

BAB II

TINJAUAN PUSTAKA & DASAR TEORI

2.1. Tinjauan Pustaka

Penelitian ini dibangun dari studi-studi yang membuktikan efektivitas CNN dalam analisis citra medis. Sarpotdar (2022) mengintegrasikan CNN dengan teknik segmentasi untuk meningkatkan sensitivitas dan spesifisitas deteksi pembesaran jantung dari citra rontgen, membuktikan bahwa kombinasi arsitektur mampu mengatasi keterbatasan model tunggal. Alzubaidi dkk. (2021) secara komprehensif meninjau keunggulan CNN dibandingkan arsitektur *deep learning* lain seperti ANN dan RNN untuk tugas berbasis citra, menegaskan CNN sebagai fondasi metodologi yang tepat untuk tugas deteksi objek medis.

Perkembangan deteksi berbasis YOLO pada citra medis dimulai dari studi Padilla dkk. (2020) yang mengaplikasikan YOLOv5 pada citra *rontgen* dada dan memperoleh akurasi 61%, sekaligus mengidentifikasi bahwa variasi dataset menjadi faktor pembatas utama. Rácz dkk. (2021) dalam ulasan sistematis terhadap 123 jurnal (2018–2024) menemukan bahwa model YOLO secara umum mencapai mAP >85% pada deteksi kanker payudara dan nodul paru, namun masih lemah terhadap lesi kecil dan langka. Elavarasu & Govindaraju (2024) kemudian membuktikan superioritas YOLOv8 atas YOLOv7 pada deteksi berbasis X-ray berkat desain *anchor-free* yang lebih adaptif terhadap variasi ukuran objek anatomis.

Beberapa penelitian secara spesifik mengevaluasi YOLOv8 untuk deteksi kardiomegali. Fistranda (2024) membandingkan varian YOLOv8 pada dataset X-ray UCI dan menemukan YOLOv8x sebagai model terbaik (akurasi 0,867), namun dengan kompleksitas komputasi tinggi. Mahmood dkk. (2025) mengembangkan YOLOv8_HEART yang mencapai mAP 0,848 dengan kecepatan 151 FPS, unggul dalam efisiensi inferensi *real-time*, meskipun tidak mengeksplorasi optimalisasi backbone. Nguyen dkk. (2023) menemukan bahwa U-Net mampu melampaui YOLOv5 untuk segmentasi kardiomegali (*Precision/Recall* 78%/93%),

mengindikasikan bahwa arsitektur *backbone* memegang peran signifikan dalam performa deteksi pada citra *rontgen* dada.

Tren terkini mengarah pada penguatan *backbone* YOLOv8 melalui integrasi modul tambahan. Alam dkk. (2024) secara khusus membuktikan bahwa backbone ResNet-50 memberikan representasi fitur yang lebih dalam dan signifikan memperbaiki performa pada citra kontras rendah kondisi yang dominan pada radiografi dada. Li dkk. (2025) mengintegrasikan mekanisme *self-attention* ke SA-YOLOv8 dan meningkatkan mAP@0.5 sebesar 7,4% untuk deteksi lesi jantung pada MRI *cardiac*. Nurmaini dkk. (2025) menggabungkan YOLOv8 dengan CBAM dan ResNeXt untuk deteksi pembesaran jantung janin pada USG dengan mAP tinggi, sementara De Haro dkk. (2025) membuktikan bahwa YOLOv8 dapat dimanfaatkan bahkan untuk pembersihan dataset guna meningkatkan kualitas segmentasi penyakit jantung LVNC.

Berbagai studi menegaskan bahwa performa YOLOv8 sangat dipengaruhi oleh strategi optimalisasi, bukan hanya desain arsitektur. Ragab dkk. (2024) dalam *systematic literature review* dari PubMed, ScienceDirect, dan IEEE menekankan bahwa kombinasi *pre-processing*, augmentasi data, dan optimasi *hyperparameter* merupakan faktor kritis yang membedakan model berkinerja tinggi. Billah dkk. (2024) membandingkan YOLOv8 dan YOLOv10 pada CT scan dan menemukan bahwa YOLOv8 menunjukkan konvergensi pelatihan lebih cepat meskipun YOLOv10 unggul dalam akurasi akhir. Khalili dkk. (2025) mengembangkan Mamba-YOLOvX *multimodal* pada CXR dan mencapai *state-of-the-art* mAP@0.5 sebesar 0,615, sementara Rajaraman dkk. (2025) menunjukkan bahwa pendekatan *ensemble* YOLOv8/v9 pada dataset JSRT dan RSNA mampu meningkatkan performa lebih lanjut melalui diversifikasi model.

Berdasarkan kajian penelitian-penelitian terdahulu, terdapat gap penelitian yang belum terjawab: belum ada studi yang secara komprehensif mengeksplorasi optimalisasi arsitektur hybrid YOLOv8-ResNet50 yang dikombinasikan dengan optimalisasi augmentasi data dan hyperparameter tuning secara sistematis dan terukur untuk deteksi kardiomegali pada citra *rontgen* dada. Mayoritas penelitian

terdahulu hanya mengoptimalkan satu aspek (arsitektur saja, atau augmentasi saja), tanpa mengintegrasikan ketiganya dalam satu framework yang koheren. Penelitian ini mengisi gap tersebut dengan mengusulkan framework optimalisasi tiga-tahap yang mencakup modifikasi backbone, eksplorasi augmentasi data, dan hyperparameter tuning sistematis untuk meningkatkan akurasi prediksi kardiomegali secara menyeluruh.

2.2. Dasar Teori

2.2.1. *Computer-Aided Diagnosis (CAD)*

Computer-Aided Diagnosis (CAD) adalah teknologi yang menggabungkan pemrosesan citra, pembelajaran mesin, dan visi komputer untuk membantu profesional medis dalam proses pengambilan keputusan diagnostik. Sistem CAD menganalisis data pasien, baik berupa citra medis maupun data non-citra, untuk menilai kondisi pasien dan mendukung klinisi dalam proses pengambilan keputusan. Dalam beberapa dekade terakhir, CAD telah menjadi bidang penelitian utama, terutama dengan integrasi metode pembelajaran mesin dan pembelajaran mendalam. Teknologi ini memungkinkan analisis data pasien yang lebih efisien dan akurat, memberikan penilaian kondisi pasien yang dapat membantu klinisi dalam proses pengambilan keputusan. Dengan kemajuan teknologi pembelajaran mendalam, sistem CAD telah berkembang untuk menangani tugas-tugas klinis yang lebih kompleks. Integrasi teknologi ini telah meningkatkan kinerja CAD dan memungkinkan pengembangan sistem untuk berbagai aplikasi medis, termasuk deteksi dini penyakit dan penilaian prognosis (Chan dkk., 2020).

