

BAB 2

TINJAUAN PUSTAKA

2.1 Penelitian Terdahulu

Dalam upaya mengembangkan sistem klasifikasi kualitas tebu berdasarkan tingkat kebersihan tebu menggunakan *deep learning*, penting untuk memperhatikan penelitian-penelitian terdahulu yang relevan. Melalui kajian literatur terhadap studi-studi sebelumnya, diperoleh landasan yang kuat untuk mendukung penelitian ini. Berikut merupakan penelitian terdahulu yang relevan dengan fokus dan ruang lingkup penelitian ini.

Penelitian dilakukan oleh Yanto et al. (2021) dengan judul “Klasifikasi Kematangan Buah Jeruk Manis Menggunakan *Convolutional Neural Network* (CNN)”. Penelitian ini bertujuan untuk mengatasi penilaian tingkat kematangan buah jeruk manis yang selama ini masih bersifat subjektif dan tidak konsisten dengan mengembangkan sistem klasifikasi otomatis menggunakan CNN yang mampu menilai kematangan buah berdasarkan warna dan tekstur citra visual. Model CNN dilatih menggunakan dataset gambar jeruk manis yang telah diklasifikasikan menurut tingkat kematangan dan diuji untuk mengklasifikasikan gambar baru dengan hasil yang cukup akurat. Penelitian ini menyimpulkan bahwa pendekatan berbasis CNN berpotensi besar untuk diterapkan dalam bidang pertanian guna meningkatkan efisiensi dan objektivitas proses penilaian kualitas buah.

Penelitian dilakukan oleh Nur & Agustin (2020) dengan judul “Klasifikasi Kematangan Tebu Menggunakan Citra Visual Batang dengan Algoritma Naïve Bayes”. Penelitian ini berawal dari permasalahan rendahnya produktivitas tebu di Indonesia akibat ketidaktepatan dalam menentukan waktu panen. Sebagai solusi, penulis mengembangkan sistem klasifikasi kematangan tebu dengan memanfaatkan citra visual batang tebu. Data citra yang digunakan sebanyak 400 gambar, dibagi menjadi data pelatihan dan pengujian. Sistem menggunakan algoritma Naïve Bayes untuk mengklasifikasikan kematangan berdasarkan tekstur batang, dan mencapai akurasi 73% untuk 30 data pengujian serta 71% untuk 100 data pengujian. Hasil ini menunjukkan bahwa pendekatan visual berbasis tekstur dapat membantu petani

dalam menentukan waktu panen secara lebih objektif.

Penelitian dilakukan oleh Nurmuslimah (2020) dengan judul “Aplikasi Metode Fuzzy Mamdani untuk Pemilihan Tebu Berkualitas pada Produksi Gula”. Penelitian ini mengangkat masalah penilaian kualitas tebu yang masih dilakukan secara manual dan cenderung subjektif di industri gula. Sebagai alternatif, penulis menerapkan metode fuzzy Mamdani sebagai sistem pendukung keputusan untuk klasifikasi mutu tebu berdasarkan parameter kemanisan, kebersihan, dan kesegaran. Sistem ini dikembangkan dengan bahasa pemrograman Python dan diuji menggunakan data simulasi. Hasil pengujian menunjukkan bahwa sistem mampu memberikan rekomendasi mutu tebu secara akurat dan konsisten. Penelitian ini menyimpulkan bahwa fuzzy Mamdani efektif untuk digunakan dalam penilaian mutu hasil pertanian dan menyarankan penerapannya di lingkungan produksi nyata dengan cakupan parameter yang lebih luas.

Penelitian oleh Sary et al., (2025) berjudul “Implementasi Algoritma YOLOv11 dan Roboflow untuk Deteksi Tingkat Kematangan Anggur Berbasis Web” mengembangkan sistem deteksi kematangan buah anggur berbasis web menggunakan YOLOv11s dan Roboflow. Dataset yang digunakan sebanyak 897 citra diperbanyak menjadi 6.135 citra melalui *preprocessing* dan augmentasi, kemudian dilatih selama 200 *epoch* menggunakan *Google colab*. Hasil penelitian menunjukkan performa yang baik dengan *precision* 0,98, *recall* 0,95, mAP 0,84, serta waktu deteksi rata-rata 1,02 detik per citra. Sistem mampu melakukan deteksi semi *real-time*, meskipun kelas anggur setengah matang masih sulit dibedakan karena kemiripan visual dengan kelas lainnya.

Penelitian dilakukan oleh Alfiano & Rahayu (2024) dengan judul “Implementasi Algoritma *Deep learning* YOLO (*You only look once*) untuk Deteksi Kualitas Kentang Segar dan Busuk Secara *Real time*”. Penelitian ini bertujuan mengatasi keterbatasan metode manual yang rawan kesalahan manusia dalam penilaian mutu. Sebagai solusinya, peneliti menerapkan algoritma YOLOv8n yang berbasis *deep learning* dan CNN untuk membangun sistem deteksi objek secara *real-time*. Sistem diimplementasikan dalam web berbasis Flask yang terhubung ke *webcam*. Pengujian dilakukan terhadap uji data untuk mengukur kinerja model deteksi. Hasil pengujian menunjukkan akurasi 100%, presisi 99,9%, *recall* 100%,

dan mAP50 sebesar 99,5%. Berdasarkan analisis tersebut, disimpulkan bahwa algoritma YOLOv8n ini efektif dan direkomendasikan untuk meningkatkan efisiensi di sektor pertanian dan industri pangan.

