

BAB II

LANDASAN TEORI

2.1 Buah Pisang Cavendish

Pisang Cavendish merupakan varietas pisang yang sangat diminati di pasar global maupun lokal. Varietas ini memiliki ciri khas berupa bentuk buah yang relatif lurus, ukuran yang seragam, serta kulit berwarna hijau saat masih muda dan berubah menjadi kuning terang ketika memasuki masa matang (Luh et al., 2022). Daging buahnya terasa manis dan bertekstur lembut, menjadikannya pilihan utama untuk konsumsi langsung maupun sebagai bahan tambahan dalam berbagai produk olahan.

Dalam sektor pertanian, pisang Cavendish menjadi komoditas andalan karena nilai ekonominya yang tinggi dan permintaannya yang stabil. Keunggulan utamanya terletak pada daya simpan yang cukup lama dibandingkan varietas pisang lain, asalkan penyimpanannya sesuai dengan standar. Seiring waktu, pisang ini akan menunjukkan perubahan fisik, seperti munculnya bintik-bintik kecoklatan, kulit yang menghitam, tekstur yang lebih lembek, serta aroma khas fermentasi yang menandakan kematangan berlebih atau pembusukan (Kesehatan Politeknik Kesehatan Makassar et al., 2024).

Menjaga kesegaran pisang Cavendish menjadi aspek penting dalam rantai distribusi, mulai dari pascapanen hingga sampai ke konsumen. Faktor lingkungan seperti suhu dan kelembaban berperan besar dalam mempertahankan kualitas buah. Untuk itu, diperlukan metode yang efektif dan akurat dalam memantau tingkat kesegaran buah secara cepat (Imansyah, 2024).

Melihat karakteristik fisiknya yang berubah dengan jelas selama proses pematangan, pisang Cavendish menjadi objek yang ideal untuk dikaji menggunakan teknologi pengolahan citra. Penerapan *Convolutional Neural Network (CNN)* dalam analisis citra dapat membantu mendeteksi perubahan visual pada buah secara otomatis, sehingga mendukung efisiensi proses sortir

dan meningkatkan kualitas produk yang sampai ke konsumen(Syaharani et al., 2024).

2.2 Pengolahan Citra Digital

Pengolahan citra digital merujuk pada serangkaian proses untuk mengubah dan menganalisis gambar guna memperoleh informasi yang bermanfaat melalui teknik komputer. Citra yang dimaksud di sini adalah representasi visual yang diubah menjadi format digital agar bisa dianalisis dan diproses lebih lanjut oleh komputer. Dalam konteks penelitian ini, pengolahan citra digunakan untuk mendeteksi tingkat kesegaran buah pisang Cavendish berdasarkan analisis gambar yang diambil dari berbagai sudut dan kondisi pencahayaan yang berbeda(Marpaung et al., n.d.).

Salah satu tujuan utama pengolahan citra adalah untuk meningkatkan kualitas gambar, seperti mengurangi noise, memperjelas detail, serta meningkatkan kontras untuk memudahkan identifikasi objek dalam gambar. Dalam konteks deteksi kesegaran pisang, pengolahan citra dapat digunakan untuk menyorot perubahan fisik pada permukaan kulit pisang, seperti perubahan warna dan munculnya bercak, yang dapat menandakan tingkat kematangan atau kerusakan pada buah.

Beberapa teknik pengolahan citra yang umum digunakan antara lain:

1. Peningkatan Kontras (*Contrast Enhancement*)

Teknik ini digunakan untuk meningkatkan perbedaan antara objek dan latar belakang gambar, sehingga objek yang ingin dianalisis dapat lebih mudah dikenali.

2. Segmentasi Citra (*Image Segmentation*)

Segmentasi citra merupakan teknik yang digunakan untuk memisahkan objek yang relevan dari latar belakang. Pada penelitian ini, segmentasi digunakan untuk mengekstraksi pisang dari gambar dan mengabaikan bagian lain yang tidak relevan.

3. Deteksi Tepi (*Edge Detection*)

Deteksi tepi adalah teknik untuk menemukan batas antara objek dengan latar belakang, yang berguna untuk mendeteksi batas-batas visual pada permukaan pisang, seperti perubahan pada kulit yang menunjukkan kerusakan atau kematangan berlebih.

4. Transformasi Warna (*Color Transformation*)

Proses ini digunakan untuk mengubah ruang warna dalam citra agar bisa lebih jelas terlihat perbedaan warna yang terjadi pada kulit pisang seiring proses pematangan atau pembusukan.

