

## BAB 2

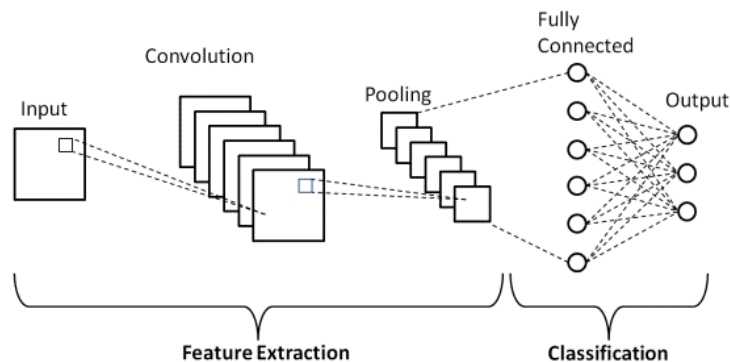
### TINJAUAN PUSTAKA

#### 2.1 Landasan Teori

##### 2.1.1 *Convolutional neural network* (CNN)

*Convolutional neural network* (CNN) adalah salah satu arsitektur *deep learning* yang secara khusus dirancang untuk memproses data berdimensi dua, seperti citra. CNN menjadi metode yang sangat populer dan banyak digunakan dalam berbagai bidang seperti klasifikasi citra, pengenalan wajah, pemrosesan suara, dan visi komputer, karena kemampuannya dalam mengekstraksi fitur-fitur penting dari data input secara otomatis tanpa perlu proses ekstraksi fitur manual.

Arsitektur CNN sendiri memiliki beberapa lapisan utama, yakni lapisan konvolusi (*convolutional layer*), lapisan aktivasi (*activation layer*), lapisan pooling (*pooling layer*), dan lapisan *fully connected*. Lapisan konvolusi merupakan komponen inti dari CNN yang berfungsi untuk melakukan ekstraksi fitur dari input citra. Pada lapisan ini, digunakan kernel atau filter yang memiliki ukuran tertentu, misalnya  $2 \times 2$  atau  $3 \times 3$ . Filter ini bergerak (*slide*) ke seluruh area citra dan melakukan operasi konvolusi, yaitu mengalikan elemen-elemen yang berkorespondensi antara filter dan area citra, lalu menjumlahkan hasilnya untuk menghasilkan sebuah nilai tunggal. Nilai-nilai ini membentuk peta fitur (*feature map*), yang mencerminkan fitur penting dari input citra. Proses ini digambarkan secara visual pada Gambar 2.1.

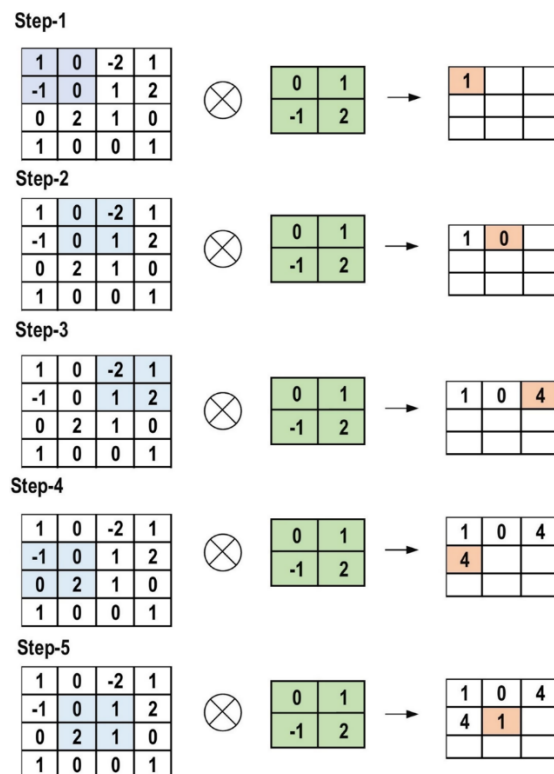


Gambar 2.1 Arsitektur CNN untuk klasifikasi Gambar

Sumber: (Pratama dkk., 2024)

Selain lapisan konvolusi, CNN juga memiliki lapisan aktivasi, seperti Rectified Linear Unit (ReLU), yang memiliki fungsi untuk menambahkan elemen non-linear dalam jaringan. Aktivasi ini memungkinkan jaringan untuk mempelajari hubungan kompleks dari fitur yang tidak dapat dijelaskan oleh fungsi linear saja. Setelah proses konvolusi dan aktivasi, data akan diproses melalui lapisan pooling. Tujuan utama dari pooling adalah untuk mereduksi dimensi dari peta fitur sambil mempertahankan informasi penting. Pooling yang sering digunakan adalah *max pooling*, yaitu mengambil nilai maksimum dari area tertentu dalam peta fitur. Dengan teknik ini, ukuran data dapat dikurangi, sehingga mempercepat proses komputasi dan mencegah *overfitting*.

Proses konvolusi pada CNN ditunjukkan lebih rinci dalam Gambar 2.2, yang menggambarkan langkah-langkah utama dari perhitungan dot product antara kernel dan citra input pada masing-masing posisi. Dalam contoh tersebut, digunakan kernel berukuran  $2 \times 2$  yang diterapkan pada citra grayscale  $4 \times 4$  tanpa padding dan dengan stride sebesar satu.



