

BAB II

TINJAUAN PUSTAKA

2.1 Penelitian Terdahulu

Berbagai sistem klasifikasi telah dikembangkan sebelumnya dengan memanfaatkan *Convolutional Neural Network* (CNN) dan teknik *Transfer Learning*. Ulasan penelitian ini akan berguna untuk memperluas pemahaman dan merumuskan konsep-konsep yang berkaitan dengan penelitian yang akan dilakukan. Salah satu penelitian yang relevan adalah penelitian oleh Fawwaz (2023) "*Implementation of Transfer Learning in CNN for Classification of Nut Type*" berfokus pada klasifikasi berbagai jenis kacang dengan menerapkan *Transfer Learning* pada model CNN seperti *InceptionV3* dan *Xception* relevan. Model *Xception* menunjukkan kinerja terbaik dengan akurasi validasi 86,36%.

Di sisi lain, penelitian oleh Zulkhoiri et al (2024) berjudul "*Investigation of Oil Palm Fruit Bunch Ripeness Classification Using Machine Learning Classifiers*" menggunakan metode *K-Nearest Neighbors* (KNN) dan *Support Vector Machine* (SVM). Keduanya bekerja berdasarkan fitur warna hasil ekstraksi HSV, RGB, dan YCbCr dari gambar buah sawit. KNN mencapai akurasi 97,40% (*hold-out*), sedangkan SVM mencapai 100% (*cross-validation*). Namun, metode ini memerlukan proses ekstraksi fitur manual dan tidak bekerja langsung dari citra mentah, sehingga tidak cocok untuk pendekatan *end-to-end* seperti pada sistem berbasis Android.

Selanjutnya "Klasifikasi Penyakit Tanaman Tomat Menggunakan Metode *Convolutional Neural Network* (CNN)" oleh Wijaya et al (2024). Penelitian ini mengklasifikasikan penyakit yang menyerang daun tanaman tomat, serta mengembangkan aplikasi berbasis web untuk klasifikasi penyakit tanaman tomat menggunakan (CNN) dan arsitektur VGG16. Model CNN yang digunakan dalam penelitian ini mencapai akurasi 82% pada data validasi dan 98% pada data pelatihan.

Selanjutnya penelitian oleh Alfatni et al (2022) berjudul "*Development of Oil Palm Fresh Fruit Bunch Grading System Using Feature-Level Fusion of Color and Texture Features*" melakukan klasifikasi kematangan buah sawit menggunakan

metode KNN, SVM, dan *Artificial Neural Network* (ANN). Penelitian ini memanfaatkan ekstraksi fitur warna (RGB, HSV) dan tekstur (GLCM, Gabor, BGLAM) sebagai input. Dari hasil pengujian, metode ANN menunjukkan performa terbaik dengan nilai AUC di atas 93%.

Selain itu, penelitian “Klasifikasi Tingkat Kematangan Buah Pisang Raja Menggunakan Metode CNN Berbasis Android” oleh Febriana and Lusiana (2024). Penelitian ini menggunakan metode *Convolutional Neural Network* pada platform Android untuk mengembangkan sistem klasifikasi tingkat kematangan buah pisang Raja dengan akurasi klasifikasi 95% menggunakan dataset 300 gambar.

Selanjutnya “Rancang Bangun Alat Klasifikasi Buah Mangga Karbitan menggunakan CNN berbasis Android” dilakukan oleh (Gumilang, Taqwa, & Salamah 2024). Penelitian ini menggunakan CNN berbasis Android untuk mengklasifikasikan kematangan mangga. Dengan akurasi rata-rata 94,4%, sistem ini dapat membedakan antara mangga yang matang secara alami dan yang kematangannya dipercepat menggunakan kalsium karbida.

Selain itu, penelitian yang dilakukan oleh (Purnama 2024) dengan judul “*Comparison of CNN Transfer Learning in Detecting Superior Local Fruit Types in Bali*”. Penelitian ini mengidentifikasi varietas buah lokal unggul di Bali, dengan membandingkan model *Transfer Learning* CNN, khususnya *MobileNetV2* dan *ResNet152*. Saat mengenali gambar buah salak Bali, model *ResNet152* menunjukkan akurasi tertinggi yakni sebesar 92%.

Selanjutnya “Implementasi *Transfer Learning* Dalam Mendeteksi Penyakit Pada Daun Gandum” dilakukan oleh Mashuri (2022) dilakukan untuk mengidentifikasi penyakit pada daun gandum, termasuk septoria dan karat strip. Penelitian ini menggunakan pendekatan *Transfer Learning* dengan lima arsitektur CNN, yaitu *InceptionV3*, *MobileNetV2*, *VGG16*, *ResNet101V2*, dan *DenseNet201*. Akurasi yang dicapai hingga 100%, dengan model *InceptionV3*, *MobileNetV2*, dan *ResNet101V2* yang merupakan model terbaik yang dihasilkan, serta *DenseNet201* dan *VGG16* masing-masing memiliki akurasi 95% dan 76%.

