



Aplikasi Pendeteksi Tingkat Kematangan Pepaya menggunakan Metode *Convolutional Neural Network* (CNN) Berbasis Android

Muhammad Sayyidin Hawibowo^{#1}, Izzati Muhimmah^{#2}

[#]Program Studi Informatika, Fakultas Teknik Industri Universitas Islam Indonesia
Jln. Kaliurang km 14.5, Sleman, Yogyakarta 55584

¹20523101@students.uii.ac.id

²izzati@uui.ac.id

Abstrak— Pepaya merupakan salah satu buah tropis yang memiliki nilai ekonomi tinggi dan memiliki kandungan gizi yang penting bagi kesehatan manusia. Kematangan pepaya sangat mempengaruhi rasa, tekstur, dan nilai jualnya. Oleh karena itu, pengembangan sistem otomatis untuk mendeteksi kematangan pepaya menjadi penting dalam industri pertanian. Dalam penelitian ini, dikembangkan sebuah aplikasi berbasis Android yang menggunakan metode *Convolutional Neural Network* (CNN) untuk mendeteksi kematangan pepaya secara otomatis. Dataset pepaya yang telah dikumpulkan digunakan untuk melatih model CNN guna mengklasifikasikan pepaya menjadi tiga kategori kematangan, yaitu belum matang, setengah matang, dan matang. Selain itu, aplikasi ini juga menyediakan prediksi masa panen pepaya berdasarkan analisis kematangan. Implementasi pada platform Android memungkinkan akses yang mudah dan cepat bagi petani atau pemilik kebun pepaya untuk memantau kematangan buah pepaya mereka. Hasil eksperimen menunjukkan bahwa model CNN yang diusulkan memiliki tingkat akurasi yang cukup tinggi yaitu sebesar 96,97% dalam mengklasifikasikan kematangan pepaya. Aplikasi ini diharapkan dapat membantu petani dalam mengoptimalkan proses panen dan pengelolaan pepaya secara efisien.

Kata kunci— Pepaya, Classification, Convolutional Neural Network, Android.

I. PENDAHULUAN

Industri pertanian dan perkebunan buah-buahan memegang peranan penting dalam ekonomi banyak negara di seluruh dunia. Di tengah lanskap pertanian Indonesia yang didominasi oleh iklim tropis, produk-produk pertanian memiliki potensi besar untuk pertumbuhan dan pengembangan. Di antara ragam produk hortikultura, buah-buahan dan sayur-sayuran memiliki peran yang sangat penting. Salah satu buah yang menonjol dalam industri ini adalah pepaya [1]. Pepaya (*Carica papaya L.*) merupakan sebuah jenis tanaman berbuah yang tergolong dalam *famili Caricaceae*. Tanaman ini berasal dari daerah tropis Amerika dan muncul melalui proses alami dengan perpaduan antara *Carica peltata Hook. & Arn.* Saat ini,

pepaya dapat dijumpai tumbuh secara meluas diberbagai wilayah tropis dan subtropis di seluruh dunia [2]. Menemukan buah pepaya tidaklah sulit, karena buah ini tersedia sepanjang tahun dan dapat dengan mudah ditemui di pasar. Selain itu, pepaya kaya akan nutrisi dan memberikan beragam manfaat bagi kesehatan tubuh [3]. Pepaya juga terkenal karena nilai ekonominya yang tinggi dan manfaat kesehatannya yang beragam. Buah pepaya mengandung kadar vitamin A dan vitamin C yang cukup tinggi, serta kaya akan mineral seperti kalsium, fosfor, magnesium, dan zat besi [1]. Dalam setiap 100 gramnya, pepaya memiliki kandungan sebesar 3,65 mg vitamin A dan 78 mg vitamin C [4].

Pepaya California atau Pepaya *Callina*, merupakan salah satu varietas terunggul dari pepaya. Pepaya ini terkenal akan rasa yang manis, daging buah yang padat, dan kemampuannya untuk tetap segar dalam waktu yang lama [5]. Varian ini biasanya memiliki berat sekitar 1 kilogram per buah dan berbentuk silindris. Salah satu karakteristik khas dari Pepaya California adalah warna oranye pada daging buahnya yang manis, sementara kulitnya berwarna hijau yang berubah menjadi kuning oranye ketika matang [6]. Salah satu keunggulan pepaya sebagai tanaman adalah kemampuannya untuk berbuah secara konsisten tanpa terpengaruh oleh perubahan musim, memungkinkannya untuk menghasilkan buah sepanjang tahun tanpa memperhatikan fluktuasi cuaca [1]. Menurut statistik yang dirilis oleh Badan Pusat Statistik (BPS), produksi pepaya di Indonesia pada tahun 2022 mencapai 1,05 juta ton, menunjukkan penurunan sekitar 10,41% dari tahun sebelumnya yang mencatat 1,17 juta ton. Sebelumnya, produksi pepaya di Indonesia cenderung fluktuatif dengan kecenderungan umumnya meningkat.

Proses pengendalian kualitas buah pepaya sebelum dan sesudah panen melibatkan langkah-langkah evaluasi visual yang tepat terhadap tingkat kematangan buah. Penilaian visual kematangan buah pepaya sebelum panen menjadi langkah krusial. Secara visual, perubahan warna kulit buah pepaya mencerminkan tingkat kematangannya: dari hijau

saat belum matang, menjadi bercak kuning saat setengah matang, dan oranye saat matang sepenuhnya. Ini merupakan indikator penting dalam menentukan waktu panen dan konsumsi [7]. Meskipun pengendalian kualitas dengan penilaian visual manual memakan waktu dan tenaga kerja yang besar, serta hasilnya sering bervariasi tergantung pada pengalaman individu, penting untuk mengembangkan metode otomatis yang efisien dalam mendeteksi tingkat kematangan buah pepaya. Hal ini menjadi krusial dalam sektor pertanian.