Gambar 2.2 Ilustrasi operasi konvolusi pada lapisan CNN

Sumber: (Alzubaidi dkk., 2021)

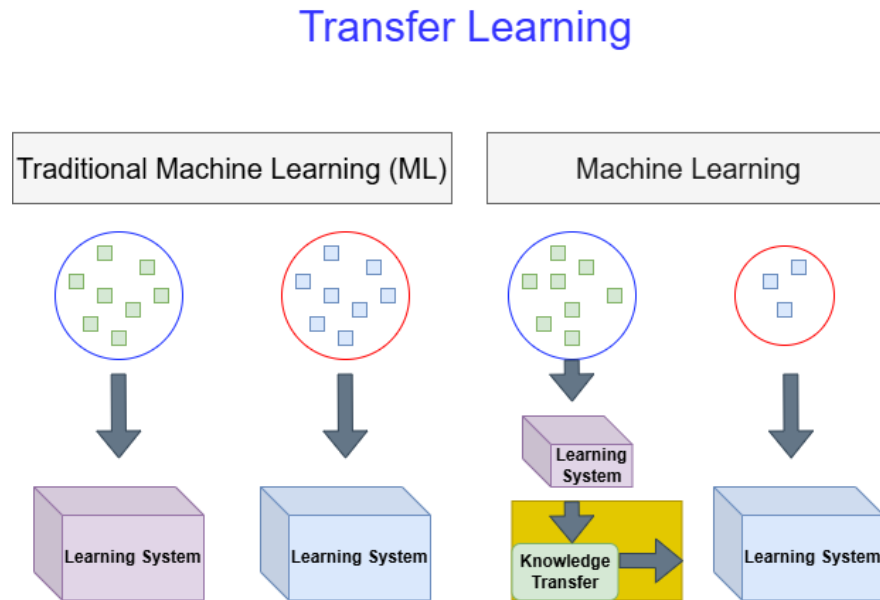
### 2.1.2 Transfer Learning

*Transfer Learning* merupakan sebuah metodologi dalam pembelajaran mesin yang memanfaatkan model *pre-trained* (model yang telah dilatih sebelumnya pada dataset tertentu) sebagai titik awal untuk menyelesaikan permasalahan baru yang memiliki karakteristik serupa. Teknik ini memungkinkan transfer pengetahuan dari model A yang telah dilatih sebelumnya ke model B yang akan diterapkan pada kasus baru, dengan cara memodifikasi dan memperbarui parameter-parameter model sesuai dengan kebutuhan dataset baru. Sebagai ilustrasi, model *pre-trained* yang telah dilatih untuk mengenali kendaraan bermotor dapat diadaptasi untuk mengenali jenis kendaraan yang lebih spesifik seperti truk atau bus (Wijaya dkk., 2021).

Salah satu tantangan utama dalam implementasi model *deep learning* adalah keterbatasan data training yang memadai. Pengumpulan dataset dalam jumlah besar merupakan pekerjaan yang membutuhkan waktu, biaya, dan sumber daya yang banyak. *Transfer Learning* menawarkan solusi yang efisien untuk mengatasi permasalahan kekurangan data training ini dengan memanfaatkan model yang telah dilatih pada dataset besar, kemudian melakukan *fine-tuning* terhadap dataset yang lebih kecil sesuai dengan domain target. Mekanisme ini dapat dianalogikan dengan hubungan guru-murid, dimana guru (model *pre-trained*) mentransfer pengetahuan yang telah diperoleh kepada murid (model baru) melalui proses pembelajaran yang bertahap. Proses *Transfer Learning* melibatkan pelatihan model CNN menggunakan dataset bervolume besar untuk memperoleh bias dan bobot (weights) yang optimal. Bobot-bobot ini kemudian ditransfer ke jaringan yang berbeda untuk proses retraining atau pengujian pada model yang serupa namun baru. Model baru dapat memanfaatkan bobot yang telah dilatih sebelumnya tanpa harus melakukan pelatihan dari awal (*training from scratch*), sehingga menghemat waktu komputasi dan meningkatkan efisiensi proses pembelajaran (Alzubaidi dkk., 2021).

Workflow *Transfer Learning* umumnya dimulai dengan pemilihan model *pre-trained* yang sesuai, diikuti dengan proses adaptasi arsitektur model sesuai kebutuhan kasus baru. Tahap selanjutnya melibatkan *fine-tuning* parameter model pada dataset target, dimana beberapa layer dapat dibekukan (*frozen*) sementara layer lainnya dilatih ulang. Pendekatan ini memungkinkan model untuk mempertahankan fitur-fitur umum yang telah dipelajari sebelumnya sambil

mempelajari karakteristik spesifik dari domain baru.(Wijaya dkk., 2021). Ilustrasi alur *transfer learning* ditunjukkan pada Gambar 2.3 berikut.

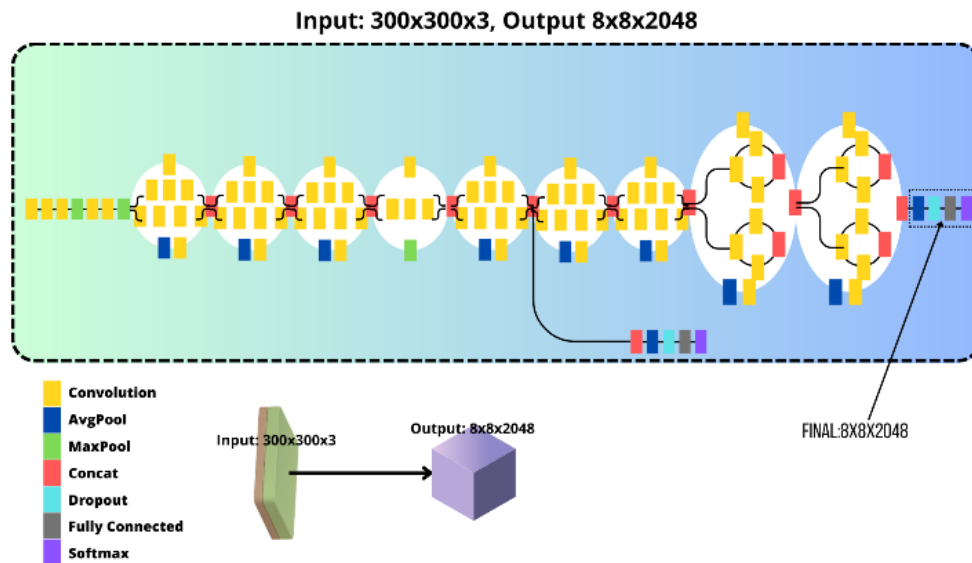


Gambar 2.3 Ilustrasi *Transfer Learning*

Sumber: (Wijaya dkk., 2021)