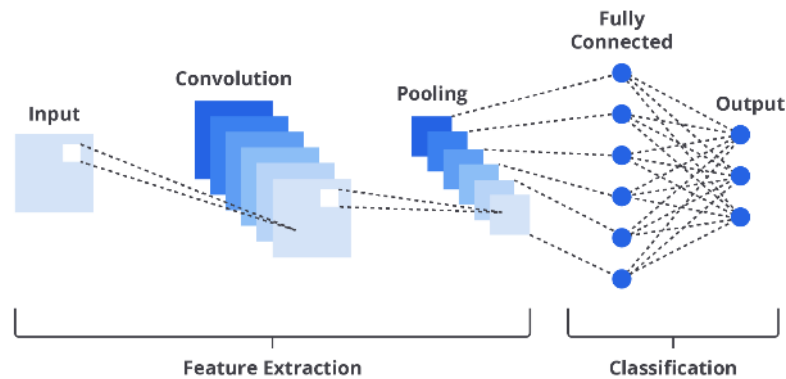
5. Pengenalan Pola (*Pattern Recognition*)

Setelah citra diproses, teknik ini digunakan untuk mengidentifikasi pola atau fitur tertentu dalam gambar yang dapat digunakan untuk mengklasifikasikan tingkat kesegaran pisang.

6. Melalui teknik-teknik pengolahan citra tersebut, informasi yang terkandung dalam gambar dapat diekstraksi dan dianalisis secara lebih efektif. Dalam penelitian ini, pengolahan citra digital menjadi langkah awal yang penting sebelum gambar diproses lebih lanjut menggunakan metode machine learning, seperti *Convolutional Neural Network (CNN)*, untuk klasifikasi tingkat kesegaran buah pisang.

2.3 Convolutional Neural Network (CNN)

Convolutional Neural Network (CNN) merupakan jenis arsitektur *jaringan saraf buatan* yang dirancang khusus untuk memproses data berbentuk gambar atau data spasial lainnya. *CNN* memiliki kemampuan yang sangat baik dalam mengenali pola-pola kompleks dalam citra digital, yang membuatnya sangat efektif dalam aplikasi seperti *pengenalan objek*, *klasifikasi gambar*, dan *deteksi objek* (Setya Nugraha & Hermawan, 2023). *CNN* berbeda dari *jaringan saraf tiruan* biasa (*feedforward neural networks*) karena dilengkapi dengan lapisan konvolusi yang secara khusus digunakan untuk menangkap fitur lokal dalam citra.

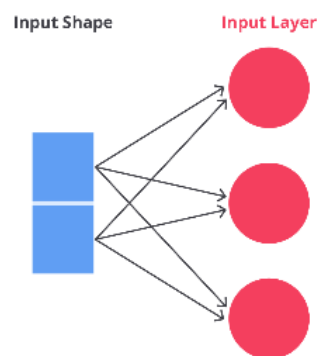


Gambar 2. 1 Alur Kerja CNN

CNN umumnya terdiri dari beberapa lapisan utama yang bekerja bersama untuk mengekstrak fitur dan melakukan klasifikasi, sebagai berikut:

1. Input Layer

Lapisan input merupakan bagian awal dari sebuah jaringan neural. Peran utamanya adalah menerima data awal yang akan diproses oleh jaringan. Setiap unit (neuron) dalam lapisan ini merepresentasikan satu fitur atau atribut dari data yang diberikan



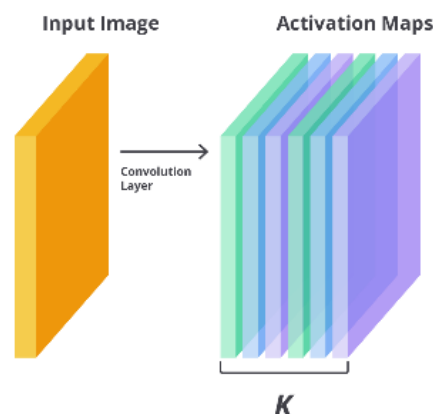
Gambar 2. 2 Input Layer

Jenis data yang masuk ke lapisan ini dapat bervariasi, tergantung pada jenis permasalahan yang ingin diselesaikan. Contohnya, dalam tugas pengenalan

gambar, input bisa berupa nilai-nilai piksel, sedangkan pada pengolahan teks, input dapat berupa representasi vektor dari kata atau karakter. Lapisan input umumnya tidak melakukan proses perhitungan yang kompleks karena fungsinya hanya menyalurkan data ke lapisan-lapisan selanjutnya. Jumlah neuron dalam lapisan ini biasanya ditentukan oleh jumlah fitur yang dimiliki oleh data.

