesar 94,6% untuk data *training* dan 97,2% untuk data *testing*. Penelitian lain tentang klasifikasi kesegaran buah [3] mendapatkan nilai akurasi 93,3%, *presicion* 93,5%, *recall* 93,34% dan *f1-score* 93,34%. Data tersebut berasal dari kaggle yang berjumlah 13599 citra. Penelitian lain tentang merancang alat pendeteksi kematangan buah nanas [4] mendapatkan hasil presentase keberhasilan klasifikasi sebesar 83,33% dengan jumlah 150 data. Namun, alat pendeteksi tidak bisa mendeteksi buah yang busuk, manis, atau segar. Penelitian [5] tentang klasifikasi tekstur kematangan buah jeruk menunjukkan hasil akurasi mencapai 96% pada *training*, 92% pada *testing* dan *epoch* sebanyak 50 dengan data berjumlah 250.

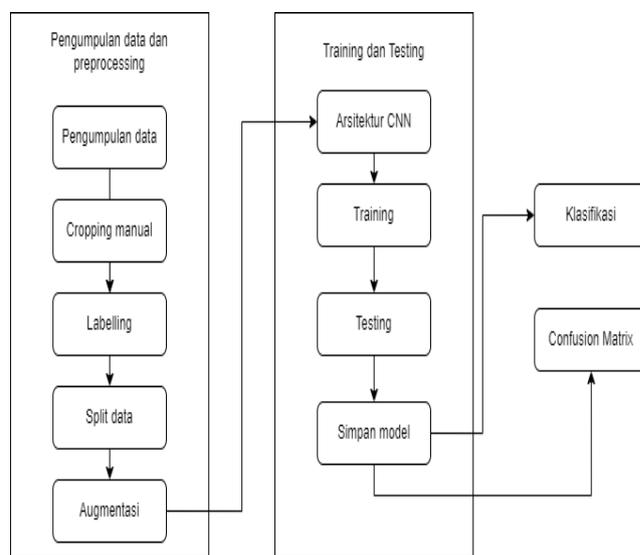
Terakhir, penelitian tentang klasifikasi kematangan pisang tanduk dengan metode HSV [6]. Penelitian tersebut menggunakan software Matlab R2021b dengan data yang diambil sendiri sebanyak 506 data latih dan 128 data uji. Kriteria tingkat kematangan dikelompokkan menjadi 4 yaitu mentah, setengah matang, matang, dan terlalu matang. Pada proses *training* menggunakan arsitektur sendiri, 200 *epoch* dan *learning rate* sebesar 0.01. Penelitian ini menghasilkan akurasi sebesar 89%. Dengan begitu, dapat diketahui bahwa penggunaan metode CNN

dalam klasifikasi citra dapat memberikan hasil akurasi yang sangat baik.

Berdasarkan permasalahan di atas, penelitian ini mengusulkan penggunaan metode CNN untuk melakukan klasifikasi kematangan buah pisang berdasarkan warna. Terdapat tiga kriteria yang digunakan yaitu mentah, matang, dan sangat matang. Perbedaan penelitian ini dengan penelitian sebelumnya yang telah disebutkan yaitu penggunaan arsitektur model yang dikembangkan sendiri dengan data sebanyak 216 citra serta penggunaan banyaknya *epoch*. Kemudian perbedaan utama dengan penelitian [6] yaitu penelitian ini hanya menggunakan citra dengan format RGB. Penelitian ini memiliki tujuan untuk mendapatkan model dengan akurasi yang baik dalam klasifikasi kematangan buah pisang sehingga dapat memberikan kontribusi dalam pengembangan teknik klasifikasi buah pisang berdasarkan warna menggunakan metode CNN.

2. METODE DAN BAHAN

Bab ini menjelaskan tentang tahap penelitian mulai dari pengumpulan dataset, *preprocessing*, *augmentasi*, *training* dan evaluasi model. Untuk lebih jelasnya pada Gambar 1 merupakan alur dari tahap penelitian.