Penelitian dilakukan oleh Ayunadin (2024) dengan judul “Implementasi *Deep learning* pada deteksi Jenis tanah berdasarkan warna menggunakan YOLO (Studi Kasus Kabupaten Gorong) ”. Penelitian ini bertujuan untuk menggantikan metode identifikasi manual yang lambat dan subjektif dengan sistem berbasis *deep learning* yang cepat dan akurat. Dataset yang digunakan terdiri dari 900 gambar tanah dengan tiga kelas yang dikumpulkan secara langsung di lapangan. Model diuji melalui konfusi matriks dan mencapai akurasi sebesar 80%. Penelitian ini menunjukkan bahwa pendekatan berbasis CNN dalam arsitektur YOLOv8 efektif dalam menyelesaikan permasalahan klasifikasi visual berbasis warna.

Pada proyek akhir ini, penelitian yang akan dilakukan untuk mengembangkan sistem klasifikasi kualitas tebu berdasarkan tingkat kebersihan secara *real-time* dengan memanfaatkan teknologi *deep learning*, khususnya algoritma deteksi objek YOLOv11s. Sistem ini bekerja dengan menangkap citra batang tebu menggunakan kamera yang dipasang di atas *conveyor* saat tebu masuk ke dalam pabrik. Gambar yang diperoleh dikirim melalui jaringan lokal ke perangkat Raspberry Pi untuk dilakukan proses *pre-processing*. Setelah itu, gambar dianalisis menggunakan model YOLOv11s yang telah dilatih secara khusus untuk mengklasifikasikan batang tebu ke dalam dua kategori, yaitu tebu bersih dan tebu kotor. Hasil deteksi dan klasifikasi dapat ditampilkan melalui antarmuka web yang menampilkan gambar hasil deteksi serta data klasifikasi secara langsung. Sistem ini diharapkan mampu meningkatkan efisiensi dan akurasi proses penilaian tebu secara otomatis di lingkungan industri. Hasil penelitian terdahulu dirangkum dalam Tabel 2.1 berikut:

Tabel 2.1 Perbandingan Variabel Penelitian

Peneliti	Fokus Penelitian	Metode / Model	Hasil
Budi Yanto, Luth Fimawahib, Asep Supriyanto, B. Herawan	Klasifikasi Tekstur Kematangan Buah Jeruk Manis Berdasarkan	<i>Convolutional Neural Network</i> (CNN) dengan <i>Multilayer</i>	Akurasi klasifikasi 97,51%, Akurasi 96%,dan pelatihan 92%. Klasifikasi menggunakan <i>smartphone</i> yang terdeteksi langsung berdasarkan warna, bobot

Peneliti	Fokus Penelitian	Metode / Model	Hasil
Hayadi, Rinanda Rizki Pratama (2021)	Tingkat Kecerahan Warna	<i>Perceptron</i> (MLP)	akan terlihat dan mampu memisahkan buah jeruk dengan baik.
Sari Nofita (2025)	Klasifikasi tingkat kematangan buah anggur	Yolov11s	<i>Precision</i> mencapai 0,98, <i>recall</i> 0,95, mAP 0,84, dan waktu deteksi 1,02 detik/citra. Sistem mampu mendeteksi kematangan anggur secara semi <i>real-time</i> .
Nur Afiq Eka Putra, Soffiana Agustin (2020)	Klasifikasi Kematangan Tebu berdasarkan Tekstur Batang	Naïve Bayes	Akurasi 73% pada 30 data uji, metode memerlukan data latih, nilai probabilitas rendah mempengaruhi hasil akhir.
S.Nurmuslimah (2020)	Pemilihan Tebu Berkualitas pada Produksi Gula.	Fuzzy Mamdani	Dapat menjadi pertimbangan pengambil keputusan dalam memilih tebu berkualitas yang akan diproduksi. Akurasi hanya 60% akibat toleransi terhadap ketidaksegaran tebu di lokasi survei yang lebih mengutamakan kelangsungan produksi.
Alfiano & Rahayu (2024)	Algoritma <i>Deep learning</i> YOLO (<i>You only look once</i>) Untuk Deteksi Kualitas Kentang Segar dan Busuk Secara <i>Real time</i>	YOLOv8n	- Model YOLOv8n mencapai <i>precision</i> 99.9%, <i>recall</i> 100%, mAP50 99.5%, dan mAP50-90 97.9% dan diimplementasikan dalam <i>Website</i> untuk deteksi secara <i>real-time</i>. - Sistem ini mampu mendeteksi kualitas kentang dengan baik.
Ayunadin (2024)	Deteksi Jenis Tanah berdasarkan warna menggunakan algoritma YOLO	YOLOv8	- Model deteksi jenis tanah menunjukkan akurasi sebesar 80%. - Sistem mampu mengidentifikasi jenis tanah dengan akurasi yang cukup tinggi.
Penelitian saat ini (Febhy Adzhany)	Klasifikasi Kualitas Tebu Berdasarkan Tingkat Kebersihan Menggunakan <i>Deep learning</i>	<i>Convolutional</i> Neural Network (CNN) dengan YOLOv11s-cls	Dapat melakukan klasifikasi kualitas tebu menjadi dua kelas, yakni bersih dan kotor melalui citra batang tebu serta data hasil dapat ditampilkan melalui antarmuka web.