### 2.1.3 InceptionV3

Arsitektur InceptionV3 merupakan sebuah *Convolutional neural network* (CNN) yang terdiri dari 42 lapisan dan dikembangkan oleh Google dengan fokus pada pencapaian kinerja tinggi dalam tugas klasifikasi citra. Model ini tersusun dari berbagai komponen utama meliputi lapisan konvolusi, pooling, dropout, fully connected, dan *softmax* yang diatur dalam struktur yang kompleks namun efisien, seperti yang tampilkan di Gambar 2.4 berikut:



Gambar 2.4 Arsitektur InceptionV3

Sumber: (Huda dkk., 2023)

InceptionV3 merupakan versi yang telah diperbarui dari model InceptionV1, yang dirancang untuk optimasi adaptasi model dengan menggunakan sejumlah inception modules. Model ini memiliki jaringan yang lebih ekstensif dibandingkan dengan InceptionV1 dan V2, sehingga mampu memberikan performa yang lebih superior dalam tugas klasifikasi. Meskipun memiliki kemampuan yang unggul, pelatihan model InceptionV3 memerlukan komputasi yang intensif dan waktu yang relatif lama. Tantangan ini dapat diatasi melalui implementasi *transfer learning*, dimana parameter model dapat didekonstruksi dengan cara menghilangkan lapisan final menggunakan teknik *transfer learning*, sehingga memungkinkan adaptasi model untuk kategori klasifikasi yang baru dengan tetap mempertahankan bobot yang telah dilatih sebelumnya (Mujahid dkk., 2022).

#### 2.1.4 Deep Learning

*Deep learning* (DL) merupakan cabang dari pembelajaran mesin (*machine learning*) yang menggunakan arsitektur jaringan saraf tiruan berlapis (*deep neural networks*) untuk melakukan proses pembelajaran secara hierarkis. DL menjadi pendekatan yang dominan dalam bidang pengenalan pola dan klasifikasi citra

karena kemampuannya untuk melakukan ekstraksi fitur secara otomatis dari data mentah, tanpa memerlukan proses rekayasa fitur manual.

Arsitektur DL terdiri dari beberapa lapisan (layer) yang tersusun secara berurutan, dimulai dari lapisan input, lapisan tersembunyi (hidden layer), hingga lapisan output. Setiap lapisan memiliki neuron yang saling terhubung dan berperan dalam mentransformasi representasi data ke bentuk yang semakin abstrak. Proses ini memungkinkan model untuk mengenali pola kompleks, seperti tekstur, bentuk, dan struktur objek pada citra, yang kemudian digunakan untuk melakukan klasifikasi secara akurat.

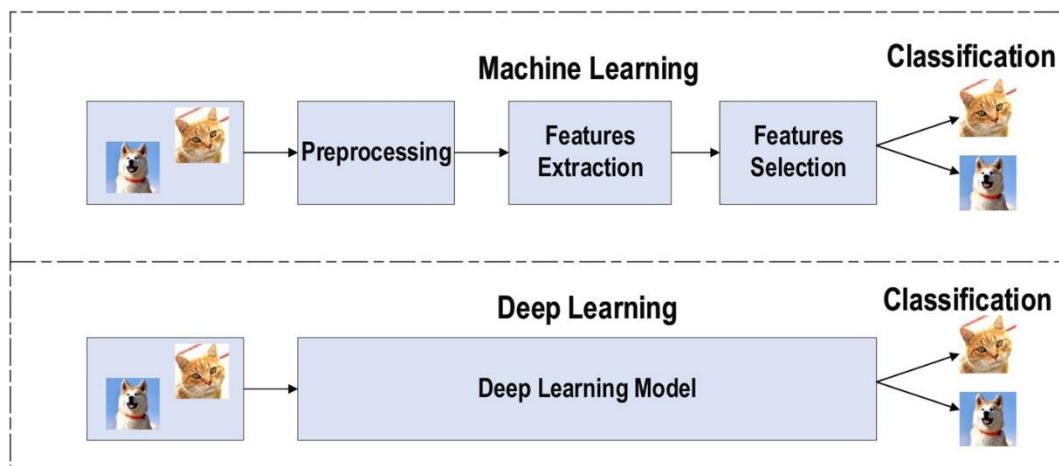
Salah satu kekuatan utama DL adalah kemampuannya mengatasi permasalahan high dimensional data melalui pembelajaran berlapis, sehingga cocok diterapkan pada bidang pengolahan citra, suara, dan teks. Dalam penelitian ini, DL digunakan sebagai dasar dalam pengembangan sistem klasifikasi varietas durian menggunakan model CNN (*Convolutional neural network*) sebagai arsitektur utama untuk mengenali pola visual pada gambar durian. CNN dipilih karena keunggulannya dalam mengekstraksi fitur lokal pada citra dan kemampuannya menghasilkan akurasi tinggi pada berbagai tugas klasifikasi visual (Alzubaidi dkk., 2021). Dalam proses pelatihan model *deep learning*, terdapat beberapa konsep penting yang menentukan bagaimana model belajar dari data, yaitu:

- 1) *Epoch* merupakan sebuah siklus lengkap dimana dataset pelatihan melalui jaringan saraf baik melalui alur maju (*forward*) maupun alur mundur (*backward*), untuk melakukan pembaruan bobot setelah setiap iterasi dilakukan. Penentuan jumlah *epoch* yang optimal menjadi bagian penting dalam proses pelatihan, jika tidak tepat dapat menyebabkan model mengalami *underfitting* maupun *overfitting* (Sari & Sucipto, 2026).
- 2) *Batch size* merupakan banyaknya jumlah sampel data yang digunakan di proses pembaruan bobot internal model untuk satu kali tahapan batch selama proses pelatihan berlangsung. Pemilihan nilai *batch size* ini memiliki pengaruh yang sangat besar pada performa model serta efisiensi durasi waktu pelatihan, penggunaan yang terlalu besar dapat menghasilkan konvergensi model yang lebih cepat namun beresiko membuat model kesulitan dalam

mempelajari pola dari data secara detail sehingga menghasilkan set bobot akhir yang kurang optimal (Sari & Sucipto, 2026).