Selain itu, penelitian berjudul “*Transfer Learning* untuk Klasifikasi Penyakit dan Hama Padi Menggunakan *MobileNetV2*” oleh Putra, Mustaqim, and Muriatmoko (2023). Membandingkan lima arsitektur untuk klasifikasi penyakit dan

hama pada tanaman padi yakni *MobileNetV2*, *NasNet Mobile*, *EfficientNet B7*, *InceptionV3*, dan VGG16. Dataset diproses dan ditingkatkan untuk mencakup tiga hama, lima penyakit, dan satu jenis padi sehat. Arsitektur *MobileNetV2* menunjukkan performa terbaik di antara kelima model yang diuji, dengan akurasi, presisi, dan *recall* masing-masing sebesar 96%, serta spesifisitas mencapai 99%.

Penelitian berjudul “Implementasi *Transfer Learning* Menggunakan *Convolutional Neural Network* dalam Klasifikasi Penyu” oleh Eka Okta Putra, Queena Fredlina, and Yudi Anggara Wijaya (2024), menggunakan CNN dengan *Transfer Learning* pada model VGG16 dan VGG19. Penelitian ini menghasilkan sistem klasifikasi spesies penyu laut yang mampu mengidentifikasi enam spesies penyu laut, dengan akurasi 91% pada data pelatihan dan 89% pada data pengujian. Perbandingan penelitian dapat dilihat pada tabel 2.1 berikut.

Tabel 2. 1 Penelitian Terdahulu

Peneliti	Judul	Input	Metode	Output
Fawwaz et al. (2023)	<i>Implementation of Transfer Learning in CNN for Classification of Nut Type</i>	Gambar kacang dari kamera	<i>InceptionV3</i> , <i>Xception</i> , CNN, <i>Transfer Learning</i> ,	Jenis kacang (6 kelas)
Zulkhoiri et al (2024)	<i>Investigation of Oil Palm Fruit Bunch Ripeness Classification Using Machine Learning Classifiers</i>	Fitur warna dari gambar buah sawit (RGB, HSV, YCbCr)	<i>K-Nearest Neighbors(KNN)</i> , <i>Support Vector Machine (SVM)</i>	Kematangan buah sawit (3 kelas)

Peneliti	Judul	Input	Metode	Output
Wijaya et al. (2024)	Klasifikasi Penyakit Tanaman Tomat Menggunakan Metode CNN	Gambar daun tomat	CNN, VGG16	Jenis penyakit daun tomat (10 kelas)
Alfatni et al (2022)	Development of Oil Palm Fresh Fruit Bunch Grading System Using Feature-Level Fusion of Color and Texture Features	Fitur warna dan tekstur (RGB, HSV, GLCM, Gabor, BGLAM)	<i>K-Nearest Neighbors(KNN)</i> , <i>Support Vector Machine (SVM)</i> , <i>Artificial Neural Network (ANN)</i>	Kematangan buah sawit
Febriana and Lusiana (2024)	Klasifikasi Tingkat Kematangan Buah Pisang Raja Menggunakan Metode CNN Berbasis Android	Gambar buah pisang	CNN, TFLite	Kematangan 3 tingkat
Gumilang, Taqwa, & Salamah (2024)	Rancang Bangun Alat Klasifikasi Buah Mangga Karbitan menggunakan CNN berbasis Android	Gambar mangga	CNN, <i>InceptionV3</i>	Matang alami vs dipercepat

Peneliti	Judul	Input	Metode	Output
Purnama (2024)	<i>Comparison of CNN Transfer Learning in Detecting Superior Local Fruit Types in Bali</i>	Gambar buah lokal	<i>MobileNetV2, ResNet152, Transfer Learning,</i>	Deteksi salak, manggis, jeruk, dll
Mashuri (2022)	Implementasi <i>Transfer Learning</i> Dalam Mendeteksi Penyakit Pada Daun Gandum	Gambar daun gandum	<i>Transfer Learning, InceptionV3, MobileNetV2, VGG16, ResNet101V2, DenseNet201</i>	Septoria, Stripe Rust
Putra, Mustaqim, and Muriatmoko (2023)	<i>Transfer Learning</i> untuk Klasifikasi Penyakit dan Hama Padi Menggunakan <i>MobileNetV2</i>	Gambar tanaman padi	<i>MobileNetV2, Transfer Learning, NasNet Mobile, EfficientNet B7, InceptionV3, VGG16</i>	5 penyakit, 3 hama, 1 padi sehat
Eka Okta Putra, Queena Fredlina, and Yudi Anggara Wijaya (2024)	Implementasi <i>Transfer Learning</i> Menggunakan <i>Convolutional Neural Network</i> dalam Klasifikasi Penyu	Gambar penyu	VGG16, VGG19, <i>Transfer Learning,</i>	6 spesies penyu