Gambar 1 Tahap Penelitian

Pengumpulan Data

Jenis pisang yang digunakan yaitu pisang yang warna kulitnya berubah menjadi kuning ketika matang seperti pisang raja, mas dan ambon. Data yang digunakan untuk *training* dan *testing* model klasifikasi kematangan buah pisang dikumpulkan secara manual menggunakan *smartphone*. Proses pengumpulan data melibatkan pengambilan foto buah satu per satu. Jumlah total data yang berhasil dikumpulkan adalah 180 gambar dengan masing-

masing kategori kematangan matang, mentah dan sangat matang terdiri dari 60 gambar. Format citra pisang RGB dengan ukuran 200x200 *pixel*. Perubahan warna kulit pisang yang diterapkan dimulai dari dominan hijau untuk pisang mentah, dominan kuning untuk matang dan muncul bercak coklat kehitaman untuk sangat matang.

Preprocessing Data

Setelah proses pengumpulan data maka dilakukan *preprocessing* data untuk mempersiapkan gambar-gambar pisang sebelum digunakan dalam pelatihan dan pengujian model. Dalam tahap ini, terdapat beberapa gambar pisang yang mengandung kumpulan gambar pisang atau bagian *background* yang tidak diinginkan. Untuk mengatasi hal tersebut dilakukan pemotongan gambar secara manual dengan tujuan untuk memastikan bahwa gambar pisang tersebut sudah siap digunakan dalam pelatihan. Proses pemotongan gambar dilakukan untuk menghapus bagian-bagian yang tidak diinginkan dalam setiap gambar pisang. Setelah proses pemotongan gambar, jumlah total data meningkat menjadi 216 data dengan masing-masing kategori kematangan matang, mentah dan sangat matang terdiri dari 72 gambar. Selanjutnya, dilakukan pembagian atau *split* data menjadi data *train*, data *validation* dan data *test*. Dari 216 data yang tersedia, kemudian dibagi sebesar 70% untuk data *train* dan 15% untuk masing-masing data *validation* dan data *test*. Data *validation* digunakan untuk melakukan validasi model serta menghindari terjadinya *overfitting*. Pada setiap *epoch*, proses pelatihan dan validasi dilaksanakan secara berurutan. Setiap kali pelatihan selesai, proses validasi akan dijalankan.

Augmentasi Data

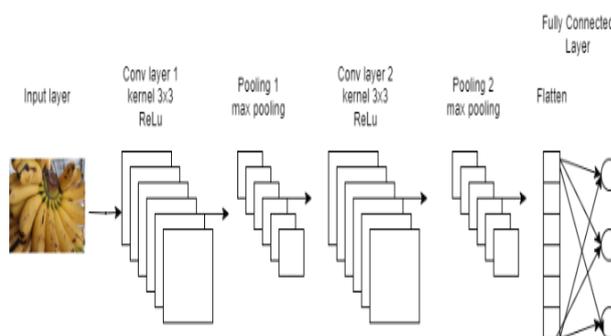
Dalam penelitian ini, karena jumlah data yang dikumpulkan masih tergolong sedikit maka dilakukan proses augmentasi data untuk meningkatkan jumlah variasi data yang ada. Augmentasi data adalah teknik yang digunakan untuk menghasilkan variasi data baru dari data yang ada dengan melakukan transformasi pada gambar-gambar tersebut. Hal ini dapat membantu meningkatkan keberagaman data yang digunakan dan mencegah *overfitting*. Proses augmentasi data dilakukan menggunakan *library ImageDataGenerator* yang merupakan bagian dari *framework keras*.

Terdapat beberapa parameter yang digunakan untuk augmentasi data. Pertama, *rescale* digunakan untuk melakukan normalisasi nilai pixel dalam gambar dengan membagi setiap pixel dengan 255. Kedua, *rotation range* untuk mengatur rentang rotasi dalam derajat. Ketiga, *width* dan *height shift range*

untuk mengatur rentang pergeseran horizontal dan vertikal dalam gambar. Keempat, *shear range* untuk mengatur rentang pergeseran sudut pada gambar. Kelima, *zoom range* untuk mengatur rentang zoom pada gambar. Keenam, *horizontal flip* untuk mengatur apakah gambar akan di-*flip* secara horizontal. Terakhir, *fill mode* untuk mengisi pixel yang kosong setelah dilakukan transformasi. Dengan menggabungkan berbagai parameter tersebut, *ImageDataGenerator* akan menghasilkan variasi gambar baru dari data yang sudah ada.