- 3) *Learning rate* merupakan parameter penting dalam algoritma optimasi proses pelatihan yang berfungsi untuk mengatur besar langkah pembaruan nilai bobot internal model pada setiap iterasi berdasarkan perhitungan gradien *loss function* (Chinthia dkk., 2025).
- 4) *Optimizer* merupakan algoritma yang digunakan dalam pemodelan *deep learning* untuk meminimalisir *loss function* untuk meningkatkan kinerja dari model. Algoritma ini memodifikasi nilai bobot (weight) dan bias saat proses backpropagation, hal tersebut dapat meminimalkan nilai kesalahan agar model dapat menghasilkan prediksi yang lebih akurat sesuai dengan data latih yang digunakan (Wardani dkk., 2023).

Perbedaan antara *Machine Learning* dan *Deep Learning* diilustrasikan pada Gambar 2.5 berikut.



Gambar 2.5 Ilustrasi perbedaan *Machine Learning* dan *Deep Learning*

Sumber: (Alzubaidi dkk., 2021)

#### 2.1.5 Preprocessing

*Preprocessing* merupakan tahap awal dalam proses pengolahan citra yang memiliki tujuan dalam mempersiapkan data untuk diproses secara optimal oleh model *deep learning*. Pada tahap ini, citra durian terlebih dahulu diubah (*resize*) ke 299 x 299 piksel. Hal ini dikarenakan model InceptionV3 secara arsitektur dirancang hanya menerima input dengan dimensi 299 x 299 agar proses ekstraksi fitur dapat berjalan dengan lancar (Szegedy dkk., 2015). Apabila ukuran gambar

bervariasi, maka distribusi informasi piksel akan menjadi tidak seragam dan dapat membuat model menjadi susah dalam mengenali pola fitur. Setelah *resize*, dilakukan proses normalisasi piksel dari 0 sampai 255 menjadi 0 atau 1. Normalisasi ini dilakukan untuk menstabilkan nilai gradien saat dilaksanakannya proses training serta mempercepat proses convergence (model sudah belajar dengan baik). Tanpa normalisasi, nilai piksel yang besar akan menyebabkan model sulit untuk belajar dan mencapai akurasi yang baik (Patro & Sahu, 2015).

#### 2.1.6 Data augmentation

*Data augmentation* merupakan teknik untuk menambah jumlah variasi data dengan melakukan transformasi pada citra tanpa mengubah objek utamanya. Tujuannya sendiri untuk mencegah kemungkinan terjadinya *overfitting*, yaitu kondisi Ketika suatu model terlalu menghafal pola dari data sehingga hal tersebut dapat membuat performa menjadi turun ketika digunakan pada data baru (Shorten & Khoshgoftaar, 2019).

Berikut teknik-teknik augmentasi yang digunakan:

- 1) Rotation berfungsi untuk memutar gambar, yang dimana memiliki tujuan untuk memberikan visualisasi kondisi dimana gambar diambil dari sudut pandang yang sedikit berbeda
- 2) Augmentasi *Flip* merupakan teknik manipulasi citra yang berkerja dengan mengubah orientasi arah objek secara otomatis, seperti membalik posisi citra objek yang semula menghadap ke kiri menjadi ke kanan (*horizontal*), maupun membalik posisi atas dan bawah (*vertikal*) tanpa mengubah nilai label asli dari citra tersebut teknik ini diterapkan agar model mampu mengenali objek meski tampil dari berbagai sudut, bukan hanya menghadap kearah tertentu.
- 3) *Zoom in/out* berfungsi untuk memperbesar maupun memperkecil gambar, teknik ini memberikan simulasi kondisi dimana jarak pengambilan gambar yang berbeda-beda, terkadang ada gambar yang diambil jauh, namun ada juga gambar yang diambil secara dekat. Hal ini dapat membantu model untuk tetap akurat walaupun resolusi objek tampak lebih besar ataupun lebih kecil.
- 4) *Brightness adjustment* berfungsi untuk mengatur tingkat pencahayaan gambar, baik menambah maupun mengurangi penerangan. Kondisi di dunia

nyata sangat berbeda-beda seperti diluar ruangan, dibawah lampu maupun ruangan gelap. Dengan memberikan variasi kecerahan, model menjadi tidak terpaku pada kondisi cahaya tertentu saat melakukan prediksi. Teknik ini membantu meningkatkan ketahanan model pada ketidakpastian lingkungan visual.

- 5) Translation merupakan teknik augmentasi data yang menggeser posisi citra kearah kiri, kanan, atas atau bawah untuk menghindari bias posisi di dalam data. Melalui pergeseran *horizontal* ataupun vertikal ini, sisa ruang kosong pada gambar dapat diisi otomatis dengan warna standar atau noise acak agar ukuran dimensi gambar asli tidak berubah setelah transformasi
- 6) *Shear range* berfungsi untuk memiringkan objek citra sepanjang sumbu *horizontal* atau vertikal, teknik ini memberikan simulasi kondisi dimana objek dalam gambar terlihat miring, baik disebabkan sudut pengambilan kamera yang tidak tegak lurus ataupun objek yang memang tidak berada pada posisi lurus. Dengan kondisi tersebut model menjadi tidak terpaku pada objek tertentu saat prediksi, sehingga meningkatkan ketahanan model terhadap variasi karakteristik baru pada objek (Lionnie & Alaydrus, 2024).

#### 2.1.7 Optimasi dan Regularisasi

Optimasi dan regularisasi model merupakan komponen penting dalam proses pelatihan *deep learning*. Optimasi berkaitan dengan proses pembaruan parameter atau bobot model agar nilai kesalahan prediksi dapat diminimalkan selama pelatihan. Pada CNN, proses pembelajaran dilakukan dengan memperbarui parameter di setiap *epoch* untuk mengurangi kesalahan antara hasil prediksi dan label sebenarnya. Salah satu hyperparameter penting dalam proses ini adalah *learning rate*, yaitu nilai yang menentukan besar langkah pembaruan parameter model. Nilai *learning rate* perlu dipilih dengan tepat karena dapat memengaruhi kestabilan proses pelatihan dan hasil pembelajaran model.

