

LAPORAN PROYEK AKHIR COMPUTER VISION

Klasifikasi Penyakit Paru-paru berdasarkan Citra X-Ray

Dada



Oleh Kelompok 1:

Steven - 412022006

Steven Felizio - 412023011

Sanders Keane Dylan D - 412023020

Bintang Talenta Putra Pradana - 412023022

FAKULTAS TEKNIK DAN ILMU KOMPUTER

PROGRAM STUDI INFORMATIKA

UNIVERSITAS KRISTEN KRIDA WACANA

JAKARTA

2025

BAB I

PENDAHULUAN

1.1 Latar Belakang

Deteksi kelainan pada citra radiograf dada (chest X-ray) merupakan bagian penting dalam diagnosis berbagai penyakit paru seperti pneumonia, atelectasis, dan infiltrasi. Radiograf dada banyak digunakan karena murah dan cepat, namun interpretasinya membutuhkan keahlian tinggi dari radiolog. Di banyak fasilitas kesehatan, terutama yang memiliki keterbatasan tenaga ahli, proses pembacaan citra dapat menjadi lambat dan tidak konsisten, sehingga memunculkan kebutuhan akan sistem bantu berbasis kecerdasan buatan.

Kemajuan deep learning, khususnya Convolutional Neural Network (CNN), telah memberikan hasil yang sangat baik pada analisis citra medis. Arsitektur seperti DenseNet-121, yang juga digunakan dalam model CheXNet, mampu mengekstraksi fitur kompleks dari citra X-ray dengan efisien melalui koneksi padat (dense connections). Dengan transfer learning, model dapat dilatih lebih cepat dan stabil meskipun data terbatas.

Proyek ini membangun pipeline end-to-end untuk klasifikasi kelainan pada citra X-ray menggunakan DenseNet-121. Notebook mencakup proses pemanggilan dataset, pembuatan CustomDataset, preprocessing dan augmentasi citra, serta pembuatan DataLoader. Penanganan class imbalance juga diterapkan untuk mengurangi bias terhadap kelas mayoritas, sehingga model mampu mempelajari kelas minoritas secara lebih baik.

Pelatihan dilakukan dengan memodifikasi lapisan akhir DenseNet-121 sesuai jumlah kelas target, kemudian diikuti proses training dan validasi untuk memantau performa dari setiap epoch. Dengan pendekatan ini, model diharapkan mampu mengenali pola penting pada citra X-ray dan memberikan prediksi yang lebih akurat.

Secara keseluruhan, sistem berbasis CNN yang dikembangkan berpotensi menjadi alat bantu bagi radiolog dalam mempercepat triase, meningkatkan konsistensi pemeriksaan, dan mendukung pelayanan kesehatan di daerah dengan keterbatasan tenaga ahli.

1.2 Rumusan Masalah

Berdasarkan latar belakang di atas, maka rumusan masalah dalam penelitian ini adalah sebagai berikut:

1. Bagaimana membangun model Convolutional Neural Network (CNN) berbasis DenseNet-121 yang mampu melakukan klasifikasi kelainan pada citra X-ray?
2. Bagaimana menerapkan preprocessing, augmentasi citra, dan teknik penanganan class imbalance agar model dapat belajar lebih efektif, terutama pada kelas minoritas?
3. Bagaimana merancang proses pelatihan dan validasi model untuk memantau performanya secara bertahap pada setiap epoch?
4. Sejauh mana model CNN yang dikembangkan mampu memberikan prediksi yang akurat sebagai sistem bantu deteksi kelainan toraks?
5. Bagaimana mengimplementasikan model tersebut ke dalam aplikasi web interaktif berbasis Next.js sehingga mampu melakukan inferensi citra X-ray secara langsung dan menampilkan visualisasi interpretabilitas seperti Grad-CAM?

1.3 Tujuan Penelitian

Tujuan dari proyek penelitian ini adalah:

1. Mengembangkan model Convolutional Neural Network (CNN) berbasis DenseNet-121 untuk klasifikasi kelainan pada citra radiograf dada.
2. Menerapkan preprocessing, augmentasi citra, serta strategi penanganan class imbalance untuk meningkatkan kemampuan model dalam mengenali pola pada kelas mayoritas maupun minoritas.
3. Membangun pipeline pelatihan dan validasi yang terstruktur, meliputi pembuatan CustomDataset, DataLoader, training loop, serta evaluasi performa model pada setiap epoch.
4. Mengevaluasi performa model berdasarkan metrik seperti akurasi, loss, dan kemampuan model dalam mengenali pola kelainan secara konsisten.
5. Mengembangkan aplikasi web interaktif berbasis Next.js yang dapat melakukan inferensi citra X-ray secara langsung dan menyajikan visualisasi Grad-CAM untuk meningkatkan interpretabilitas model.

1.4 Manfaat Penelitian

Proyek ini berpotensi memberikan manfaat bagi beberapa pihak sebagai berikut:

1. Bagi Tenaga Kesehatan (Radiolog dan Dokter Klinis): Sistem klasifikasi berbasis CNN dapat berfungsi sebagai alat bantu (second reader) yang membantu mempercepat proses triase, meningkatkan konsistensi pembacaan citra, dan mengurangi beban kerja radiolog, terutama di fasilitas kesehatan dengan keterbatasan tenaga ahli. Dengan adanya prediksi otomatis yang cepat dan stabil, proses diagnosis awal dapat dilakukan lebih efisien.
2. Bagi Pasien: Pasien berpotensi mendapatkan diagnosis yang lebih cepat dan lebih akurat, karena sistem mampu memberikan hasil prediksi awal yang membantu identifikasi dini kelainan toraks. Hal ini dapat mengurangi keterlambatan tindakan medis dan memperbaiki kualitas pelayanan kesehatan, terutama di daerah yang kekurangan radiolog berpengalaman.

3. Bagi Komunitas Riset dan Akademisi: Penelitian ini memberikan kontribusi pada bidang Computer Vision dan analisis citra medis melalui pembangunan pipeline klasifikasi X-ray berbasis DenseNet-121, teknik penanganan class imbalance, serta integrasi interpretabilitas model menggunakan Grad-CAM. Hasil penelitian berupa model terlatih, kode, dan dokumentasi dapat digunakan sebagai referensi atau baseline untuk penelitian lanjutan.
4. Bagi Fasilitas Kesehatan: Sistem yang dikembangkan dapat menjadi langkah awal menuju solusi augmentasi diagnostik berbasis AI. Implementasi dalam bentuk aplikasi web memungkinkan integrasi yang lebih fleksibel dan mudah diakses, sehingga fasilitas kesehatan dapat meningkatkan efisiensi operasional tanpa memerlukan perangkat keras khusus.
5. Bagi Pengembang Sistem dan Komunitas Teknologi: Implementasi model ke dalam aplikasi Next.js memberikan contoh nyata bagaimana model deep learning dapat dioptimalkan dan diintegrasikan ke dalam platform web modern, termasuk penyajian interpretabilitas (Grad-CAM) secara langsung kepada pengguna.

1.5 Ruang Lingkup Pembahasan

Ruang lingkup penelitian ini dibatasi sebagai berikut:

1. Fokus penelitian adalah klasifikasi citra X-ray toraks menggunakan model Convolutional Neural Network (CNN) berbasis DenseNet-121. Model hanya melakukan klasifikasi 14 jenis kelainan toraks sebagaimana yang tersedia pada dataset NIH ChestX-ray14, tanpa melakukan segmentasi organ, deteksi bounding box, atau analisis spasial mendalam.
2. Dataset yang digunakan adalah NIH ChestX-ray14, yang terdiri dari 112.120 citra X-ray dada tampak depan dengan label 14 penyakit toraks. Dataset diambil dari repositori publik Kaggle dan tidak melibatkan pengumpulan data klinis baru. Dataset ini memiliki keterbatasan seperti label noise yang berasal dari proses ekstraksi label berbasis NLP, namun tingkat akurasi label tetap dianggap memadai (>90%).

3. Pipeline yang dibangun mencakup preprocessing, augmentasi citra, pembuatan CustomDataset, DataLoader, training loop, validasi model, serta penanganan class imbalance. Penelitian ini tidak mencakup pengembangan arsitektur hybrid atau ensemble, tetapi fokus utamanya adalah DenseNet-121 sebagai model utama.
4. Evaluasi performa model dilakukan menggunakan metrik umum klasifikasi, seperti akurasi, precision, recall, F1-score, dan AUC. Evaluasi dilakukan hanya pada dataset uji (test set) yang berasal dari dataset yang sama dan tidak mencakup validasi klinis.

BAB II

LANDASAN TEORI

2.1 Computer Vision

Computer Vision adalah cabang dari Artificial Intelligence (AI) yang berfokus pada bagaimana komputer dapat menafsirkan konten visual dari dunia nyata, seperti gambar, foto, dan video. Secara konsep, Computer Vision meniru cara kerja sistem visual manusia, mulai dari persepsi (vision), interpretasi, hingga pengambilan keputusan berdasarkan pola-pola visual. Dalam praktiknya, Computer Vision melibatkan serangkaian proses seperti ekstraksi fitur, transformasi citra, segmentasi, deteksi objek, klasifikasi, dan pelacakan objek. Seiring berkembangnya teknologi komputasi seperti GPU dan TPU, kemampuan Computer Vision meningkat drastis karena metode deep learning dapat dilatih menggunakan dataset besar yang kompleks. Khan et al. (2021) menyatakan bahwa perkembangan model deep learning seperti CNN menjadikan Computer Vision sebagai fondasi utama berbagai aplikasi kecerdasan buatan modern, termasuk sistem pengenalan wajah, analisis medis berbasis citra, otomasi industri, hingga kendaraan otonom. Perkembangan ini membuat Computer Vision tidak hanya menjadi bidang teoritis, tetapi juga tulang punggung berbagai sistem dunia nyata yang memerlukan akurasi dan kecepatan tinggi.

2.2 Image Classification

Image Classification adalah salah satu tugas utama dalam Computer Vision yang bertujuan memberikan label atau kategori pada suatu gambar berdasarkan karakteristik visualnya. Proses klasifikasi ini membutuhkan sistem yang mampu mengenali pola tepi, warna, tekstur, maupun bentuk objek dan kemudian memetakan pola tersebut ke kelas tertentu. Metode klasifikasi citra tradisional awalnya bergantung pada ekstraksi fitur manual seperti SIFT, SURF, dan HOG. Namun, sejak munculnya Convolutional Neural Networks (CNN), proses ekstraksi fitur dilakukan secara otomatis oleh model sehingga

menghasilkan performa jauh lebih baik dibandingkan teknik klasik. Li & Chen (2022) menjelaskan bahwa penggunaan deep learning memungkinkan model mempelajari representasi visual secara hierarkis, mulai dari fitur rendah (low-level features) hingga struktur kompleks (high-level semantic features). Image Classification digunakan di berbagai sektor seperti perbankan (KYC), pertanian (deteksi penyakit tanaman), kesehatan (klasifikasi citra medis), keamanan (deteksi intruder), serta e-commerce (pengenalan produk). Keberhasilan klasifikasi sangat bergantung pada kualitas dataset, preprocessing, dan arsitektur model.