2.2 Landasan Teori

2.2.1 Tanaman Tebu

Tebu (*Saccharum officinarum* L.) merupakan tanaman semusim yang termasuk dalam famili *Poaceae* (rumput-rumputan), dikenal luas karena kandungan gula yang tersimpan di dalam batangnya. Tanaman ini terdiri dari bagian-bagian penting seperti batang, daun, akar, dan bunga. Tebu telah dibudidayakan sejak berabad-abad lalu dan berperan dalam lahirnya industri gula komersial, khususnya sejak abad ke-19. Indonesia sebagai negara beriklim tropis memiliki kondisi yang sangat mendukung untuk budidaya tebu secara luas (Endrizal & Meilin, 2022).

Sebagai komoditas perkebunan yang strategis, tebu memiliki kontribusi penting dalam pemenuhan kebutuhan gula nasional. Ketergantungan masyarakat terhadap gula menjadikan pengelolaan dan peningkatan produksi tebu sebagai isu yang krusial. Namun demikian, berbagai tantangan masih dihadapi dalam proses produksi, mulai dari pemilihan varietas, penggunaan bibit, hingga pengolahan hasil panen. Gula sendiri dikategorikan sebagai bahan pangan pokok yang distribusinya diawasi pemerintah. Meski memiliki potensi besar, produksi dalam negeri belum mampu memenuhi seluruh kebutuhan konsumsi, sehingga Indonesia masih mengandalkan impor gula dari luar negeri untuk menutup kekurangannya (Endrizal & Meilin, 2022).

Dalam menilai kualitas tebu, terdapat tiga parameter utama yang dikenal dengan istilah MBS, yaitu manis, bersih, dan segar. Kriteria "manis" merujuk pada tingkat kematangan tebu yang cukup, biasanya ditandai dengan faktor kemasakan sebesar 25-30 persen, serta koefisien daya tahan dan koefisien peningkatan yang idealnya berada pada rentang 90-100 persen. Selanjutnya, aspek "bersih" menekankan pada rendahnya kadar kotoran atau unsur non-tebu yang menempel, dengan batas toleransi maksimal sebesar lima persen. Sementara itu, kriteria "segar" mengacu pada waktu penanganan setelah panen, di mana tebu sebaiknya digiling dalam waktu kurang dari 36 jam setelah ditebang. Dari ketiga parameter tersebut, kesegaran merupakan faktor yang paling sulit untuk diukur secara langsung karena bergantung pada waktu dan kondisi logistik setelah pemanenan (Nurmuslimah, 2020)

2.2.2 Deep learning

Deep learning adalah suatu pendekatan pembelajaran yang menggunakan jaringan saraf tiruan (*artificial neural network*) yang memiliki banyak lapisan (*multi-layer*). Jaringan itu dirancang menyerupai cara kerja otak manusia, di mana setiap neuron saling terhubung dan membentuk jaringan kompleks. Metode ini, yang juga dikenal dengan istilah *deep structured learning*, *hierarchical learning*, atau *deep neural learning*. Metode ini bekerja dengan memanfaatkan transformasi *non-linear* secara bertingkat. *Deep learning* dapat dianggap sebagai gabungan dari konsep *machine learning* dan kecerdasan buatan berbasis jaringan saraf. Beberapa algoritma yang termasuk dalam kategori *Deep learning*, diantaranya *Convolutional Neural Networks* (CNN), *Restricted Boltzmann Machine* (RBM), *Deep Belief Networks* (DBN), dan *Stacked Autoencoders* (Nugroho et al., 2020).

Menurut Rachman & Santoso (2021) *Deep learning* dihadirkan sebagai solusi untuk mengatasi berbagai keterbatasan yang dimiliki oleh metode *machine learning* konvensional. Salah satu keunggulannya adalah kemampuannya dalam melakukan rekayasa fitur (*feature engineering*) secara otomatis, tanpa perlu campur tangan manusia. Selain itu, *deep learning* mampu menghasilkan prediksi yang lebih akurat seiring bertambahnya jumlah data, sehingga semakin efektif digunakan. *Deep learning* memiliki sejumlah kelebihan, di antaranya:

- 1) *Universal*, dapat diterapkan pada hampir keseluruhan bidang aplikasi.
- 2) *Robust*, tahan terhadap beragam variasi data yang mencerminkan kondisi nyata di lapangan.
- 3) *Generalization*, dapat digunakan untuk berbagai jenis data dan aplikasi melalui kemampuan *transfer learning*
- 4) *Scalability* yang tinggi, sebagaimana ditunjukkan pada arsitektur jaringan ResNet milik Microsoft yang memiliki hingga 1.202 lapisan dan mampu berjalan di sistem komputasi tunggal.