Perancangan Arsitektur Model

Pada tahap ini dilakukan perancangan arsitektur model CNN. CNN merupakan salah satu kategori jaringan saraf yang paling populer terutama untuk data dimensi tinggi seperti gambar dan video [7]. Pemilihan arsitektur yang digunakan bervariasi tergantung pada kasus yang ada. Oleh karena itu, penelitian ini telah dirancang sebuah arsitektur CNN yang cocok dengan permasalahan.



Gambar 2. Arsitektur CNN

Arsitektur CNN yang dirancang terdiri dari lapisan konvolusi, *max pooling* dan *fully connected*. Gambar 2 menunjukkan arsitektur CNN yang diterapkan. Secara keseluruhan arsitektur model CNN telah diringkas dalam Tabel 1.

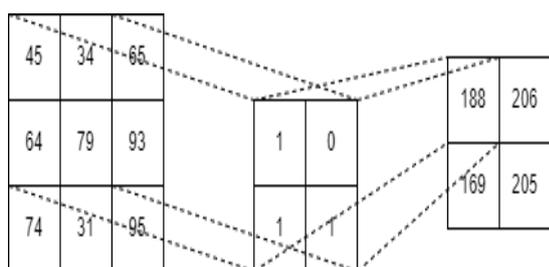
Feature maps sederhananya merupakan *output* dari proses konvolusi. Tanda (-) pada kolom *stride* berarti tidak adanya pengaturan atau masukan secara manual berapa banyak *stride* yang digunakan.

Convolution

Convolution merupakan lapisan dasar untuk melakukan proses konvolusi [8]. Dalam konvolusi diperlukan citra masukan, *filter* atau kernel, dan *feature map*. Pada pelatihan model, setiap gambar akan melalui proses ini dan fitur-fitur akan dilakukan ekstraksi. Citra akan diproses sesuai dengan parameter yang digunakan. Berikut ilustrasi dari proses *convolution* terdapat pada Gambar 3.

Tabel 1 Ringkasan Arsitektur CNN

Layer	Featur <i>e</i> Maps	Size	Kern <i>el</i> Size	Strid <i>e</i>	Activatio <i>n</i>
Input	3	200x200	-	-	-
Conv	8	200x200	3x3	1	ReLU
Poolin <i>g</i>	8	100x100	2x2	-	-
Conv	16	100x100	3x3	1	ReLU
Poolin <i>g</i>	16	50x50	2x2	-	-
Conv	32	50x50	3x3	1	ReLU
Poolin <i>g</i>	32	25x25	2x2	-	-
FC Outpu <i>t</i>	-	3	-	-	Softmax



Gambar 3. Ilustrasi Proses Konvolusi

Pada Gambar 3 menunjukkan bahwa terdapat sebuah matriks citra 3x3 melewati filter berukuran 2x2 dengan *stride* 1 kemudian menghasilkan *feature map* berukuran 2x2. Proses tersebut dapat terjadi dengan cara mengalikan setiap elemen atau nilai *pixel* dari wilayah citra dengan filter kemudian dijumlahkan. Penelitian ini menggunakan data citra RGB dengan ukuran *pixel* sebesar 200x200 dengan *stride* 1 pada layer Conv serta filter berukuran 3x3 yang sudah dijelaskan pada Tabel 1.