Pemodelan deteksi objek menggunakan CNN telah dilakukan untuk beberapa bidang misal pada penelitian [8] menggunakan algoritma *Convolutional Neural Network* (CNN) untuk klasifikasi jenis tanah berbasis android di Kecamatan Bawen, Jawa Tengah. Model yang dibuat memiliki akurasi *training* 97% dan akurasi *testing* 95%. Data gambar diambil dari wilayah Kecamatan Bawen dan data publik dari *website Kaggle*. Adapun pada penelitian yang dilakukan oleh Sunarti [9] penelitian ini bertujuan untuk mengklasifikasi dua jenis daun herbal, yaitu daun katuk (*sauropus androgynus*) dan daun kelor (*moringa oleifera*), menggunakan metode ekstraksi *Fourier Descriptor* dan metode klasifikasi *Naïve Bayes* dengan tiga jenis kernel (*Gaussian, Bernoulli, Multinomial*) serta *Convolutional Neural Network*. Penelitian terkait lainnya yaitu [10] dalam penelitiannya mengusulkan lima arsitektur *Convolutional Neural Network* (CNN) untuk mengidentifikasi tumor otak pada gambar MRI. Hasil penelitian menunjukkan bahwa arsitektur c memiliki akurasi tertinggi sebesar 82,2%, sedangkan CNN ResNet50 memiliki akurasi tertinggi sebesar 94,1%. Arsitektur yang diusulkan berkinerja baik dengan jumlah lapisan konvolusi yang tinggi. Terdapat perbedaan dalam tiga penelitian yang telah dilakukan. Penelitian [8] menggunakan data gambar jenis tanah untuk klasifikasi dengan akurasi tertinggi 86,67%. Sementara itu, penelitian [9] fokus pada klasifikasi citra daun menggunakan ekstraksi fitur dan metode klasifikasi *Naïve Bayes* serta CNN. Penelitian [10] membandingkan pengaruh jumlah layer konvolusi dan jenis *pooling layer* pada model CNN untuk klasifikasi citra dengan akurasi tertinggi 95%. Dengan demikian, ketiga penelitian tersebut memberikan kontribusi yang beragam dalam pengembangan teknik klasifikasi citra dan pengolahan data. Pada penelitian [11] juga menggunakan teknik *Deep Learning*, dimana model *Convolutional Neural Network* (CNN) digunakan untuk ekstraksi citra pada gambar dua dimensi. Data citra yang digunakan meliputi motif batik corak insang, dayak, ikat celup, dan megamendung, dengan total 1320 data latih, 80 data validasi, dan 120 data uji. Dalam proses klasifikasi, jika salah satu kelas mencapai persentase 70% (*passing grade*), maka objek tersebut diklasifikasikan ke dalam kelas tersebut; jika tidak, maka dikategorikan sebagai "Objek Lainnya". Hasil akhir penelitian menunjukkan bahwa Aplikasi Klasifikasi Motif Batik berbasis Android berjalan sesuai harapan, baik dalam fungsionalitas aplikasi maupun proses klasifikasinya.

Salah satu solusi yang muncul adalah pemanfaatan *Convolutional Neural Networks* (CNN) dalam aplikasi Android. CNN adalah algoritma *deep learning* yang efektif dalam pengenalan gambar dan dapat mengenali pola visual pada buah pepaya yang menunjukkan tanda-tanda kematangan. Jaringan ini dirancang khusus untuk memproses citra sebagai masukan, dengan lapisan konvolusi sebagai komponen utamanya. Pendekatan ini meningkatkan efisiensi pembelajaran citra dan mempermudah aplikasinya. CNN dikenal memiliki tingkat akurasi tinggi dalam pemrosesan data citra [12]. Penerapan aplikasi Android yang menggunakan metode CNN untuk mendeteksi kematangan buah pepaya berpotensi untuk meningkatkan efisiensi dalam industri pertanian dengan mengoptimalkan proses pengendalian kualitas.

Dengan dampak positifnya terhadap industri pertanian, aplikasi yang dapat mendeteksi kematangan buah pepaya diharapkan dapat meningkatkan kualitas hasil panen, mengurangi pemborosan makanan, dan mendukung keberlanjutan sektor pertanian secara keseluruhan. Selain itu, aplikasi ini memberikan manfaat berupa penghematan waktu dan sumber daya bagi petani, memungkinkan mereka untuk fokus pada aspek-aspek lain dari pertanian seperti perawatan tanaman, irigasi, dan manajemen lahan. Tidak hanya fitur pendeteksi kematangan, fitur pengingat dan prediksi yang ada dalam aplikasi juga akan membantu petani untuk mengetahui waktu panen yang tepat. Inovasi ini adalah contoh konkret bagaimana teknologi dapat memberikan kontribusi besar dalam meningkatkan produktivitas dan keberlanjutan sektor pertanian.

II. PENELITIAN YANG TERKAIT

Beberapa penelitian sebelumnya diantaranya adalah penelitian yang dilakukan oleh Agustina [1] mengembangkan metode deteksi kematangan buah pepaya menggunakan algoritma YOLO (*You Only Look Once*). Tahapannya meliputi pengumpulan citra pepaya, pemberian label, dan pelatihan model. Hasil penelitian menunjukkan tingkat keberhasilan yang tinggi, dengan 27 dari 30 data berhasil terdeteksi dengan benar, memberikan akurasi sebesar 93%. Meskipun demikian, penelitian mengidentifikasi beberapa kendala seperti pencahayaan yang kurang optimal dan sudut kamera yang miring. Secara keseluruhan, penelitian ini memberikan kontribusi dalam pengembangan metode deteksi kematangan buah pepaya dengan YOLO, meskipun masih membutuhkan analisis lebih mendalam dan perbandingan dengan metode lain untuk memperbaiki keterbatasan yang ada. Penelitian yang dilakukan oleh Masturoh [13] mengembangkan metode klasifikasi menggunakan *Convolutional Neural Network* (CNN) untuk mengidentifikasi tingkat kematangan buah pepaya. Dengan menggunakan dataset berisi 1.500 gambar buah pepaya yang sudah di petik diambil melalui ponsel, penulis mengimplementasikan dan melatih model CNN. Evaluasi kinerja metode klasifikasi dilakukan dengan berbagai metrik seperti *confusion matrix, precision, recall, accuracy*, dan *kappa score*. Hasilnya menunjukkan bahwa model CNN mencapai akurasi sebesar 97,69%,

menunjukkan efektivitas dalam mengklasifikasikan tingkat kematangan buah pepaya secara akurat.

Penelitian yang dilakukan oleh Aminudin [7] menggunakan metode *K-Nearest Neighbor* (KNN) untuk mendeteksi tingkat kematangan pepaya jenis Bangkok berdasarkan warna kulit buah. Menggunakan 12 citra latih dan 12 citra uji, penelitian ini menggunakan ekstraksi statistika warna seperti rerata *red*, rerata *green*, standar deviasi *red*, dan lainnya. Hasilnya menunjukkan akurasi sebesar 75% dengan $K=5$ dan 75% juga dengan $K=7$. Dalam pengujian, 4 data uji diidentifikasi sebagai buah pepaya mentah dan 3 data uji sebagai setengah matang. Penelitian ini dapat diperluas dengan lebih banyak dataset untuk meningkatkan konsistensi dan performa model. Sedangkan pada penelitian oleh Hossen [14] berfokus pada klasifikasi dan pengenalan penyakit pepaya menggunakan teknik *deep learning*, khususnya model CNN yang diimplementasikan dengan modul *Keras API*. Dengan mengumpulkan dataset gambar penyakit pepaya dan melakukan preprocessing, penelitian ini mencapai akurasi 91% dalam mendeteksi dan mengklasifikasikan penyakit pepaya. Model CNN juga menunjukkan akurasi, *recall*, presisi, dan skor F1 sebesar 0,909. Dibandingkan dengan teknik lain seperti *K-means clustering* dan *random forests*, model CNN memberikan hasil yang lebih baik dengan akurasi rata-rata 91%. Penelitian ini memberikan wawasan penting tentang potensi model CNN dalam identifikasi penyakit pepaya dengan akurasi tinggi.