2.2.3 PT Perkebunan Nusantara

Holding Perkebunan Nusantara (2022) menyatakan “PT Perkebunan Nusantara III (Persero) atau PTPN III (Persero), merupakan Badan Usaha Milik Negara (BUMN) Holding Perkebunan yang bergerak di bidang pengelolaan, pengolahan dan pemasaran hasil komoditi perkebunan. Komoditi perkebunan yang

diusahakan adalah kelapa sawit, karet, tebu, teh, kopi, kakao, tembakau, aneka kayuan, buah-buahan dan aneka tanaman lainnya. Saat ini PT Perkebunan Nusantara III (Persero) telah memiliki brand nasional produk hilirisasi komoditi perkebunan dengan nama “Nusakita” disamping beberapa brand lain yang dimiliki oleh Anak Perusahaan dari PTPN Group”.

Riset Perkebunan Nusantara, 2023 menyatakan “PT Riset Perkebunan Nusantara adalah perusahaan berbasis riset Indonesia, anak perusahaan dari Holding Perkebunan Nusantara, di bidang tanaman perkebunan yang menghasilkan inovasi dan peningkatan kapasitas keahlian pada semua aspek agribisnis perkebunan dengan memberikan layanan terpadu (*one stop serving*) untuk kebutuhan bibit, sarana produksi, teknologi, jasa, analisis ekonomi dan kebijakan untuk kejayaan industri perkebunan nasional.” PT Riset Perkebunan Nusantara (RPN) mengelola sejumlah pusat penelitian di berbagai daerah, salah satunya adalah Pusat Penelitian Perkebunan Gula Indonesia (P3GI) yang berlokasi di Pasuruan, Jawa Timur.

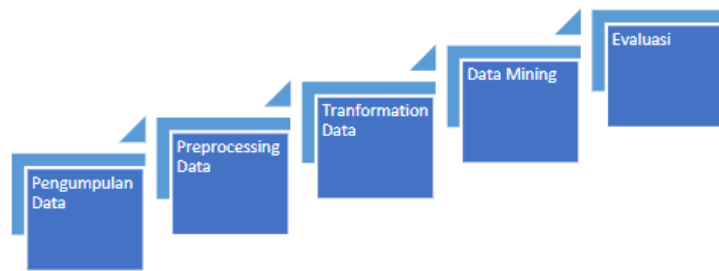
P3GI (2023) menyatakan “Pusat Penelitian Perkebunan Gula Indonesia (P3GI), adalah lembaga penelitian pergulaan di Indonesia. Lembaga penelitian yang berpusat di Pasuruan, Jawa Timur, ini bertugas melaksanakan penelitian, menghasilkan mengkaji teknologi dan produk pergulaan dan pemanis bagi kemajuan masyarakat gula, khususnya petani tebu dan pabrik gula, dan memberikan bantuan teknis kepada klien”.

2.2.4 Knowledge Discovery in Database (KDD)

Knowledge Discovery in Database adalah metode untuk memperoleh informasi berupa data dari *database* yang sudah ada. Terdapat tabel yang saling berhubungan dalam basis data. Data yang diperoleh melalui proses KDD dapat digunakan sebagai sumber data untuk pengambilan keputusan. Istilah KDD dan *data mining* sering digunakan untuk mendeskripsikan proses pengambilan informasi yang tersembunyi dari *database* yang besar. Keduanya saling terkait, namun memiliki makna yang berbeda. *Data mining* merupakan salah satu langkah yang terdapat dalam proses KDD (Leza et al., 2024).

Berikut adalah tahapan metode penelitian menggunakan metode *Knowledge Discovery in Database* (KDD) beserta aktivitas dan deskripsi

aktivitasnya (Astiadewi et al., 2025):



Gambar 2.1 Tahapan KDD

Berdasarkan Gambar 2.1 terdapat beberapa tahapan yang dilakukan secara berurutan untuk memperoleh informasi dan pengetahuan dari data yang diolah. Adapun tahapan-tahapan dalam proses KDD diantaranya :

1) *Data Selection*

Tahapan awal yang dilakukan yakni data *selection*. *Data selection* merupakan proses pemilihan serta menganalisis data yang relevan dari data yang sudah dikumpulkan. Hal ini dilakukan karena sering ditemukan ternyata tidak semua data dibutuhkan dalam proses data mining, proses seleksi dilakukan pada data objek gambar.

2) *Preprocessing*

Selanjutnya setelah data berhasil diperoleh, dilakukan tahap *preprocessing*. Pada tahap ini, data diolah kembali dengan cara memotong bagian tertentu dari gambar (*cropping*) dan mengubah ukuran gambar (*resizing*) agar keseluruhan data memiliki ukuran yang sama. Proses ini penting dilakukan untuk memastikan konsistensi input sebelum digunakan dalam pelatihan model.

3) *Transformation*

Tahap ini merupakan tahapan untuk mengubah data menjadi format yang sudah siap untuk dianalisis. Proses ini dilakukan dengan menambahkan label atau anotasi pada citra. Setelah data diberi label, dataset dibagi kedalam 3 bagian, yakni *train* (data latih), *test* (data uji), dan *valid* (data validasi), sesuai kebutuhan proses pelatihan model.