2. Convolutional Layers (Lapisan Konvolusi)

Lapisan konvolusi merupakan bagian awal dari arsitektur Convolutional Neural Network (CNN) yang berperan untuk mengekstraksi fitur dari gambar input. Pada tahap ini, hubungan antar piksel dalam gambar tetap dipertahankan, sehingga pola lokal dapat dikenali dengan baik.



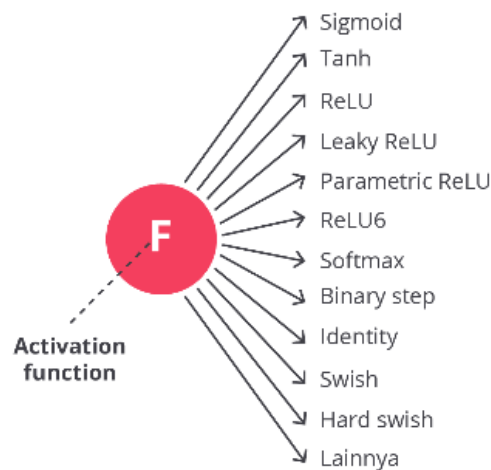
Gambar 2. 3 Convolutional Layers

Proses konvolusi melibatkan operasi matematika antara dua komponen utama: matriks citra dan filter (atau kernel). Filter ini bergerak melintasi gambar dan melakukan perhitungan untuk menghasilkan activation map atau feature map, yang merepresentasikan fitur penting dari gambar tersebut, seperti tepi, tekstur, atau pola tertentu. Setiap filter belajar mengenali karakteristik tertentu, dan semakin dalam lapisan konvolusi, fitur yang diidentifikasi menjadi semakin kompleks.

3. Activation Layers

Setelah setiap lapisan konvolusi dalam CNN, biasanya diterapkan fungsi aktivasi non-linear seperti ReLU, ELU, atau Leaky ReLU. Fungsi aktivasi ini tidak hanya mengubah output dari lapisan sebelumnya, tetapi juga menambahkan kemampuan non-linear ke dalam jaringan, yang penting untuk mengenali pola-pola kompleks dalam data.

Meskipun sering disebut sebagai "lapisan", activation layer sebenarnya tidak memiliki parameter yang dipelajari seperti bobot atau bias. Oleh karena itu, kadang-kadang tidak ditampilkan secara eksplisit dalam diagram arsitektur jaringan karena keberadaannya dianggap sudah melekat pada setiap proses konvolusi.

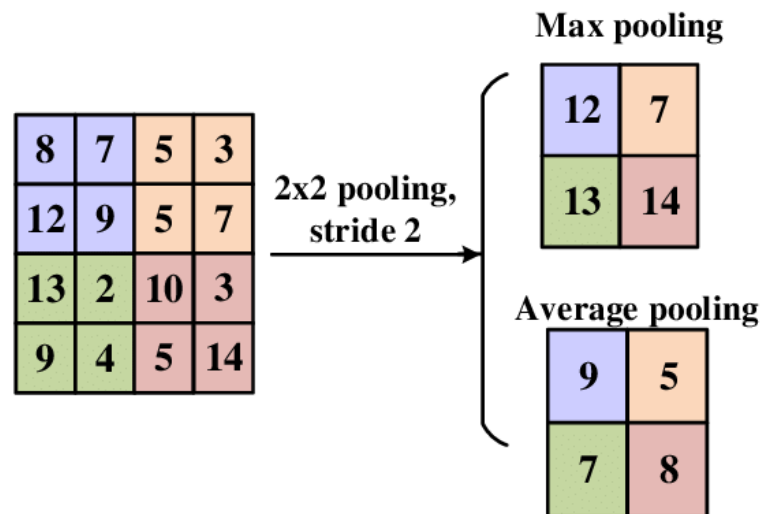


Gambar 2. 4 Activation Layers

Dalam sebuah jaringan neural, setiap lapisan terdiri dari sejumlah neuron yang memproses data masuk dan menghasilkan output. Namun, agar jaringan bisa belajar dari data dengan struktur non-linear, diperlukan fungsi aktivasi untuk mengubah output tersebut. Fungsi ini memungkinkan jaringan untuk menangani berbagai permasalahan yang tidak dapat diselesaikan hanya dengan operasi linear saja.

4. Pooling Layer

Pooling layer merupakan salah satu komponen penting dalam Convolutional Neural Network (CNN) yang berfungsi untuk mengecilkan dimensi spasial dari hasil ekstraksi fitur. Tujuan utama dari proses ini adalah untuk mengurangi jumlah parameter dan beban komputasi, membantu mencegah overfitting, serta mempercepat proses pelatihan model.