Penelitian selanjutnya oleh Matsuroh [15] mengembangkan sistem komputer yang dapat mendeteksi kematangan buah jeruk berdasarkan tingkat kecerahan warna. Dalam penelitian ini, langkah-langkahnya meliputi identifikasi masalah klasifikasi jeruk manis, pengumpulan dataset 100 gambar jeruk berkualitas dan tidak berkualitas, pelatihan model CNN, dan pengujian model dengan 10 citra. Hasilnya menunjukkan akurasi klasifikasi sebesar 97,52% pada dataset 100 gambar dan 92% pada pengujian dengan 10 citra. Meskipun demikian, penelitian ini mungkin perlu diperluas dan diuji lebih lanjut dengan dataset yang lebih besar dan variasi citra yang lebih luas untuk memastikan konsistensi dan kemampuan model dalam situasi dunia nyata. Dalam penelitian yang dilakukan oleh Damayanti [16] juga bertujuan untuk mengembangkan sebuah program yang dapat mengidentifikasi tingkat kematangan buah mangga Badami berdasarkan warnanya menggunakan metode *Convolutional Neural Networks* (CNN) dalam pemrosesan citra digital. Tujuan utamanya adalah memastikan bahwa mangga yang akan dikonsumsi telah mencapai kematangan yang optimal. Pengujian dilakukan menggunakan 25 citra sebagai data uji dan 179 citra sebagai data latih dari total 204 citra yang tersedia, dengan hasil akurasi model mencapai 97,2%. Namun, menurut penilaian saya, distribusi citra yang tidak seimbang antara mangga Badami yang busuk, mentah, dan matang bisa memengaruhi kinerja model. Misalnya, terdapat 35 citra mangga Badami busuk, 75 citra mentah, dan 94 citra matang. Sebaiknya, distribusi

ini disetarakan untuk melatih model agar dapat memberikan hasil yang lebih akurat.

III. METODE PENELITIAN

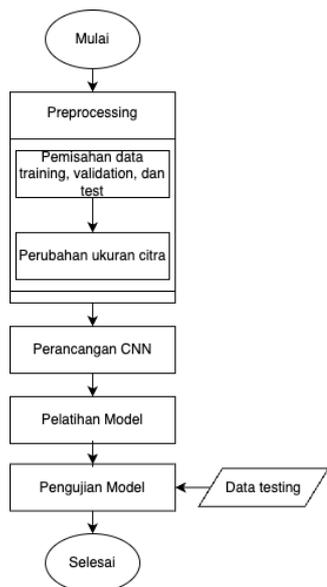
A. Pengumpulan Data

Pengumpulan data atau citra pepaya dilakukan secara langsung di kebun petani pepaya yang terletak di wilayah Sleman, Yogyakarta. Proses pengambilan gambar dilakukan menggunakan kamera ponsel dengan resolusi 13MP. Teknik pengambilan gambar dilakukan dengan menempatkan kamera smartphone pada jarak sekitar 10 cm sejajar dari buah pepaya yang terdapat pada pohonnya. Selama proses pengumpulan data, citra pepaya diambil dalam tiga variasi tingkat kematangan, yakni setengah matang, matang, dan mentah, untuk memperoleh representasi yang komprehensif dari berbagai kondisi kematangan buah pepaya. Proses pengambilan data ini memastikan bahwa dataset yang terkumpul mencakup variasi yang cukup untuk pengembangan model klasifikasi atau analisis yang akurat terkait dengan kematangan buah pepaya. Sehingga data pada penelitian ini menggunakan data primer yang didapatkan melalui foto digital. Jumlah dataset dalam penelitian ini seluruhnya berjumlah 315 data yang telah dipisahkan berdasarkan tingkat kematangan. Penentuan tingkat kematangan dilakukan juga oleh petani pepaya kebun *ken-ken farm* yang berada di Sanggragan, Sleman mempunyai pengalaman kerja sejak tahun 2019.

B. Pembuatan Model Convolutional Neural Network

Pemilihan *Convolutional Neural Network* (CNN) sebagai metode dalam penelitian ini didasarkan pada temuan dari studi sebelumnya yang menunjukkan bahwa CNN mampu mengklasifikasikan data berupa gambar dengan tingkat akurasi yang tinggi [12]. Keunggulan utama CNN terletak pada kemampuannya untuk secara efektif mengekstraksi fitur-fitur penting dari gambar secara otomatis, sehingga sangat cocok untuk tugas-tugas klasifikasi citra seperti yang ingin dicapai dalam penelitian ini. Selain itu, metode CNN telah terbukti berhasil dalam berbagai aplikasi pengenalan pola dan visi komputer, sehingga menjadi pilihan yang tepat dalam mengatasi permasalahan klasifikasi kematangan buah pepaya. Model dalam penelitian ini dievaluasi menggunakan metode data testing yang terpisah dari data latih dan validasi. Data testing digunakan untuk mengevaluasi kinerja model yang telah dibangun dengan mengukur *loss* dan akurasi pada dataset yang tidak pernah dilihat oleh model sebelumnya. Hasil evaluasi ini memberikan gambaran tentang seberapa baik model mampu melakukan klasifikasi pada data baru yang belum pernah diproses sebelumnya, sehingga memberikan ukuran objektif terhadap kemampuan generalisasi model. Arsitektur CNN yang digunakan dalam penelitian ini terdiri dari beberapa *layer* yang tersusun secara berurutan. Pertama, terdapat *layer Rescaling* yang berfungsi untuk melakukan normalisasi skala pada nilai-nilai piksel gambar, membantu proses pelatihan. Selanjutnya, terdapat tiga *layer Conv2D* yang

menggunakan *kernel* 3x3 untuk mengekstraksi fitur-fitur gambar. Setelah setiap *layer* konvolusi, dilakukan operasi *MaxPooling2D* dengan menggunakan filter 2x2 untuk mengurangi dimensi *spasial output*. Hasilnya kemudian diubah menjadi vektor satu dimensi menggunakan *layer Flatten*, sehingga siap untuk diproses lebih lanjut. Terakhir, terdapat dua *layer Dense* yang bertujuan untuk melakukan klasifikasi, dengan jumlah *neuron* 128 pada layer pertama dan 3 neuron pada layer terakhir, sesuai dengan jumlah kelas yang akan diklasifikasikan.