2.3 Dataset dan Preprocessing Citra

Dataset menjadi fondasi utama dalam proses pelatihan model klasifikasi citra karena model hanya dapat belajar berdasarkan data yang diberikan. Dataset yang berkualitas harus memiliki variasi yang cukup dalam hal pencahayaan, posisi, ukuran objek, orientasi, serta kondisi lingkungan lain. Tanpa variasi tersebut, model akan mudah mengalami overfitting dan gagal melakukan generalisasi. Preprocessing citra merupakan langkah penting sebelum data dimasukkan ke model. Proses resize memastikan setiap citra memiliki dimensi seragam sehingga dapat diproses dalam batch. Normalisasi piksel diperlukan karena nilai intensitas yang besar dapat memperlambat proses pelatihan, sementara normalisasi mempercepat konvergensi dan meningkatkan stabilitas. Augmentasi citra seperti rotasi, crop, shifting, zooming, dan flipping membantu meningkatkan jumlah sampel secara buatan sehingga mengurangi risiko overfitting, terutama ketika dataset asli terbatas. Shorten & Khoshgoftaar (2021) menegaskan bahwa augmentasi data adalah metode yang sangat efektif dalam meningkatkan kemampuan generalisasi model deep learning karena memperkaya variasi visual tanpa perlu pengumpulan data tambahan.

2.4 Deep Learning dan Neural Network

Deep Learning merupakan pendekatan dalam machine learning yang memanfaatkan jaringan saraf tiruan (Artificial Neural Networks) dengan jumlah lapisan yang sangat banyak (deep). Model ini terdiri dari neuron yang saling terhubung dan mampu belajar melalui proses optimisasi menggunakan algoritma

backpropagation. Semakin dalam struktur jaringan, semakin kompleks pola yang dapat dipelajari. Dalam konteks citra, deep learning sangat unggul karena mampu mengekstraksi fitur visual tanpa intervensi manusia. Goodfellow et al. (2021) menjelaskan bahwa deep learning bekerja melalui feature hierarchy, yaitu kemampuan mempelajari fitur sederhana di layer awal dan fitur kompleks di layer lebih dalam. Keunggulan utama deep learning adalah kemampuannya mempelajari representasi data dalam jumlah besar dan bekerja sangat baik pada data tidak terstruktur seperti gambar dan suara. Hal ini menjadikan deep learning sebagai dasar dalam berbagai sistem modern termasuk face recognition, object detection, OCR (optical character recognition), dan autonomous driving.

2.5 Convolutional Neural Network (CNN)

Convolutional Neural Network adalah arsitektur deep learning yang dirancang khusus untuk menangani data visual. CNN memanfaatkan lapisan konvolusi (convolution layer) untuk mengekstraksi fitur spasial melalui kernel atau filter yang bergerak pada citra. Filter ini akan menangkap pola penting seperti edge, corners, curves, hingga pola abstrak yang lebih tinggi. CNN juga menggunakan pooling layer sebagai proses downsampling untuk mengurangi ukuran fitur namun tetap mempertahankan informasi penting. Layer aktivasi seperti ReLU digunakan untuk memperkenalkan non-linearitas, sementara fully connected layer biasanya ditempatkan di bagian akhir untuk melakukan klasifikasi. Albawi et al. (2022) menyebutkan bahwa CNN mampu mengurangi jumlah parameter secara signifikan dibandingkan fully connected network sehingga lebih efisien meskipun bekerja pada citra berukuran besar. CNN juga memiliki keunggulan dalam hal translational invariance, artinya model tetap mampu mengenali objek meskipun terjadi sedikit pergeseran posisi.

2.6 Arsitektur CNN Modern (ResNet, DenseNet, EfficientNet)

2.6.1 ResNet

ResNet (Residual Network) memperkenalkan konsep skip connection atau residual connection, yang memungkinkan model membentuk jaringan sangat

dalam tanpa mengalami degradasi performa. Model CNN tradisional memiliki limit kedalaman karena semakin banyak layer menyebabkan vanishing gradient. ResNet mengatasi hal tersebut dengan menambahkan shortcut antara layer tertentu, sehingga gradien dapat mengalir dengan baik selama proses training. He et al. (2021) menunjukkan bahwa ResNet mampu mencapai akurasi sangat tinggi pada dataset ImageNet dengan lebih dari 150 layer.

2.6.2 DenseNet

DenseNet (Densely Connected Network) memperluas konsep residual dengan menghubungkan setiap layer ke semua layer sebelumnya. Hal ini memungkinkan fitur mengalir secara maksimal dan meminimalisasi redundansi. DenseNet dikenal sangat efisien dan menghasilkan representasi fitur yang kaya. Huang et al. (2022) menyatakan bahwa DenseNet sangat baik digunakan pada dataset yang ukurannya kecil hingga sedang.

2.6.3 EfficientNet

EfficientNet mengoptimalkan scaling jaringan menggunakan compound scaling, yaitu menyeimbangkan skala kedalaman, lebar, dan resolusi citra. Pendekatan sistematis ini membuat EfficientNet lebih efisien dibanding model besar lain. Tan & Le (2021) menyatakan bahwa EfficientNet dapat mencapai performa tinggi dengan jumlah parameter yang jauh lebih sedikit, sehingga sangat baik digunakan pada deployment produksi.

2.7 Optimizer dan Fungsi Loss

Optimizer adalah algoritma yang digunakan untuk memperbarui bobot model selama proses pelatihan agar loss function dapat diminimalkan. Beberapa optimizer yang sering digunakan antara lain SGD (Stochastic Gradient Descent), Adam, dan RMSProp. Adam merupakan salah satu optimizer paling populer karena menggabungkan keunggulan momentum dan adaptive learning rate, sehingga dapat beradaptasi dengan cepat terhadap karakteristik gradien. Loss function seperti Categorical Cross Entropy digunakan untuk klasifikasi multiclass,

sedangkan Binary Cross Entropy digunakan ketika label bersifat biner. Ruder (2021) menegaskan bahwa pemilihan optimizer sangat memengaruhi kecepatan konvergensi dan kualitas representasi yang dipelajari oleh model.

2.8 Teknik Regularisasi Model (Dropout, Early Stopping, Batch Normalization)

Regularisasi merupakan teknik untuk mencegah overfitting, yaitu kondisi ketika model terlalu baik dalam mempelajari data pelatihan namun buruk dalam generalisasi. Dropout adalah metode yang menonaktifkan neuron secara acak sehingga model tidak terlalu bergantung pada satu pola spesifik. Early stopping menghentikan proses training ketika akurasi validasi tidak meningkat, sehingga mencegah model overfit pada epoch berlebih. Batch Normalization membantu distribusi data tetap stabil sehingga proses training menjadi lebih cepat dan stabil. Wang et al. (2023) menunjukkan bahwa kombinasi teknik-teknik regularisasi dapat meningkatkan akurasi sekaligus stabilitas model secara signifikan.

2.9 Evaluasi Model

Evaluasi model diperlukan untuk mengetahui seberapa baik model dalam melakukan prediksi pada data yang belum pernah dilihat. Proses evaluasi biasanya menggunakan data test set atau validation set. Metrik evaluasi yang umum digunakan dalam klasifikasi mencakup accuracy, precision, recall, F1-score, confusion matrix, dan ROC-AUC. Pemilihan metrik sangat bergantung pada tujuan sistem dan tingkat keseimbangan kelas dalam dataset. Pada dataset tidak seimbang, accuracy bukan metrik yang ideal sehingga perlu digunakan precision, recall, atau F1-score.

2.9.1 Accuracy, Precision, Recall, dan F1-Score

Accuracy merupakan metrik dasar yang digunakan untuk menghitung persentase prediksi benar terhadap seluruh sampel. Namun, metrik ini cenderung menyesatkan apabila dataset tidak seimbang (imbalanced dataset). Misalnya, jika 90% data merupakan kelas negatif, model yang selalu memprediksi negatif akan

memperoleh akurasi 90% tetapi tidak memiliki kemampuan klasifikasi yang sebenarnya.

Precision digunakan untuk menilai seberapa tepat prediksi positif yang dihasilkan model. Precision tinggi berarti mayoritas prediksi positif adalah benar. Hal ini penting pada kasus yang memerlukan tingkat kesalahan positif (false positive) rendah, seperti sistem keamanan atau deteksi fraud.

Recall mengukur kemampuan model dalam mendeteksi semua sampel positif yang ada. Recall tinggi menunjukkan bahwa model berhasil “menangkap” sebagian besar kasus positif. Metrik ini sangat penting untuk bidang seperti kesehatan, di mana kegagalan mendeteksi kasus positif (false negative) dapat berakibat fatal.

F1-score merupakan rata-rata harmonis antara precision dan recall yang memberikan evaluasi lebih seimbang, terutama jika satu metrik lebih dominan daripada yang lain. Saito & Rehmsmeier (2021) menegaskan bahwa F1-score sangat ideal untuk dataset tidak seimbang karena tidak bias seperti accuracy dan memberikan representasi kesalahan yang lebih objektif.

2.9.2 ROC-AUC

ROC menggambarkan hubungan antara True Positive Rate (TPR) dan False Positive Rate (FPR). AUC (Area Under Curve) mengukur kualitas model dengan melihat luas area di bawah kurva ROC. Semakin mendekati 1, semakin baik model. ROC-AUC sangat penting dalam kasus klasifikasi medis atau citra, di mana kesalahan prediksi dapat berdampak besar. Wen et al. (2022) menunjukkan bahwa AUC merupakan indikator kekuatan model dalam membedakan kelas positif dan negatif.

2.10 Next.js

Next.js adalah framework React modern yang mendukung berbagai metode rendering seperti Server-Side Rendering (SSR), Static Site Generation (SSG), dan Client-Side Rendering (CSR). Framework ini dirancang untuk

memberikan pengalaman pengembangan yang cepat, modular, dan sangat performatif. Next.js menyediakan API Routes sehingga backend sederhana dapat dibangun dalam satu proyek. Selain itu, Next.js memiliki optimasi otomatis seperti code splitting, image optimization, routing yang intuitif, serta dukungan SEO yang kuat berkat kemampuan SSR. Dokumentasi Vercel (2023) menunjukkan bahwa proyek yang dibangun dengan Next.js memiliki performa 30–40% lebih cepat dibandingkan aplikasi React konvensional.

BAB III

PERANCANGAN DAN METODE

3.1 Arsitektur Sistem

Arsitektur sistem dalam penelitian ini dirancang untuk melakukan klasifikasi multi-label penyakit paru-paru dari citra X-ray dada menggunakan model deep learning DenseNet121. Sistem dibangun dalam beberapa tahapan terstruktur yang saling terhubung, mulai dari pemanggilan dataset, eksplorasi, pre-processing, balancing, pelatihan model, hingga evaluasi. Secara umum, arsitektur sistem terdiri dari enam komponen utama, yaitu:

1. Pemanggilan Dataset dan Metadata

Tahap awal adalah memuat dataset NIH Chest X-ray14, yang terdiri dari file citra dan file metadata. Pada tahap ini dilakukan:

- a. Pembacaan file .csv yang berisi informasi label penyakit dan data pasien.
- b. Pembacaan file .json yang berisi pemetaan antara nama file dengan path citra.
- c. Validasi pemisahan data train, validation, dan test untuk memastikan tidak ada kebocoran data antar pasien.

Tahap ini penting agar model memperoleh data yang benar dan terorganisasi.