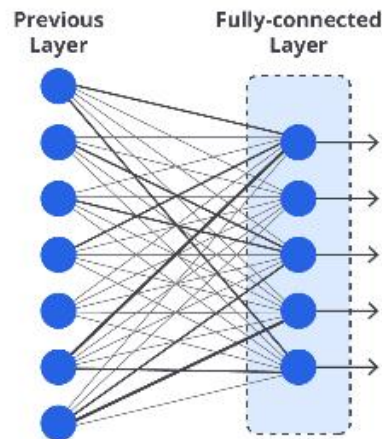
Gambar 2. 5 Pooling layer

Cara kerjanya cukup sederhana: pooling akan mengambil potongan kecil dari representasi gambar (biasanya ukuran 2x2 atau 3x3) dan merangkum informasi di dalamnya menjadi satu nilai. Proses ini bisa dilakukan dengan beberapa metode, yang paling umum adalah max pooling (mengambil nilai tertinggi) dan average pooling (mengambil nilai rata-rata). Dengan menyaring informasi yang paling penting, pooling layer membantu jaringan fokus pada fitur yang paling relevan, sekaligus membuat model lebih tahan terhadap pergeseran kecil dalam gambar.

5. Fully Connected Layers

Fully connected layer, atau sering disebut dense layer, merupakan bagian akhir dari arsitektur jaringan neural, terutama dalam CNN. Lapisan ini biasanya ditempatkan setelah serangkaian lapisan konvolusi dan pooling.

Dalam lapisan ini, setiap neuron terhubung langsung ke seluruh neuron pada lapisan sebelumnya, dengan setiap koneksi memiliki bobot yang akan disesuaikan selama proses pelatihan.



Gambar 2. 6 Fully Connected layers

Tujuan utama dari fully connected layer adalah mengintegrasikan semua fitur yang telah diekstraksi oleh lapisan sebelumnya dan menggunakannya untuk menghasilkan prediksi akhir. Misalnya, dalam klasifikasi gambar, lapisan ini bertugas untuk mengolah representasi fitur dan menentukan kelas mana yang paling sesuai berdasarkan informasi tersebut. Kekuatan lapisan ini terletak pada kemampuannya dalam membuat keputusan berdasarkan kombinasi kompleks dari fitur-fitur input yang telah diproses sebelumnya.

6. Output Layers

Output layer merupakan lapisan terakhir dalam sebuah arsitektur neural network, termasuk Convolutional Neural Network (CNN). Lapisan ini bertanggung jawab untuk menghasilkan prediksi akhir berdasarkan informasi yang telah diproses dan dianalisis oleh lapisan-lapisan sebelumnya.

Jumlah neuron pada output layer bergantung pada jenis tugas yang dijalankan oleh jaringan. Misalnya, untuk klasifikasi citra dengan 4 kelas, output layer akan memiliki 4 neuron masing-masing mewakili satu kelas. Aktivasi yang digunakan juga menyesuaikan, seperti softmax untuk multi-class classification atau sigmoid untuk binary classification.

2.3.1 Keunggulan CNN dalam Pengolahan Citra

1. Pengenalan Pola Otomatis

CNN dapat mengidentifikasi pola-pola visual yang relevan (seperti warna, tekstur, dan bentuk) tanpa memerlukan pemrograman manual fitur. Dengan demikian, *CNN* dapat bekerja secara otomatis pada gambar tanpa perlu prosedur ekstraksi fitur yang rumit (Arkadia et al., 2021)

2. Kebebasan dari Proses *Pre-processing* Manual

Karena *CNN* dapat belajar untuk mengekstraksi fitur penting dari citra secara otomatis, tidak diperlukan banyak *pre-processing* gambar. Ini mengurangi waktu dan usaha yang diperlukan untuk menyiapkan data.

3. Invarian terhadap Posisi dan Skala

CNN dapat mengenali objek atau pola dalam gambar meskipun ada perubahan posisi, skala, atau rotasi. Hal ini membuat *CNN* sangat efektif dalam aplikasi dunia nyata di mana objek atau fitur dalam gambar dapat muncul dalam berbagai orientasi.

2.3.2 Penerapan CNN dalam Deteksi Kesegaran Pisang

Dalam konteks penelitian ini, *CNN* digunakan untuk mendeteksi tingkat kesegaran buah pisang Cavendish dengan menganalisis gambar pisang yang diambil pada berbagai tingkat kematangan (Fatahna et al., 2025). *CNN* akan mempelajari pola-pola visual yang menunjukkan perubahan warna kulit, tekstur, dan keberadaan bercak hitam atau coklat yang menandakan tingkat kematangan atau kerusakan buah. Dengan menggunakan dataset gambar yang berlabel, *CNN* dapat dilatih untuk membedakan pisang yang segar, matang, dan busuk.