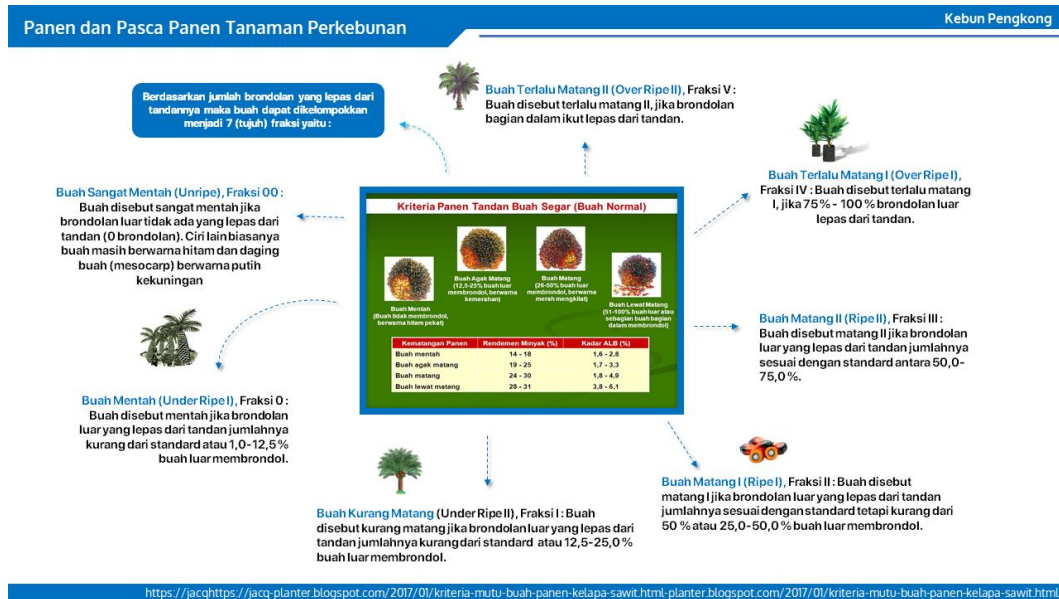
Berdasarkan perbandingan dari berbagai penelitian tersebut, metode CNN dengan pendekatan *Transfer Learning* terbukti unggul dalam klasifikasi berbasis citra karena mampu mengenali fitur secara otomatis tanpa proses ekstraksi manual. Di antara berbagai arsitektur, *MobileNetV2* dipilih karena ringan, cepat, dan efisien untuk dijalankan di perangkat Android, menjadikannya paling sesuai untuk studi kasus klasifikasi kematangan buah kelapa sawit secara langsung melalui kamera ponsel ataupun dari galeri.

2.2 Landasan Teori

2.2.1 Kematangan Kelapa Sawit

Kesegaran dan tingkat kematangan buah kelapa sawit memiliki peran penting dalam menentukan kualitas serta jumlah minyak yang dihasilkan. Jika buah dipanen pada saat kematangannya optimal, maka minyak yang dihasilkan akan lebih banyak dan kualitasnya lebih baik (Salambue and Shiddiq 2019). Sebaliknya, jumlah dan kualitas minyak yang dihasilkan dapat berkurang jika buah dipanen saat masih mentah atau lewat matang.

Seiring bertambahnya tingkat kematangan, warna buah kelapa sawit mengalami perubahan akibat berkurangnya pigmen seperti klorofil, yang memberikan warna hijau, serta antosianin, yang memberi warna merah keunguan. Saat buah mencapai kematangan, kulitnya berubah menjadi oranye kemerahan. Oleh karena itu, kandungan antosianin dapat dimanfaatkan sebagai indikator untuk menilai tingkat kematangan buah. Selain itu, pigmen alami seperti karoten yang berwarna oranye-merah juga meningkat konsentrasinya seiring dengan kematangan buah (Raj et al. 2021). Perubahan warna ini menjadi indikator penting dalam menentukan waktu panen, di mana pengamatan visual terhadap warna kulit buah masih menjadi metode umum yang digunakan untuk menilai tingkat kematangan buah kelapa sawit (Himmah, Widyaningsih, and Maysaroh 2020).



Gambar 2. 1 Kematangan Buah Kelapa Sawit
Sumber: (Pengkong 2021)

Dapat dilihat pada gambar 2.1 diatas, yang mana buah mentah ditandai dengan sedikit atau tidak adanya brondolan yang lepas serta warna buah yang cenderung hitam kemerahan. Buah matang memiliki ciri khas warna merah mengkilat hingga orange dengan pelepasan brondolan antara 25% hingga 75% dari bagian luar tandan. Sementara itu, buah lewat matang menunjukkan tingkat pelepasan brondolan lebih dari 75%, bahkan hingga buah bagian dalam ikut lepas, dan warna dominan orange. Klasifikasi ini menjadi dasar penting dalam pengembangan sistem informasi berbasis *mobile* untuk mengidentifikasi tingkat kematangan buah kelapa sawit menggunakan metode *Transfer Learning* pada arsitektur *Convolutional Neural Network* (CNN), untuk membedakan kategori buah mentah, matang, dan lewat matang.