Fungsi Aktivasi

Setiap lapisan *convolution* menerapkan fungsi aktivasi dan menghasilkan *activation map* sebagai *output*. Fungsi aktivasi diterapkan agar model jaringan syaraf tidak memperoleh nilai berupa non-linear [9]. Beberapa fungsi yang umum digunakan meliputi *sigmoid*, *Tanh*, *softmax* dan *ReLU*. Secara umum, fungsi aktivasi *ReLU* banyak digunakan oleh para peneliti karena dianggap memiliki performa terbaik. Fungsi aktivasi *ReLU* melakukan perhitungan dengan *threshold* nol, yang berarti nilai

input akan menjadi *x* jika lebih dari nol dan nilai input akan menjadi 0 jika kurang dari nol [9].

Fungsi aktivasi *sigmoid* dan *softmax* biasanya digunakan pada lapisan akhir. Fungsi *sigmoid* berperan dalam mengubah nilai *input* menjadi rentang antara 0 dan 1. Umumnya, fungsi aktivasi *sigmoid* sering diterapkan dalam pengklasifikasian dua kelas. Sementara itu, fungsi aktivasi *softmax* digunakan lebih sering dalam klasifikasi dengan dua kelas atau lebih [9]. Oleh karena itu, penelitian ini menggunakan fungsi aktivasi *softmax* pada lapisan akhir untuk mengklasifikasikan tiga kelas. Berikut adalah persamaan dari fungsi aktivasi *softmax*.

$$S(y_i) = \frac{e^{y_i}}{\sum_j e^{y_j}} \quad (1)$$

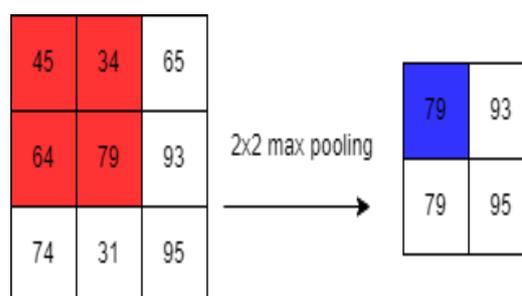
Dimana :

y = nilai input lapisan sebelumnya

i, j = indeks unit dan lapisan

Pooling

Pooling adalah salah satu teknik yang berfungsi untuk mengurangi ukuran matriks atau *feature map*. Hal ini dilakukan dengan menerapkan operasi *pooling* pada wilayah tertentu dari *feature map* untuk mendapatkan nilai yang mewakili wilayah tersebut [10]. Hasil dari proses *pooling* berupa *feature map* yang nantinya akan disambungkan ke lapisan *fully-connected*. Ukuran *pooling* yang sering digunakan berupa 2x2. Terdapat dua macam *pooling*, yaitu *max pooling* dan *average pooling* [11]. Penelitian ini menggunakan *max pooling* pada lapisan *pooling*. Cara kerja *max pooling* yaitu dengan mengambil angka *pixel* terbesar pada setiap wilayah citra sedangkan *average pooling* menghitung rata-rata dari setiap wilayah. Gambar 4 merupakan ilustrasi tentang cara kerja *max pooling*.



Gambar 4. Ilustrasi Proses Max Pooling

Seperti yang terlihat pada Gambar 4, lapisan *max pooling* berukuran 2x2 melewati citra yang ditandai dengan warna merah kemudian diambil nilai tertinggi. Hasil *max pooling* ditandai dengan warna biru.

Flatten

Lapisan *flatten* berfungsi sebagai penghubung antara lapisan konvolusi dan *fully-connected*. *Flatten*

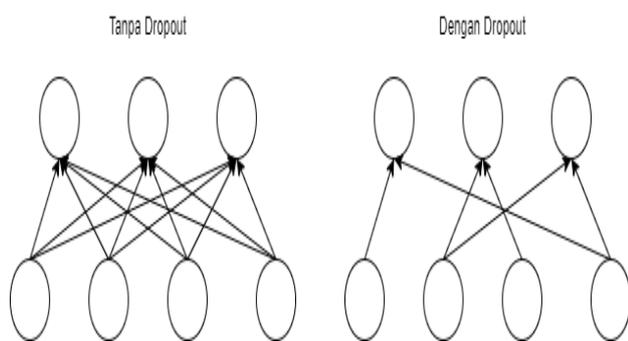
digunakan untuk mengubah *feature map* dari proses konvolusi ke dalam bentuk *array* satu dimensi [12], [13]. Dengan menggunakan *flatten*, struktur spasial dari *feature map* diubah menjadi representasi vektor yang dapat diteruskan ke lapisan *fully connected* untuk perhitungan lebih lanjut.