4) *Data mining*

Pada tahap ini, algoritma YOLOv11 digunakan untuk pelatihan model menggunakan data yang telah dilakukan pelabelan sebelumnya. Proses pelatihan ini dilakukan agar model dapat mengenali dan mempelajari pola dari data yang telah disiapkan.

5) *Evaluation*

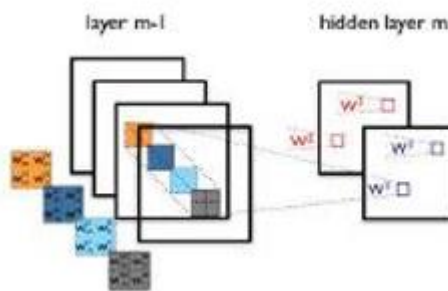
Tahap terakhir dalam proses ini adalah *evaluation*, yaitu evaluasi terhadap kinerja modul yang telah dilatih. Penilaian dilakukan dengan menggunakan *Confusion matrix* seperti *accuracy*, *precision*, *recall*, dan *F1-score*.

2.2.5 *Convolutional Neural Network (CNN)*

Convolutional Neural Network (CNN) merupakan jenis arsitektur jaringan saraf tiruan yang dirancang untuk memproses data visual. CNN terdiri dari beberapa lapisan utama seperti lapisan konvolusi, *pooling*, dan *fully connected* yang bekerja bersama untuk mengenali pola dan mengekstraksi fitur dari gambar maupun data sejenis lainnya. Keunggulan utama CNN terletak pada kemampuannya dalam mengekstraksi ciri penting secara otomatis tanpa memerlukan campur tangan manusia, serta efisiensinya dalam penggunaan memori dan tingkat kompleksitas yang lebih rendah dibandingkan dengan jaringan saraf tiruan lainnya (Rahmadani et al., 2023).

2.2.5.1 *Konsep Convolutional Neural Network (CNN)*

Cara kerja CNN memiliki kesamaan pada *Multilayer Perceptron (MLP)*, namun dalam CNN setiap neuron dipresentasikan ke dalam bentuk dua dimensi, tidak sama seperti MLP setiap neuron hanya berukuran satu dimensi. Operasi linear pada CNN menggunakan operasi konvolusi, sedangkan bobot tidak lagi satu dimensi saja, tetapi berbentuk empat dimensi yang merupakan kumpulan kernel konvolusi seperti pada Gambar 2.2. Dimensi bobot pada CNN adalah neuron input x neuron output x tinggi x lebar. Karena sifat proses konvolusi, maka CNN hanya bisa dipakai pada data yang memiliki struktur dua dimensi seperti citra dan suara (Cahya et al., 2021).



Gambar 2.2 Kernel Konvolusi
(Sumber : Astiadewi et al., 2025)

2.2.5.2 Arsitektur Jaringan Convolutional Neural Network (CNN)

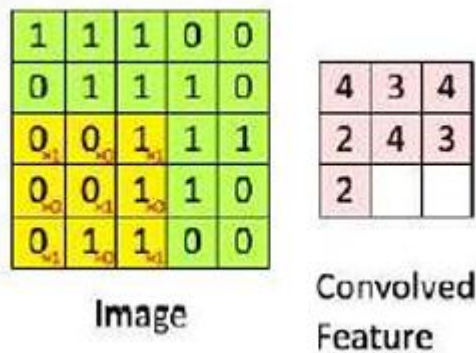
JST terdiri dari berbagai *layer* dan beberapa neuron di tiap masing-masing *layer*. Hal tersebut tidak bisa ditentukan dengan aturan yang pasti dan berlaku berbeda-beda dalam data yang berbeda. Sebuah CNN terdiri dari beberapa *layer*. Berdasarkan arsitektur, ada empat macam *layer* utama dalam sebuah CNN hanya saja yang diterapkan pada penelitian ini ada tiga macam lapisan diantaranya :

1) *Convolution Layer*

Proses citra yang dimanipulasi menggunakan eksternal mask dan dapat menghasilkan citra baru dimana proses ini dilakukan dengan mereduksi dimensi sebuah citra dengan menggunakan operasi *convolutional* yang bertujuan untuk mengekstrak fitur penting dari citra melalui proses *encoding*. Jadi, *Convolution Layer* ialah sebuah hasil kali dari elemen dan memiliki kesesuaian di dalam dua vector (Kurniadi et al., 2021).

Convolution Layer melakukan operasi konvolusi pada output dari *layer* sebelumnya. *Layer* tersebut merupakan proses utama yang mendasari sebuah CNN. Bobot di *layer* tersebut menspesifikasikan kernel konvolusi yang dipakai, sehingga kernel konvolusi bisa dilatih berdasarkan input pada CNN.

Tujuan melakukan konvolusi terhadap data citra yaitu untuk mengekstraksi fitur dari citra input. Konvolusi dapat menghasilkan transformasi linear dari data input sesuai informasi spasial pada data. Bobot di *layer* tersebut menspesifikasikan kernel konvolusi yang dipakai, sehingga kernel konvolusi bisa dilatih berdasarkan input pada CNN. Penggambaran operasi konvolusi dapat dilihat pada Gambar 2.3 (Cahya et al., 2021).