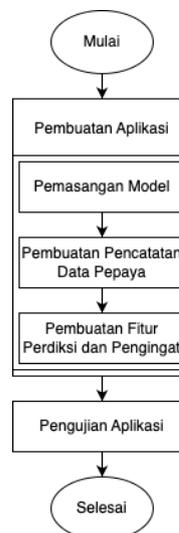
Gambar. 1 Alur pembuatan model CNN

Pada Gambar 1 terdapat alur pembuatan model CNN yang dimulai dengan proses *preprocessing*. Dalam proses ini, data dipisahkan menjadi data pelatihan, validasi, dan pengujian menggunakan kode *Python*. Pemisahan dilakukan secara acak dengan proporsi 70% untuk pelatihan model, 10% untuk validasi, dan 20% untuk pengujian. Selain pemisahan data, citra juga diubah ukurannya agar memiliki ukuran yang seragam untuk pelatihan model. Setelah tahap *preprocessing* selesai, langkah selanjutnya adalah merancang arsitektur CNN. Setelah itu, model dilatih menggunakan data pelatihan, dan akhirnya diuji menggunakan data pengujian.

C. Pembuatan Aplikasi Android

Setelah pembuatan model selesai, langkah selanjutnya adalah mengembangkan aplikasi Android. Model CNN yang telah dibuat akan digunakan oleh aplikasi untuk mendeteksi tingkat kematangan pepaya. Untuk memasukkan model ke dalam aplikasi, perlu mengonversi format model menjadi format *TensorFlow Lite* agar dapat diintegrasikan dengan baik. Setelah berhasil memasang model, langkah berikutnya adalah membuat fitur prediksi tingkat kematangan berdasarkan perhitungan hari menurut jurnal [17]. Kemudian, dilanjutkan dengan pembuatan fitur pencatatan data pepaya, sehingga pengguna dapat melihat jumlah pepaya dengan tingkat kematangan tertentu di

kebun mereka. Terakhir, aplikasi akan dilengkapi dengan fitur notifikasi atau alarm yang dapat digunakan sebagai pengingat bagi pengguna. Alur pembuatan aplikasi android dapat dilihat pada Gambar 2.



Gambar. 1 Alur pembuatan aplikasi android

IV. HASIL DAN PEMBAHASAN

Dalam penelitian ini, sebuah pendekatan diimplementasikan untuk mengklasifikasikan tiga tingkat kematangan buah pepaya, yaitu matang, mentah, dan setengah matang, menggunakan metode *Convolutional Neural Network (CNN)*. Dataset yang digunakan akan dibagi menjadi tiga bagian: data training, data validasi, dan data testing. Data training dan validasi akan digunakan dalam proses pengembangan model, sementara data testing akan digunakan untuk mengevaluasi kinerja model yang telah dibangun. Sebagai parameter utama dalam menilai keberhasilan model, digunakan nilai akurasi. Setelah pembuatan model selesai dilanjutkan dengan pembuatan aplikasi android beserta fitur-fitur yang ada didalamnya.

Proses awal penelitian melibatkan pengelompokan dataset yang sudah dikategorikan sesuai tingkat kematangannya. Dataset ini dibagi menjadi tiga bagian utama, yaitu data latih (*train*), data validasi (*validation*), dan data uji (*test*). Penggunaan proporsi yang tepat antara data latih, validasi, dan uji dilakukan untuk memastikan keseimbangan dan keberagaman dalam pembelajaran model. Pembagian dataset ini dilakukan menggunakan kode *python* untuk memisahkan data secara random berdasarkan kategori kematangannya.

Jumlah materi matang	: 104
Jumlah train matang	: 72
Jumlah val matang	: 10
Jumlah test matang	: 22
Jumlah materi setengah matang	: 102
Jumlah train setengah matang	: 71
Jumlah val setengah matang	: 10
Jumlah test setengah matang	: 21
Jumlah materi mentah	: 109
Jumlah train mentah	: 76
Jumlah val mentah	: 10
Jumlah test mentah	: 23

Gambar. 3 Hasil pembagian data set

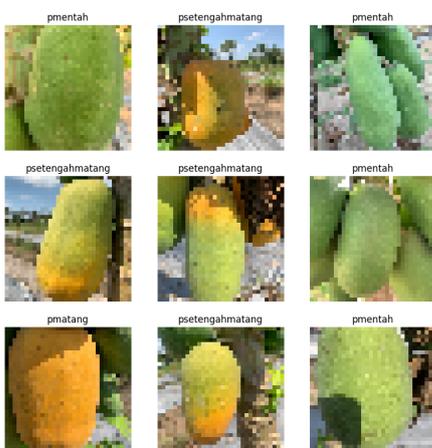
Pada Gambar 3 terlihat hasil pembagian data yang dilakukan secara proporsional, di mana sebagian besar data (70%) digunakan untuk pelatihan model, sementara sebagian kecil dialokasikan untuk validasi (10%) dan pengujian (20%). Proses pembagian data ini penting untuk mencegah *overfitting* dan memastikan keberagaman representasi data dalam setiap tahapan pembelajaran.

```
img_height, img_width = 32, 32
batch_size = 20

train_ds = tf.keras.utils.image_dataset_from_directory(
    train_dir,
    image_size = (img_height, img_width),
    batch_size = batch_size
)
val_ds = tf.keras.utils.image_dataset_from_directory(
    validation_dir,
    image_size = (img_height, img_width),
    batch_size = batch_size
)
test_ds = tf.keras.utils.image_dataset_from_directory(
    test_dir,
    image_size = (img_height, img_width),
    batch_size = batch_size
)
```

Gambar. 4 Pengolahan dan perubahan ukuran citra

Pada Gambar 4 pengolahan citra dilakukan menggunakan *TensorFlow*, sebuah *framework* yang kuat untuk pembelajaran mesin. Gambar-gambar buah dalam dataset diproses menjadi format yang dapat diakses oleh model, dengan mengubah resolusi gambar menjadi 32x32 piksel untuk mempercepat proses pembelajaran. Selanjutnya ukuran *batch* yang akan digunakan dalam pelatihan (*batch_size* = 20). Ini adalah parameter yang akan mempengaruhi seberapa banyak data gambar yang dimuat dalam setiap iterasi saat melatih model. Proses ini melibatkan memuat gambar-gambar dari direktori yang ditentukan dan mengelompokkannya menjadi *batch*. Parameter-parameter yang diberikan termasuk direktori untuk setiap set data (pelatihan, validasi, dan pengujian), ukuran gambar yang diinginkan (dalam hal ini, 32x32 piksel), dan jumlah sampel gambar dalam setiap batch. Dataset untuk masing-masing set kemudian disimpan dalam variabel *train_ds*, *val_ds*, dan *test_ds*.