Implementasi CAD dalam pencitraan medis mencakup berbagai tahapan, termasuk pra-pemrosesan citra, segmentasi, ekstraksi fitur, dan klasifikasi. Pada tahap pra-pemrosesan, citra medis dipersiapkan untuk mengurangi artefak dan meningkatkan kualitas gambar. Segmentasi kemudian digunakan untuk membedakan struktur yang berbeda dalam citra, seperti organ atau lesi. Setelah itu, fitur-fitur penting diekstraksi dan digunakan dalam algoritma klasifikasi untuk menilai kondisi pasien (Mathews & Mathews, 2020).

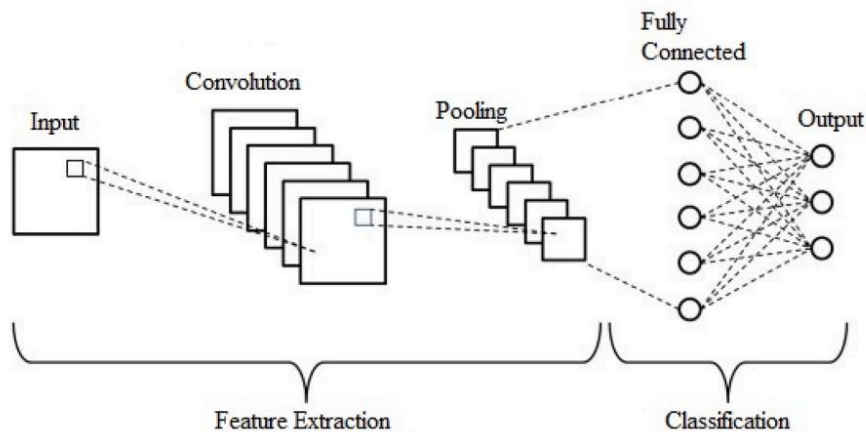
Tujuan utama sistem CAD adalah meningkatkan akurasi diagnosis dan mengurangi waktu yang dibutuhkan dalam interpretasi citra medis. Dengan bantuan

CAD, radiolog dapat lebih cepat dan tepat dalam mengidentifikasi kondisi medis, yang pada akhirnya meningkatkan kualitas perawatan pasien.

2.2.2. *Convolutional Neural Network*

Convolutional Neural Network (CNN) merupakan salah satu algoritma dari pembelajaran mendalam yang memanfaatkan arsitektur dari jaringan syaraf tiruan. CNN cocok digunakan untuk melakukan pemrosesan dan pengenalan pola pada citra. CNN merupakan pembelajaran mendalam hasil dari pengembangan multilayer perceptron (MLP) (Bouwman dkk., 2019). Terdapat kesamaan antara cara kerja CNN dan MLP. Namun, perbedaannya CNN merepresentasikan neuron dalam bentuk dua dimensi, sedangkan neuron pada MLP direpresentasikan hanya satu dimensi (Kusumaningrum dkk., 2020).

Di antara metode pembelajaran mendalam, manfaat utama jaringan CNN untuk klasifikasi citra meliputi: (1) CNN memiliki parameter yang relatif lebih sedikit sehingga pelatihan dapat dilakukan dengan cepat, pengurangan risiko *overfitting*, dan kebutuhan data latih yang lebih sedikit, (2) dapat mendeteksi fitur yang lebih, di mana saja pada citra jika pernah mempelajari filter yang dapat mendeteksi fitur. Karena citra memiliki fitur yang berulang, CNN dapat digeneralisasi dengan relative lebih baik, (3) lapisan bawah CNN mendeteksi fitur tingkat bawah dan lapisan atas menggabungkan fitur ini menjadi fitur tingkat tinggi yang lebih besar. Arsitektur CNN dapat bekerja dengan baik pada masukan citra dibandingkan dengan metode pembelajaran mendalam lainnya. Arsitektur CNN memiliki perbedaan pada tahap sebelum dilakukannya proses pelatihan. Bentuk awal input dari CNN adalah berupa matriks citra yang dikonversi menjadi lebih kecil dan memiliki nilai unik untuk selanjutnya dapat masuk ke proses pelatihan. Arsitektur CNN secara umum ditunjukkan secara rinci di Gambar 2.1.



Gambar 2.1 Arsitektur CNN

Sumber : diadopsi dari Islam dkk (2024)

Arsitektur CNN memiliki beberapa komponen yang diadopsi oleh Habib & Qureshi (2022), diantaranya:

1. Lapisan Konvolusi

Lapisan konvolusi merupakan lapisan yang digunakan untuk memperkecil ukuran citra yang dilakukan melalui filtering. Filter akan membagi citra masukan menjadi beberapa kotak kecil (misal 3x3 atau 5x5) dan menggabungkan kisi-kisi kecil ini menjadi keluaran berupa feature map.

2. Pooling

Pooling merupakan lapisan yang berfungsi untuk mengurangi masalah sensitivitas pada keluaran *feature map*. Lapisan pooling mengambil nilai yang mewakili setiap grid untuk menyusun matriks citra yang telah direduksi. Tujuan dari pooling adalah untuk mendapatkan nilai paling unik yang mewakili citra input. Dua teknik pooling yang sering digunakan yaitu max pooling dan average pooling. Max pooling yaitu mengambil nilai tertinggi dari grid yang telah ditentukan. Sedangkan *average* pooling yaitu melakukan perhitungan rata-rata dari nilai-nilai yang berada pada grid yang telah ditentukan.

3. *Fully Connected Layer*

Fully connected layer bekerja dengan melakukan transformasi matriks ke dalam bentuk flatten / 1 dimensi. Hasil 1 dimensi tersebut digunakan sebagai input

awal untuk dilatih menggunakan teknik klasifikasi (misal jaringan syaraf tiruan *backpropagation* atau *feed forward*) dalam mencari output klasifikasi.

2.2.3. YOLOv8 (*You Only Look Once version 8*)

YOLOv8 merupakan iterasi terbaru dari keluarga algoritma YOLO yang dikembangkan oleh *Ultralytics* pada Januari 2023. Algoritma ini dirancang untuk melakukan deteksi objek secara real-time dengan keseimbangan optimal antara kecepatan inferensi dan akurasi prediksi, menjadikannya solusi praktis untuk aplikasi industri dan penelitian yang memerlukan respons cepat tanpa mengorbankan performa (Terven dkk., 2023).