Model *CNN* yang dilatih dapat diintegrasikan dalam sistem deteksi berbasis website untuk memungkinkan pengguna mengunggah gambar buah pisang

dan mendapatkan hasil prediksi tingkat kesegaran secara otomatis. Hal ini memungkinkan proses pengujian kualitas yang cepat, efisien, dan lebih konsisten dibandingkan dengan metode inspeksi manual.

2.4 Streamlit Sebagai Platform Website

Streamlit adalah sebuah *framework open-source* yang memungkinkan pengembang untuk dengan cepat membuat aplikasi web interaktif berbasis Python, tanpa memerlukan keterampilan khusus dalam pengembangan *front-end* atau *back-end*. *Streamlit* dirancang untuk menyederhanakan proses pembuatan aplikasi web, memungkinkan pengembang untuk fokus pada pembuatan aplikasi fungsional dengan sedikit konfigurasi. Penggunaan *Streamlit* sangat cocok untuk aplikasi yang berhubungan dengan data, seperti sistem deteksi berbasis *machine learning* atau *visualisasi data* (Christian & Idrus, 2023).

2.4.1 Keunggulan Streamlit dalam Pengembangan Website

1. Kemudahan penggunaan *streamlit* memungkinkan pengembang untuk membuat antarmuka pengguna (*UI*) hanya dengan menulis kode Python. Hal ini sangat berguna untuk pengembang yang tidak memiliki pengalaman dalam pengembangan web tradisional, seperti *HTML*, *CSS*, atau *JavaScript*. Dengan *Streamlit*, pembuatan *UI* menjadi jauh lebih sederhana dan cepat, hanya dengan menggunakan *widget* seperti tombol, *slider*, dan grafik.
2. Integrasi dengan model *machine learning* salah satu keunggulan utama *Streamlit* adalah kemampuannya untuk secara langsung mengintegrasikan model *machine learning* atau *deep learning* yang telah dibangun menggunakan pustaka Python seperti *TensorFlow*, *Keras*, atau *scikit-learn*. Model-model ini dapat dengan mudah di-load ke dalam aplikasi *Streamlit* untuk memberikan fungsionalitas seperti prediksi atau klasifikasi berdasarkan input pengguna.
3. Interaktivitas tinggi dengan *streamlit*, aplikasi dapat dibuat interaktif dengan mudah. Pengguna dapat memberikan input, seperti mengunggah gambar, memilih parameter melalui *slider*, atau memasukkan data lainnya. Hasil dari input tersebut dapat ditampilkan

secara langsung pada antarmuka pengguna tanpa perlu memuat ulang halaman, membuat aplikasi lebih responsif dan *user-friendly*.

4. Penyederhanaan proses *deployment Streamlit* memudahkan proses *deployment* aplikasi web. Pengembang cukup menjalankan aplikasi pada *server* lokal, dan aplikasi dapat langsung diakses melalui *browser*. Selain itu, *Streamlit* juga mendukung integrasi dengan layanan *cloud* seperti *Heroku*, *AWS*, atau *Google Cloud Platform*, yang memungkinkan aplikasi dapat diakses secara publik di internet dengan mudah.
5. Visualisasi data yang cepat dan efisien Salah satu fitur utama *Streamlit* adalah kemampuannya untuk melakukan visualisasi data secara cepat dan efisien. Dengan menggunakan pustaka visualisasi seperti *Matplotlib*, *Plotly*, atau *Altair*, pengembang dapat menyajikan data dalam berbagai bentuk, seperti grafik atau diagram. Hal ini sangat bermanfaat untuk aplikasi yang memerlukan tampilan hasil analisis atau prediksi, seperti visualisasi kesegaran buah pisang yang terdeteksi oleh model *CNN*.
6. Pengembangan dan debugging cepat *streamlit* menyediakan pengalaman pengembangan yang sangat cepat karena setiap perubahan dalam kode akan langsung diterapkan dan dapat dilihat di *browser*. Pengembang tidak perlu melakukan proses *refresh* atau *reload* halaman secara manual. Hal ini memudahkan proses *debugging* dan eksperimen dengan aplikasi secara langsung.

2.4.2 Penerapan Streamlit pada Sistem Deteksi Kesegaran Pisang

Untuk aplikasi deteksi kesegaran pisang menggunakan *CNN*, *Streamlit* akan berfungsi sebagai platform untuk membuat antarmuka pengguna yang memungkinkan pengguna untuk mengunggah gambar pisang. Begitu gambar diunggah, aplikasi akan memproses gambar tersebut menggunakan model *CNN* untuk menentukan tingkat kesegaran (*segar*, *matang*, *mentah*, atau *busuk*) dan menampilkan hasil prediksi kepada pengguna.