Gambar 2.3 Operasi Konvolusi
(Sumber : Azmi et al., 2023)

Kotak hijau secara keseluruhan adalah gambar yang akan dilakukan konvolusi. Kernel bergerak dari sudut kiri atas ke kanan bawah. Sehingga hasil konvolusi dari Gambar 2.3 dapat dilihat dari gambar di sebelah kanan. Tujuannya dilakukan konvolusi pada data gambar yaitu untuk mengekstraksi fitur dari gambar input (Azmi et al., 2023).

2) *Activation layer*

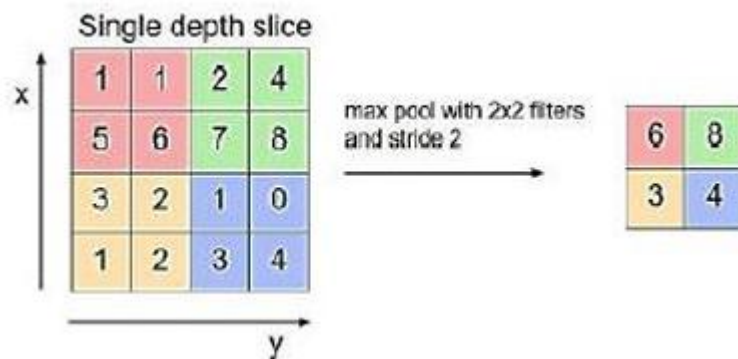
Pada lapisan ini, terjadi proses pengubahan nilai nilai *feature map* pada jarak tertentu tergantung pada fungsi aktivasi yang dipakai, yaitu fungsi aktivasi ReLu. Pada dasarnya fungsi ReLu (*Rectified Linear Unit*) melakukan “*threshol*” dari 0 hingga *infinity*. Fungsi ini menjadi salah satu fungsi yang populer saat ini (Azmi et al., 2023).

3) *Subsampling layer / pooling layer*

Pooling layer merupakan salah satu proses dalam metode CNN. *Layer* ini melakukan proses reduksi sebuah ukuran dari citra yang memiliki tujuan untuk meningkatkan persamaan dari posisi fitur. Dalam proses algoritma CNN *max pooling* merupakan metode *Pooling* yang paling banyak digunakan. *Max Pooling* menggunakan nilai maksimal dari sebuah grid atau bagian kecil dari *Convolution Layer* untuk menyusun citra sehingga citra tersebut tereduksi (Kurniadi et al., 2021).

Subsampling merupakan proses mereduksi ukuran sebuah data citra. Pada pengolahan citra, *Subsampling* bertujuan untuk meningkatkan kemampuan model dalam mengenali fitur meskipun terjadi perubahan posisi objek. *Max pooling* membagi output dari *Convolution Layer* menghasilkan beberapa grid kecil

selanjutnya mengambil nilai maksimal dari setiap grid untuk menyusun matriks citra yang telah direduksi seperti yang ditunjukkan pada Gambar 2.4. Kemudian hasil dari proses tersebut dapat dilihat di kumpulan grid sebelah kanannya (Cahya et al., 2021).



Gambar 2.4 Operasi *Max Pooling*
(Sumber : Cahya et al 2021)

Kotak yang berwarna merah, hijau, kuning dan biru pada sisi kiri merupakan kelompok kotak yang akan dipilih nilai maksimumnya. Sehingga hasil dari proses tersebut dapat dilihat pada kumpulan kotak disebelah kanannya (Azmi et al., 2023).

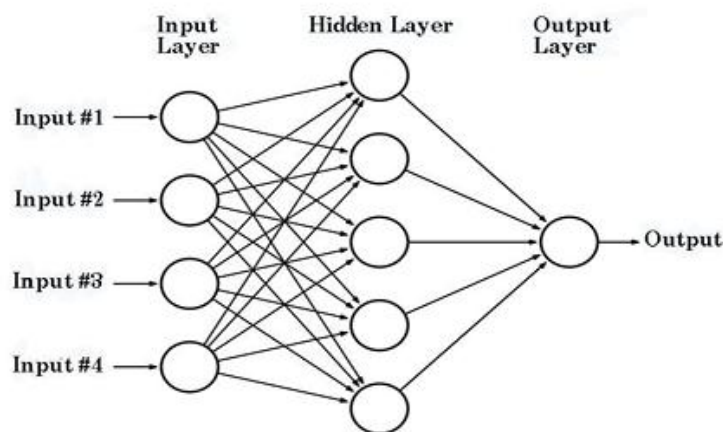
4) *Fully connected layer*

Fully connected layer adalah *layer* yang biasa digunakan untuk MLP dan memiliki tujuan dalam melakukan transformasi dimensi data sehingga data tersebut dapat diklasifikasikan secara linear. Neuron pada *Convolution Layer* ditransformasikan menjadi data satu dimensi, setelah itu neuron tersebut diinput ke dalam sebuah *fully connected layer*. Akibat dari perubahan neuron pada *Convolution Layer* menjadi satu dimensi menyebabkan data kehilangan informasi spasialnya dan tidak reversible, *Fully connected layer* hanya dapat diimplementasikan di akhir jaringan.