Tabel 2. 2 Deskripsi Kematangan Buah Kelapa Sawit

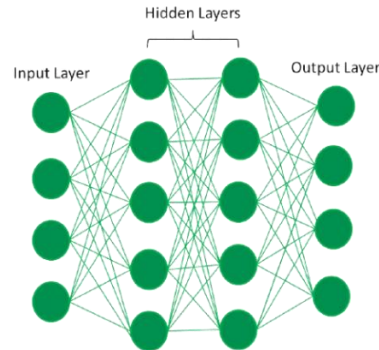
No	Gambar	Deskripsi
1		<i>Unripe oil palm fruit</i> atau buah kelapa sawit yang belum matang yakni berwarna kehitaman atau oranye kekuningan. Kualitas minyak dari buah ini ketika dipanen, akan memiliki rendemen (jumlah) minyak yang rendah. Ciri utamanya yakni membrondol kurang dari 1 brondol per kilogram janjang.
2		<i>Ripe oil palm fruits</i> atau buah kelapa sawit yang sudah matang sempurna yakni berwarna Kemerahan mengkilap, Warna ini menandakan bahwa buah sudah siap dipanen. Kualitas minyak ini akan menghasilkan rendemen minyak yang tinggi dan lebih baik, karena kandungan dengan komposisi yang lebih stabil serta optimal.
3		<i>Overripe palm fruit</i> atau buah kelapa sawit yang lewat matang yakni memiliki warna merah tua dan membrondol lebih dari 50% hingga maksimal 75%. Rendemen minyak mungkin tetap tinggi, namun kualitasnya rendah dan dapat menyebabkan kerugian panen jika tidak segera diambil.

Sumber: (Pardamean 2021)

2.2.2 Deep Learning

Deep Learning adalah kumpulan algoritma dalam *Machine Learning* yang dirancang untuk mempelajari data menggunakan jaringan syaraf tiruan, di mana setiap lapisan dalam jaringan mempelajari fitur dari data secara bertingkat. Lapisan awal mempelajari fitur dasar, sedangkan lapisan berikutnya membentuk

pemahaman yang lebih kompleks berdasarkan informasi dari lapisan sebelumnya. Hal ini memungkinkan algoritma untuk mengidentifikasi pola pengumpulan informasi yang lebih sederhana (Akmal et al. 2025).



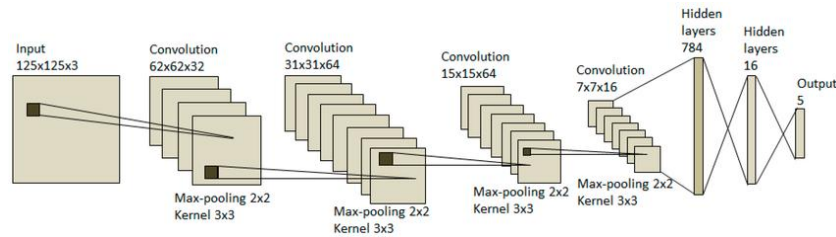
Gambar 2. 2 *Deep Learning*

Sumber: (Saumyasaxena2730 2025)

Pada Gambar 2.2 diatas terdiri dari tiga komponen utama yakni lapisan masukan (*input layers*), lapisan tersembunyi (*hidden layer*), dan lapisan keluaran (*output layer*). Lapisan masukan berfungsi untuk menerima data awal, seperti nilai piksel pada gambar, lalu data tersebut diproses melalui serangkaian lapisan tersembunyi yang terdiri dari neuron-neuron dan fungsi aktivasi, yang bertujuan untuk mengenali pola atau fitur penting dalam data. Setiap neuron terhubung dengan neuron lain melalui bobot yang diubah selama fase pelatihan sehingga jaringan dapat menghasilkan keluaran yang akurat dan menampilkan hasil klasifikasi akhir.

2.2.3 *Convolutional Neural Network (CNN)*

CNN adalah salah satu bentuk jaringan saraf tiruan yang paling umum digunakan dalam pengolahan citra digital. CNN telah terbukti berkinerja sangat baik dalam berbagai tugas visual, termasuk deteksi objek, segmentasi gambar, dan klasifikasi. CNN dirancang khusus untuk menangani data visual seperti foto (Ayeni 2022). CNN menggunakan teknik *Deep Learning*, yang merupakan perpanjangan dari *Multilayer Learning*, untuk menganalisis dataset besar dan kompleks serta secara otomatis menemukan ciri-ciri penting tanpa memerlukan ekstraksi ciri oleh manusia (Harahap et al. 2020).



Gambar 2. 3 *Convolutional Neural Network*
Sumber: (Phung and Rhee 2019)

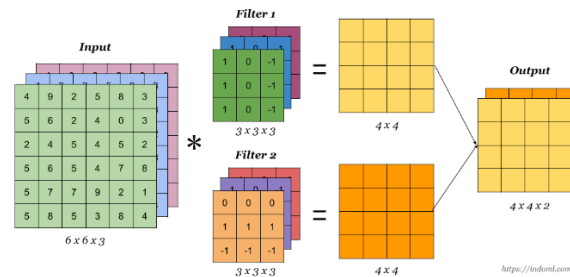
Arsitektur *Convolutional Neural Network* (CNN) yang ditunjukkan pada Gambar 2.3 ini dibagi menjadi beberapa tahap utama yakni lapisan input hingga lapisan output. Dimensi awal gambar masukan adalah 125x125x3, karena ukuran pixelnya 125x125, dan tiga saluran warna RGB. Lapisan max-pooling 2x2 digunakan untuk mengurangi dimensi spasial setelah gambar di proses melalui banyak lapisan konvolusi dengan filter 3x3. Hasil dari metode ini adalah 62x62x32 untuk lapisan pertama, 31x31x64 untuk lapisan kedua, 15x15x64 untuk lapisan ketiga, dan 7x7x16 untuk lapisan keempat. Setelah keluaran konvolusi diubah menjadi vektor berdimensi 784, keluaran tersebut diproses oleh dua lapisan tersembunyi, yang masing-masing memiliki 784 dan 16 neuron. Terakhir, lapisan output menggunakan fungsi aktivasi *Softmax* untuk menghitung probabilitas klasifikasi gambar dan memiliki 5 neuron, yang sesuai dengan jumlah kelas.