Gambar. 5 Visualisasi data

Pada Gambar 5 dilakukan visualisasi data untuk memahami representasi gambar dalam dataset. Melalui plot gambar-gambar buah dari dataset latih, penelitian ini memberikan gambaran tentang keberagaman gambar yang

digunakan untuk melatih model. Dengan demikian, tahap eksplorasi data ini membantu peneliti memahami karakteristik dataset dan mempersiapkan pemodelan dengan lebih baik.

```
# Define the input shape
input_shape = (None, 32, 32, 3)

# Define the model architecture
model = tf.keras.Sequential([
    tf.keras.layers.Rescaling(1./255, input_shape=input_shape[1:]),
    tf.keras.layers.Conv2D(32, 3, activation="relu"),
    tf.keras.layers.MaxPooling2D(),
    tf.keras.layers.Conv2D(32, 3, activation="relu"),
    tf.keras.layers.MaxPooling2D(),
    tf.keras.layers.Conv2D(32, 3, activation="relu"),
    tf.keras.layers.MaxPooling2D(),
    tf.keras.layers.Flatten(),
    tf.keras.layers.Dense(128, activation="relu"),
    tf.keras.layers.Dense(3)
])

# Build the model with the specified input shape
model.build(input_shape)

# Print the model summary
model.summary()
```

Gambar. 6 Pembuatan model

Pada Gambar 6, didefinisikan arsitektur model jaringan saraf konvolusi (CNN) untuk mengklasifikasikan gambar-gambar dalam dataset. Dimulai dengan mendefinisikan bentuk input yang diharapkan untuk model, yaitu gambar-gambar berukuran 32x32 piksel dengan 3 saluran warna (RGB). Selanjutnya, menentukan arsitektur model dengan menggunakan beberapa layer yang diatur secara berurutan. Ini termasuk *layer-layer konvolusi* untuk mengekstraksi fitur-fitur gambar, *layer max pooling* untuk mengurangi dimensi spasial *output*, serta *layer-layer fully connected* untuk melakukan klasifikasi. Setelah model didefinisikan, kemudian dibangun menggunakan bentuk input yang telah ditentukan sebelumnya dan mencetak ringkasan dari arsitektur model tersebut. Dengan langkah-langkah ini, model siap untuk dikompilasi, dilatih, dan dievaluasi untuk menghasilkan kemampuan klasifikasi yang diinginkan.

Model: "sequential"

Layer (type)	Output Shape	Param #
rescaling (Rescaling)	(None, 32, 32, 3)	0
conv2d (Conv2D)	(None, 30, 30, 32)	896
max_pooling2d (MaxPooling2D)	(None, 15, 15, 32)	0
conv2d_1 (Conv2D)	(None, 13, 13, 32)	9248
max_pooling2d_1 (MaxPooling2D)	(None, 6, 6, 32)	0
conv2d_2 (Conv2D)	(None, 4, 4, 32)	9248
max_pooling2d_2 (MaxPooling2D)	(None, 2, 2, 32)	0
flatten (Flatten)	(None, 128)	0
dense (Dense)	(None, 128)	16512
dense_1 (Dense)	(None, 3)	387

=====
 Total params: 36291 (141.76 KB)
 Trainable params: 36291 (141.76 KB)
 Non-trainable params: 0 (0.00 Byte)

Gambar. 7 Arsitektur CNN

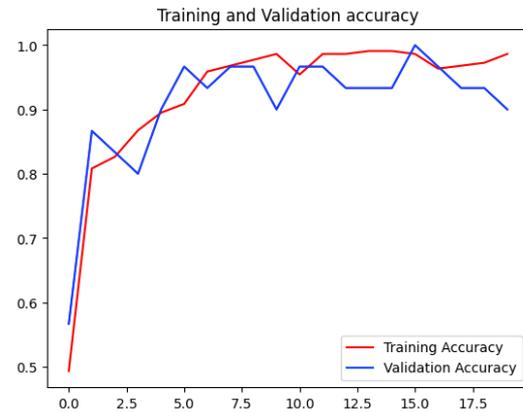
Pada Gambar 7 menunjukkan sebuah model jaringan saraf yang dirancang dengan arsitektur sequential untuk melakukan klasifikasi gambar. Dalam model ini, terdapat beberapa layer yang disusun secara berurutan. Pertama, terdapat layer Rescaling yang bertujuan untuk melakukan normalisasi skala pada nilai-nilai piksel gambar, sehingga memudahkan proses pelatihan. Kemudian, terdapat tiga layer Conv2D yang masing-masing menggunakan kernel 3x3 untuk mengekstraksi fitur-fitur gambar. Setelah setiap layer konvolusi, dilakukan operasi *max pooling* dengan menggunakan filter 2x2 untuk mengurangi dimensi spasial *output*. Selanjutnya, hasilnya diubah menjadi vektor satu dimensi menggunakan *layer Flatten*. Terakhir, terdapat dua *layer Dense* yang bertujuan untuk melakukan klasifikasi, dengan jumlah *neuron* 128 pada layer pertama dan 3 *neuron* pada layer terakhir, sesuai dengan jumlah kelas yang akan diklasifikasikan. Total parameter yang dapat di-train dalam model ini adalah 36,291, dan keseluruhan model memiliki ukuran memori sekitar 141.76 *kilobyte*. Semua parameter dalam model dapat di-train, karena menggunakan lapisan-lapisan yang dapat di-train seperti lapisan konvolusi dan *dense*. Dengan demikian, model ini siap untuk dikompilasi, dilatih, dan dievaluasi untuk melakukan klasifikasi gambar sesuai dengan tujuannya.

```
Epoch 1/20 [====] - 21s 2s/step - loss: 1.0424 - accuracy: 0.4932 - val_loss: 0.9313 - val_accuracy: 0.5667
Epoch 2/20 [====] - 7s 159ms/step - loss: 0.6834 - accuracy: 0.8082 - val_loss: 0.4378 - val_accuracy: 0.8667
Epoch 3/20 [====] - 6s 246ms/step - loss: 0.3943 - accuracy: 0.8265 - val_loss: 0.4281 - val_accuracy: 0.8333
Epoch 4/20 [====] - 5s 168ms/step - loss: 0.2823 - accuracy: 0.8676 - val_loss: 0.4667 - val_accuracy: 0.8000
Epoch 5/20 [====] - 7s 209ms/step - loss: 0.2300 - accuracy: 0.8950 - val_loss: 0.3365 - val_accuracy: 0.9000
Epoch 6/20 [====] - 5s 158ms/step - loss: 0.1895 - accuracy: 0.9087 - val_loss: 0.1379 - val_accuracy: 0.9667
Epoch 7/20 [====] - 5s 155ms/step - loss: 0.1435 - accuracy: 0.9089 - val_loss: 0.1272 - val_accuracy: 0.9333
Epoch 8/20 [====] - 5s 156ms/step - loss: 0.1018 - accuracy: 0.9600 - val_loss: 0.1156 - val_accuracy: 0.9667
Epoch 9/20 [====] - 6s 228ms/step - loss: 0.0943 - accuracy: 0.9772 - val_loss: 0.0954 - val_accuracy: 0.9667
Epoch 10/20 [====] - 5s 162ms/step - loss: 0.0690 - accuracy: 0.9863 - val_loss: 0.2326 - val_accuracy: 0.9000
Epoch 11/20 [====] - 6s 228ms/step - loss: 0.0996 - accuracy: 0.9543 - val_loss: 0.1454 - val_accuracy: 0.9667
Epoch 12/20 [====] - 6s 221ms/step - loss: 0.0541 - accuracy: 0.9863 - val_loss: 0.0677 - val_accuracy: 0.9667
Epoch 13/20 [====] - 5s 169ms/step - loss: 0.0621 - accuracy: 0.9863 - val_loss: 0.1800 - val_accuracy: 0.9333
Epoch 14/20 [====] - 6s 220ms/step - loss: 0.0434 - accuracy: 0.9909 - val_loss: 0.1565 - val_accuracy: 0.9333
Epoch 15/20 [====] - 7s 236ms/step - loss: 0.0439 - accuracy: 0.9909 - val_loss: 0.1526 - val_accuracy: 0.9333
Epoch 16/20 [====] - 5s 163ms/step - loss: 0.0389 - accuracy: 0.9863 - val_loss: 0.0295 - val_accuracy: 1.0000
Epoch 17/20 [====] - 7s 159ms/step - loss: 0.0826 - accuracy: 0.9625 - val_loss: 0.0414 - val_accuracy: 0.9667
Epoch 18/20 [====] - 6s 264ms/step - loss: 0.0705 - accuracy: 0.9600 - val_loss: 0.1401 - val_accuracy: 0.9333
Epoch 19/20 [====] - 6s 222ms/step - loss: 0.0719 - accuracy: 0.9726 - val_loss: 0.1176 - val_accuracy: 0.9333
Epoch 20/20 [====] - 6s 224ms/step - loss: 0.0525 - accuracy: 0.9863 - val_loss: 0.4452 - val_accuracy: 0.9000
```