Regularisasi diimplementasikan untuk mengurangi risiko *overfitting*, yaitu kondisi ketika model memiliki performa tinggi pada data latih, tetapi kurang mampu memberikan hasil yang baik pada data baru atau data uji. Pada model CNN, beberapa teknik regularisasi dapat diterapkan untuk membantu meningkatkan

kemampuan generalisasi model. Menurut (Alzubaidi dkk., 2021) beberapa pendekatan seperti dropout, batch normalization, weight decay, dan *data augmentation* dapat digunakan untuk membantu mengurangi risiko *overfitting* pada model CNN. Beberapa teknik optimasi dan regularisasi yang digunakan dalam proyek ini adalah sebagai berikut:

- 1) *Fine-tuning* merupakan salah satu teknik dalam *transfer learning* yang dimana bobot awal (*original weight*) dari jaringan saraf tiruan yang telah dilatih sebelumnya (*pretrained model*) disesuaikan atau dilatih kembali agar dapat beradaptasi ke karakteristik dataset target yang baru (Kandel & Castelli, 2020).
- 2) *Warmup Learning rate* merupakan sebuah strategi dalam pelatihan jaringan saraf dimana proses training diawali dengan *learning rate* kecil, lalu kemudian ditingkatkan kembali ke nilai *learning rate* awal setelah proses pelatihan berjalan stabil (He dkk., 2018).
- 3) *Dropout* adalah teknik regularisasi yang digunakan untuk meningkatkan kemampuan generalisasi model dengan cara menonaktifkan beberapa jaringan neuron secara acak pada setiap *epoch* pelatihan. Mekanisme ini memaksa jaringan untuk mempelajari representasi fitur yang saling independen selama pelatihan, sedangkan pada fase pengujian seluruh neuron akan kembali aktif secara penuh (Alzubaidi dkk., 2021).
- 4) L2 Regularization adalah teknik regularisasi standar dalam *deep learning* yang berkerja secara langsung pada arsitektur dan parameter model untuk menangani *overfitting*. Teknik ini diterapkan untuk mengontrol bobot selama pelatihan, sehingga model dapat meningkatkan performa generalisasinya pada data uji yang belum pernah dilihat sebelumnya (Alzubaidi dkk., 2021).

#### 2.1.8 Confusion matrix

*Confusion matrix* merupakan sebuah tabel yang digunakan untuk memberikan evaluasi kinerja model klasifikasi dengan cara membandingkan label actual dan label prediksi yang dihasilkan dari model. Menurut (Sathyanarayanan, 2024) *confusion matrix* merupakan sebuah tabel yang berbentuk persegi berukuran  $N \times N$ , dimana  $N$  menunjukkan jumlah output classes. Setiap baris yang ada pada

matriks ini menunjukkan jumlah data yang nantinya akan diprediksi model sebagai kelas tertentu, sedangkan setiap kolom menunjukkan jumlah data kelas aktual. Dengan kata lain baris merepresentasikan kelas prediksi, sedangkan kolom merepresentasikan kelas aktual. Secara umum *confusion matrix* untuk klasifikasi biner ditunjukkan dalam bentuk Tabel 2.1 berikut

Tabel 2.1 *Confusion matrix*

|                | <b>Prediksi Positif</b> | <b>Prediksi Negatif</b> |
|----------------|-------------------------|-------------------------|
| Aktual positif | TP (True Positive)      | FN (False Positive)     |
| Aktual negatif | FN (False Negative)     | TN (True Negative)      |

Sedangkan untuk kasus multi-kelas, *confusion matrix* diperluas ke ukuran  $K \times K$  (dimana  $K$  adalah jumlah kelas). Setiap sel  $(i,j)$  menunjukkan jumlah sampel dari kelas actual  $i$  yang diprediksi sebagai kelas  $j$ . Berdasarkan komponen TP, TN, FP, FN, metrik evaluasi utama dapat dirumuskan sebagai berikut:

$$Akurasi = \frac{TP+TN}{TP+TN+FP+FN} \quad 2-1$$

Akurasi memberikan gambaran terkait seberapa besar proporsi prediksi benar terhadap keseluruhan sampel. Akurasi biasa berguna saat kelas di dataset setara, namun akurasi dari klasifikasi mempunyai limitasi, terutama terhadap dataset yang tidak setara.

$$Presisi = \frac{TP}{TP+FP} \quad 2-2$$

Jika terdapat kasus dimana ada lebih dari 2 output kelas, biasanya akurasi klasifikasi tidak memberikan informasi yang akurat terhadap kelas yang diprediksi. Dari kasus tersebut presisi menjadi jawaban yang lebih tepat. Presisi mengukur ketepatan prediksi positif, dan juga tidak bisa menilai kinerja model dengan baik pada kasus negative. Selain itu, presisi juga tidak melakukan perhitungan pada false negative, yaitu kasus yang sebenarnya positif tetapi diprediksi negative oleh model.

$$Recall = \frac{TP}{TP+FN} \quad 2-3$$

Recall mengukur seberapa banyak data positif aktual yang berhasil terdeteksi oleh model.

$$F1-Score = 2 \times \frac{Presisi \times Recall}{Presisi + Recall} \quad 2-4$$

F-1 score merupakan rata-rata harmonic dari presisi dan recall, berguna ketika data tidak seimbang. Selain itu analisis *confusion matrix* pada klasifikasi multi-kelas dapat membantu mengidentifikasi kelas mana yang cenderung sering tertukar(*misclassification*) dan kelas mana yang diprediksi dengan stabil.