CNN diakui karena kemampuannya mendeteksi pola visual melalui ekstraksi fitur otomatis, sehingga menghilangkan kebutuhan akan proses desain fitur manual. Seiring bertambahnya kedalaman jaringan, setiap lapisan konvolusi menangkap fitur yang lebih kompleks. Alzubaidi et al (2021) menyatakan bahwa CNN lebih unggul dalam identifikasi pola dan klasifikasi gambar karena dapat menciptakan representasi hierarkis data visual. Selain itu, desain CNN mempertahankan struktur spasial input sambil meningkatkan efisiensi komputasi melalui penggunaan konektivitas lokal dan berbagi parameter (Nawrocka, Nawrocki, and Kot 2023). Berikut tiga bagian inti pada *convolutional neural network* yaitu *Convolution Layer*, *Pooling Layer*, *Fully Connected Layer* (Zewen Li et al. 2021)

1) *Convolution Layer*

Langkah krusial dalam ekstraksi fitur model *Convolutional Neural Network* (CNN) adalah *Convolutional Layer*. Hasil dari proses ini dikenal sebagai peta fitur

atau (*feature map*), yang menampilkan ciri-ciri input yang penting. Ketika kernel konvolusi (filter) diterapkan dengan ukuran tertentu, informasi di bagian tepi (*border*) sering kali hilang. Untuk mengatasi hal ini, digunakan *padding*, yaitu menambah nol di sekitar input sehingga ukuran output tetap terjaga. Selain itu, *stride* (langkah pergeseran filter) digunakan untuk menggambarkan beberapa proses rapat yang dilakukan. Semakin besar nilai *stride*, semakin jarang filter diterapkan, sehingga kepadatan peta fitur menjadi lebih rendah (Zewen Li et al. 2021). Seperti yang ditunjukkan pada gambar 2.4 dibawah.



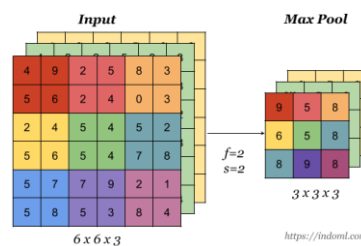
Gambar 2. 4 *Convolution layer*
Sumber: (Priyono 2018)

2) *Pooling Layer*

Pooling adalah teknik yang digunakan untuk mengurangi kompleksitas dan menghilangkan redundansi (juga dikenal sebagai *down-sampling*). *Pooling* berfungsi untuk mengambil informasi penting dari area lokal peta fitur dengan mempertahankan nilai-nilai utama sekaligus mengurangi ukuran data. Dua jenis *pooling* yang paling sering digunakan adalah:

- Max Pooling*: Menghitung jumlah maksimum dari setiap area kecil.
- Average Pooling*: Menghitung rata-rata setiap wilayah kecil.

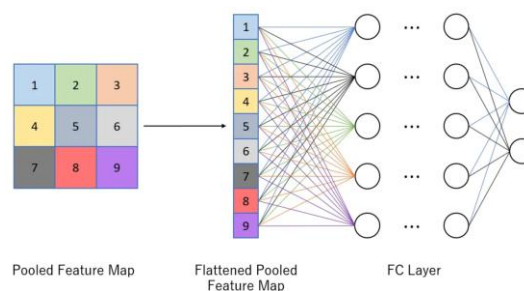
Pooling membuat model lebih efisien, tidak terlalu sensitif terhadap kesalahan kecil, dan lebih mudah digunakan (Zewen Li et al. 2021), seperti yang tertera pada gambar 2.5 dibawah.



Gambar 2. 5 *Pooling Layer*
Sumber: (Priyono 2018)

3) *Fully connected layer*

Setelah peta fitur dihasilkan dari lapisan *Convolution* dan diperkecil (*down sampling*) melalui *pooling*, langkah selanjutnya adalah meratakan (*flattening*) peta fitur menjadi vektor satu dimensi (1D). Vektor ini kemudian diteruskan ke satu atau lebih lapisan yang terhubung penuh (*fully connected layer* / *dense layer*). Pada lapisan ini, Setiap neuron terhubung dengan semua neuron di lapisan sebelumnya dan bobot koneksi ini dapat dipelajari selama proses pelatihan. *Fully connected Layer* berfungsi untuk menggabungkan semua fitur yang telah diekstraksi sebelumnya dan menghasilkan output akhir model. Tugas klasifikasi pada *fully connected layer* terakhir adalah menghasilkan jumlah neuron output yang sesuai dengan jumlah kelas yang akan diprediksi. Setiap *fully connected layer* biasanya diikuti oleh fungsi aktivasi nonlinier seperti ReLU, kecuali lapisan output (Zewen Li et al. 2021). Struktur dari *fully connected layer* terakhir dapat dilihat pada Gambar 2.6 dibawah.