Berbeda dengan pendekatan deteksi objek tradisional yang menggunakan sliding window atau region proposal *networks*, YOLOv8 memperlakukan deteksi objek sebagai masalah regresi tunggal, di mana seluruh gambar input diproses dalam satu kali *forward pass* untuk memprediksi *bounding box* koordinat, probabilitas kelas objek, dan skor kepercayaan (*confidence score*) secara simultan. Pendekatan *unified* ini memungkinkan YOLOv8 mencapai kecepatan deteksi yang sangat tinggi hingga mencapai beberapa ratus frame per detik pada perangkat performa tinggi (Reis dkk., 2024).

Arsitektur *backbone* YOLOv8 menggunakan *CSPDarknet53* yang diperkuat dengan koneksi *Cross-Stage Partial* (CSP) untuk meningkatkan aliran informasi antar layer dan memfasilitasi ekstraksi fitur hierarkis multi-skala dari gambar input. Desain ini memungkinkan model untuk menangkap informasi semantik pada berbagai tingkat abstraksi, mulai dari detail tekstur tingkat rendah hingga konsep objek tingkat tinggi, yang sangat penting untuk deteksi akurat pada citra medis dengan kontras dan noise yang bervariasi (Reis dkk., 2024).

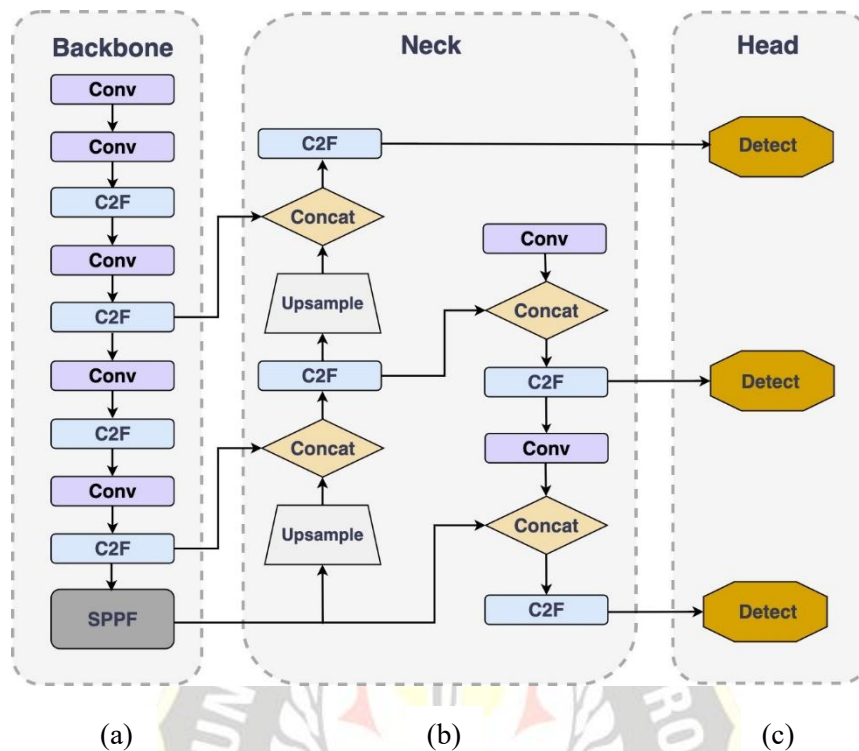
Komponen *Neck* pada YOLOv8 mengadopsi modul C2f yang inovatif sebagai pengganti *Feature Pyramid Network* (FPN) tradisional, dengan tujuan menggabungkan fitur semantik tingkat tinggi dengan informasi spasial tingkat rendah secara lebih efektif. Strategi fusi multi-scale ini meningkatkan kemampuan model dalam mendeteksi objek dengan berbagai ukuran, terutama objek-objek kecil seperti lesi atau anomali di citra medis yang sering terlewatkan oleh model

konvensional. *Head* YOLOv8 mengimplementasikan pendekatan anchor-free detection yang memprediksi langsung pusat lokasi objek tanpa bergantung pada anchor box yang telah ditentukan sebelumnya. Penghapusan anchor boxes ini menyederhanakan proses pelatihan model, mengurangi hyperparameter yang perlu dioptimasi, dan meningkatkan adaptabilitas terhadap objek dengan berbagai rasio aspek dan skala, yang memberikan keuntungan signifikan ketika diterapkan pada dataset medis dengan morfologi objek yang heterogen (Terven dkk., 2023).

YOLOv8 mengintegrasikan teknik *advanced* data augmentation termasuk MixUp dan Mosaic augmentation untuk meningkatkan robustness model terhadap variasi lighting, noise, dan deformasi citra, serta mengimplementasikan *self-attention mechanism* yang membantu model memahami hubungan spasial dan semantik kompleks antar fitur dalam gambar. Kedua inovasi ini sangat relevan untuk aplikasi pencitraan medis di mana dataset seringkali terbatas dan variabilitas klinis tinggi (Widayani dkk., 2024).

YOLOv8 tersedia dalam berbagai ukuran model (YOLOv8-nano, *small*, *medium*, *large*, *extra-large*) yang memungkinkan peneliti dan praktisi untuk memilih trade-off optimal antara akurasi dan kecepatan sesuai kebutuhan spesifik aplikasi. Variasi ini mengakomodasi kebutuhan komputasi yang beragam, mulai dari perangkat edge dengan sumber daya terbatas seperti smartphone dan microcontroller hingga server performa tinggi untuk processing batch atau *inference cloud-based* (Billah dkk., 2024).

Dalam domain pencitraan medis, YOLOv8 telah menunjukkan performa superior pada berbagai modalitas imaging termasuk X-ray, CT-scan, dan MRI untuk tugas-tugas seperti deteksi lesi, segmentasi organ, dan klasifikasi penyakit. Studi empiris melaporkan bahwa YOLOv8 mencapai presisi > 0.92 , recall > 0.90 , dan *mean Average Precision* (mAP) > 0.93 pada tugas deteksi tumor otak, serta akurasi 99.8% pada klasifikasi penyakit paru-paru termasuk deteksi COVID-19. Keunggulan ini menjadikan YOLOv8 sebagai pilihan utama untuk aplikasi diagnosis medis yang memerlukan akurasi tinggi dan kecepatan real-time (Widayani dkk., 2024). Arsitektur YOLOv8 secara umum ditunjukkan secara rinci di Gambar 2.2.