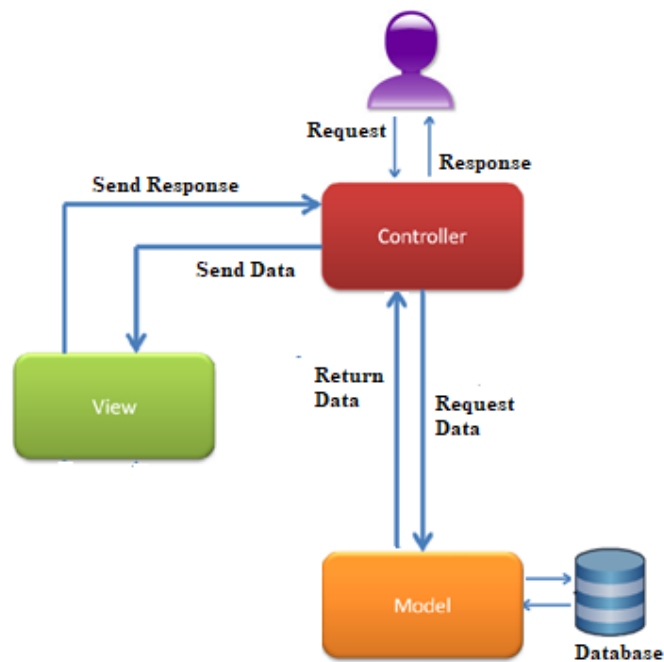
#### 2.1.9 Framework Laravel

*Framework* Laravel merupakan sebuah kerangka kerja (*framework*) aplikasi web berbasis PHP yang menerapkan pola arsitektur Model-View-Controller (MVC). Laravel dibuat untuk memisahkan logika bisnis, tampilan antarmuka, dan pengelolaan data dalam aplikasi web agar pengembangannya lebih terstruktur, mudah dikelola dan diperluas. Laravel memberikan tool set serta arsitektur yang mendukung pengembangan aplikasi web dengan kode yang bersih dan efisien (Khantwal & Sahi, 2023).

Arsitektur Model-View-Controller (MVC) Laravel serta sistem routing-controller-model memberikan struktur yang unggul terhadap PHP native dalam hal *maintainability* dan *scalability* (Ariyanto dkk., 2024). Laravel digunakan sebagai kerangka dasar pengembangan *website*, namun juga cocok diintegrasikan dengan model klasifikasi citra berbasis *machine learning*, melalui API yang menghubungkan *frontend web* dan *backend* inferensi. Struktur utama Laravel meliputi tiga komponen:

- 1) Model bertugas mengelola akses dan interaksi dengan basis data serta representasi domain data
- 2) View bertugas menampilkan antarmuka pengguna dengan template engine Blade.
- 3) Controller bertugas untuk menjadi penghubung Model dan View, menangani permintaan pengguna, memproses logika bisnis, serta mengarahkan pengguna ke tampilan yang sesuai.

Dengan arsitektur ini, dalam penelitian sistem klasifikasi sistem durian, Laravel akan dimanfaatkan sebagai *backend website* untuk menerima gambar yang diupload, memanggil model klasifikasi, dan menampilkan hasil klasifikasi ke pengguna melalui antarmuka yang responsif. Arsitektur *Model-View-Controller* ditunjukkan pada Gambar 2.6 berikut.

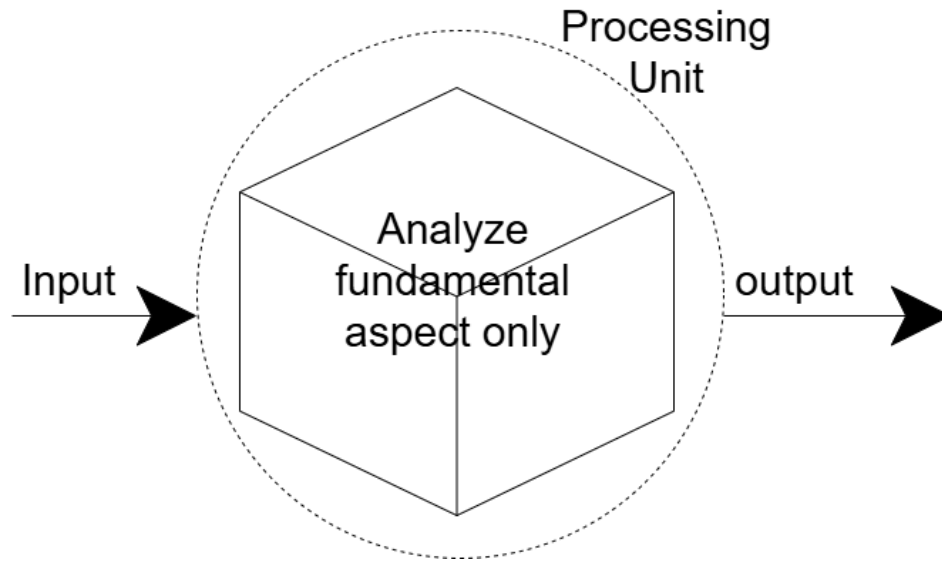


Gambar 2.6 Arsitektur Model-View-Controller

Sumber: (Sinaga dkk., 2021)

#### 2.1.10 Black box testing

Blackbox testing merupakan metode pengujian perangkat lunak yang memiliki fokus dalam fungsionalitas sistem tanpa melihat struktur internal kode program. penguji hanya memberikan input dan melakukan evaluasi output yang dihasilkan. Menurut (Melo dkk., 2021) metode ini berfungsi untuk menguji kualitas perangkat lunak tanpa memperhatikan struktur kode dan program, dengan fokus pada apakah output yang dihasilkan sudah sesuai dengan yang diharapkan atau tidak. Beberapa Teknik yang ada di dalam *black box testing* diantaranya adalah equivalence partitioning, boundary value analysis, decision table testing, dan state transition testing. Representasi *Black Box Testing* ditunjukkan pada Gambar 2.7 berikut.