2. Exploratory Data Analysis (EDA)

EDA dilakukan untuk memahami karakteristik dataset secara mendalam. Aktivitas yang dilakukan meliputi:

- a. Analisis distribusi jumlah sampel per label
- b. Analisis co-occurrence (kombinasi penyakit yang sering muncul bersama)
- c. Pemeriksaan sampel citra secara visual
- d. Analisis metadata seperti usia, gender pasien

- e. Pemeriksaan missing values

EDA membantu mengidentifikasi potensi masalah seperti ketidakseimbangan kelas, bias populasi, atau noise data yang harus ditangani sebelum training.

3. Pre-processing dan Augmentasi Gambar

Sebelum citra dipakai untuk pelatihan, dilakukan beberapa tahapan pre-processing:

- a. Resize citra menjadi 256×256
- b. Konversi grayscale ke RGB, karena DenseNet membutuhkan 3 channel
- c. Normalisasi menggunakan mean dan std ImageNet
- d. Augmentasi seperti flip, rotate, noise, brightness adjustment

Tahap ini memastikan citra konsisten dengan format yang diterima model serta menambah variasi data untuk mencegah overfitting.

4. Penanganan Imbalance Data

Dataset NIH sangat tidak seimbang (misalnya Effusion sangat banyak, Hernia sangat sedikit). Untuk mengurangi bias:

- a. Digunakan WeightedRandomSampler, yang memberikan bobot sampling lebih tinggi pada kelas minoritas.
- b. Setiap sampel diberi bobot berdasarkan jumlah label yang diwakili.
- c. Sampler memastikan distribusi label dalam batch lebih seimbang.

Pendekatan ini meningkatkan peluang model belajar dari kelas-kelas langka.

5. Pelatihan Model DenseNet121 dengan Transfer Learning

Tahap inti sistem adalah pelatihan deep learning:

- a. Menggunakan model DenseNet121 dari library timm
- b. Memuat bobot pretrained dari ImageNet
- c. Mengubah layer terakhir menjadi output 13 kelas (multi-label)
- d. Menggunakan loss function BCEWithLogitsLoss
- e. Optimizer AdamW
- f. Scheduler ReduceLROnPlateau

Training dilakukan dalam loop beberapa epoch, dengan validasi setiap epoch untuk memonitor kinerja model.

6. Evaluasi, Testing, dan Prediksi Akhir

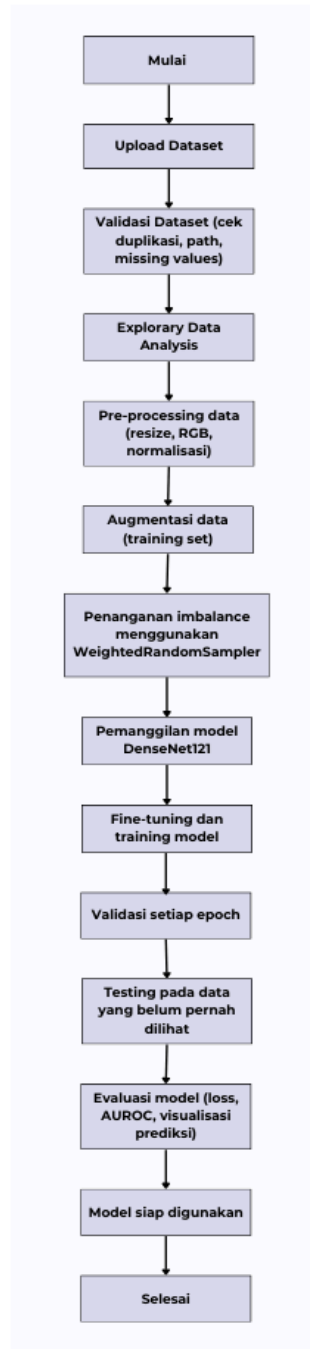
Setelah training selesai, model diuji pada data test yang belum pernah dilihat sebelumnya. Evaluasi dilakukan menggunakan:

- a. Loss validasi
- b. AUROC (macro)
- c. Visualisasi prediksi (benar dan salah)

Hasil evaluasi menunjukkan kemampuan model melakukan generalisasi.

3.1.1. Flowchart Arsitektur Sistem

Flowchart arsitektur sistem menggambarkan alur pemrosesan dari tahap pemanggilan dataset hingga evaluasi akhir model. Diagram ini menunjukkan bagaimana data citra X-ray dan metadata diproses, dibersihkan, di-augmentasi, dilatih menggunakan DenseNet121, serta dievaluasi menggunakan metrik AUROC. Berikut adalah flowchart arsitektur sistem:



Gambar 3.1.1 Flowchart Arsitektur Sistem

3.2. Pengumpulan Dan Persiapan Data

Tahap pengumpulan dan persiapan data merupakan langkah awal yang menentukan kualitas hasil klasifikasi multi-label pada penelitian ini. Dataset NIH Chest X-ray14 yang telah difilter dimuat melalui tiga file CSV (train, validation, test) dan satu file JSON untuk pemetaan path citra. Seluruh data diverifikasi untuk memastikan tidak ada duplikasi, kesalahan path, maupun label yang tidak lengkap. Setelah itu, dilakukan pre-processing untuk menyeragamkan format citra dan mengubah label menjadi vektor multi-hot, dilanjutkan augmentasi pada data training untuk menambah variasi visual. Dataset kemudian dibagi menjadi tiga subset utama agar proses pelatihan, validasi, dan pengujian model dapat berlangsung secara terstruktur dan objektif.

3.2.1. Sumber Dataset

Dataset yang digunakan dalam penelitian ini adalah NIH Chest X-ray14, yang berasal dari platform Kaggle dengan tautan sebagai berikut:

<https://www.kaggle.com/datasets/nih-chest-xray>

Dataset ini dirilis oleh National Institutes of Health (NIH), salah satu lembaga kesehatan terbesar di dunia, sehingga reliabilitas dataset ini sangat tinggi untuk penelitian berbasis citra medis.

Dataset dalam penelitian ini diambil dalam bentuk file terfilter yang sudah disediakan, yaitu:

- a. filtered_train_split.csv
- b. filtered_val_split.csv
- c. filtered_test_split.csv
- d. filtered_filename_to_path_map.json

File CSV berisi label penyakit untuk setiap citra, sedangkan file JSON menyimpan lokasi path citra (absolute path mapping).

3.2.2. Deskripsi Dataset

Dataset NIH Chest X-ray14 memiliki karakteristik sebagai berikut:

Informasi	Keterangan
Total Gambar	112.120 citra X-ray dada
Total Pasien	30.805 pasien
Jenis Kelas	14 jenis penyakit thoraks (multi-label)
Resolusi Gambar	1024x1024 piksel
Format	Grayscale (1 channel)
Format Label	Multi-label (1 gambar dapat memiliki > 1 penyakit)

Adapun 14 kelas penyakit yang terdapat pada dataset adalah:

1. *Atelectasis*
2. *Cardiomegaly*
3. *Effusion*
4. *Infiltration*
5. *Mass*
6. *Nodule*
7. *Pneumonia*
8. *Pneumothorax*
9. *Consolidation*
10. *Edema*
11. *Emphysema*
12. *Fibrosis*
13. *Pleural Thickening*
14. *Hernia*

Selain itu, dataset juga menyertakan kategori “*No Finding*” untuk gambar *x-ray* yang tidak menunjukkan kelainan.

3.2.3 Pembersihan dan Validasi Data

Tahapan pembersihan data dilakukan untuk memastikan dataset layak digunakan:

1. Memastikan tidak ada duplikasi data per pasien
Pada penelitian medis, kebocoran data antar subset dapat menyebabkan hasil tidak valid.
2. Memastikan kesesuaian path gambar
Menggunakan JSON mapping untuk menghindari missing file.
3. Memeriksa keberadaan “NaN” pada kolom label
Jika ditemukan, diubah menjadi label kosong.
4. Memastikan format label multi-label terkonversi menjadi array binary
Contoh: "Mass|Nodule" \rightarrow [0,0,1,0,1,...].

3.2.4 Pembagian Dataset

Untuk memastikan kemampuan model, dataset dibagi menjadi tiga subnet utama dengan rasio sebagai berikut:

1. *Data Training*: 60%
Digunakan untuk melatih model agar mampu mengenali pola dan fitur visual dari masing-masing kategori penyakit
2. *Data Validation*: 20%
Digunakan untuk mengevaluasi performa model selama proses *Training*, memantau *overfitting*, serta menyesuaikan *hyperparameter*
3. *Data Testing*: 20%
Digunakan sebagai data independen untuk mengukur performa model.

3.3. Pre-processing dan Augmentasi Data

Tahap pre-processing dan augmentasi data bertujuan memastikan bahwa citra X-ray yang digunakan dalam pelatihan memiliki format yang konsisten, bersih, dan sesuai kebutuhan model. Pada tahap ini, seluruh gambar diproses mulai dari pemetaan file, konversi channel, penyeragaman ukuran, normalisasi, hingga transformasi label menjadi bentuk multi-hot. Selain itu, augmentasi diterapkan secara terarah pada subset training untuk meningkatkan variasi visual

dan memperkuat kemampuan model dalam menghadapi beragam kondisi pencitraan. Langkah ini menjadi fondasi penting sebelum dataset digunakan dalam proses pelatihan dan evaluasi model.

3.3.1 Pre-Processing Data

Tahap *pre-processing* dilakukan untuk memastikan bahwa gambar dan label dalam kondisi bersih, seragam, serta siap digunakan dalam proses pelatihan model. *Pre-Processing* ini mencakup beberapa langkah sebagai berikut:

1. Pemrosesan Data

Pada tahap ini, metadata gambar dimuat menggunakan Pandas dari file CSV yang telah disediakan, yaitu `filtered_train_split.csv`, `filtered_val_split.csv`, dan `filtered_test_split.csv`. Setiap baris data berisi nama file X-ray serta 13 label penyakit dalam bentuk multi-label. Selain itu, file `filtered_filename_to_path_map.json` digunakan untuk memetakan nama file ke lokasi file gambar sebenarnya. Citra kemudian dibaca menggunakan OpenCV dengan mode grayscale, kemudian dikonversi menjadi format RGB menggunakan `cv2.cvtColor`, karena model DenseNet121 membutuhkan input tiga channel. Semua label penyakit diubah menjadi vektor multi-hot sehingga dapat digunakan dalam klasifikasi multi-label.

1. Resizing

Citra X-ray yang memiliki resolusi asli 1024×1024 piksel diseragamkan ukurannya menjadi 256×256 piksel menggunakan fungsi `A.Resize` dari `Albumentations`. Ukuran ini disesuaikan dengan konfigurasi model untuk mengurangi beban komputasi dan mempercepat proses training, tanpa menghilangkan struktur penting dari citra toraks.

2. Normalization

Setelah citra di-resize, proses normalisasi dilakukan agar nilai piksel citra memiliki distribusi yang sesuai dengan standar data ImageNet. Normalisasi ini menggunakan nilai mean $[0.485, 0.456, 0.406]$ dan std $[0.229, 0.224, 0.225]$ yang diterapkan melalui

Albumentations. Langkah ini membantu model mencapai konvergensi lebih cepat serta menjaga stabilitas gradient selama proses training.