Fully Connected Layer

Pada lapisan *fully connected*, setiap *neuron* dari lapisan sebelumnya terkoneksi secara keseluruhan ke setiap *neuron* di lapisan berikutnya. Setiap nilai yang dihasilkan oleh *neuron* tersebut berkontribusi dalam memprediksi sejauh mana suatu nilai cocok dengan kelas tertentu. *Output* dari lapisan terakhir di *fully connected* adalah hasil klasifikasi citra *input*. Dalam konteks pengenalan citra, lapisan *fully connected* bertanggung jawab untuk menggabungkan informasi yang diperoleh dari lapisan sebelumnya dan menghasilkan prediksi akhir tentang kelas apa yang paling sesuai dengan citra yang diinputkan.

Dropout

Dropout merupakan sebuah metode regularisasi yang bertujuan untuk membantu mengurangi resiko *overfitting* dengan meningkatkan akurasi pengujian dengan mengorbankan akurasi pelatihan [14]. *Dropout* akan memilih secara acak *neuron-neuron* yang tidak akan digunakan selama proses pelatihan model. Hal ini membantu menghindari ketergantungan yang terlalu kuat antar neuron dan meningkatkan kemampuan generalisasi model. Selain itu, *dropout* akan mengurangi beban komputasi dan mempercepat proses pelatihan model. Dengan pernyataan tersebut maka penelitian ini menggunakan *dropout* dengan *rate* sebesar 0.3. Pada Gambar 5 merupakan perbedaan antara penggunaan *dropout* dan yang tidak menggunakan *dropout*.



Gambar 5. Ilustrasi Penggunaan Dropout

Evaluasi Confusion Matrix

Tahap akhir sesudah pelatihan model yaitu evaluasi. Evaluasi bertujuan untuk menilai performa model dalam melakukan klasifikasi. *Confusion matrix* merupakan sebuah metode yang berfungsi untuk mengevaluasi performa model yang mencakup nilai

akurasi, *precision*, *recall* dan *f1-score*. Perhitungan ini melibatkan empat parameter penting yaitu *true positive*, *true negative*, *false positive* dan *false negative* [15]. Tabel 2 menunjukkan perhitungan *confusion matrix*. Untuk rumus perhitungan performa dapat ditemukan dalam persamaan (1), (2), (3) dan (4).

$$accuracy = \frac{TP+TN}{TP+TN+FP+FN} \times 100\% \quad (1)$$

$$recall = \frac{TP}{FP+TP} \times 100\% \quad (2)$$

$$precision = \frac{TP}{FN+TP} \times 100\% \quad (3)$$

$$f1 - score = 2 \times \frac{recall \times precision}{recall+precision} \quad (4)$$

Dalam persamaan 1, 2, dan 3, TP mewakili jumlah prediksi benar positif, TN mewakili jumlah prediksi benar negatif, FP mewakili jumlah prediksi yang salah positif dan FN mewakili jumlah prediksi yang salah negatif. Dengan menggunakan persamaan tersebut, dapat dihitung dan dievaluasi kinerja model.

Tabel 2. Confusion Matrix

		Nilai Aktual	
		Negatif	Positif
Nilai Prediksi	Negatif	True Negative (TN)	False Positive (FP)
	Positif	False Negative (FN)	True Positive (TP)