Gambar 2.7 Representasi *Black box testing*

Sumber: (Khan & Farmeena, 2012)

#### 2.1.11 *User acceptance testing (UAT)*

*User acceptance testing (UAT)* merupakan tahap akhir dalam proses pengujian perangkat lunak yang memiliki tujuan untuk memastikan apakah sistem yang dikembangkan telah memenuhi kebutuhan dan ekspektasi pengguna akhir. Menurut (Pillai & Hemamalini, 2022) prosedur *user acceptance testing* memastikan bahwa persyaratan kualitas perangkat lunak dari klien/pengguna telah terpenuhi, sekaligus mengurangi biaya perbaikan cacat dan waktu pengembangan saat diterapkan lebih awal dalam *Software Development Life Cycle (SDLC)*. *User acceptance testing* memiliki peran penting dikarenakan hasilnya akan menentukan jika sistem sudah siap digunakan secara operasional, atau masih memerlukan perbaikan lanjutan.

UAT pada penelitian ini disusun dengan skala likert menggunakan 4 alternatif jawaban, yaitu Sangat Setuju (skor 4), Setuju (skor 3), Tidak Setuju (skor 2), dan Sangat Tidak Setuju (skor 1). Pilihan netral sengaja tidak digunakan agar responden terdorong untuk memilih salah satu dari pendapat (Prayogo dkk., 2025). Data dari kuesioner ini kemudian dianalisis dengan rumus 2-5 yang digunakan untuk menghitung nilai rata-rata sebagai berikut:

$$\bar{x} = \frac{\sum(\text{Nilai Kuesioner})}{n} \quad 2-5$$

Dimana (nilai kuesioner) merupakan jumlah total skor dari semua responden pada suatu pernyataan, dan n merupakan total responden yang mengisi kuesioner. Selanjutnya, untuk mendapatkan hasil dalam bentuk persentase, digunakan rumus 2-6 berikut:

$$P = \frac{\bar{x}}{4} \times 100\% \quad 2-6$$

P merupakan persentase tingkat penerimaan pengguna pada pernyataan yang telah diujikan. Hasil persentase ini dibandingkan dengan tabel interpretasi skor pada tabel 2.2 untuk menentukan kategori penerimaannya.

Tabel 2.2 Interpretasi Skor

| Kriteria            | Persentase   |
|---------------------|--------------|
| Sangat Tidak Setuju | 0% - 24,99%  |
| Tidak Setuju        | 25% - 49,99% |
| Setuju              | 50% - 74,99% |
| Sangat Setuju       | 75% - 100%   |

## 2.2 Penelitian Terdahulu

Ringkasan penelitian terdahulu berikut menggambarkan hasil yang relevan untuk klasifikasi citra. Penelitian terdahulu yang telah dilakukan sebelumnya berkaitan dengan judul proyek akhir ini, antara lain:

Aherwadi dkk. (2022) melakukan penelitian untuk prediksi kematangan, kualitas, dan umur simpan buah menggunakan pendekatan *deep learning* berbasis citra digital, dengan studi kasus pada buah pisang. Penelitian ini menggunakan dua dataset, yaitu dataset citra pisang yang dikumpulkan sendiri sebanyak 2.100 citra dan dataset publik Fruit360, serta menerapkan teknik *data augmentation* untuk meningkatkan jumlah data hingga 18.900 citra. Dalam proses pengolahan data, fitur visual buah diekstraksi menggunakan arsitektur *Convolutional neural network* pra-latih, termasuk InceptionV3, yang kemudian dimanfaatkan untuk mengidentifikasi tingkat kematangan buah (mentah, matang, dan terlalu matang). Hasil penelitian menunjukkan bahwa pendekatan CNN yang digunakan mampu mencapai akurasi validasi hingga 99,36% pada dataset hasil augmentasi, serta menunjukkan kemampuan yang baik dalam mengenali perubahan visual buah berdasarkan warna, tekstur, dan pola permukaan. Temuan ini menunjukkan bahwa pemanfaatan fitur visual hasil ekstraksi CNN seperti InceptionV3 efektif untuk klasifikasi karakteristik buah, sehingga relevan untuk diterapkan pada penelitian klasifikasi varietas durian yang juga bergantung pada perbedaan visual antar varietas yang relatif kompleks.

Muhammed dkk. (2022) melakukan penelitian untuk klasifikasi dan penilaian kualitas (grading) buah mangga hasil panen menggunakan arsitektur *Convolutional neural network* (CNN). Penelitian ini menggunakan dataset citra delapan varietas mangga, yang diklasifikasikan berdasarkan ciri visual seperti warna, bentuk, ukuran, dan tekstur. Proses pelatihan model dilakukan dengan menerapkan teknik *data augmentation* berupa rotasi, translasi, zooming, shearing, dan *horizontal flip* untuk meningkatkan variasi data. Hasil penelitian menunjukkan bahwa penggunaan arsitektur InceptionV3 mampu mencapai akurasi klasifikasi varietas hingga 99,2% serta akurasi grading sebesar 96,7%, dengan nilai precision, recall, dan F1-score yang tinggi. Hasil tersebut menunjukkan bahwa InceptionV3 efektif dalam mengekstraksi fitur visual buah secara detail, sehingga pendekatan yang digunakan

relevan untuk diterapkan pada penelitian klasifikasi varietas durian yang juga memiliki perbedaan visual antar varietas yang relatif halus.

Arafin dkk. (2022) melakukan penelitian untuk klasifikasi cacat pada permukaan beton menggunakan beberapa arsitektur *Convolutional neural network* (CNN), yaitu VGG-19, ResNet-50, InceptionV3, Xception, dan MobileNetV2. Penelitian ini menggunakan dataset besar yang terdiri dari 4.087 citra retakan (crack) dan 1.100 citra spalling yang dikumpulkan dari berbagai sumber nyata tanpa menerapkan augmentasi data, sehingga merepresentasikan kondisi lapangan secara realistis. Selain itu, dilakukan analisis sensitivitas hiperparameter dengan membandingkan optimizer SGD dan RMSprop serta beberapa nilai *learning rate*. Hasil penelitian menunjukkan bahwa InceptionV3 mencapai performa terbaik dengan akurasi sebesar 91%, precision 83%, dan recall 100% pada konfigurasi optimizer SGD dan *learning rate* 0,001, mengungguli ResNet-50, Xception, MobileNetV2, dan VGG-19. Temuan ini membuktikan bahwa InceptionV3 memiliki kemampuan ekstraksi fitur visual multi-skala yang lebih optimal dalam mengenali pola tekstur kompleks, sehingga relevan untuk diterapkan pada klasifikasi objek biologis dengan karakteristik visual yang berbeda-beda, termasuk klasifikasi varietas durian.