Layer tersebut merupakan *layer* yang biasanya dipakai dalam penerapan MLP dan bertujuan untuk melakukan transformasi pada dimensi data agar dapat diklasifikasikan secara linear. Disetiap neuron pada *Convolution Layer* perlu ditransformasi menjadi data satu dimensi terlebih dahulu sebelum dapat dimasukkan kedalam sebuah *fully connected layer*. Dengan itu menyebabkan kehilangan data informasi spasialnya dan menjadi tidak reversibel, *Fully connected layer* hanya bisa diimplementasikan di akhir jaringan. *Convolution Layer* dengan ukuran kernel 1 x 1 menjalankan fungsi yang sama dengan sebuah *Fully connected*

layer namun dengan tetap mempertahankan karakter spasial dari data (Cahya et al., 2021).

Lapisan *Fully-connected* atau lebih dikenal dengan *dense layer* yang dapat dilihat pada Gambar 2.5 adalah lapisan dimana semua neuron aktivitas dari lapisan sebelumnya terhubung semua dengan neuron di lapisan selanjutnya seperti halnya jaringan syaraf tiruan (Azmi et al., 2023).



Gambar 2.5 *Lapisan Fully-connected*
(Sumber : Azmi et al., 2023)

2.2.6 *You Only Look Once (YOLO)*

You only look once (YOLO) merupakan algoritma yang dirancang secara khusus untuk mendeteksi objek secara *real-time* dengan menggabungkan prediksi posisi objek (*bounding box*) dan klasifikasi kelas objek dalam satu tahap pemrosesan. Pendekatan ini memungkinkan sistem untuk langsung menganalisis seluruh gambar secara menyeluruh dan menghasilkan deteksi objek dengan sangat cepat. Kecepatan dan efisiensi yang ditawarkan membuat YOLO menjadi salah satu metode paling populer untuk aplikasi yang membutuhkan respons waktu nyata, seperti deteksi objek pada video maupun pengolahan citra digital (Alfiano & Rahayu, 2024).

Di sisi lain, YOLO juga dikenal sebagai salah satu metode dalam algoritma *machine learning* yang berfokus pada pendeteksian objek dalam sebuah *frame* gambar. Metode ini pertama kali diperkenalkan oleh Joseph Redmon, Santosh Divvala, dan Ross Girshick pada tahun 2015 melalui publikasi *You only look once: Unified, Real-time Object Detection*. Tidak seperti metode konvensional yang

memisahkan tahap klasifikasi dan pencarian lokasi objek, YOLO menyatukan proses tersebut ke dalam satu jaringan saraf (*neural network*). Dengan pendekatan regresi langsung dari piksel gambar, YOLO memprediksi koordinat *bounding box* dan kelas objek sekaligus. Sistem ini bekerja dengan membagi gambar menjadi grid-grid ($N \times N$), yang kemudian dievaluasi secara paralel untuk mendeteksi objek dalam gambar secara efisien (Hadi et al., 2021).

2.2.7 Raspberry Pi

Raspberry Pi merupakan komputer mini yang awalnya dirancang di Inggris untuk mendukung pembelajaran ilmu komputer di sekolah-sekolah, khususnya di negara berkembang. Seiring waktu, penggunaannya meluas ke berbagai bidang seperti robotika, pembuatan super komputer, dan perangkat portabel lainnya. Walaupun berukuran kecil, Raspberry Pi memiliki kemampuan yang menyerupai komputer konvensional dan dilengkapi dengan berbagai fitur input-output layaknya mikrokontroler. Salah satu varian unggulannya adalah Raspberry Pi 4 Model B, yang menawarkan peningkatan signifikan dalam hal prosesor, memori, konektivitas, serta efisiensi daya dibandingkan generasi sebelumnya. Kelebihan utamanya terletak pada pin GPIO (*General Purpose Input Output*) yang memungkinkan integrasi dengan berbagai sensor dan modul eksternal, menjadikannya perangkat favorit dalam berbagai proyek berbasis sensor (Muhammad & Stefanie, 2023).

2.2.8 Python

Python adalah bahasa pemrograman multiparadigma yang bersifat dinamis dan berorientasi objek, serta dapat dimanfaatkan dalam berbagai jenis pengembangan perangkat lunak. Bahasa ini memiliki kemampuan integrasi yang baik dengan bahasa pemrograman lain maupun berbagai alat bantu. Python dilengkapi dengan pustaka standar yang luas dan fleksibel, sehingga mudah dipelajari dalam waktu singkat. Sebagai bahasa interpretatif serbaguna, Python dirancang dengan filosofi yang menekankan keterbacaan kode. Dengan sintaksis yang sederhana namun kuat, Python menggabungkan keunggulan fungsional dengan pustaka bawaan yang kaya dan mendukung berbagai kebutuhan pengembangan (Nugroho et al., 2020).

2.2.9 Confusion Matrix

Confusion Matrix adalah tabel yang membandingkan hasil prediksi model dengan data asli, memberikan informasi tentang jumlah prediksi yang benar (*True Positive* dan *True Negative*) serta kesalahan prediksi (*False Positive* dan *False Negative*) (Valerian et al., 2025).