3. HASIL DAN PEMBAHASAN

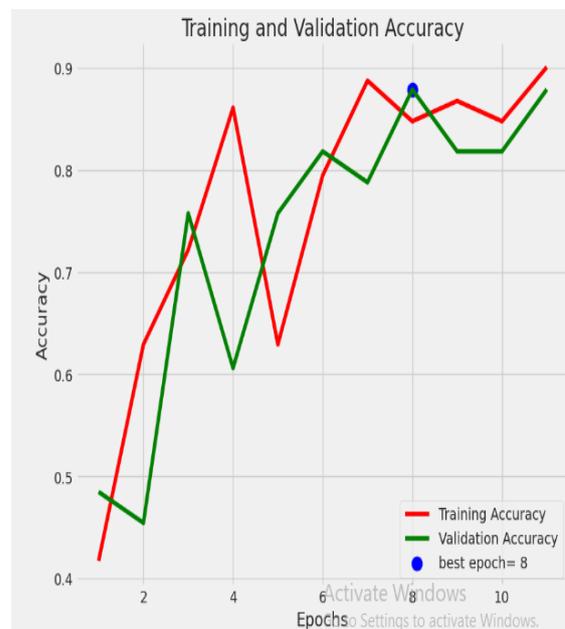
Setelah proses pelatihan selesai, model yang telah terbentuk siap digunakan untuk proses pengujian. Model yang telah dilatih akan menerima data baru sebagai *input* dan menghasilkan prediksi berdasarkan apa yang telah dipelajari selama proses pelatihan. Dengan begitu, dapat dilakukan evaluasi seberapa baik model bekerja pada data yang tidak ada dalam pelatihan. Pada setiap proses pelatihan terdapat nilai akurasi dan *loss* yang dihasilkan. *Loss* merupakan nilai yang menggambarkan sejauh mana prediksi model tersebut meleset dari nilai yang sebenarnya, sedangkan nilai akurasi menggambarkan sejauh mana prediksi model tersebut tepat. Semakin kecil nilai *loss* maka model yang dihasilkan cenderung lebih baik, sedangkan nilai akurasi semakin tinggi menunjukkan kemampuan model dalam melakukan prediksi dengan benar. Tujuan dari pembuatan model yaitu untuk menemukan bobot yang menghasilkan angka

loss yang rendah dan akurasi tinggi.

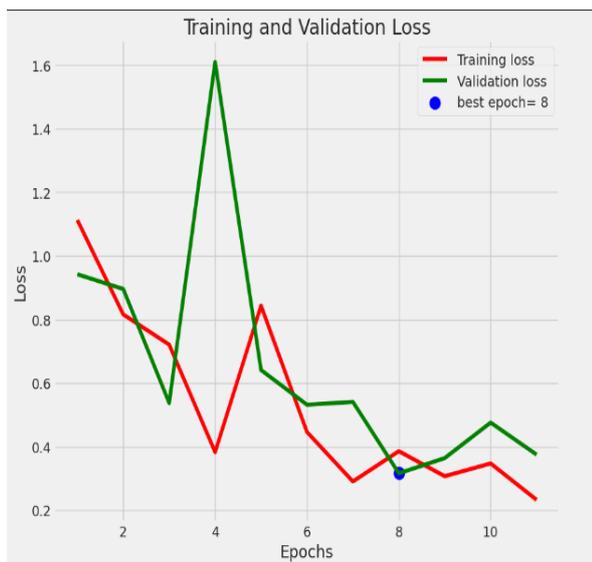
Tabel 3. Classification Report Model

	Pre cisi on	Recal l	F1- Score	Su ppo rt
Matang	0.75	0.90	0.82	10
Mentah	1.00	1.00	1.00	10
Sangat Matang	0.90	0.75	0.82	12
Accurac y Macro avg	0.88	0.88	0.88	32
Weighte d avg	0.88	0.88	0.88	32

evaluasi *classification report* terdapat pada Tabel 3.



Gambar 7 Grafik Akurasi



Gambar 6. Grafik Loss

Gambar 8 menunjukkan hasil prediksi model dibandingkan dengan label asli. Dalam kategori mentah terdapat 10 data yang diprediksi dengan benar. Sementara dalam kategori matang terdapat 9 data diprediksi dengan tepat dan hanya 1 data yang diprediksi kurang tepat. Untuk kategori sangat matang terdapat 9 data diprediksi dengan tepat dan 3 data yang diprediksi kurang tepat. Penelitian ini menggunakan parameter yang berbeda dibandingkan dengan penelitian [6].