Wulandari dkk. (2025) melakukan penelitian untuk klasifikasi citra cuaca menggunakan algoritma *Convolutional neural network* (CNN) dengan membandingkan tiga arsitektur, yaitu InceptionV3, DenseNet169, dan NASNetMobile. Penelitian ini menggunakan dataset citra cuaca dari Kaggle yang terdiri dari empat kelas, yaitu cloudy, rain, shine, dan sunrise, serta menerapkan tahapan *preprocessing* dan augmentasi data. Hasil penelitian menunjukkan bahwa InceptionV3 memperoleh akurasi sebesar 97,94% pada data latih, 92,34% pada data validasi, dan 93,81% pada data uji, sedangkan DenseNet169 memperoleh akurasi 98,09% pada data latih, 88,46% pada data validasi, dan 92,33% pada data uji. Sementara itu, NASNetMobile mencapai akurasi 96,51% pada data latih, 87,82% pada data validasi, dan 89,97% pada data uji. Berdasarkan hasil tersebut, InceptionV3 menunjukkan performa yang paling konsisten pada seluruh tahap pengujian, sehingga dinilai lebih stabil dalam mengklasifikasikan citra dengan

variasi visual tinggi dan relevan untuk diterapkan pada klasifikasi objek biologis yang kompleks, termasuk klasifikasi varietas durian.

Maeda-Gutiérrez dkk. (2025) melakukan penelitian untuk klasifikasi penyakit daun tanaman talas (*Colocasia esculenta*) menggunakan arsitektur InceptionV3, ResNet-50, dan Vision Transformer (ViT). Penelitian ini menerapkan pendekatan 5-fold cross validation serta evaluasi multi-metrik yang meliputi akurasi, precision, recall, F1-score, dan specificity. Hasil penelitian menunjukkan bahwa InceptionV3 mencapai performa tertinggi dengan akurasi 99,85%, F1-score 99,85%, dan specificity 99,92%, sedangkan ResNet-50 memperoleh akurasi 99,67% dengan F1-score 99,66%, dan Vision Transformer (ViT) mencapai akurasi 99,70% dengan F1-score 99,69%. Perbedaan hasil tersebut menunjukkan bahwa InceptionV3 lebih unggul dalam menangkap fitur visual lokal seperti tekstur dan pola daun dibandingkan ResNet-50 dan ViT, sehingga dinilai lebih efektif untuk klasifikasi objek biologis dengan karakteristik visual yang rumit, termasuk klasifikasi varietas durian. Perbandingan hasil penelitian terdahulu dirangkum pada Tabel 2.3 berikut.

Tabel 2.3 Penelitian Terdahulu

| No | Nama Peneliti        | Judul Penelitian                                                                   | Algoritma   | Hasil                                                                                                                                                                                    |
|----|----------------------|------------------------------------------------------------------------------------|-------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1  | Aherwadi dkk. (2022) | Prediction of Fruit Maturity, Quality, and Its Life Using Deep Learning Algorithms | InceptionV3 | InceptionV3 mampu mencapai akurasi validasi hingga 99,36% pada dataset hasil augmentasi dan efektif dalam mengenali perubahan visual buah berdasarkan warna, tesktur dan pola permukaan. |

| No | Nama Peneliti            | Judul Penelitian                                                                                            | Algoritma                                                 | Hasil                                                                                                                                                                              |
|----|--------------------------|-------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 2  | Muhammad dkk.<br>(2022)  | Classification and Grading of Harvested Mangoes Using <i>Convolutional neural network</i>                   | InceptionV3                                               | Penggunaan InceptionV3 menghasilkan akurasi klasifikasi varietas mangga sebesar 99,2% dan akurasi grading sebesar 96,7%, dengan nilai precision, recall, dan F1-score yang tinggi. |
| 3  | Arafin dkk.<br>(2022)    | Performance Comparison of Multiple <i>Convolutional neural networks</i> for Concrete Defects Classification | VGG-19, ResNet-50, InceptionV3, Xception, dan MobileNetV2 | InceptionV3 mencapai performa terbaik dengan akurasi 91%, precision 83%, dan recall 100% pada konfigurasi <i>optimizer</i> SGD dan <i>learning rate</i> 0,001.                     |
| 4  | Wulandari dkk.<br>(2025) | Comparative Analysis of Weather Image Classification Using CNN Algorithm with InceptionV3,                  | InceptionV3, DenseNet169, dan NASNetMobile                | InceptionV3 menunjukkan performa paling konsisten dengan akurasi 97,94% (data latih), 92,34%                                                                                       |

| No | Nama Peneliti               | Judul Penelitian                                                                                                                                   | Algoritma                                            | Hasil                                                                                                                                    |
|----|-----------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------|
|    |                             | DenseNet169 and NASNetMobile Architecture Models                                                                                                   |                                                      | (validasi), dan 93,81% (data uji), lebih stabil dibandingkan model lainnya.                                                              |
| 5  | Maeda-Gutiérrez dkk. (2025) | AI-Driven Plant Health Assessment: A Comparative Analysis of Inception V3, ResNet-50 and ViT with SHAP for Accurate Disease Identification in Taro | InceptionV3, ResNet-50, dan Vision Transformer (ViT) | InceptionV3 memperoleh performa tertinggi dengan akurasi 99,85%, F1-score 99,85%, dan specificity 99,92%, mengungguli ResNet-50 dan ViT. |