Dengan menggunakan *Streamlit*, sistem deteksi ini bisa lebih interaktif dan mudah diakses oleh pengguna dari berbagai latar belakang. Platform berbasis *website* ini juga akan memungkinkan pengembangan lebih lanjut, seperti penambahan fitur lainnya untuk memvisualisasikan hasil analisis, memberikan rekomendasi penyimpanan, atau memungkinkan pengguna untuk mengunggah *dataset* untuk pelatihan model secara personal.

2.5 Framework, Bahasa Pemrograman, Pustaka Python, dan Tools

Dalam penelitian ini, berbagai *framework* dan alat digunakan untuk pengolahan citra, pelatihan model *machine learning* (terutama *CNN*), serta pengembangan aplikasi *website*. Berikut adalah *framework* dan alat yang digunakan:

2.5.1 Framework

Framework digunakan sebagai fondasi dalam membangun model pembelajaran mesin dan mengembangkan aplikasi *website* berbasis *Python*. *Framework* yang digunakan dalam penelitian ini meliputi:

1. TensorFlow

TensorFlow merupakan *framework open-source* yang dikembangkan oleh Google untuk membangun dan melatih model *machine learning*, termasuk *deep learning*. Dalam penelitian ini, *TensorFlow* digunakan untuk membangun arsitektur *CNN*, melatih model dengan *dataset* citra buah pisang, serta melakukan evaluasi kinerja model. *TensorFlow* dipilih karena kemampuannya dalam menangani komputasi numerik tingkat tinggi dan skalabilitasnya dalam pelatihan model besar.

2. Keras

Keras adalah *Application Programming Interface (API)* tingkat tinggi yang berjalan di atas *TensorFlow*. *Keras* menyederhanakan proses pembuatan dan pelatihan jaringan saraf tiruan dengan menyediakan antarmuka yang lebih intuitif. Dalam penelitian ini, *Keras* digunakan untuk mendefinisikan arsitektur *CNN* secara modular dan memudahkan eksperimen dengan berbagai konfigurasi model.

3. Streamlit

Streamlit adalah *framework open-source* untuk membangun aplikasi *website* interaktif secara cepat menggunakan *Python*. *Streamlit* memungkinkan integrasi langsung antara model machine learning dengan tampilan antarmuka *website*. Dalam penelitian ini, *Streamlit* digunakan untuk mengembangkan aplikasi *website* yang berfungsi sebagai media input citra dan menampilkan hasil prediksi kesegaran buah pisang secara *real-time*.

2.5.2 Bahasa Pemrograman

Python

Python dipilih sebagai bahasa utama karena memiliki ekosistem pustaka yang sangat kaya untuk pengembangan *machine learning*, pengolahan citra, serta pengembangan aplikasi *website*. *Python* juga menawarkan sintaks yang sederhana dan fleksibel, sehingga mempercepat proses pengembangan dan memudahkan dokumentasi(Sanner, n.d.).

2.5.3 Pustaka Python

Pustaka *Python* yang digunakan dalam penelitian ini mendukung proses *pre-processing* data, pelatihan model, analisis performa, serta visualisasi data. Pustaka yang digunakan antara lain:

1. OpenCV

OpenCV (Open Source Computer Vision Library) digunakan untuk berbagai operasi pengolahan citra seperti *resizing*, augmentasi gambar, konversi warna, dan manipulasi gambar lainnya sebelum data digunakan untuk melatih model *CNN*.

2. Pillow (*PIL*)

Pillow merupakan pustaka *Python* untuk memproses gambar digital. Dalam penelitian ini, *Pillow* digunakan untuk membuka, memanipulasi, dan menyimpan citra selama proses *pre-processing*.

3. Matplotlib

Matplotlib adalah pustaka untuk membuat visualisasi data dalam bentuk grafik dua dimensi. *Matplotlib* digunakan untuk memvisualisasikan perkembangan akurasi dan *loss* selama proses pelatihan model.

4. NumPy

NumPy adalah pustaka *Python* untuk komputasi numerik. *NumPy* digunakan dalam penelitian ini untuk memproses *array* data gambar dan mendukung berbagai operasi matematis pada *dataset*.

5. Pandas

Pandas digunakan untuk memanipulasi dan menganalisis data dalam bentuk tabel (*dataframe*), terutama jika diperlukan dalam pengelolaan *metadata* gambar atau analisis tambahan.