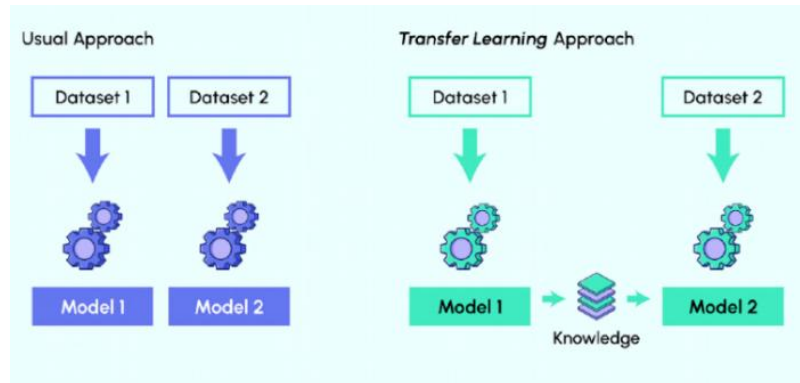


Gambar 2. 6 *Fully connected layer*
Sumber: (Aramendia 2024)

2.2.4 Pengembangan Model Klasifikasi

1) *Transfer Learning*

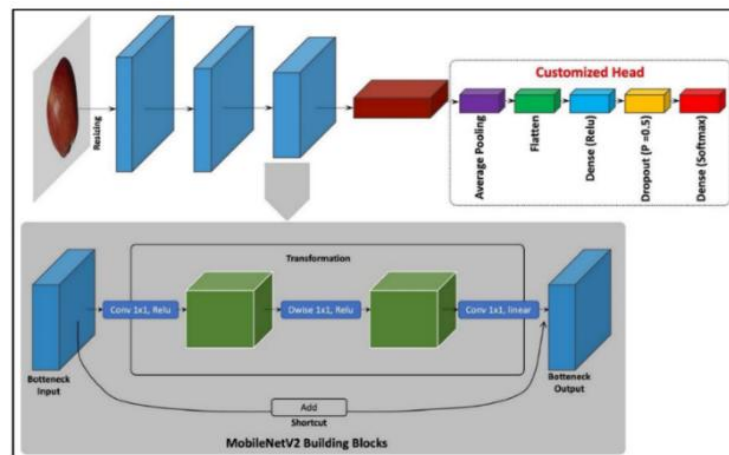
Kecepatan klasifikasi dapat terhambat oleh waktu komputasi yang relatif lama untuk pelatihan algoritma *Deep Learning* berbasis CNN. Dalam mengatasi keterbatasan ini, *Transfer Learning* perlu digunakan, yang mana *Transfer Learning* sendiri merupakan teknik yang mempercepat proses pelatihan dengan menggunakan model yang telah dilatih sebelumnya pada dataset yang besar, yang mengharuskan pengguna untuk melatih ulang model tersebut dengan sejumlah kecil data berdasarkan kebutuhan masing-masing (Mardianto, Dewi, and Risma 2024). Seperti yang ditunjukkan pada Gambar 2.7 dibawah.



Gambar 2. 7 *Transfer Learning*
Sumber: (Maulid 2023)

2) Arsitektur *MobileNetV2*

Arsitektur CNN salah satunya *MobileNetV2*, yang diciptakan khusus untuk perangkat dengan sumber daya terbatas, termasuk aplikasi android. Arsitektur ini menggabungkan pendekatan residual terbalik, kemacetan linier, dan algoritma konvolusi yang dapat dipisahkan secara mendalam dengan cara yang ringan dan efektif. Hal ini memungkinkan penghitungan parameter yang lebih rendah dan bebas komputasi. Menurut penelitian oleh Elinda Lusyana Puji Ristanti (2024), *MobileNetV2* lebih baik dari arsitektur VGG16 dalam klasifikasi penyakit yang menyerang daun cabai, dengan akurasi 92% dibandingkan dengan 74% untuk VGG16. Menurut hasil ini, *MobileNetV2* sangat akurat, cepat secara komputasi, dan efisien model, yang membuatnya ideal untuk digunakan dalam sistem klasifikasi berbasis *mobile*. Seperti yang ditunjukkan pada Gambar 2.8, sistem klasifikasi kematangan kelapa sawit juga akan diterapkan dengan memanfaatkan arsitektur ini.



Gambar 2. 8 Arsitektur *MobileNetV2*
Sumber: (Gulzar 2023)