Gambar. 8 Proses pelatihan model

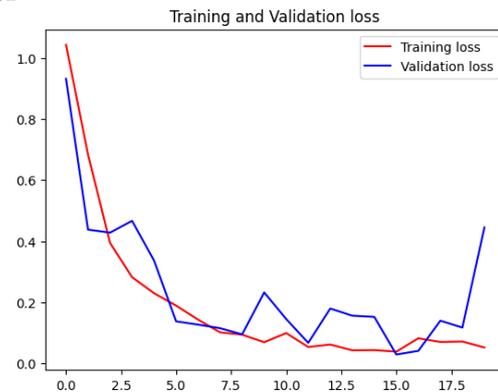
Pada tahap pelatihan model Gambar 8, dilakukan penggunaan 20 *epoch* dengan data latih dan validasi yang telah disiapkan sebelumnya. Proses pelatihan ini memungkinkan model untuk belajar dari data latih dan meningkatkan kinerjanya dalam mengklasifikasikan gambar. Setiap *epoch* mencakup iterasi di mana model mengevaluasi dan menyesuaikan dirinya berdasarkan *gradien loss function*. Hasil pelatihan menunjukkan bahwa pada *epoch* pertama, *loss* pada data latih adalah 1.0424 dengan akurasi sebesar 49.32%, sedangkan *loss* pada data validasi adalah 0.9313 dengan akurasi sebesar 56.67%. Selama 20 *epoch*, model mengalami peningkatan kinerja secara bertahap, di mana akurasi pada data latih meningkat dari 49.32% menjadi 98.63%, dan akurasi pada data validasi meningkat dari 56.67% menjadi 90.00%. Hal ini menunjukkan bahwa model berhasil belajar dari data latih dan mampu menggeneralisasi pengetahuannya pada data baru yang belum pernah dilihat sebelumnya, seperti yang ditunjukkan oleh akurasi yang konsisten pada data validasi.

Proses pelatihan ini merupakan langkah penting dalam membangun model yang dapat diandalkan dalam melakukan klasifikasi gambar dengan akurasi yang tinggi.



Gambar. 9 Grafik akurasi pelatihan dan validasi

Dari Gambar 9, kita bisa melihat bahwa akurasi pelatihan (digambarkan dengan garis merah) dan akurasi validasi (digambarkan dengan garis biru) keduanya meningkat selama periode awal pelatihan. Akurasi pelatihan nampaknya sedikit lebih tinggi daripada akurasi validasi sepanjang proses, yang merupakan indikasi biasa dari *overfitting*, namun perbedaannya tidak signifikan, yang menunjukkan bahwa model cukup baik dalam menggeneralisasi



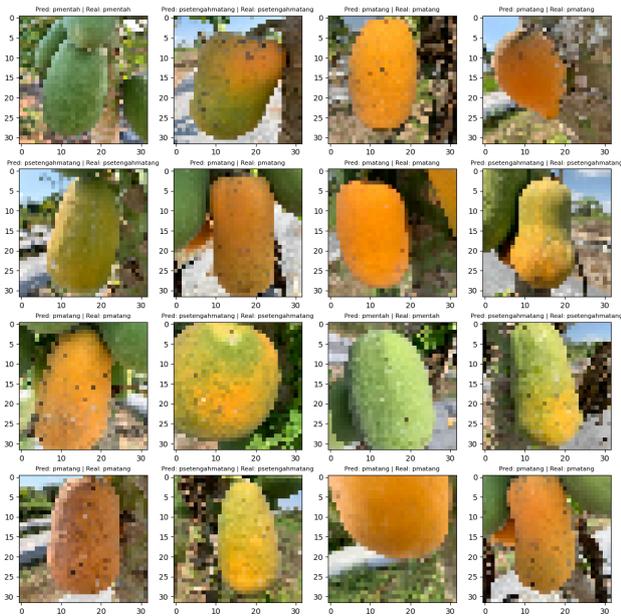
Gambar. 10 Grafik kerugian pelatihan dan validasi

Gambar 10 ini menggambarkan kerugian pelatihan (*training loss*) dan kerugian validasi (*validation loss*) dari sebuah model *machine learning* selama proses pelatihan. Garis merah mewakili kerugian pelatihan, dan garis biru mewakili kerugian validasi. Keduanya menurun tajam pada awal pelatihan, menunjukkan bahwa model dengan cepat belajar dari data. Setelah penurunan awal ini, kerugian pelatihan dan validasi berfluktuasi namun secara umum terus menurun, menandakan perbaikan bertahap dalam prediksi model.

```
model.evaluate(test_ds)
4/4 [====] - 14s 7ms/step - loss: 0.0932 - accuracy: 0.9697
[0.09320225566625595, 0.9696969985961914]
```

Gambar. 11 Evaluasi menggunakan data testing

Pada Gambar 11 model dievaluasi menggunakan data *test* yang telah dipersiapkan sebelumnya. Hasil evaluasi menunjukkan bahwa model memiliki *loss* sebesar 0.0932 dan akurasi sebesar 96.97% saat diterapkan pada data *test*. Hasil evaluasi ini mengindikasikan bahwa model yang telah dilatih mampu melakukan klasifikasi gambar dengan tingkat akurasi yang tinggi, mendekati 97%, saat dihadapkan pada data baru yang tidak pernah dilihat sebelumnya.