6. Seaborn

Seaborn merupakan pustaka visualisasi statistik berbasis *Matplotlib* yang mempermudah pembuatan grafik yang lebih informatif dan menarik, seperti *plotting confusion matrix* untuk evaluasi performa model.

2.5.4 Tools

Tools yang digunakan dalam penelitian ini berfungsi untuk menunjang proses pengembangan, pelatihan model, eksperimen, serta *deployment* aplikasi. *Tools* yang digunakan meliputi:

1. Google Colab

Google Colab adalah platform *cloud* berbasis *Jupyter Notebook* yang disediakan oleh Google, memungkinkan pengguna untuk menulis dan menjalankan kode *Python* secara gratis dengan dukungan *GPU* dan *TPU*. Dalam penelitian ini, *Google Colab* digunakan untuk proses pelatihan model *CNN*, dengan memanfaatkan akses *GPU* gratis untuk mempercepat komputasi.

2. Anaconda

Anaconda adalah distribusi *Python* yang dirancang untuk sains data dan *machine learning*. *Anaconda* menyediakan manajemen lingkungan (*environment management*) yang memudahkan pengelolaan berbagai versi pustaka dan *dependensi* proyek. Dalam penelitian ini, *Anaconda* digunakan untuk mengatur lingkungan pengembangan lokal yang stabil dan terisolasi.

3. Jupyter Notebook

Jupyter Notebook adalah aplikasi *website* yang memungkinkan pembuatan dokumen berisi kode interaktif, visualisasi, dan narasi teks. Dalam penelitian ini, *Jupyter Notebook* digunakan untuk menyusun, menjalankan, dan mendokumentasikan eksperimen secara sistematis.

4. Google Drive

Google Drive digunakan sebagai media penyimpanan *dataset*, file model, dan hasil eksperimen selama penggunaan *Google Colab*, sehingga memudahkan sinkronisasi dan pengelolaan file penelitian.

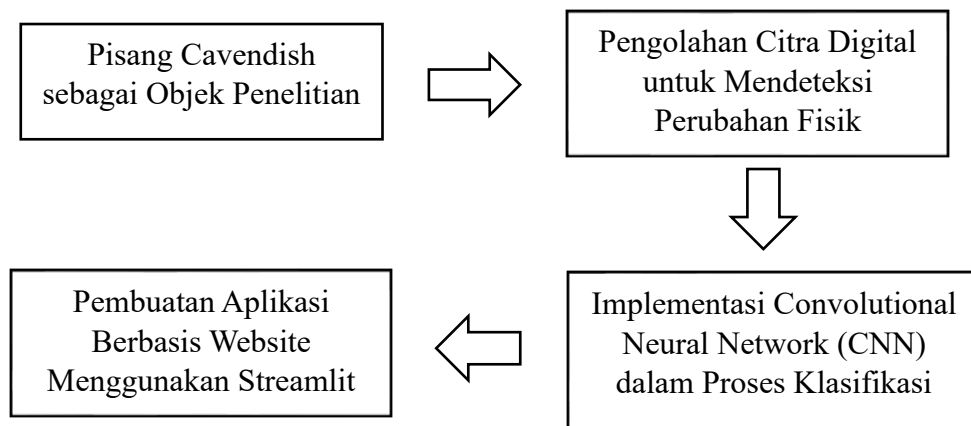
2.6 Metode Pengembangan Perangkat Lunak

2.6.1 Definisi Software Development Life Cycle (SDLC)

Software Development Life Cycle (SDLC) adalah kerangka proses yang mendefinisikan tahapan-tahapan terstruktur untuk membangun perangkat lunak secara sistematis mulai dari analisis kebutuhan hingga pemeliharaan. Tujuan SDLC adalah menjamin bahwa perangkat lunak dibangun tepat guna (sesuai kebutuhan), tepat waktu, terukur kualitasnya, dan terkelola risikonya. Dalam penelitian ini, SDLC digunakan sebagai acuan metodologis agar pengembangan aplikasi deteksi kesegaran pisang berjalan terarah, terdokumentasi, dan dapat direplikasi (Sagita & Surbakti, n.d.).