Gambar 2.2 Arsitektur YOLOv8 diadopsi oleh Yao dkk (2024), (a) *Backbone*, (b) *Neck*, (c) *Head*

Arsitektur YOLOv8 memiliki beberapa komponen yaitu sebagai berikut :

a. *Backbone*

Bagian *Backbone* bertindak sebagai ekstraksi fitur citra menggunakan CSPDarknet53 yang dilengkapi koneksi *Cross-Stage Partial* (CSP) serta lapisan **Conv**, **C2F**, dan **SPPF**. Struktur ini memungkinkan model mempelajari informasi hierarkis dari citra input sehingga dapat menangkap informasi semantik lebih mendalam baik dari detail tekstur maupun konsep visual tingkat abstrak. Desain *Backbone* ini sangat penting dalam meningkatkan akurasi pada modalitas medis (X-ray, MRI, CT) di mana kontras dan *noise* citra sering bervariasi.

b. *Neck*

Komponen *Neck* yang memegang peranan penting dalam YOLOv8 adalah modul **C2F** serta fitur fusi multi-skala dengan operasi **Concat** dan **Upsample**. Fusi fitur ini berfungsi menggabungkan informasi dari berbagai tingkat (*high-*

level dan *low-level*), sehingga YOLOv8 dapat mendeteksi objek dengan berbagai ukuran, bahkan pada objek kecil seperti lesi pada citra medis yang sering luput dari deteksi model konvensional.

c. *Head*

Head pada YOLOv8 mengadopsi pendekatan *anchor-free* dengan prediksi langsung pusat objek tanpa bergantung pada anchor box. Arsitektur ini menyederhanakan proses pelatihan model, mengurangi jumlah *hyperparameter*, serta meningkatkan adaptabilitas model pada dataset medis yang memiliki morfologi objek heterogen dan bentuk irregular, memungkinkan deteksi objek dengan rasio aspek dan skala bervariasi secara optimal.

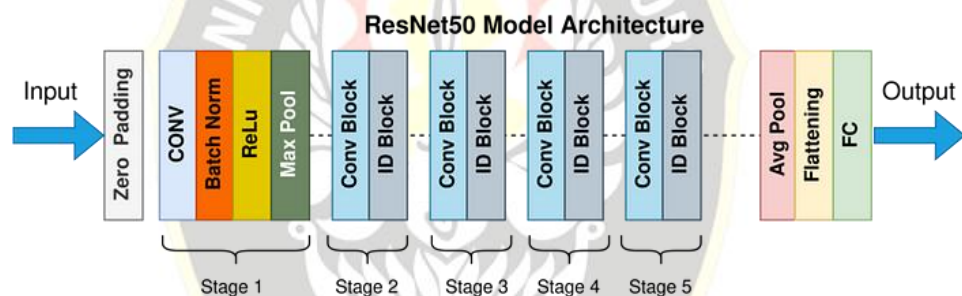
Dalam konteks kalkulasi CTR untuk deteksi kardiomegali, mekanisme *anchor-free detection* YOLOv8 memberikan keunggulan spesifik dibandingkan pendekatan berbasis *anchor* tradisional. Jantung dan rongga dada sebagai objek target memiliki variasi rasio aspek yang relatif konsisten namun ukurannya bervariasi antar pasien tergantung postur dan kondisi klinis. Pendekatan *anchor-free* memungkinkan YOLOv8 memprediksi bounding box secara langsung dari pusat objek tanpa terikat pada ukuran anchor yang telah ditentukan sebelumnya, sehingga lebih adaptif terhadap variasi ukuran anatomis antar pasien. Output berupa koordinat *bounding box* (x, y, w, h) dari YOLOv8 inilah yang kemudian dikonversi secara langsung menjadi lebar jantung dan lebar rongga dada untuk kalkulasi CTR otomatis menjadikan YOLOv8 sebagai komponen yang tidak hanya mendeteksi keberadaan objek tetapi juga menghasilkan informasi kuantitatif yang dibutuhkan untuk diagnosis klinis.

2.2.4. RESNET-50

ResNet-50 adalah jaringan saraf konvolusi yang memiliki kedalaman 50 lapisan. *Shortcut connection* adalah sebuah konsep baru dari salah satu arsitektur CNN yaitu ResNet-50 (Faiz Nashrullah dkk., 2020). Algoritma Resnet-50 memiliki karakteristik yang berbeda dari koneksi lompat dan struktur pembelajaran residual.

Ini dapat memecahkan masalah pelatihan jaringan dalam, penghilangan gradien, dan ledakan gradien dengan lebih baik (Jia dkk., 2023).

Hasil penelitian menunjukkan bahwa implementasi arsitektur ResNet-50 pada klasifikasi tumor otak menggunakan citra MRI menghasilkan tingkat akurasi yang sangat tinggi. Pada studi yang dilakukan oleh Ali dkk (2025), model ResNet-50 berhasil mengklasifikasikan tumor otak (glioma, meningioma, dan tumor hipofisis) dengan akurasi mencapai 99,88%, jauh melampaui model CNN klasik yang hanya mencapai 94,55%. Temuan ini membuktikan bahwa ResNet-50 sangat efektif dalam menangkap fitur kompleks pada citra medis dan mampu memberikan hasil klasifikasi yang sangat akurat dan andal, sehingga dapat menjadi alat pendukung penting bagi profesional medis dalam proses diagnosis dan perencanaan perawatan pasien (Ali dkk., 2025). Arsitektur ResNet-50 ditunjukkan pada Gambar 2.3.



Gambar 2.3 Arsitektur ResNet -50 diadopsi oleh Chandra dkk (2024)

ResNet-50 merupakan jaringan konvolusional dalam (deep convolutional network) yang menggunakan pendekatan residual *learning* melalui *shortcut connections* untuk mengatasi degradasi akurasi pada jaringan yang semakin dalam. Gambar di atas menggambarkan alur utama dari ResNet-50 yang terdiri dari beberapa blok penting, yang dijelaskan sebagai berikut:

a. *Zero Padding*

Tahapan pertama adalah *Zero Padding*, yang digunakan untuk menambahkan piksel nol di sekitar batas citra input. Hal ini dilakukan agar ukuran citra tetap konstan setelah proses konvolusi dan untuk menjaga informasi spasial dari fitur di tepi gambar.

b. *Convolution + Batch Normalization + ReLU + Max Pooling*

Convolution (CONV): Lapisan konvolusi awal bertujuan untuk mengekstraksi fitur dasar dari input citra (misalnya tepi, tekstur, atau pola sederhana). *Batch Normalization* (Batch Norm): Menormalkan output dari layer konvolusi agar mempercepat pelatihan dan menstabilkan distribusi aktivasi. *ReLU* (*Rectified Linear Unit*): Fungsi aktivasi non-linear yang memperkenalkan kompleksitas dan membantu model belajar representasi data yang lebih baik. *Max Pooling* (*Max Pool*): Mengurangi dimensi fitur dan mempercepat komputasi dengan mengambil nilai maksimum dari wilayah lokal fitur map.

c. *Convolutional Block* (Conv Block)

Blok konvolusi ini adalah bagian penting dalam ResNet yang mengandung:

- Beberapa layer konvolusi dengan kernel 1×1 , 3×3 , dan 1×1 (dikenal sebagai *bottleneck architecture*),
- Shortcut connection yang melakukan penyesuaian dimensi menggunakan layer konvolusi 1×1 karena terjadi perubahan ukuran (misalnya akibat stride).