Gambar. 12 Visualisasi hasil prediksi menggunakan data testing

Pada tahap ini, dilakukan visualisasi hasil prediksi model terhadap data *test*. Gambar-gambar dari data *test* ditampilkan bersama dengan label prediksi yang dihasilkan oleh model serta label sesungguhnya. Dengan visualisasi ini, kita dapat melihat secara langsung sejauh mana model mampu mengklasifikasikan gambar-gambar pada data *test* dengan benar. Pada Gambar 12 menggunakan 16 data *test* secara random untuk pengujian menunjukan hasil yang memuaskan karena hasil prediksi semua benar.

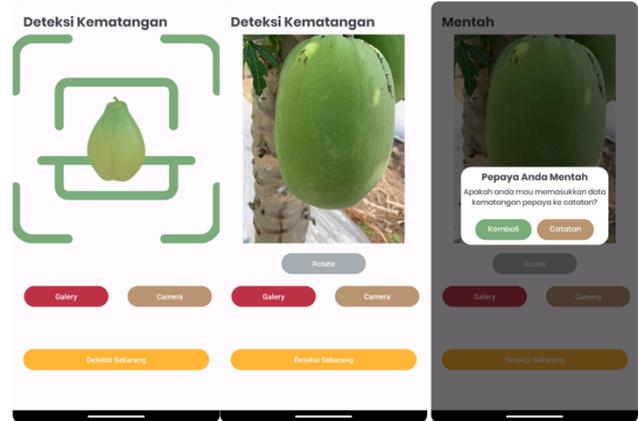
```

converter = tf.lite.TFLiteConverter.from_keras_model(model)
tflite_model = converter.convert()

with open("modelbaru.tflite", 'wb') as f:
    f.write(tflite_model)
    
```

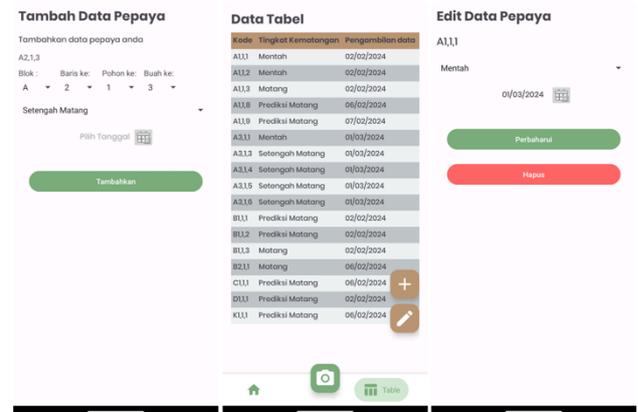
Gambar. 13 Mengubah format model

Setelah berhasil melatih model dan mencapai tingkat akurasi yang tinggi serta kesalahan yang rendah, model kemudian diubah menjadi format *TensorFlow Lite*. Langkah ini tergambar dalam Gambar 13. Model yang telah diubah ini akan digunakan untuk mengimplementasikan *Convolutional Neural Network* (CNN) pada platform berbasis Android.



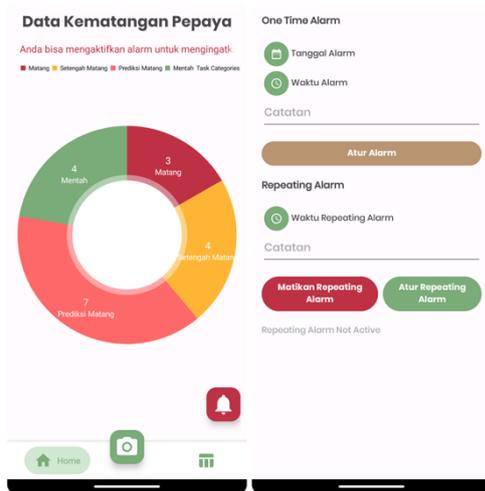
Gambar. 14 Implementasi model kedalam android

Setelah merubah format *file* model menjadi *.tflite* kemudian model tersebut diimplementasikan pada aplikasi. Dalam implementasi pengguna bisa memilih untuk menginput gambar melalui *gallery* atau menggunakan kamera. Jika sudah menginput gambar pengguna bisa menekan tombol deteksi untuk mendeteksi tingkat kematangan papaya. Pada Gambar 14 terlihat pemasangan model untuk deteksi dari aplikasi sudah berjalan dengan baik, pada ahir proses terlihat *dialog box* yang menunjukkan tingkat kematangan papaya yang sedang dideteksi.



Gambar.15 Implementasi fitur pencatatan

Pada Gambar 15 terlihat fitur pencatatan data pepaya. Di sini, pengguna dapat memasukkan posisi pepaya yang berada di blok apa, baris berapa, pohon keberapa, dan buah keberapa. Pengguna juga diminta memasukkan jenis kematangan pepaya berdasarkan hasil deteksi yang dilakukan sebelumnya. Terakhir, pengguna diminta memasukkan tanggal pengambilan data pepaya. Selain itu, pengguna juga dapat melihat dan mengedit data pepaya yang telah dimasukkan, sehingga dapat diketahui berapa pepaya yang sudah matang, mentah, dan setengah matang.



Gambar. 16 Implementasi fitur prediksi dan pengingat

Terlihat pada Gambar 16 dalam implementasi fitur prediksi tingkat kematangan berdasarkan perhitungan hari [17], aplikasi ini hanya akan memprediksi buah yang memiliki tingkat kematangan setengah matang menjadi prediksi matang. Ketika status buah berubah menjadi prediksi matang, pengguna akan menerima notifikasi bahwa buah tersebut telah diprediksi matang. Selanjutnya, untuk fitur pengingat atau alarm, pengguna dapat mengatur sendiri waktu alarm atau notifikasi. Aplikasi juga menyediakan alarm berulang untuk pengecekan secara berkala dalam jangka waktu 4 hari. Dari Gambar 14, Gambar 15, dan Gambar 16 dapat dilihat bahwa seluruh fitur yang ada dalam aplikasi sudah berjalan dengan baik sehingga sudah dapat digunakan oleh pengguna.

Dalam tahap pengujian, disini dilakukan uji coba dengan mengambil lima sampel citra dari setiap klasifikasi pepaya. Dengan demikian, total sampel pengujian dalam proses ini adalah 15 sampel. Penting untuk dicatat bahwa citra-citra yang digunakan dalam pengujian ini harus dalam kondisi yang baik atau normal. Citra yang diuji dalam pengujian ini diambil melalui gambar yang sudah tersimpan pada galeri.

TABEL I
TABEL PENGUJIAN FITUR DETEKSI

No	Gambar	Hasil Klasifikasi	Benar/Salah
1		Mentah	Benar
2		Mentah	Benar

No	Gambar	Hasil Klasifikasi	Benar/Salah
3		Mentah	Benar
4		Mentah	Benar
5		Mentah	Benar
6		Setengah Matang	Benar
7		Setengah Matang	Benar
8		Setengah Matang	Benar
9		Setengah Matang	Benar
10		Setengah Matang	Benar
11		Matang	Benar
12		Matang	Benar

No	Gambar	Hasil Klasifikasi	Benar/Salah
13		Matang	Benar
14		Matang	Benar
15		Matang	Benar

Hasil pengujian pada Tabel 1 menunjukkan bahwa dari 15 data gambar pepaya yang diuji, semuanya sesuai dengan hasil klasifikasi yang diharapkan. Ini menunjukkan bahwa proses pengujian berjalan dengan baik dan citra-citra pepaya telah berhasil diklasifikasikan dengan benar.