2.7 Kerangka Berfikir

1. Pisang Cavendish sebagai Objek Penelitian
 - a. Pisang Cavendish memiliki karakteristik fisik yang berubah seiring tingkat kematangan (warna kulit, tekstur, bintik-bintik coklat).
 - b. Perubahan ini bisa dimanfaatkan sebagai indikator visual untuk mendeteksi tingkat kesegaran pisang.
2. Pengolahan Citra Digital untuk Mendeteksi Perubahan Fisik
 - a. Gambar pisang diambil dari berbagai sudut dan kondisi pencahayaan.
 - b. Teknik pengolahan citra (peningkatan kontras, segmentasi, deteksi tepi, transformasi warna, dan pengenalan pola) digunakan untuk mengekstraksi informasi visual terkait perubahan fisik pisang.
3. Implementasi *Convolutional Neural Network (CNN)* dalam Proses Klasifikasi
 - a. *CNN* digunakan untuk mempelajari dan mengenali pola-pola visual pada kulit pisang, seperti perubahan warna dan munculnya bercak.
 - b. *CNN* melakukan klasifikasi tingkat kesegaran pisang (segar, matang, busuk) secara otomatis berdasarkan citra yang sudah diproses.
4. Pembuatan Aplikasi Berbasis *Website* Menggunakan *Streamlit*
 - a. *Streamlit* digunakan untuk membangun antarmuka aplikasi yang memungkinkan pengguna mengunggah gambar pisang.
 - b. Gambar yang diunggah diproses oleh model *CNN* untuk menampilkan hasil prediksi tingkat kesegaran secara interaktif dan *real-time*.
 - c. *Streamlit* mempermudah integrasi model *machine learning* ke dalam aplikasi *website* tanpa perlu keahlian *web development* lanjutan.



Gambar 2. 7 Kerangka Berfikir

2.7 Penelitian Terdahulu

Tabel 2. 1 Penelitian Terkait

No	Peneliti, Tahun	Objek Penelitian	Motode	Hasil Penelitian	Kelebihan & Kekurangan
1	Fitriani et al. (2021)	Deteksi Kematangan Buah Pisang Menggunakan CNN	CNN (Convolutional Neural Network)	Akurasi mencapai 92% dalam membedakan pisang matang dan mentah	Akurasi tinggi, namun belum berbasis web
2	Prasetyo & Nugroho (2020)	Klasifikasi Buah Apel Menggunakan Deep Learning	CNN+Augmentasi Data	Model mampu membedakan 3 kelas dengan akurasi 89%	Fokus pada apel, metode dapat diadaptasi
3	Sari et al. (2022)	Sistem Deteksi Buah Busuk dengan Deep Learning	CNN+MobileNet	Akurasi 90% dan ringan dijalankan pada mobile	Kompatibel di mobile, namun dataset terbatas
4	Rachmawati et al. (2021)	Implementasi CNN pada Website Deteksi	CNN+Streamlit	Deteksi sayuran dengan akurasi 88%,	Berbasis web, namun fokus pada sayuran

		Sayuran Segar		sistem berbasis web	
5	Gunawan et al. (2023)	Klasifikasi Kesegaran Buah Menggunakan Transfer Learning	Transfer Learning (ResNet50)	Akurasi 94% untuk 3 kelas buah (segar, hampir busuk, busuk)	Akurasi tinggi, namun tidak spesifik pada pisang
6	Sharma & Gupta (2021)	Fruit Freshness Detection Using Deep Learning	CNN+ VGG16	Akurasi 91%, mendeteksi kesegaran berbagai buah	Metode kuat, tapi belum dikembangkan untuk web app

2.7.1 Persamaan Penelitian Terdahulu

Penelitian terkait umumnya memanfaatkan *deep learning*, khususnya *CNN*, untuk mendeteksi atau mengklasifikasi tingkat kematangan atau kesegaran buah. *CNN* sering digunakan secara langsung atau dikombinasikan dengan *MobileNet*, *VGG16*, atau *Transfer Learning*. Tujuannya adalah meningkatkan akurasi dan mengotomatisasi penilaian kondisi buah.

2.7.2 Perbedaan Penelitian Terdahulu

Perbedaan dari penelitian terdahulu terletak pada objek, metode, dan implementasinya. Beberapa penelitian seperti Fitriani et al. (2021) fokus pada pisang, sedangkan lainnya meneliti apel, sayuran, atau berbagai jenis buah. Metode yang digunakan bervariasi, mulai dari *CNN* murni, kombinasi dengan *MobileNet* dan *VGG16*, hingga *Transfer Learning* seperti *ResNet50*. Selain itu, hanya Rachmawati et al. (2021) yang mengembangkan sistem berbasis web, sementara yang lain belum mengarah ke implementasi aplikasi web. Ada juga yang dioptimalkan untuk perangkat mobile seperti penelitian Sari et al. (2022).

Teknologi *deep learning* seperti *CNN* telah digunakan untuk mendeteksi kesegaran buah, namun masih sedikit yang fokus pada buah pisang atau berbasis web. Ini membuka peluang untuk mengembangkan sistem deteksi kesegaran pisang yang akurat dan mudah diakses.