Conv Block digunakan di awal setiap stage baru ketika dimensi fitur berubah.

d. *Identity Block* (ID Block)

Blok identitas digunakan untuk mempertahankan dimensi fitur yang sama, dengan struktur serupa Conv Block tetapi tanpa konvolusi penyesuaian pada shortcut-nya. Di sini, informasi dari input langsung dijumlahkan dengan hasil konvolusi, memungkinkan pembelajaran residual dilakukan secara efektif.

ResNet-50 memiliki susunan Conv Block dan ID Block sebagai berikut:

- 1 Conv Block + 2 ID Block (Stage 1)
- 1 Conv Block + 3 ID Block (Stage 2)
- 1 Conv Block + 5 ID Block (Stage 3)
- 1 Conv Block + 2 ID Block (Stage 4)

e. *Average Pooling*

Setelah seluruh blok residual, dilakukan *Average Pooling* untuk mereduksi ukuran fitur map menjadi satu nilai rata-rata per channel. Hal ini mengurangi kompleksitas sebelum masuk ke layer klasifikasi akhir.

f. *Flattening*

Output dari *average pooling* di flatten menjadi vektor satu dimensi untuk diproses oleh layer fully connected (dense layer).

g. *Fully Connected (FC) Layer*

Layer terakhir berupa fully connected layer (biasanya diikuti oleh softmax jika untuk klasifikasi). Layer ini menghasilkan prediksi akhir berdasarkan representasi fitur yang telah dibentuk dari seluruh lapisan sebelumnya.

Dalam konteks deteksi kardiomegali pada citra rontgen dada, mekanisme residual learning ResNet-50 memberikan kontribusi spesifik yang krusial. Citra rontgen dada memiliki karakteristik kontras rendah dengan perbedaan visual yang sangat halus antara jantung normal dan yang mengalami pembesaran. Pada jaringan *deep learning* konvensional tanpa *skip connection*, sinyal gradien yang membawa informasi perbedaan halus ini berpotensi menghilang sebelum mencapai lapisan awal (*vanishing gradient*), sehingga fitur-fitur halus tersebut tidak terekstraksi dengan baik. *Shortcut connection* pada ResNet-50 memastikan bahwa informasi dari lapisan awal tetap mengalir langsung ke lapisan yang lebih dalam melalui identity mapping, memungkinkan model mempertahankan detail fitur tekstur halus sekaligus membangun representasi semantik tingkat tinggi secara hierarkis. Inilah yang menjadi justifikasi teknis mengapa ResNet-50 dipilih sebagai *backbone* pengganti CSPDarknet53 kemampuannya mempertahankan gradien pada jaringan dalam sangat relevan untuk mengekstraksi perbedaan morfologis halus pada citra radiografi dada.

SEKOLAH PASCASARJANA

2.2.5. Kardiomegali

Kardiomegali adalah sebuah keadaan anatomis dimana besarnya jantung lebih besar dari ukuran jantung normal, yaitu suatu keadaan diameter transversal dari gambaran jantung $\geq 50\%$ dari besar rongga dada atau $CTR > 50\%$ pada proyeksi posterior-anterior foto toraks. Kardiomegali terjadi ketika besar jantung $> 50\%$ dari diameter internal rongga dada (Zhou dkk., 2021).

Pengukuran dilakukan oleh para peneliti dan ditinjau oleh komite radiologi di Universitas Jeddah, Departemen Pencitraan Medis dan Ilmu Radiasi. Ukuran

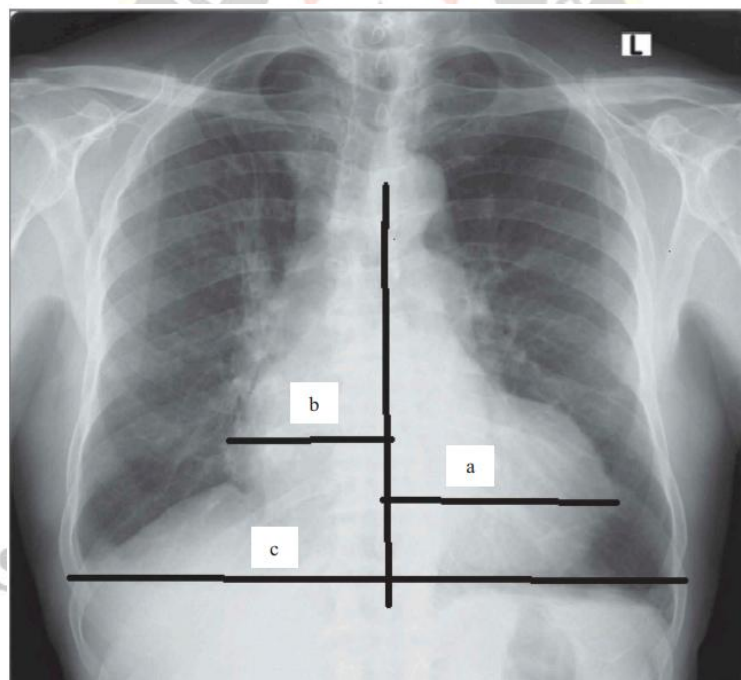
jantung diperkirakan dari rontgen dada *posteroanterior* (PA) dengan menghitung *Cardiothoracic Ratio* (CTR), yaitu rasio antara diameter transversal maksimum jantung dan lebar maksimum toraks di atas sudut kostofrenikus, yang diukur dari tepi dalam tulang rusuk. CTR dihitung secara manual dan didokumentasikan untuk setiap pasien, seperti yang ditunjukkan pada Gambar 1. Pada orang dewasa, CTR normal adalah maksimal 0,5 (Candemir dkk., 2018).

CTR dihitung menggunakan persamaan (1) :

$$CTR = \frac{a + b}{c} \quad (1)$$

dengan *a* adalah jarak dari batas kanan jantung ke garis tengah, *b* adalah jarak dari batas kiri jantung ke garis tengah serta *c* adalah diameter toraks maksimum (TD) di atas sudut kostofrenikus yang diukur dari tepi dalam tulang rusuk.

Penjelasan lebih rinci tentang CTR akan ditunjukkan pada Gambar 2.4.