V. KESIMPULAN

Kesimpulan dari penelitian ini adalah pembangunan aplikasi klasifikasi tingkat kematangan pepaya dengan menggunakan algoritma CNN telah menunjukkan kemajuan yang signifikan dalam penerapan teknologi untuk meningkatkan efisiensi di industri pertanian. Aplikasi ini berhasil memproses data gambar dari galeri. Hasil pengujian menggunakan data testing menunjukkan akurasi sebesar 96.97% pada model CNN dalam mengklasifikasikan tingkat kematangan pepaya. Meskipun pengujian hanya dilakukan pada sejumlah data gambar pepaya tertentu, hasil klasifikasi menunjukkan kesesuaian yang baik dengan harapan, menandakan keberhasilan proses pengujian. Seluruh fitur aplikasi berjalan dengan baik, termasuk pencatatan dan prediksi, memberikan nilai tambah bagi pengguna dalam mengelola data pepaya secara efektif. Namun, untuk penelitian selanjutnya, disarankan untuk melakukan evaluasi lebih lanjut terhadap aplikasi dengan uji coba menggunakan dataset yang lebih beragam, serta mengeksplorasi potensi pengembangan fitur tambahan untuk mendukung kegiatan pertanian secara lebih baik.

REFERENSI

[1] F. Agustina and M. Sukron, "Deteksi Kematangan Buah Pepaya Menggunakan Algoritma YOLO Berbasis Android," *Jurnal Ilmiah Infokam*, vol. 18, no. 2, pp. 70–78, 2022.
 [2] S. Febjislamy, K. Suketi, and R. Yuniarti, "Karakterisasi morfologi bunga, buah, dan kualitas buah tiga genotipe pepaya hibrida," *Buletin Agrohorti*, vol. 6, no. 1, pp. 112–119, 2018.

[3] E. Ellif, S. H. Sitorus, and R. Hidayati, "Klasifikasi Kematangan Pepaya Menggunakan Ruang Warna HSV dan Metode Naive Bayes Classifier," *Coding Jurnal Komputer dan Aplikasi*, vol. 9, no. 01, pp. 66–75, 2021.
 [4] L. A. Wardani, I. G. P. S. Wijaya, and F. Bimantoro, "KLASIFIKASI JENIS DAN TINGKAT KEMATANGAN BUAH PEPAYA BERDASARKAN FITUR WARNA, TEKSTUR DAN BENTUK MENGGUNAKAN SUPPORT VECTOR MACHINE," *Jurnal Teknologi Informasi, Komputer, dan Aplikasinya (JTika)*, vol. 4, no. 1, pp. 75–87, 2022.
 [5] T. Agustin, S. Suyudi, and H. Nuryaman, "Kinerja Kelembagaan Agribisnis Pepaya California," *Jurnal Agristan*, vol. 1, no. 2, 2019.
 [6] M. R. Dwiantara, "PREFERENSI KONSUMEN TERHADAP BUAH PEPAYA CALIFORNIA DAN," 2020.
 [7] A. Aminudin, "KLASIFIKASI TINGKAT KEMATANGAN BUAH PEPAYA MENGGUNAKAN METODE K-NEAREST NEIGHBOR BERDASARKAN WARNA KULIT BUAH," 2019.
 [8] Y. P. Astuti, E. R. Subhiyakto, I. Wardatunizza, and E. Kartikadarma, "Implementasi Algoritma Convolutional Neural Network (CNN) Untuk Klasifikasi Jenis Tanah Berbasis Android," *Jurnal Informatika: Jurnal Pengembangan IT*, vol. 8, no. 3, pp. 220–225, 2023.
 [9] S. P. Backar, P. Purnawansyah, H. Darwis, and W. Astuti, "Hybrid Fourier Descriptor Naive Bayes dan CNN pada Klasifikasi Daun Herbal," *Jurnal Informatika: Jurnal Pengembangan IT*, vol. 8, no. 2, pp. 126–133, 2023.
 [10] N. Nafiyah, "Identifikasi Tumor Otak Citra MRI dengan Convolutional Neural Network," *Jurnal Informatika: Jurnal Pengembangan IT*, vol. 8, no. 3, pp. 213–219, 2023.
 [11] I. Maulana, H. Sastypratiwi, H. Muhardi, N. Safriadi, and H. Sujaini, "Implementasi Convolutional Neural Network (CNN) untuk Klasifikasi Motif Batik pada Aplikasi Computer Vision Berbasis Android," *JEPIN (Jurnal Edukasi dan Penelitian Informatika)*, vol. 9, no. 3, pp. 384–393.
 [12] N. Nurmalasari, Y. A. Setiawan, W. Astuti, M. R. R. Saclan, S. Masturoh, and T. Haryanti, "Classification for Papaya Fruit Maturity Level with Convolutional Neural Network," *Jurnal Riset Informatika*, vol. 5, no. 3, pp. 331–338, 2023.
 [13] S. Masturoh and T. Haryanti, "CLASSIFICATION FOR PAPAAYA FRUIT MATURITY LEVEL WITH CONVOLUTIONAL NEURAL NETWORK".
 [14] M. S. Hossen, I. Haque, M. S. Islam, M. T. Ahmed, M. J. Nime, and M. A. Islam, "Deep learning based classification of papaya disease recognition," in *Proceedings of the 3rd International Conference on Intelligent Sustainable Systems, ICISS 2020*, Institute of Electrical and Electronics Engineers Inc., Dec. 2020, pp. 945–951. doi: 10.1109/ICISS49785.2020.9316106.
 [15] B. Yanto, L. Fimawahib, A. Supriyanto, B. H. Hayadi, and R. R. Pratama, "Klasifikasi Tekstur Kematangan Buah Jeruk Manis Berdasarkan Tingkat Kecerahan Warna dengan Metode Deep Learning Convolutional Neural Network," *INOVTEK Polbeng-Seri Informatika*, vol. 6, no. 2, pp. 259–268, 2021.
 [16] S. A. Damayanti, A. Arkadia, and D. S. Prasvita, "Klasifikasi Buah Mangga Badami Untuk Menentukan Tingkat Kematangan dengan Metode CNN," in *Prosiding Seminar Nasional Mahasiswa Bidang Ilmu Komputer dan Aplikasinya*, 2021, pp. 158–165.
 [17] A. Suryana and R. Wiradinata, "Pengaruh konsentrasi kitosan terhadap lama simpan dan mutu pada dua tingkat kematangan pepaya Callina (*Carica papaya L.*)," 2013.