Gambar 6 dan 7 menunjukkan hasil grafik akurasi dan *loss* selama pelatihan dengan menerapkan batasan maksimal 15 *epoch*. Dalam pelatihan juga menggunakan *library ModelCheckpoint* untuk memonitor *epoch* dengan akurasi terbaik dan *EarlyStopping*. *EarlyStopping* berfungsi untuk menghentikan pelatihan jika sudah mencapai kondisi tertentu seperti tidak adanya peningkatan akurasi dalam beberapa *epoch* terakhir.

Terlihat pada grafik 6 dan 7 *best epoch* untuk grafik *loss* terdapat pada *epoch* ke 8 dan berlaku juga untuk grafik akurasi. Selanjutnya, pengujian model yang telah dilatih menggunakan data *test*. Hasil pengujian ini berupa *classification report* yang menunjukkan tingkat akurasi sebesar 88%. Gambar 8 merupakan hasil *confusion matrix* dari evaluasi model. Hasil



Gambar 8. Confusion Matrix

Pada penelitian ini dilakukan pelatihan dengan batasan maksimal 15 *epoch*, *learning rate* sebesar 0.001 dan 32 data uji. Sementara penelitian [6] menggunakan 128 data uji, *learning rate* 0.01 dan 200 *epoch*. Terdapat perbedaan juga terdapat pada jumlah kelas kategori pada penelitian sebelumnya yang berjumlah 4 kelas. Akurasi model pada penelitian ini memiliki selisih 1 % lebih rendah dibandingkan penelitian sebelumnya yang menggunakan metode HSV dan mendapatkan akurasi sebesar 89%. Perbedaan dalam parameter dan metode yang digunakan dapat mempengaruhi performa dan akurasi model yang dihasilkan.

4. KESIMPULAN

Mengacu pada hasil penelitian dapat diambil kesimpulan bahwa akurasi model pada penelitian ini terhitung lebih rendah dibandingkan penelitian sebelumnya dan lebih tinggi dari penelitian [4]. Namun, perlu diperhatikan bahwa terdapat perbedaan signifikan dalam parameter pelatihan seperti jumlah *epoch* dan jumlah data yang digunakan. Hal ini menyebabkan pelatihan model pada penelitian ini berlangsung lebih cepat. Salah satu perbedaan atau kekurangan dari penelitian ini jika dibandingkan dengan penelitian yang menggunakan HSV adalah tidak adanya satu kategori yang dimasukkan yaitu kategori setengah matang. Selain itu terdapat perbedaan dalam jenis pisang yang digunakan. Penelitian sebelumnya fokus pada jenis pisang tanduk sedangkan penelitian ini tidak terpaku pada satu jenis pisang dikarenakan keterbatasan data yang dapat diambil. Untuk penelitian selanjutnya, disarankan untuk memperbanyak jumlah data yang digunakan, menambah kategori kelas kematangan dan melakukan klasifikasi berdasarkan jenis pisang yang berbeda. Hal tersebut akan membantu meningkatkan keberagaman data dan memperoleh pemahaman yang lebih mendalam mengenai klasifikasi pisang.