Gambar 2.4 Rontgen dada posteroanterior (PA), (a) jarak dari batas kanan jantung ke garis tengah, (b) jarak dari batas kiri jantung ke garis tengah, (c) diameter toraks maksimum

Cardiothoracic Ratio (CTR) (*a* = jarak dari batas kanan jantung ke garis tengah, *b* = jarak dari batas kiri jantung ke garis tengah, dan *c* = diameter toraks maksimum) (Candemir dkk., 2018).

2.2.6. *Random Search*

Random Search merupakan metode optimasi yang bekerja dengan cara memilih kombinasi nilai parameter secara acak dari ruang pencarian (*search space*) yang telah didefinisikan sebelumnya. Proses ini dilakukan sebanyak iterasi yang ditentukan oleh peneliti, setiap kombinasi dievaluasi menggunakan metrik performa tertentu, dan kombinasi terbaik dipilih sebagai konfigurasi parameter yang optimal. Berbeda dengan *Grid Search* yang mengevaluasi seluruh kemungkinan kombinasi secara sistematis dan *exhaustif*, *Random Search* tidak memerlukan penjelajahan menyeluruh sehingga jauh lebih efisien secara komputasi, terutama pada ruang pencarian berdimensi tinggi (Rom dkk., 2024).

Prinsip kerja *Random Search* dapat dijelaskan melalui tiga tahapan utama. Pertama, peneliti mendefinisikan distribusi atau rentang nilai untuk setiap parameter yang akan dioptimalkan. Kedua, algoritma secara acak mengambil sampel kombinasi parameter dari distribusi tersebut. Ketiga, setiap kombinasi dievaluasi menggunakan metrik validasi yang dipilih, dan proses ini diulang sebanyak iterasi yang ditetapkan hingga diperoleh kombinasi parameter terbaik (Rom dkk., 2024).

Kelebihan utama *Random Search* dibandingkan metode lainnya meliputi: Efisiensi komputasi, yaitu tidak perlu mengeksplorasi seluruh ruang kombinasi, sehingga waktu komputasi jauh lebih singkat dari *Grid Search*. Skalabilitas tinggi, yaitu sangat efektif pada ruang pencarian berdimensi tinggi. Sebagai ilustrasi, pada arsitektur AlexNet dengan 145.152.000 kemungkinan kombinasi, *Grid Search* memerlukan lebih dari 24 jam, sementara *Random Search* dapat menemukan solusi optimal dalam batas waktu yang sama (Rom dkk., 2024).

Fleksibilitas distribusi, yaitu mendukung berbagai jenis distribusi nilai parameter. Hasil kompetitif, yaitu terbukti menghasilkan performa setara atau lebih baik dibandingkan *Grid Search* dengan anggaran komputasi yang lebih kecil, seperti CNN yang mencapai akurasi 0,9923 menggunakan *Random Search* dibandingkan *Grid Search* dengan akurasi 0,9907 (Rom dkk., 2024).

Dalam implementasinya menggunakan Python, *Random Search* dapat diterapkan melalui *RandomizedSearchCV* dari library *Scikitlearn*. Proses kerjanya

melibatkan definisi ruang pencarian parameter, penentuan jumlah iterasi (n_iter), penerapan *k-fold cross-validation*, dan pemilihan kombinasi terbaik berdasarkan metrik evaluasi yang ditetapkan. Putri dkk. (2023) membuktikan bahwa penerapan RandomSearchCV pada model *Adaptive Boosting* untuk prediksi pasien gagal jantung menghasilkan akurasi 85% dan AUC 0,897 kategori *good classification*, menunjukkan bahwa Random Search mampu menemukan konfigurasi parameter optimal secara efisien bahkan pada data medis yang kompleks.

Dalam penelitian ini, *Random Search* diterapkan pada dua tahap optimasi sekaligus. Pertama, pada pencarian konfigurasi augmentasi data optimal, yaitu menentukan kombinasi terbaik dari parameter augmentasi seperti *nilai flip*, *scale*, *translate*, *hue*, *saturation*, dan *brightness* yang menghasilkan peningkatan performa deteksi terbaik pada model hybrid YOLOv8-ResNet50. Kedua, pada *hyperparameter tuning* pelatihan model, yaitu mengoptimalkan kombinasi *learning rate*, *batch size*, *momentum*, dan *weight decay* secara efisien tanpa harus mencoba seluruh kemungkinan kombinasi. Penerapan Random Search pada dua tahap ini memungkinkan eksplorasi ruang konfigurasi yang jauh lebih luas dibandingkan pendekatan manual, sekaligus tetap menjaga efisiensi komputasi dalam proses pelatihan model.

2.2.7. Augmentasi Data

Augmentasi data (*data augmentation*) adalah teknik yang digunakan untuk memperluas variasi dataset pelatihan secara artifisial tanpa menambah jumlah data asli, dengan cara menerapkan serangkaian transformasi pada citra yang sudah ada. Teknik ini terbukti secara signifikan meningkatkan kinerja dan kemampuan generalisasi model deep learning, khususnya pada kondisi dataset medis yang terbatas dalam jumlah (Islam dkk., 2024). Augmentasi sangat krusial dalam domain citra medis karena pengumpulan data klinis memerlukan proses yang panjang dan membutuhkan validasi ahli.

Augmentasi data pada citra dapat dikelompokkan menjadi dua kategori utama. Pertama, transformasi geometris, yaitu perubahan tata letak spasial citra seperti flipping horizontal (*flipplr*), perubahan skala (*scale*), translasi posisi

(*translate*), rotasi, dan *mosaic augmentation* yang menggabungkan empat citra berbeda dalam satu kanvas pelatihan. Kedua, transformasi fotometris, yaitu perubahan pada properti warna dan kecerahan citra melalui manipulasi ruang warna HSV (*Hue-Saturation-Value*), meliputi parameter *hsvh* (rona warna), *hsvs* (saturasi), dan *hsvv* (kecerahan) (Shorten & Khoshgoftaar, 2021). Pemilihan jenis dan intensitas augmentasi tidak dapat dilakukan secara sembarangan, melainkan harus disesuaikan dengan karakteristik modalitas citra medis yang digunakan. Ragab dkk. (2024) menegaskan bahwa augmentasi pada citra rontgen dada harus mempertahankan realisme klinis sehingga transformasi ekstrem seperti rotasi bebas, shear, dan perspective sebaiknya dinonaktifkan karena tidak mencerminkan variasi yang mungkin ditemui dalam praktik radiologi nyata. Pada citra rontgen dada, augmentasi yang direkomendasikan bersifat konservatif, mengutamakan variasi kecerahan dan kontras fotometris serta transformasi geometris ringan yang tetap menjaga integritas posisi anatomis organ (Islam dkk., 2024).