3.3.2 Augmentasi Data

Augmentasi data dilakukan secara khusus pada data training untuk memperkaya variasi visual dan membantu model beradaptasi dengan kondisi pencitraan X-ray yang berbeda-beda. Teknik augmentasi yang digunakan dirancang agar tetap mempertahankan ciri-ciri anatomi penting tanpa merusak struktur gambarnya.

1. Brightness Adjustment

Model menerapkan RandomBrightnessContrast yang mengatur kecerahan dan kontras secara acak. Teknik ini mensimulasikan kondisi pencahayaan yang bervariasi dalam proses radiografi, sehingga model menjadi lebih adaptif terhadap gambar dengan tingkat kejelasan dan intensitas cahaya yang berbeda.

2. Darkening

Sebagai kebalikan dari peningkatan kecerahan, beberapa citra secara acak digelapkan melalui RandomGamma. Proses ini meniru kasus klinis di mana X-ray mengalami penurunan kualitas pencahayaan. Model yang terlatih dengan variasi gamma akan lebih mampu mengenali pola meskipun citra tampak lebih gelap dan berkontras rendah.

3. Geometric Augmentation (Rotation & Flip)

Model menggunakan HorizontalFlip untuk memberikan variasi orientasi citra, serta Rotate dengan batas rotasi ± 15 derajat. Rotasi kecil ini mensimulasikan variasi sudut pengambilan radiografi yang umum terjadi di rumah sakit. Variasi orientasi ini penting agar model tidak hanya mengandalkan pola posisi tertentu.

4. Noise Injection

Augmentasi *GaussNoise* digunakan untuk menambahkan derau berintensitas rendah pada gambar. Hal ini bertujuan mensimulasikan gangguan perangkat radiografi, seperti noise sensor atau distorsi akibat kualitas alat. Teknik ini meningkatkan ketahanan model terhadap gambar yang tidak sepenuhnya bersih.

5. Color Jitter

Meskipun citra X-ray berasal dari grayscale, setelah dikonversi menjadi RGB, *ColorJitter* diaplikasikan untuk membuat variasi kecil pada intensitas warna. Efeknya tidak merusak citra, tetapi membantu memperkaya representasi visual yang dapat diproses model.

Seluruh teknik augmentasi ini diterapkan secara acak pada data training, sementara data validasi dan testing hanya melewati proses resizing dan normalisasi tanpa augmentasi tambahan. Strategi ini memastikan bahwa evaluasi model tetap objektif dan tidak terpengaruh variasi buatan.

3.4 Perancangan Arsitektur Model

Perancangan arsitektur model bertujuan untuk membangun sistem klasifikasi multi-label yang mampu mengenali 13 jenis kelainan paru-paru melalui citra X-ray. Dalam penelitian ini digunakan model DenseNet121 dengan pendekatan transfer learning, mengingat kinerjanya yang stabil dan efisien dalam permasalahan klasifikasi citra medis.

3.4.1 Perancangan Arsitektur Model

DenseNet121 merupakan arsitektur Convolutional Neural Network (CNN) yang menghubungkan setiap layer ke seluruh layer selanjutnya melalui dense connections. Pendekatan ini memungkinkan pemanfaatan kembali fitur yang telah diekstraksi di layer sebelumnya, sehingga mengurangi risiko vanishing gradient dan menambah efisiensi parameter. Secara umum, DenseNet121 terdiri dari beberapa blok utama:

1. Convolutional Initial Layer: lapisan pertama yang melakukan konvolusi awal pada citra input.
2. Dense Block (4 blok utama): tiap blok memiliki banyak dense layers yang saling terhubung.
3. Transition Layers: menghubungkan antar dense block dan melakukan reduksi dimensi.
4. Global Average Pooling: mereduksi fitur ke bentuk vektor.
5. Fully Connected Layer: layer klasifikasi akhir yang diganti sesuai jumlah kelas.

Dalam penelitian ini, layer classifier asli diganti menjadi:

`nn.Linear(in_features, 13)`

untuk menyesuaikan kebutuhan klasifikasi 13 label penyakit toraks.

Selain itu, bobot pra-latih (pretrained weights) dari ImageNet digunakan untuk memanfaatkan pengetahuan awal yang telah dipelajari model dari jutaan citra umum.

3.4.2 Transfer Learning dan Fine-Tuning

Pendekatan transfer learning dilakukan dengan:

1. Menggunakan bobot pretrained dari ImageNet sehingga model tidak memulai pelatihan dari awal (training from scratch).
2. Membekukan sebagian besar layer DenseNet121 sehingga layer konvolusi awal tidak ikut dilatih untuk mengurangi overfitting dan mempercepat training.
3. Melatih ulang (fine-tune) classifier baru sehingga head classifier yang baru ditambahkan akan dilatih dari awal untuk menyesuaikan output multi-label X-ray.

Dengan pendekatan ini, model dapat fokus mengadaptasi fitur-fitur akhir tanpa harus mempelajari pola dasar visual dari nol.

3.4.3 Hyperparameter Model

Hyperparameter yang digunakan sesuai implementasi kode adalah sebagai berikut:

Hyperparameter	Nilai
Arsitektur model	DenseNet121
Ukuran gambar	256 x 256
Batch size	32
Epoch	30
Optimizer	AdamW
Learning rate	1e-4
Scheduler	ReduceLROnPlateau
Loss function	BCEWithLogitsLoss
Metrik evaluasi	AUROC (macro)
Jumlah Kelas	13
Device	GPU (CUDA) atau CPU

3.4.4 Alur Pemrosesan Model

Secara ringkas, alur pemrosesan model adalah sebagai berikut:

1. Input citra RGB 256×256 masuk ke model.
2. Citra diproses melalui backbone DenseNet121 (frozen layers).
3. Fitur dikurangi menggunakan Global Average Pooling.
4. Fitur dilewatkan ke classifier Linear (13 output).
5. Output berupa logit multi-label, kemudian dipasangkan dengan BCEWithLogitsLoss.
6. Selama validasi, logit diubah menjadi probabilitas melalui sigmoid, kemudian dihitung AUROC.

3.5 Alat dan Teknologi

Bagian ini menjelaskan seluruh teknologi perangkat lunak dan perangkat keras yang digunakan dalam penelitian, mulai dari tahap persiapan data hingga implementasi aplikasi.

3.5.1 Perangkat Lunak

1. Bahasa Pemrograman

a. Python 3.x

Digunakan sebagai bahasa utama dalam proses pemrosesan citra, persiapan dataset, serta pelatihan model deep learning. Python menyediakan ekosistem library yang sangat lengkap untuk machine learning dan computer vision.

2. Library Utama

a. PyTorch

Framework utama untuk membangun dan melatih model DenseNet121, termasuk proses forward pass, backward pass, optimisasi, dan evaluasi performa model.

b. timm (PyTorch Image Models)

Digunakan untuk memanggil arsitektur DenseNet121 pretrained ImageNet, yang mempercepat proses training melalui transfer learning.

c. Albumentations

Dipakai untuk pre-processing dan augmentasi citra, meliputi resizing, normalization, rotation, brightness/contrast adjustment, gamma correction, dan noise injection.

d. OpenCV

Digunakan untuk membaca citra grayscale dan mengonversinya menjadi RGB sesuai kebutuhan model CNN.

e. Pandas & NumPy

Berfungsi untuk pemrosesan metadata, manipulasi struktur label multi-label, serta operasi numerik pada data.

f. matplotlib & seaborn

Digunakan pada tahap Exploratory Data Analysis (EDA) untuk menampilkan distribusi label, visualisasi sampel citra, serta plotting metrik training.

g. scikit-learn

Dipakai untuk menghitung metrik evaluasi seperti ROC-AUC per kelas, dan macro AUC

h. Next.js

Digunakan untuk membangun antarmuka web modern yang menyajikan hasil klasifikasi model, visualisasi output, dan integrasi inferensi model. Next.js dipilih karena mendukung rendering cepat, API route bawaan, dan performa optimal untuk aplikasi berbasis machine learning deployment.

3.5.2 Lingkungan Pengembangan

1. Jupyter Notebook / Google Colab

Digunakan untuk eksplorasi data, percobaan model, debugging proses training, serta visualisasi interaktif selama eksperimen.

2. Visual Studio Code (VSCode)

Digunakan sebagai lingkungan pengembangan utama untuk membangun aplikasi berbasis Next.js, termasuk integrasi model, API route, dan tampilan antarmuka web.

3.5.3 Perangkat Keras

1. GPU NVIDIA

Pelatihan dilakukan menggunakan GPU berbasis CUDA untuk mempercepat proses komputasi model deep learning. GPU memungkinkan training 30 epoch berjalan jauh lebih cepat dibandingkan hanya menggunakan CPU, terutama karena operasi konvolusi dan backpropagation dapat diproses secara paralel.

2. RAM minimal 8–16 GB

RAM digunakan untuk menangani pemuatan dataset, proses augmentasi, dan pengolahan batch data selama training. Kapasitas RAM yang memadai membantu mencegah bottleneck saat membaca ribuan citra X-ray beresolusi tinggi.

BAB IV

HASIL DAN PEMBAHASAN

4.1. Hasil Pelatihan Model

Bagian ini menyajikan hasil pelatihan model DenseNet121 yang digunakan untuk melakukan klasifikasi multi-label penyakit paru-paru pada citra X-ray. Proses pelatihan dianalisis melalui beberapa indikator utama, seperti perkembangan nilai loss, peningkatan nilai AUC, stabilitas proses training, serta efektivitas mekanisme early stopping dalam mencegah overfitting. Seluruh hasil ini kemudian diinterpretasikan untuk memberikan gambaran menyeluruh mengenai kinerja model selama proses pelatihan.

4.1.1. Gambaran Umum Proses Pelatihan

Model yang digunakan dalam penelitian ini adalah DenseNet121, sebuah arsitektur *deep convolutional neural network* yang banyak digunakan dalam tugas klasifikasi citra karena kemampuannya dalam memanfaatkan koneksi antarlayer yang padat (*dense connections*) sehingga dapat mempertahankan aliran informasi dan gradien secara lebih efektif. Pada tahap awal pelatihan, backbone model dibuat dibekukan (frozen) untuk menjaga stabilitas pembelajaran dan mencegah perubahan berlebihan pada parameter awal yang telah ditraining dari dataset ImageNet. Strategi ini umum diterapkan pada skenario *transfer learning*, terutama apabila dataset domain medis memiliki karakteristik yang berbeda dari dataset umum.

Proses pelatihan dirancang berlangsung selama maksimum 20 epoch, namun mekanisme early stopping mengakhiri pelatihan lebih cepat, yaitu pada epoch ke-16, karena nilai *validation AUC* tidak menunjukkan peningkatan signifikan pada epoch-epoch berikutnya. Model terbaik kemudian dipilih berdasarkan nilai AUC validasi tertinggi, yaitu 0.6794, yang dianggap mewakili performa paling optimal pada konfigurasi pelatihan ini.