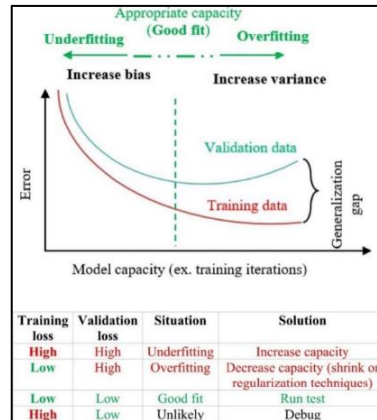
3) Data Augmentation

Data *Augmentation* merupakan teknik yang digunakan untuk memperluas dan memvariasikan data pelatihan dengan menghasilkan data sintetis dari data yang sudah ada. Teknik ini berperan penting dalam meningkatkan generalisasi dan mengurangi *overfitting*, terutama saat data berlabel terbatas (Shorten, Khoshgoftaar, and Furht 2021). Dalam praktiknya, augmentasi dilakukan dengan memodifikasi data asli menjadi versi baru yang tetap mempertahankan label, sehingga memungkinkan model belajar dari beragam pola tanpa memerlukan tambahan data nyata. Augmentasi data merujuk pada metode memperluas kumpulan data yang tidak lengkap dengan menyediakan titik data yang hilang untuk meningkatkan analisis (Jacob Murel 2024). Pada pengolahan citra, seperti dalam evaluasi kematangan buah kelapa sawit, augmentasi gambar seperti rotasi, *flipping*, *zoom*, dan *cropping* digunakan untuk menciptakan variasi visual, sehingga meningkatkan ketahanan dan akurasi model dalam menghadapi berbagai kondisi lingkungan.

2.2.5 Evaluasi Model Deep Learning

Evaluasi kinerja model *Deep Learning* dilakukan dengan membandingkan performa model pada data pelatihan (*training*) dan data validasi atau pengujian (*validation/testing*) untuk menilai kemampuan generalisasi model. Zhang et al (2017) menegaskan bahwa konsistensi performa antara data *training* dan *testing* merupakan indikator utama generalisasi, yang dikenal sebagai *generalization gap*. Gap yang kecil menunjukkan bahwa model tidak sekadar menghafal data *training*, tetapi mampu mempelajari pola yang dapat diterapkan pada data baru.

Jiang et al (2019) menjelaskan bahwa *generalization gap* didefinisikan sebagai selisih kinerja antara data *training* dan data *testing*, namun tidak terdapat ambang batas numerik baku yang secara universal menyatakan suatu model aman dari *overfitting*. Oleh karena itu, evaluasi dilakukan secara kualitatif dengan mengamati kestabilan dan kedekatan kurva *training* dan *validation*. Gap yang relatif kecil dan stabil umumnya diasosiasikan dengan kemampuan generalisasi yang baik.



Gambar 2. 9 Evaluasi Model *Deep Learning*
 Sumber: (AlThuwaynee et al. 2021)

Analisis kurva loss dan *accuracy* selama pelatihan merupakan pendekatan standar dalam evaluasi model *Deep Learning*. Penurunan *training loss* yang diikuti oleh *validation loss* yang stabil atau menurun menunjukkan proses pembelajaran yang efektif, sedangkan kondisi ketika *training loss* menurun tetapi *validation loss* meningkat mengindikasikan terjadinya *overfitting*. Berdasarkan landasan tersebut, evaluasi model dalam penelitian ini difokuskan pada analisis pola kurva *training loss*, *validation loss*, *training accuracy*, dan *validation accuracy*, serta besarnya *generalization gap*, sebagai dasar penilaian kemampuan generalisasi model klasifikasi tingkat kematangan buah kelapa sawit.

2.2.6 Confusion Matrix

Confusion Matrix merupakan metode yang digunakan untuk mengevaluasi kinerja klasifikasi suatu model dan menentukan persentase prediksi yang akurat dan tidak akurat dengan cara membandingkan nilai aktual dengan nilai yang diharapkan. Berikut ini adalah empat komponen utama *Confusion Matrix*:

- 1) *True Positive* (TP) merupakan jumlah data positif yang diharapkan menjadi *true positives*.
- 2) *True Negative* (TN) merupakan kuantitas data negatif yang diharapkan menjadi *true negatives*.
- 3) *False Positive* (FP) merupakan kuantitas data negatif yang secara keliru diproyeksikan sebagai positif.
- 4) *False Negative* (FN) merupakan kuantitas data positif yang secara keliru diproyeksikan sebagai negatif.

Nilai-nilai ini memungkinkan penggunaan beberapa metrik untuk menilai kinerja model kategorisasi seperti:

		Predicted Class		
		Positive	Negative	
Actual Class	Positive	True Positive (TP)	False Negative (FN) <i>Type II Error</i>	Sensitivity $\frac{TP}{(TP + FN)}$
	Negative	False Positive (FP) <i>Type I Error</i>	True Negative (TN)	Specificity $\frac{TN}{(TN + FP)}$
		Precision $\frac{TP}{(TP + FP)}$	Negative Predictive Value $\frac{TN}{(TN + FN)}$	Accuracy $\frac{TP + TN}{(TP + TN + FP + FN)}$

Gambar 2. 10 *Confusion Matrix*
Sumber: (Bonnet 2023)

Berdasarkan nilai *confusion matrix* pada Gambar 2.10, kinerja model klasifikasi dapat dihitung untuk memperoleh nilai akurasi, presisi, dan *recall*. Nilai akurasi menunjukkan tingkat ketepatan model dalam mengklasifikasikan data dengan benar dibandingkan keseluruhan data.

$$Akurasi = \frac{TP + TN}{TP + TN + FP + FN} \times 100\%$$

Melakukan komparasi data yang terprediksi benar positif dengan keseluruhan data hasil terprediksi positif serta mendeskripsikan kadar keakuratan antara hasil prediksi model dengan data yang diminta merupakan nilai presisi.

$$Presisi = \frac{TP}{FP + TP} \times 100\%$$

Recall mendeskripsikan berapa persen data terprediksi benar positif dengan keseluruhan data benar positif dari keberhasilan model klasifikasi yang didapat.