Dalam penelitian ini, parameter augmentasi tidak ditetapkan secara manual, melainkan diperoleh melalui proses eksplorasi sistematis menggunakan metode Random Search pada tujuh parameter aktif: *hsvh*, *hsvs*, *hsvv*, *translate*, *scale*, *flip*, dan *mosaic*. Proses augmentasi diterapkan secara on-the-fly pada setiap batch pelatihan sehingga meskipun jumlah file citra tetap, variasi citra yang dilihat model bertambah signifikan pada 150 epoch dengan 563 citra latih, model berpotensi memproses hingga 84.450 variasi citra unik. Mekanisme ini secara langsung berkontribusi pada kemampuan generalisasi model dan meminimalkan risiko overfitting (Adedigba dkk., 2021).

Dalam konteks dataset medis yang terbatas, augmentasi data memegang peran kritis sebagai strategi regularisasi implisit yang mencegah *overfitting* sekaligus meningkatkan kemampuan generalisasi model. Shorten & Khoshgoftaar (2019) dalam tinjauan komprehensif mereka menegaskan bahwa augmentasi geometris seperti *flipping*, *scaling*, dan translasi bekerja dengan mengekspos model terhadap variasi transformasi yang mungkin ditemui pada data uji nyata, sehingga model tidak hanya belajar dari satu perspektif representasi citra. Namun demikian, perlu dipahami bahwa augmentasi berbasis transformasi geometris pada dasarnya

menghasilkan turunan dari citra yang sama (*intra-sample variation*), sehingga keragaman yang dihasilkan bersifat variasi posisi dan orientasi, bukan variasi *inter-sample* yang sesungguhnya. Oleh karena itu, pemilihan jenis augmentasi yang tepat dan teroptimasi menjadi krusial agar variasi yang dihasilkan benar-benar relevan dengan kondisi klinis yang mungkin ditemui, seperti perbedaan posisi pasien, jarak pengambilan foto, maupun variasi perangkat radiografi.

2.2.8. Confusion Matrix

Confusion matrix merupakan teknik evaluasi visual yang digunakan dalam *machine learning* (Xu dkk., 2020). *Confusion matrix* bekerja dengan cara membandingkan suatu hasil klasifikasi terhadap nilai sebenarnya. *Confusion matrix* memberikan informasi mengenai perbandingan hasil klasifikasi dari model yang telah dibentuk oleh sistem. Gambar 2.5 menunjukkan visualisasi perbandingan menggunakan *confusion matrix*.



The diagram shows a 2x2 matrix with 'Nilai Prediksi' (Predicted Value) on the vertical axis and 'Nilai Aktual' (Actual Value) on the horizontal axis. The matrix cells are labeled as follows:

		Nilai Aktual	
		Positive	Negative
Nilai Prediksi	Positive	TP	FP
	Negative	FN	TN

Gambar 2.5 Confusion Matrix

Confusion matrix terdiri atas nilai *True Positive* (TP), *True Negative* (TN), *False Positive* (FP), dan *False Negative* (FN). TP menunjukkan suatu data yang bernilai positif yang berhasil terklasifikasi dalam kelas positif. TN menunjukkan suatu data yang bernilai negatif yang berhasil terklasifikasi dalam kelas negatif. FP menunjukkan suatu data yang bernilai negatif yang terklasifikasi salah ke dalam kelas positif. Sedangkan, FN menunjukkan suatu data yang bernilai positif yang terklasifikasi salah ke dalam kelas negatif.

Rumus yang ditunjukkan pada formula adalah sebagai berikut:

$$Akurasi = \frac{TP + TN}{TP + TN + FP + FN} \quad (2)$$

$$Presisi = \frac{TP}{TP + FP} \quad (3)$$

$$Recall = \frac{TP}{TP + FN} \quad (4)$$

$$F1 = \frac{2 \times Presisi \times Recall}{Presisi + Recall} \quad (5)$$

2.2.9. Mean Average Precision (mAP)

Mean Average Precision (mAP) merupakan metrik evaluasi utama yang digunakan secara luas dalam tugas deteksi objek berbasis *deep learning*, termasuk pada arsitektur YOLO. mAP mengukur kemampuan model secara komprehensif dalam mendeteksi dan melokalisasi objek secara akurat pada seluruh kelas yang diujikan. Berbeda dengan metrik akurasi pada klasifikasi biasa, mAP memperhitungkan baik ketepatan prediksi (*precision*) maupun kelengkapan deteksi (*recall*) sekaligus, sehingga lebih representatif untuk mengevaluasi performa sistem deteksi objek medis (Subha dkk., 2025).

Sebelum menghitung mAP, terlebih dahulu dihitung nilai *Average Precision* (AP) untuk setiap kelas. AP merepresentasikan luas area di bawah kurva *precision-recall* (kurva PR) yang diperoleh dengan memvariasikan ambang batas *confidence score* dari 0 hingga 1. Semakin besar luas area di bawah kurva PR, semakin baik kemampuan model dalam mendeteksi objek dari kelas tersebut. Kurva PR yang ideal mendekati sudut kanan atas, menunjukkan nilai *precision* dan *recall* yang keduanya tinggi secara bersamaan (Reis dkk., 2024).

Nilai AP untuk setiap kelas dihitung dengan mengintegrasikan kurva *precision-recall*, yang secara matematis dapat dinyatakan sebagai berikut:

$$AP = \int_0^1 p(r) dr \quad (6)$$

di mana $p(r)$ adalah nilai *precision* pada nilai *recall* r . Selanjutnya, mAP diperoleh dengan merata-ratakan nilai AP dari seluruh kelas yang dievaluasi, sebagaimana ditunjukkan pada Persamaan berikut:

$$mAP = \frac{1}{N} \sum_{k=1}^N AP_k \quad (7)$$

di mana N adalah jumlah kelas dan AP_k adalah nilai *Average Precision* untuk kelas ke- k .

Dalam konteks deteksi objek modern, perhitungan mAP selalu dikaitkan dengan nilai *Intersection over Union* (IoU), yaitu rasio antara luas area irisan (*intersection*) dan luas area gabungan (*union*) antara *bounding box* prediksi dan *bounding box ground truth*. IoU digunakan sebagai kriteria untuk menentukan apakah suatu prediksi diklasifikasikan sebagai *True Positive* (TP) atau *False Positive* (FP). Persamaan IoU dituliskan sebagai berikut:

$$IoU = \frac{\text{Area of Intersection}}{\text{Area of Union}} \quad (8)$$

Terdapat dua varian mAP yang umum digunakan, yaitu mAP50 dan mAP50-95. mAP50 mengukur akurasi deteksi pada satu ambang IoU tetap sebesar 0,50, artinya suatu prediksi dianggap benar apabila IoU antara *bounding box* prediksi dan *ground truth* $\geq 0,5$. Sementara itu, mAP50-95 merupakan rata-rata mAP yang dihitung pada rentang ambang IoU dari 0,50 hingga 0,95 dengan interval 0,05 (10 threshold), sehingga lebih ketat dan komprehensif dalam mengukur kemampuan lokalisasi model (Subha dkk, 2025). Perbandingan kedua metrik ini disajikan pada Tabel 2.1 berikut.