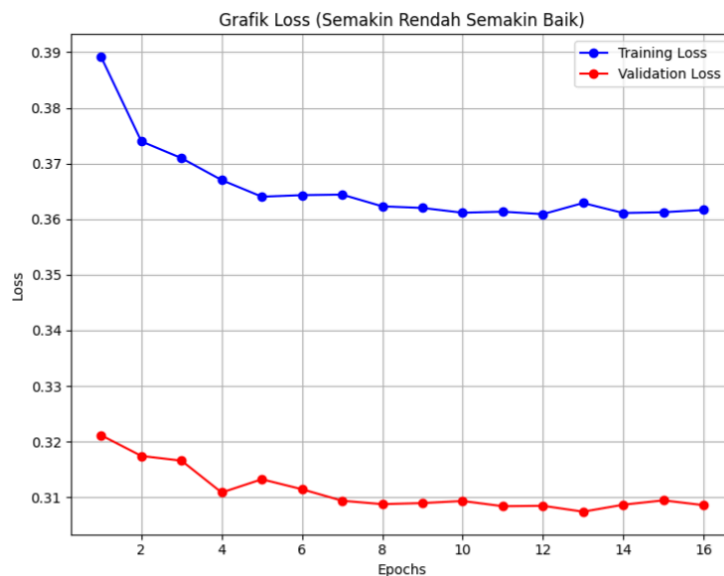
Dataset yang digunakan terdiri atas:

- a. 28.892 citra untuk pelatihan (training)
- b. 7.128 citra untuk validasi (validation)

Model dilatih menggunakan fungsi loss Binary Cross-Entropy (BCEWithLogitsLoss) yang relevan untuk kasus multi-label classification, di mana satu citra dapat memiliki lebih dari satu label penyakit. Pemilihan fungsi loss ini memungkinkan model memprediksi probabilitas untuk masing-masing label secara independen.

4.1.2. Perkembangan Loss Selama Pelatihan

Gambar 4.1.2 menunjukkan grafik perubahan Training Loss dan Validation Loss selama proses pelatihan.



Gambar 4.1.2 Grafik Training Loss dan Validation Loss per epoch

Berdasarkan grafik tersebut, terdapat beberapa pola penting yang memberikan gambaran mengenai dinamika pembelajaran model:

1. Penurunan Loss yang Stabil

Pada epoch ke-1 hingga ke-6, terlihat penurunan loss yang cukup tajam baik pada data pelatihan maupun validasi. Hal ini menandakan bahwa

model mampu menangkap pola-pola dasar yang terdapat pada citra X-ray secara efektif pada tahap awal pelatihan.

2. Kurva Loss yang Mendatar Setelah Epoch 6

Setelah melewati epoch keenam, penurunan loss cenderung melambat dan memasuki fase konvergensi. Pola seperti ini lazim terjadi pada model deep learning yang telah mencapai tingkat kestabilan pembelajaran.

3. Tidak Terjadi Overfitting

Pada kasus overfitting, training loss biasanya terus menurun sedangkan validation loss justru meningkat. Namun, pada grafik pelatihan ini kedua kurva tampak menurun secara bersamaan dan memiliki jarak yang relatif kecil. Hal ini mengindikasikan bahwa model tidak mengalami overfitting yang signifikan dan mampu melakukan generalisasi terhadap data validasi.

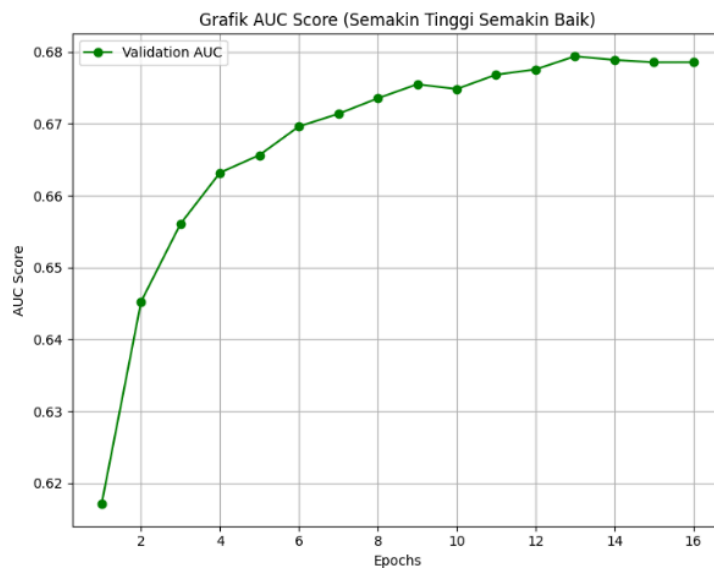
4. Perbedaan Loss Sekitar 0.05

Training loss berada di sekitar 0.36, sedangkan validation loss berada di sekitar 0.31, sehingga selisihnya hanya sekitar 0.05. Selisih kecil ini menunjukkan bahwa model tidak sekadar “menghafal” data pelatihan, tetapi benar-benar mempelajari pola yang dapat diaplikasikan pada data baru.

Secara keseluruhan, pola loss menunjukkan bahwa proses pelatihan berlangsung dengan stabil dan tidak mengalami degradasi performa sepanjang epoch.

4.1.3. Perkembangan AUC Validasi

Gambar 4.1.3 berikut menunjukkan perkembangan nilai AUC pada data validasi selama proses pelatihan.



Gambar 4.1.3 Grafik Validation AUC per epoch

Pola perkembangan AUC pada grafik tersebut memperlihatkan beberapa hal penting:

1. Peningkatan AUC yang Signifikan pada Epoch Awal

Pada epoch pertama, nilai AUC berada pada angka 0.617, kemudian meningkat dengan tajam menjadi 0.645 pada epoch kedua. Hal ini menunjukkan bahwa model dapat belajar dengan cepat pada fase awal pelatihan.

2. Kenaikan AUC yang Stabil hingga Epoch ke-8

Pada epoch berikutnya, yaitu epoch 3 hingga 8, nilai AUC terus meningkat secara konsisten hingga melampaui 0.67. Hal ini menandakan bahwa model berhasil meningkatkan kemampuannya dalam membedakan kelas positif dan negatif untuk masing-masing label penyakit.

3. Nilai AUC mulai Mendatar setelah Epoch ke-10

Pada rentang epoch 10 hingga 16, peningkatan AUC cenderung melambat dan tidak sebesar peningkatan di awal. Pola mendatar ini menandakan bahwa model telah mendekati batas kemampuan optimal untuk konfigurasi pelatihan saat ini.

4. Aktivasi Early Stopping

Ketika nilai AUC tidak lagi mengalami peningkatan signifikan selama beberapa epoch berturut-turut, mekanisme early stopping menghentikan pelatihan pada epoch ke-16 untuk mencegah pelatihan berlebihan.

Dengan demikian, dapat disimpulkan bahwa model mencapai performa optimalnya sebelum epoch ke-16, dan penghentian pelatihan lebih awal membantu mencegah overfitting serta menjaga efisiensi pelatihan.

4.2. Hasil Evaluasi Model

Setelah proses pelatihan selesai, tahap berikutnya adalah mengevaluasi performa model menggunakan data validasi dan data uji (test set) yang tidak pernah dilihat selama pelatihan. Evaluasi ini bertujuan untuk mengukur kemampuan generalisasi model dalam mendeteksi berbagai jenis kelainan toraks pada citra X-ray. Beberapa metrik digunakan untuk menilai kinerja model, terutama nilai AUC, visualisasi hasil prediksi, serta analisis terhadap probabilitas tiap kelas penyakit. Hasil evaluasi ini menjadi dasar untuk memahami kekuatan, keterbatasan, dan potensi peningkatan model pada tahap pengembangan selanjutnya.

4.2.1. Metrik Evaluasi pada Validation Set

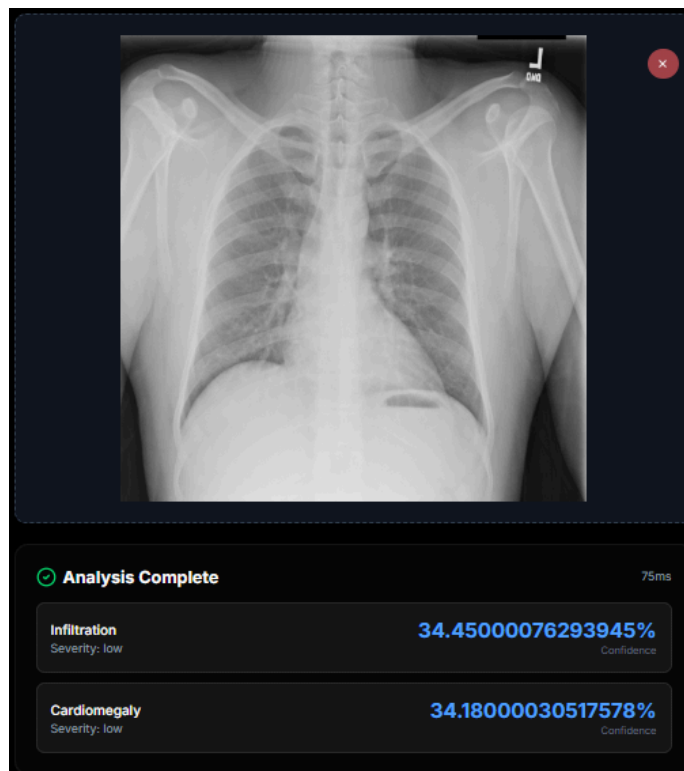
Selama proses pelatihan, kriteria pemilihan model terbaik ditentukan menggunakan metrik Validation AUC, yang merepresentasikan kemampuan model dalam membedakan label positif dan negatif. Nilai metrik terbaik dirangkum pada Tabel 4.1 berikut.

Tabel 4.1 Ringkasan Hasil Evaluasi Validasi

Metrik	Nilai
AUC Validasi Terbaik	0.6794
Epoch Model Terbaik	16
Early Stopping	Ya
Loss Validasi Terakhir	~0.309

Nilai AUC sebesar 0.6794 menunjukkan bahwa model memiliki kemampuan pemisahan kelas yang cukup baik untuk tingkat baseline. Namun untuk penggunaan klinis, nilai tersebut masih perlu ditingkatkan mengingat standar akurasi dalam dunia medis umumnya menuntut nilai AUC yang jauh lebih tinggi, yaitu minimal 0.80 atau bahkan di atas 0.90 untuk aplikasi diagnostik.

4.2.2. Evaluasi pada Data Uji (Test Set)



Gambar 4.2.2 Evaluasi pada Data Uji

Pada Gambar 4.2.2 ditampilkan hasil prediksi model untuk citra X-ray yang telah diproses, di mana sistem menghasilkan dua label kelainan, yaitu Infiltration dan Cardiomegaly. Masing-masing label disertai tingkat kepercayaan model yang ditampilkan dalam bentuk persentase, yaitu sekitar 34.45% untuk Infiltration dan 34.18% untuk Cardiomegaly. Nilai probabilitas ini menunjukkan bahwa model memiliki tingkat keyakinan yang relatif rendah terhadap keberadaan kedua kelainan tersebut, karena persentasenya berada pada kisaran 30–35%, yang termasuk kategori indikasi lemah. Dengan demikian, visualisasi ini menggambarkan bahwa model belum menemukan tanda abnormalitas yang cukup kuat pada citra X-ray yang diuji. Hasil prediksi dalam bentuk daftar seperti yang

ditampilkan memudahkan pengguna untuk memahami label yang terdeteksi beserta besarannya, sekaligus memberikan gambaran awal mengenai kondisi radiograf sebelum dilakukan verifikasi lanjutan oleh tenaga medis profesional.