DAFTAR PUSTAKA

- [1] "Terus Dorong Peningkatan Konsumsi Buah Nusantara, Pemerintah Gelar Kembali Gelar Buah Nusantara (GBN) ke-7 Tahun 2022 - Kementerian Koordinator Bidang Perekonomian Republik Indonesia." <https://www.ekon.go.id/publikasi/detail/4450/terus-dorong-peningkatan-konsumsi-buah-nusantara-pemerintah-gelar-kembali-gelar-buah-nusantara-gbn-ke-7-tahun-2022> (diakses 15 Juni 2023).
- [2] S. A. Damayanti, A. Arkadia, dan D. S. Pravista, "Klasifikasi Buah Mangga Badami Untuk Menentukan Tingkat Kematangan dengan Metode CNN," *Prosiding Seminar Nasional Mahasiswa Ilmu Komputer dan Aplikasinya*, vol. 2, no. 2, hlm. 158-165, 2021.
- [3] F. Paraijun, R. N. Aziza dan D. Kuswardani, "Implementasi Algoritma Convolutional Neural Network Dalam Mengklasifikasi Kesegaran Buah Berdasarkan Citra Buah," *KILAT*, vol. 11, no. 1, hlm. 1-9, 2022.
- [4] Y. B. E. Purba, N. F. Saragih, S. Sitepu dan A. Gea, "Implementasi Algoritma Convolutional Neural Network Dalam Mengklasifikasi Kesegaran Buah Nanas Dengan Menggunakan Mikrokontroler Dengan Metode Convolutional Neural Network (CNN)," *Jurnal Ilmiah Teknik Informatika METHOTIKA*, vol. 2, no. 1, hlm. 13-21, 2022.
- [5] B. Yanto, L. Firmawahib, A. Supriyanto, B. H. Hayadi dan R. R. Pratama, "Klasifikasi Tekstur Kematangan Buah Jeruk Manis Berdasarkan Tingkat Kecerahan Waran dengan Metode Deep Learning Convolutional Neural Network," *INOVTEK Polberg-Seri Informatika*, vol. 6, no. 2, hlm. 259-268, 2021.
- [6] S. Mala dan Rasiban, "Klasifikasi Kematangan Buah Pisang Tanduk Berdasarkan Warna Menggunakan Metode Hue Saturation Value (HSV)," *JUPITER (Jurnal Penelitian Ilmu dan Teknik Komputer)*, vol. 1, no. 1, hlm. 197-207, 2022.
- [7] S. Khan, H. Rahmani, S. A. A. Shah, dan M. Bennamoun, *A Guide to Convolutional Neural Networks for Computer Vision*. dalam *Synthesis Lectures on Computer Vision*. Cham: Springer International Publishing, 2018. doi: 10.1007/978-3-031-01821-3.
- [8] A. Asrafil, A. Paliwang, M. R. D. Septian, M. Cahyanti dan R. Swedia, "Klasifikasi Penyakit Tanaman Apel Dari Citra Daun Dengan Convolutional Neural Network," *SEBATIK*, vol. 24, no. 2, hlm. 207-212, 2020.
- [9] Y. Rizki., R. M. Taufiq, H. Mukhtar dan D. Putri, "Klasifikasi Pola Kain Tenun Melayu Menggunakan Faster R-CNN," *IT Journal Research and Development*, vol. 5, no. 2, hlm. 215 - 225, 2021.
- [10] A. Ramdan, V. Zilvan, E. Suryawati, H. F. Pardede dan V. P. Rahadi, "Tea Clone Classification Using Deep CNN with Residual and Densely Connections," *Jurnal Teknologi dan Sistem Komputer*, vol. 8, no. 4, hlm. 289-296, 2020.
- [11] M. B. Bejiga, A. Zeggada, A. Nouffidj, dan F. Melgani, "A Convolutional Neural Network Approach for Assisting Avalanche Search and Rescue Operations with UAV Imagery," *Remote Sensing*, vol. 9, no. 2, hlm. 100, Feb 2017, doi: 10.3390/rs9020100.

- [12] A. Rosebrock, *Deep learning for computer vision with python: Starter bundle*, 1 ed. PyImageSearch, 2017.
- [13] M. L. Prasetyo dkk., "Face recognition using the convolutional neural network for Barrier Gate System," *ijIM: International Journal of Interactive Mobile Technologies*, vol. 15, no. 10, hlm. 138-153, 2021.
- [14] H. Abhirawa, J. Jondri, dan A. Arifianto, "Pengenalan Wajah Menggunakan Convolutional Neural Network," *eProceedings of Engineering*, vol. 4, no. 3, hlm. 4907-4916, 2017.
- [15] A. Kurniadi, "Implementasi Convolutional Neural Network Untuk Klasifikasi Varietas Pada Citra Daun Sawi Menggunakan Keras," *Doubleclick*, vol. 4, no. 1, hlm. 25, Agu 2020, doi: 10.25273/doubleclick.v4i1.5812.