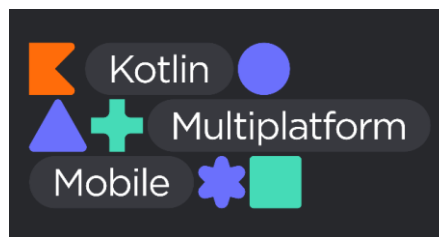
$$Recall = \frac{TP}{FN + TP} \times 100\%$$

Evaluasi dilakukan untuk menilai kinerja klasifikasi model terhadap citra buah kelapa sawit dengan membandingkan label aktual dan nilai prediksi yang dihasilkan, sehingga diperoleh nilai akurasi sebagai metrik utama performa model (Alfredo and Suharjito 2022).

2.2.7 Pengembangan Aplikasi Mobile dengan Kotlin

Kotlin adalah bahasa pemrograman resmi terbaru untuk pengembangan aplikasi Android. Bahasa ini memiliki sintaks yang luas, sepenuhnya kompatibel

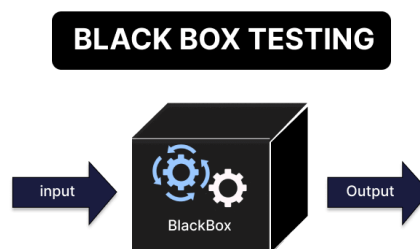
dengan Java, aman dari *null-pointer*, dan berfungsi dengan baik untuk mendesain antarmuka pengguna. Dibandingkan dengan Java, Kotlin menawarkan kinerja yang kompetitif untuk pengembangan aplikasi Android, termasuk manajemen dan konsumsi memori yang efektif, menurut penelitian yang diterbitkan dalam Jurnal KDI, Ekspresi *lambda*, *ekstensi*, dan *coroutine* hanyalah beberapa fitur canggih *Kotlin* yang mempercepat pengembangan aplikasi yang canggih, responsif, dan bermanfaat. Oleh karena itu, *Kotlin* sangat disarankan untuk mengembangkan aplikasi seluler berbasis Android saat ini (Sanjaya and Susilo 2024).



Gambar 2. 11 Bahasa Pemrograman *Kotlin*
Sumber: (Yasin 2021)

2.2.8 Pengujian *Black Box*

Pengujian perangkat lunak yang berfokus pada input dan output tanpa memeriksa arsitektur dasar program dikenal sebagai pengujian "*Black Box Testing*". Teknologi ini sering digunakan pada tahap akhir pengembangan untuk menjamin bahwa semua fungsionalitas yang diharapkan pengguna diimplementasikan dengan benar. Menurut Rambe and Prihantoro (2022), pengujian *Black Box* dilakukan tanpa mengetahui struktur internal aplikasi yang diuji. Pengujian ini hanya melihat output yang dihasilkan dari input yang diberikan.

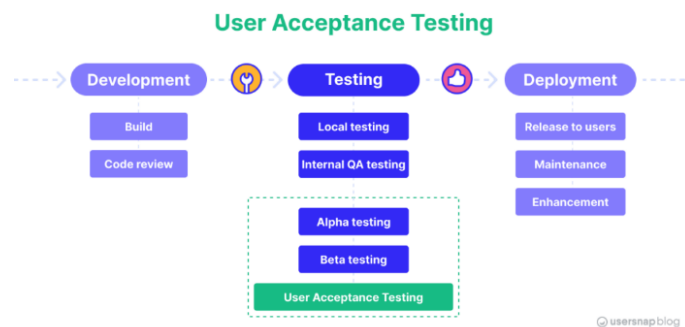


Gambar 2. 12 *Black Box Testing*
Sumber: (Nawar 2025)

Pengujian *Black Box* seperti gambar 2.11 diatas dilakukan pada fitur input gambar, hasil klasifikasi, dan antarmuka aplikasi untuk memastikan bahwa sistem memberikan hasil sesuai ekspektasi pengguna.

2.2.9 Pengujian UAT (User Acceptance Testing)

User Acceptance Testing (UAT) merupakan tahap akhir dalam proses pengujian sistem sebelum implementasi, yang bertujuan untuk memastikan bahwa aplikasi telah berjalan sesuai kebutuhan dan harapan pengguna akhir. Pengujian ini dilakukan langsung oleh pengguna atau pihak terkait menggunakan skenario uji yang mencerminkan kondisi penggunaan sebenarnya. Fokus utama UAT bukan hanya pada aspek teknis, melainkan pada kesesuaian fungsionalitas sistem terhadap tujuan penggunaannya. Jika sistem lolos UAT, maka akan diberikan persetujuan melalui dokumen *sign-off* sebagai bukti bahwa sistem siap diimplementasikan, sebaliknya jika ditemukan kekurangan, pengembang harus melakukan revisi sebelum sistem dapat diterima (Hady, Haryono, and Rahayu 2020).



Gambar 2. 13 *User Acceptance Testing*
Sumber: (Moneta 2022)