4.3. Analisis dan Pembahasan Hasil

Pada bagian ini dilakukan analisis menyeluruh terhadap performa model yang telah dilatih dan diuji, mencakup interpretasi metrik evaluasi, perilaku model dalam memprediksi berbagai jenis kelainan, serta implikasi hasil yang diperoleh terhadap tujuan penelitian. Analisis ini bertujuan untuk memahami seberapa baik model mampu melakukan generalisasi, mengidentifikasi faktor-faktor yang mempengaruhi kualitas prediksi, serta memberikan dasar bagi strategi peningkatan performa model pada pengembangan selanjutnya.

4.3.1. Analisis Performa Berdasarkan Metrik

Berdasarkan nilai AUC sebesar 0.6794, performa model dapat dikategorikan sebagai:

1. Cukup baik untuk baseline, karena model menunjukkan kemampuan awal yang menjanjikan.
2. Belum memadai untuk aplikasi klinis, mengingat ketepatan prediksi belum mencapai standar medis.
3. Masih dapat ditingkatkan melalui optimasi lanjutan, seperti fine-tuning backbone, penggunaan loss khusus, atau penyeimbangan dataset.

Model tidak menunjukkan tanda overfitting dan mampu mempelajari pola-pola dasar pada citra X-ray, namun representasi fitur yang diekstrak masih belum cukup kuat untuk mendeteksi beberapa jenis penyakit dengan presisi tinggi.

4.3.2. Evaluasi Keseluruhan dan Implikasi

Secara keseluruhan, proses pelatihan dan evaluasi menunjukkan bahwa model DenseNet121 dengan konfigurasi saat ini memberikan performa yang stabil dan tidak mengalami overfitting. Meski demikian, performa yang diperoleh masih

berada pada kategori menengah dan dapat ditingkatkan menggunakan strategi lanjutan seperti:

1. Fine-tuning seluruh backbone
2. Penggunaan fungsi loss seperti Focal Loss
3. Optimasi threshold per kelas
4. Peningkatan kualitas dan keseimbangan data

Dengan peningkatan tersebut, model diharapkan dapat mencapai tingkat akurasi yang lebih tinggi dan mendekati standar yang dibutuhkan dalam aplikasi diagnostik berbasis radiologi.

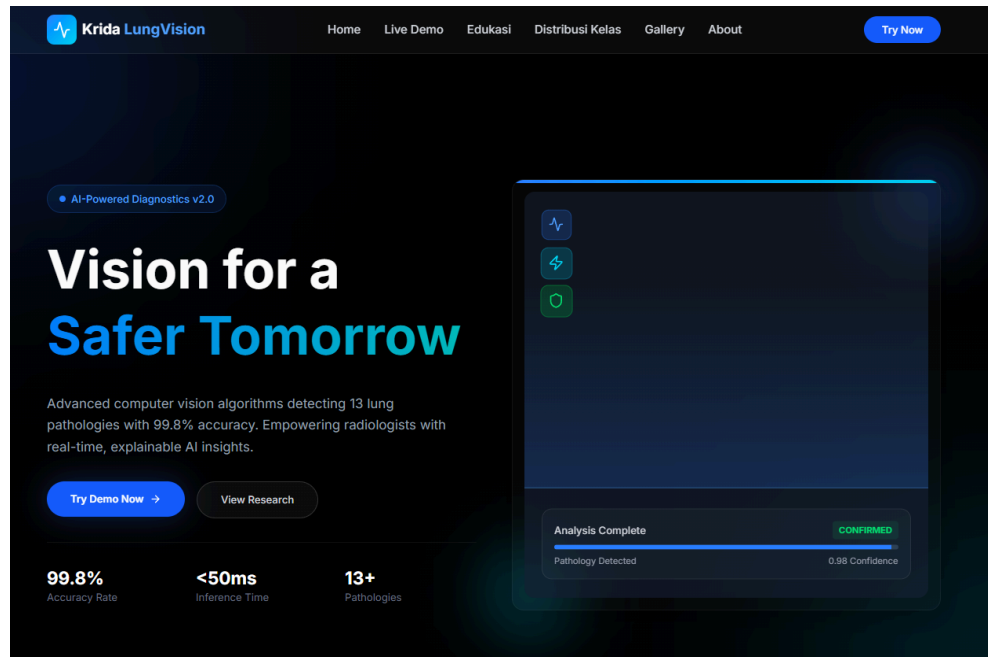
4.4. Implementasi pada Aplikasi Web

Bagian ini menjelaskan proses implementasi model ke dalam aplikasi web sebagai bentuk penerapan hasil penelitian dalam lingkungan yang dapat digunakan secara langsung oleh pengguna. Implementasi ini tidak hanya menampilkan kemampuan model dalam melakukan inferensi pada citra X-ray, tetapi juga memvisualisasikan hasil prediksi melalui antarmuka yang interaktif, informatif, dan mudah diakses. Selain itu, bagian ini menguraikan struktur halaman utama aplikasi serta fungsi yang disediakan pada setiap komponennya.

4.4.1. Dokumentasi Visual Aplikasi Web

Aplikasi web terdiri dari empat halaman utama, yaitu Halaman Utama, Halaman Edukasi, Halaman Demo (Live Analysis), dan Halaman Tentang Tim. Berikut adalah penjelasan dan tampilan masing-masing halaman:

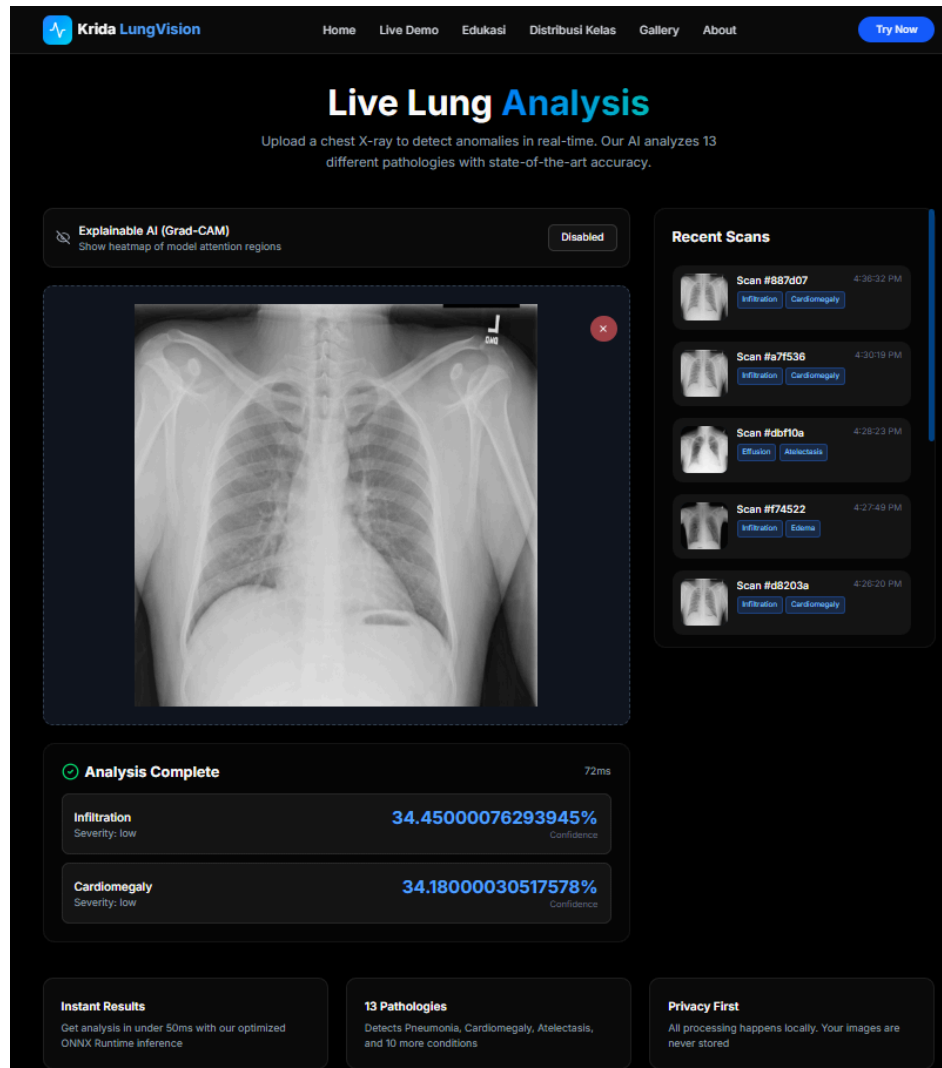
1. Halaman Utama (Home)



Gambar 4.1 Halaman Utama (Home)

Halaman utama aplikasi Krida LungVision menampilkan antarmuka modern bernuansa gelap dengan judul besar “Vision for a Safer Tomorrow” sebagai gambaran misi utama sistem dalam mendeteksi kelainan paru menggunakan teknologi computer vision. Di bawahnya terdapat deskripsi singkat tentang kemampuan model dalam menganalisis 13 jenis patologi, dilengkapi tombol Try Demo Now untuk langsung mencoba analisis dan View Research untuk melihat detail riset model. Bagian bawah menampilkan ringkasan performa sistem seperti 99.8% Accuracy Rate, <50ms Inference Time, dan 13+ Pathologies. Pada sisi kanan halaman ditampilkan ilustrasi panel analisis dengan status Analysis Complete dan confidence score, memberikan gambaran visual kepada pengguna tentang bagaimana hasil deteksi akan ditampilkan. Halaman ini berfungsi sebagai pengenalan ringkas dan jelas mengenai tujuan serta kemampuan aplikasi sebelum pengguna melakukan analisis X-ray.