Tabel 2.1 Perbandingan mAP50 dengan mAP50-90

Metrik	Ambang IoU	Karakteristik
mAP50	IoU = 0,50	Lebih longgar; standar komparasi antar penelitian
mAP50-95	IoU = 0,50–0,95	Lebih ketat; mencerminkan presisi lokalisasi sesungguhnya

Dalam penelitian ini, mAP50-95 dipilih sebagai metrik utama karena secara langsung mencerminkan kualitas lokalisasi *bounding box* jantung dan rongga toraks yang dihasilkan model. Semakin presisi posisi *bounding box* yang dideteksi, semakin akurat pula hasil perhitungan *Cardiothoracic Ratio* (CTR) yang diperoleh dari koordinat *bounding box* tersebut (Ragab dkk., 2024). Dengan demikian, peningkatan nilai mAP50-95 secara langsung berimplikasi pada keandalan sistem diagnosis kardiomegali yang dikembangkan.

2.2.10. Framework Flask

Flask adalah *micro web framework* berbasis bahasa pemrograman Python yang dikembangkan oleh komunitas Pallets dan tersedia secara resmi melalui repositori Python Package Index (PyPI). Flask dirancang dengan prinsip minimalis dan modular, di mana pengembang hanya menggunakan komponen yang benar-benar diperlukan tanpa ketergantungan terhadap pustaka eksternal yang besar. Pendekatan ini menjadikan Flask sangat ringan, fleksibel, dan mudah dikustomisasi sesuai kebutuhan aplikasi, mulai dari layanan *web* sederhana hingga sistem berbasis kecerdasan buatan yang kompleks (Pallets Projects, 2023).

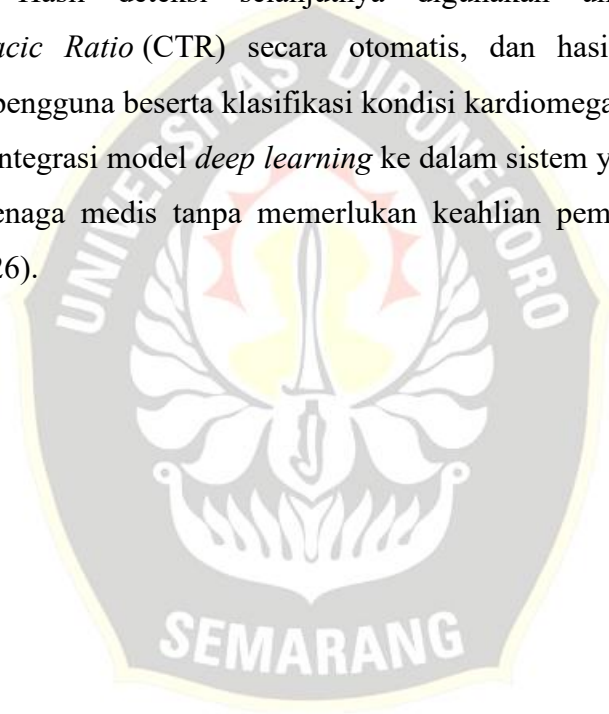
Arsitektur Flask mengikuti pola *request-response* berbasis protokol HTTP, di mana setiap URL dipetakan ke fungsi Python tertentu yang disebut *view function* melalui mekanisme *routing*. Flask mendukung metode HTTP seperti GET dan POST yang memungkinkan komunikasi data antara antarmuka pengguna (*frontend*) dan logika pemrosesan di sisi server (*backend*). Flask juga dilengkapi dengan *template engine* Jinja2 yang memungkinkan pembuatan antarmuka web dinamis dengan memisahkan logika bisnis dari tampilan antarmuka secara terstruktur (Aswathi & Nagappan, 2024).

Salah satu keunggulan Flask yang menjadikannya populer dalam pengembangan sistem berbasis *deep learning* adalah kemudahannya dalam mengintegrasikan pustaka Python seperti TensorFlow, PyTorch, dan Ultralytics YOLOv8 ke dalam alur kerja aplikasi web. Model *deep learning* yang telah dilatih dapat dimuat langsung ke dalam aplikasi Flask dan dieksekusi secara *real-time* ketika menerima input dari pengguna berupa unggahan citra medis. Hasil inferensi model selanjutnya dikembalikan kepada pengguna dalam bentuk respons JSON atau halaman web yang telah dirender melalui antarmuka yang interaktif (Nursyifa dkk, 2024).

Penerapan Flask dalam sistem deteksi objek berbasis YOLO untuk keperluan medis telah dibuktikan oleh berbagai penelitian. Pebrianata dkk. (2025) membuktikan bahwa integrasi YOLOv11 dengan Flask berhasil membangun antarmuka deteksi tumor otak secara *real-time* tanpa menurunkan nilai F1-*score* model, menunjukkan bahwa Flask tidak menambah latensi yang signifikan

pada proses inferensi. Senada dengan itu, Siddiqui dkk. (2024) mengembangkan sistem klasifikasi diagnosis medis dengan *backend* berbasis Flask yang mampu memproses input data pasien dan menghasilkan prediksi klasifikasi melalui integrasi model *machine learning* secara *real-time* dalam satu platform terpadu.

Dalam penelitian ini, Flask digunakan sebagai *backend framework* untuk membangun aplikasi web prediksi kardiomegali. Aplikasi menerima input berupa citra rontgen dada yang diunggah oleh tenaga medis, kemudian meneruskannya ke model hybrid YOLOv8-ResNet50 untuk proses deteksi *bounding box* jantung dan rongga toraks. Hasil deteksi selanjutnya digunakan untuk menghitung nilai *Cardiothoracic Ratio* (CTR) secara otomatis, dan hasilnya ditampilkan kembali kepada pengguna beserta klasifikasi kondisi kardiomegali. Pendekatan ini memungkinkan integrasi model *deep learning* ke dalam sistem yang dapat diakses langsung oleh tenaga medis tanpa memerlukan keahlian pemrograman khusus (Harish dkk., 2026).



SEKOLAH PASCASARJANA