Menurut (Karimi, 2021) *Confusion matrix* adalah alat untuk analisis prediktif dalam *machine learning*. Untuk memeriksa kinerja berdasarkan klasifikasi model *machine learning*, *Confusion matrix* menunjukkan rincian yang lebih rinci tentang benar dan salah klasifikasi untuk setiap kelas. Baris-baris matriks bersesuaian dengan label kebenaran dasar, dan kolom mewakili prediksi.

		True Class	
		Positive	Negative
Predicted Class	Positive	TP	FP
	Negative	FN	TN

Gambar 2.6 *Confusion matrix*
(Sumber : Karimi, 2021)

Dapat dilihat pada Gambar 2.6, elemen matriks dicirikan berdasarkan label yang diprediksi (positif, negatif) dan hasil perbandingan prediksi dengan label kelas yang sebenarnya (benar, salah):

1. *True Positive* (TP) adalah jumlah data dengan nilai aktual positif dan nilai prediksi positif.
2. *True Negative* (TN) adalah jumlah data dengan nilai aktual positif dan nilai prediksi negatif.
3. *False Positive* (FP) adalah jumlah data dengan nilai aktual negatif dan nilai prediksi positif.
4. *False Negative* (FN) adalah jumlah data dengan nilai aktual negatif dan nilai prediksi negatif.

Dari hasil prediksi yang telah diperoleh, dapat dilakukan perhitungan untuk memperoleh nilai *accuracy*, *recall*, *specificity*, *precision*, *negative predictive value* (NPV), dan *F1-score*. Setelah nilai masing-masing metrik diperoleh, hasil evaluasi selanjutnya diinterpretasikan untuk mengetahui tingkat performa model klasifikasi. Pada penelitian ini, interpretasi yang digunakan mengacu pada metrik *accuracy*, *precision*, *recall*, dan *F1-score*, sebagaimana disajikan pada Tabel 2.2.

Tabel 2.2 Interpretasi Nilai Metrik Evaluasi Model Klasifikasi
(Sumber : Diadaptasi dari Harristyando (2024))

Metrik	Rentang Nilai	Interpretasi
<i>Accuracy</i>	1	Model mampu mengklasifikasikan seluruh data dengan benar (sempurna).
	0,7–0,9	Model memiliki performa yang baik dengan sebagian besar data berhasil diklasifikasikan secara benar.
	< 0,5	Model memiliki performa yang rendah sehingga masih memerlukan perbaikan.
<i>Precision</i>	1	Seluruh prediksi positif yang dihasilkan model merupakan prediksi yang benar.
	0,7–0,9	Model memiliki ketepatan yang baik dalam memprediksi kelas positif, meskipun masih terdapat beberapa <i>false positive</i> .
	< 0,5	Model menghasilkan banyak <i>false positive</i> , sehingga ketepatan prediksi positif rendah.
<i>Recall</i>	1	Model berhasil mengidentifikasi seluruh data positif dengan benar.
	0,7–0,9	Model mampu mengidentifikasi sebagian besar data positif, meskipun masih terdapat beberapa <i>false negative</i> .
	≤ 0,5	Model gagal mengidentifikasi banyak data positif sehingga performanya rendah.
<i>F1-Score</i>	1	Model memiliki keseimbangan sempurna antara <i>precision</i> dan <i>recall</i> .
	0,7–0,9	Model memiliki keseimbangan yang baik antara <i>precision</i> dan <i>recall</i> , namun masih terdapat ruang untuk peningkatan.
	< 0,5	Model memiliki keseimbangan yang buruk antara <i>precision</i> dan <i>recall</i> .

Berdasarkan Tabel 2.2, semakin tinggi nilai *accuracy*, *precision*, *recall*, dan *F1-score* atau semakin mendekati nilai 1, maka semakin baik performa model klasifikasi yang dihasilkan (Harristyando, 2024). Dalam penelitian ini, interpretasi tersebut digunakan sebagai acuan umum dalam menganalisis hasil evaluasi model. Oleh karena itu, interpretasi tersebut tidak dimaksudkan sebagai standar atau batasan baku yang berlaku untuk seluruh penelitian maupun kasus klasifikasi.

2.2.10 *Black Box Testing*

Black box testing adalah metode pengujian perangkat lunak yang berfokus pada pengujian fungsi-fungsi pada sistem tanpa perlu mengetahui struktur kode / logika kodingan dari aplikasi tersebut. Metode ini mengevaluasi perangkat lunak berdasarkan input dan output yang diharapkan, untuk memastikan sistem bekerja sesuai dengan spesifikasi yang telah ditentukan. Metode ini memiliki kelebihan berupa menemukan kesalahan pada sisi fungsional meskipun tidak melibatkan pemeriksaan terhadap kode sumber. Namun demikian, terdapat kekurangan berupa ketidakmampuan metode ini dalam mendeteksi kesalahan pada level internal kode program. Oleh karena itu, metode ini akan lebih efektif jika dikombinasikan dengan teknik pengujian lainnya guna memperoleh hasil evaluasi sistem yang lebih menyeluruh. *Black box testing* sangat sesuai digunakan untuk menguji tampilan antarmuka pengguna serta interaksi antar bagian dalam sistem (M. A. M. A. Rachman et al., 2025).