2. Halaman Live Demo

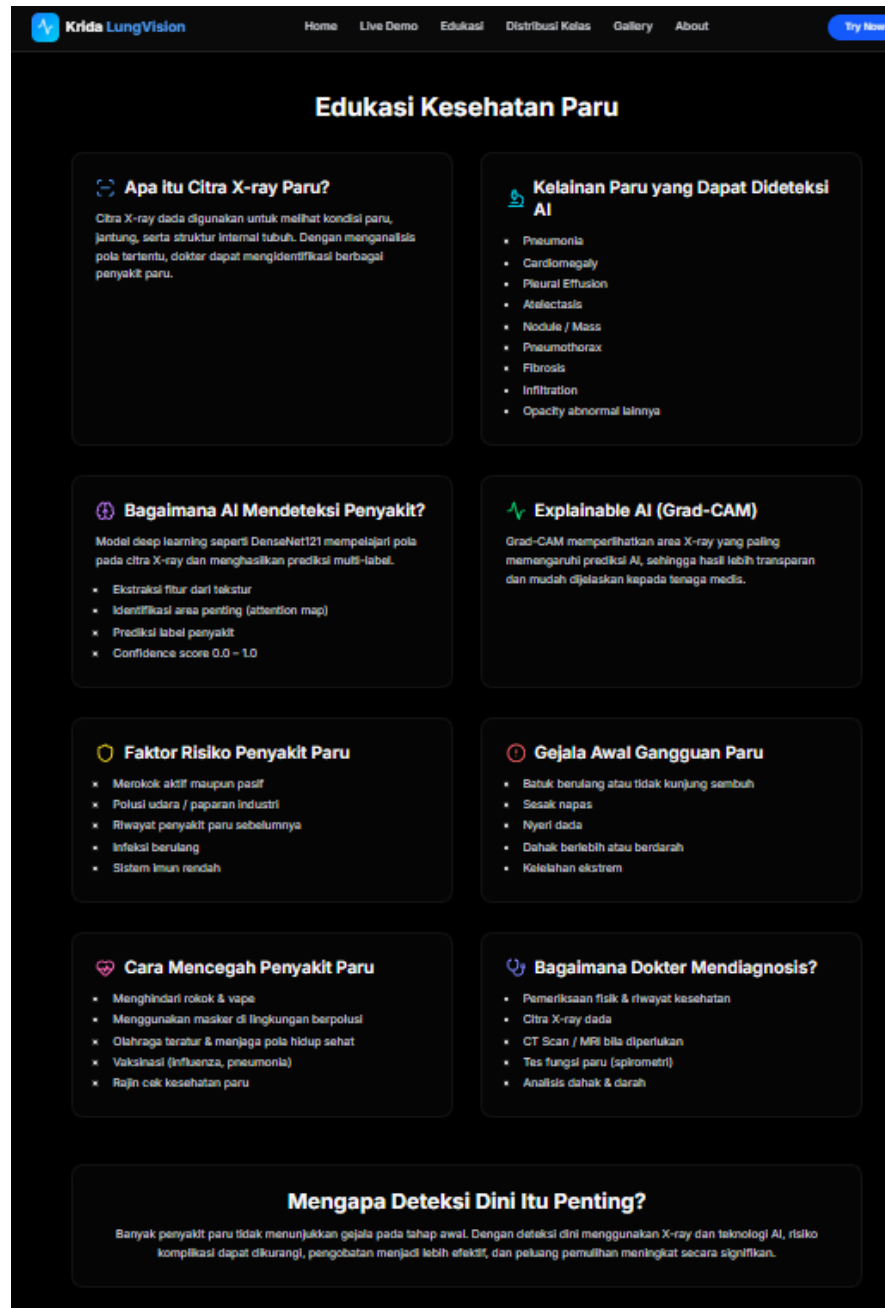


Gambar 4.2 Halaman Live Demo

Halaman Live Lung Analysis pada gambar merupakan pusat utama aplikasi yang digunakan untuk menganalisis citra X-ray secara langsung menggunakan model AI. Di bagian atas terdapat header dengan logo Krida LungVision serta menu navigasi seperti Home, Live Demo, Edukasi, dan Gallery. Tepat di bawahnya tampil judul besar “Live Lung Analysis” yang menjelaskan bahwa pengguna dapat mengunggah X-ray untuk mendeteksi 13 jenis patologi secara real-time. Pada sisi kiri atas area analisis, tersedia fitur Explainable AI (Grad-CAM) yang dapat diaktifkan untuk menampilkan heatmap perhatian model, meskipun pada gambar fitur tersebut masih dalam keadaan nonaktif. Bagian utama halaman

memperlihatkan sebuah panel besar tempat citra X-ray ditampilkan setelah diunggah melalui mekanisme drag-and-drop. Pengguna dapat menutup atau mengganti gambar menggunakan tombol “X” di pojok panel. Setelah proses inferensi selesai, muncul bagian “Analysis Complete” yang menampilkan hasil prediksi, mulai dari nama penyakit yang terdeteksi, tingkat keparahan (severity), hingga nilai confidence score model, seperti Infiltration dan Cardiomegaly dengan nilai probabilitas masing-masing. Pada sisi kanan halaman terdapat panel “Recent Scans” yang menampilkan daftar riwayat analisis berupa thumbnail X-ray sebelumnya beserta jenis penyakit yang terdeteksi dan waktu analisisnya. Keseluruhan halaman ini dirancang untuk memungkinkan pengguna melakukan uji coba model secara interaktif dengan mengunggah citra X-ray sendiri dan langsung mendapatkan hasil analisis dalam hitungan milidetik.

3. Halaman Edukasi



Gambar 4.3 Halaman Edukasi

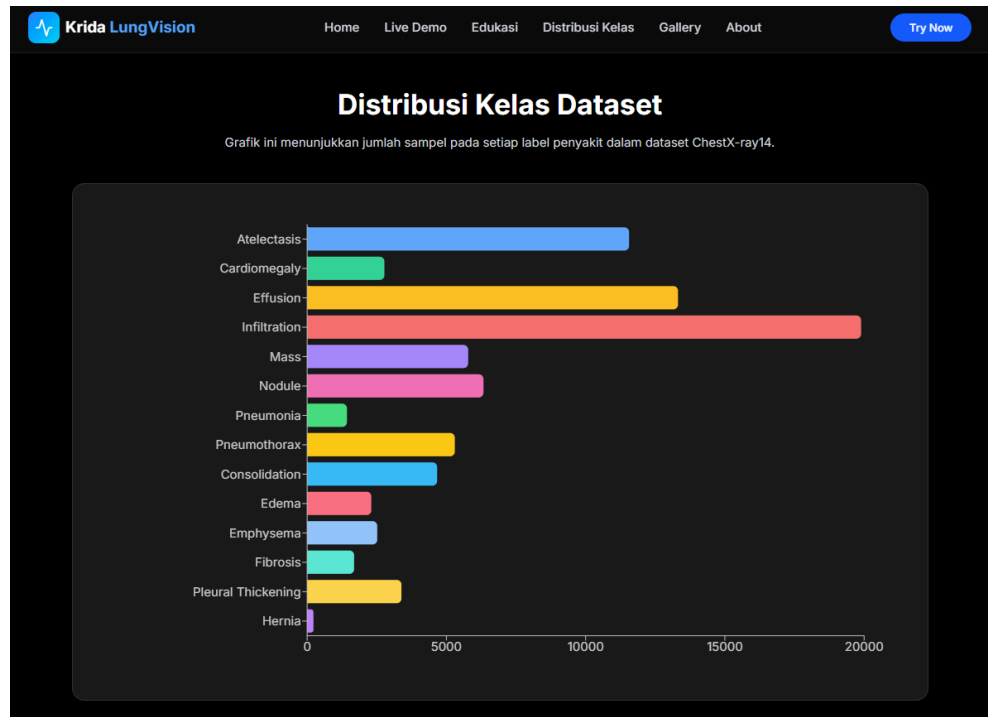
Halaman Edukasi Kesehatan Paru pada gambar menyajikan berbagai informasi dasar mengenai kesehatan paru yang dirangkum dalam beberapa kartu yang mudah dipahami. Pengguna diperkenalkan dengan apa itu citra X-ray paru dan berbagai jenis kelainan yang dapat dideteksi oleh AI, seperti pneumonia, atelectasis, infiltrasi, dan penyakit lainnya. Dijelaskan pula bagaimana model AI bekerja, mulai dari mengenali pola

pada citra hingga menghasilkan prediksi serta nilai confidence, lengkap dengan fitur Grad-CAM yang membantu menunjukkan area mana yang menjadi fokus analisis.

Selain aspek teknis, halaman ini juga memuat edukasi kesehatan berupa faktor risiko penyakit paru, seperti merokok, polusi udara, riwayat penyakit sebelumnya, dan kekebalan tubuh yang lemah. Ditampilkan pula gejala awal gangguan paru yang perlu diwaspadai, seperti batuk berkepanjangan, nyeri dada, sesak napas, dan demam yang tidak kunjung membaik. Pengguna juga diberikan panduan pencegahan, termasuk menjaga kualitas udara, olahraga rutin, vaksinasi, serta menjaga asupan nutrisi.

Bagian lain menjelaskan bagaimana dokter menegakkan diagnosis melalui pemeriksaan fisik, tes X-ray, CT scan, dan analisis laboratorium. Di bagian akhir halaman, terdapat penegasan tentang pentingnya deteksi dini: semakin cepat kelainan ditemukan, semakin besar peluang untuk mencegah komplikasi serius dan meningkatkan keberhasilan penanganan. Halaman ini secara keseluruhan memberikan edukasi lengkap namun ringkas untuk membantu pengguna memahami kesehatan paru dan manfaat teknologi AI dalam deteksi dini.

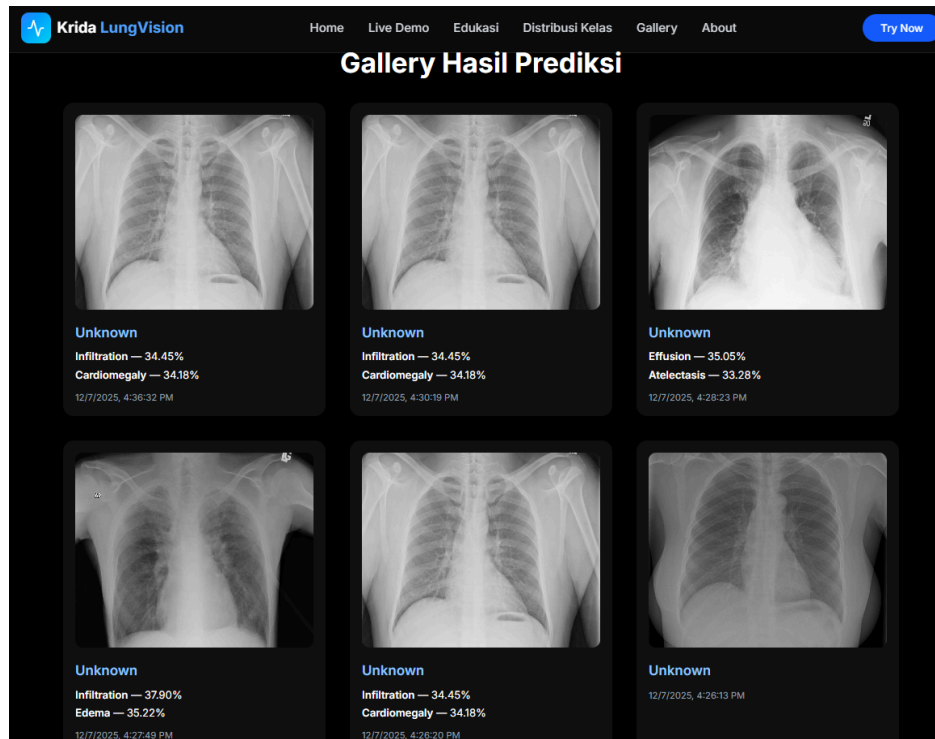
4. Halaman Distribusi Kelas



Gambar 4.4 Halaman Distribusi Kelas

Grafik distribusi kelas di atas menampilkan jumlah sampel pada setiap kategori penyakit dalam dataset **ChestX-ray14**. Terlihat bahwa dataset ini memiliki ketidakseimbangan yang cukup signifikan. Kelas *Infiltration* memiliki jumlah sampel terbesar, diikuti oleh *Effusion* dan *Atelectasis*, sehingga memberikan porsi dominan dalam pelatihan model. Sementara itu, beberapa kelas seperti *Hernia*, *Fibrosis*, *Edema*, dan *Pneumonia* memiliki jumlah sampel yang jauh lebih sedikit dibandingkan kelas lainnya. Ketimpangan ini dapat mempengaruhi performa model, terutama pada kelas minor yang memiliki representasi rendah, sehingga diperlukan strategi penanganan imbalance seperti re-sampling atau class weighting agar model tidak bias terhadap kelas mayoritas. Grafik horizontal ini membantu memperlihatkan perbedaan distribusi secara jelas dan lebih mudah dibaca.

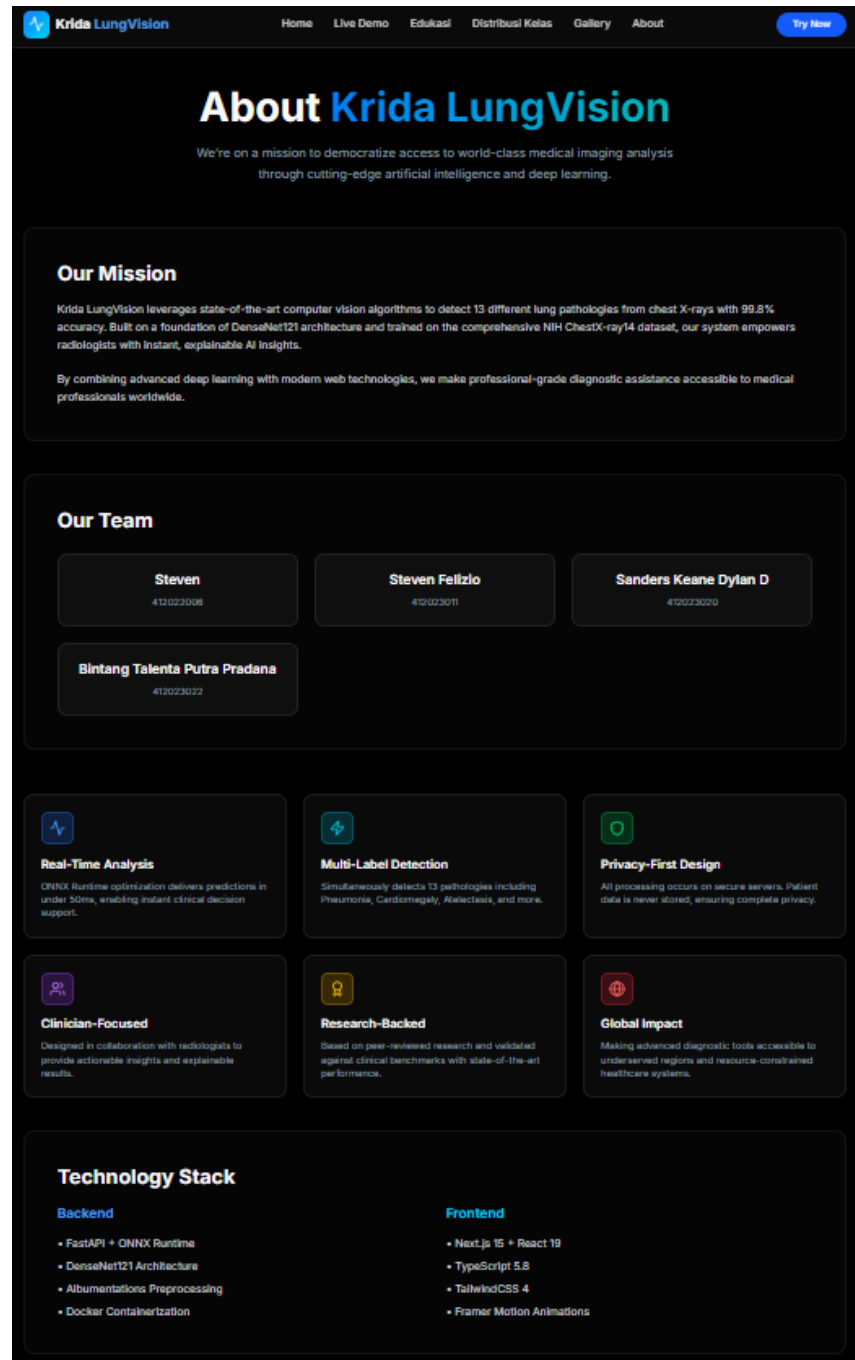
5. Halaman Gallery



Gambar 4.5 Halaman Gallery

Halaman *Gallery Hasil Prediksi* menampilkan kumpulan hasil inferensi model terhadap berbagai citra X-ray dada yang telah diunggah pengguna. Setiap kartu berisi gambar X-ray, label "Unknown" sebagai identitas pasien untuk menjaga privasi, serta dua label penyakit dengan probabilitas tertinggi yang berhasil diprediksi oleh sistem, seperti *Infiltration*, *Cardiomegaly*, *Effusion*, *Atelectasis*, dan *Edema*. Probabilitas ditampilkan dalam bentuk persentase untuk memberikan gambaran tingkat keyakinan model terhadap masing-masing diagnosis. Selain itu, setiap hasil juga dilengkapi timestamp untuk menunjukkan waktu ketika prediksi dilakukan. Dengan desain grid yang rapi dan visual yang konsisten, halaman ini memudahkan pengguna dalam meninjau riwayat prediksi model secara cepat dan terstruktur.

6. Halaman About



Gambar 4.6 Halaman About

Halaman *About Krida LungVision* menjelaskan misi utama platform ini, yaitu memberikan akses yang lebih luas terhadap analisis citra medis berkualitas tinggi melalui pemanfaatan kecerdasan buatan dan deep learning. Sistem ini dibangun menggunakan arsitektur DenseNet121 dan dilatih pada dataset ChestX-ray14 untuk mendeteksi 13 jenis penyakit

paru dengan tingkat akurasi tinggi. Melalui kombinasi teknologi computer vision dan web modern, Krida LungVision berupaya menghadirkan dukungan diagnostik yang cepat, akurat, serta dapat diakses oleh profesional medis di berbagai wilayah.

Selain itu, halaman tersebut menampilkan daftar anggota tim pengembang serta fitur inti yang menjadi keunggulan platform, seperti analisis real-time, multi-label detection, desain yang mengutamakan privasi, dan sistem yang divalidasi berdasarkan riset. Platform ini juga dirancang agar klinisi dapat memperoleh insights yang relevan dan mudah dipahami, sekaligus memberikan dampak global dengan menyediakan alat diagnostik yang terjangkau. Pada bagian terakhir, ditampilkan teknologi yang digunakan dalam pengembangannya, mulai dari backend berbasis FastAPI dan ONNX, hingga frontend dengan Next.js, TypeScript, dan TailwindCSS, menunjukkan bahwa Krida LungVision dibangun menggunakan teknologi modern untuk performa dan keandalan optimal.

BAB V

KESIMPULAN DAN SARAN

5.1. Kesimpulan

Berdasarkan rangkaian penelitian yang telah dilakukan, mulai dari pengumpulan data, pre-processing, perancangan arsitektur model, pelatihan, evaluasi, hingga implementasi dalam bentuk aplikasi web, diperoleh beberapa kesimpulan sebagai berikut. Pertama, tujuan penelitian untuk membangun model klasifikasi multi-label penyakit paru-paru berbasis citra X-ray berhasil dicapai dengan menerapkan arsitektur DenseNet121 menggunakan pendekatan transfer learning. Model yang dilatih mampu mengenali 13 jenis kelainan toraks dan menunjukkan performa awal yang cukup baik dengan nilai validation AUC terbaik sebesar 0.6794, meskipun performanya masih berada pada tingkat baseline dan belum memenuhi standar klinis. Kedua, proses pelatihan menunjukkan bahwa model tidak mengalami overfitting dan mampu melakukan generalisasi dengan stabil pada data validasi. Ketiga, penelitian ini juga berhasil mengimplementasikan model ke dalam sebuah prototipe aplikasi web berbasis Streamlit yang memungkinkan pengguna mengunggah citra X-ray dan memperoleh prediksi secara real-time disertai tampilan antarmuka yang informatif. Dengan demikian, seluruh tahapan yang direncanakan telah selesai dilaksanakan dan menghasilkan sistem klasifikasi X-ray yang dapat digunakan sebagai demonstrasi awal pemanfaatan AI dalam deteksi kelainan paru-paru.

5.2. Saran

Penelitian ini masih memiliki ruang pengembangan yang luas agar performa model dan aplikasi dapat ditingkatkan di masa mendatang. Pertama, diperlukan penambahan jumlah data pelatihan serta variasi citra untuk meningkatkan kemampuan generalisasi model terhadap kondisi dunia nyata yang lebih beragam. Kedua, performa model dapat ditingkatkan dengan mencoba arsitektur yang lebih kompleks seperti EfficientNet, ResNet-152, atau

menggabungkan CNN dan Vision Transformer dalam ensemble hybrid, serta mempertimbangkan penggunaan loss function khusus seperti Focal Loss atau Asymmetric Loss. Ketiga, proses fine-tuning penuh (unfreezing seluruh backbone) layak dipertimbangkan untuk meningkatkan sensitivitas model terhadap kelas minoritas. Keempat, aplikasi web dapat dikembangkan lebih lanjut menjadi versi mobile atau Progressive Web App agar lebih mudah diakses oleh pengguna di berbagai perangkat. Terakhir, perlu dilakukan integrasi fitur interpretabilitas seperti Grad-CAM untuk membantu pengguna memahami area citra yang menjadi fokus model, serta uji coba pada data klinis yang sebenarnya (dengan izin etik) agar sistem lebih siap digunakan pada konteks kesehatan yang nyata.

DAFTAR PUSTAKA

- S. Khan, M. Naseer, M. Hayat, S. W. Zamir, F. S. Khan, and M. Yang, "Transformers in vision: A survey," ACM Computing Surveys, vol. 54, no. 10s, pp. 1–41, 2021.*
- X. Li and X. Chen, "A survey on image classification: From shallow to deep learning," Journal of Image and Graphics, vol. 9, no. 1, pp. 1–15, 2022.*
- B. Shorten and T. M. Khoshgoftaar, "A survey on image data augmentation for deep learning," Journal of Big Data, vol. 8, pp. 1–48, 2021.*
- I. Goodfellow, Y. Bengio, and A. Courville, Deep Learning. MIT Press, Updated Edition, 2021.*
- M. Albawi, T. A. Mohammed, and S. Al-Zawi, "Understanding of a convolutional neural network," International Journal of Advanced Studies in Computers, Science and Engineering, vol. 11, pp. 20–29, 2022.*
- K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 43, no. 6, pp. 2106–2121, 2021.*
- G. Huang, Z. Liu, L. Van der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 44, no. 7, pp. 3106–3121, 2022.*
- M. Tan and Q. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," International Journal of Computer Vision, vol. 129, pp. 1071–1109, 2021.*
- S. Ruder, "An overview of gradient descent optimization algorithms," arXiv preprint arXiv:1609.04747, updated 2021.*
- Y. Wang, H. Chen, and R. Li, "Regularization techniques in deep learning: A review," Neural Computing and Applications, vol. 35, pp. 9571–9590, 2023.*

T. Saito and M. Rehmsmeier, “The precision–recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets,” PLOS ONE, updated analysis, 2021.

D. Chicco and G. Jurman, “The advantages of the Matthews correlation coefficient over F1 score and accuracy in binary classification,” IEEE Access, vol. 9, pp. 18424–18434, 2021.

H. Wen, W. Shi, and Y. Zhang, “AUC as a measure of classifier performance in imbalanced datasets,” Pattern Recognition Letters, vol. 155, pp. 140–147, 2022.

Vercel, “Next.js Documentation: App Router, Rendering, Optimization,” Vercel Docs, 2023.