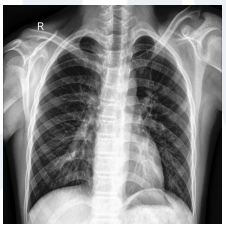
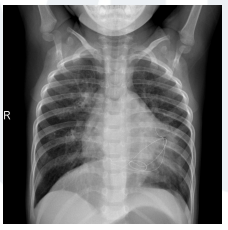
# BAB III
# METODOLOGI PENELITIAN

## 3.1 Gambaran Umum Objek Penelitian

Penelitian ini bertujuan untuk mengeksplorasi penggunaan CNN dalam melakukan analisis citra radiografi (X-ray) paru-paru guna mendeteksi penyakit paru-paru secara efisien dan akurat. Dataset yang digunakan dalam penelitian ini diperoleh dari repositori publik *Kaggle*, yang terdiri atas 8.022 citra X-ray bagian dada [67]. Seluruh citra tersebut dibagi menjadi dua bagian utama, yaitu data pelatihan dan data validasi, dengan masing-masing citra berasal dari pasien yang berbeda. Pada data pelatihan, setiap citra diberi label dengan format (penyakit)-(nomor ID citra) untuk mempermudah pengelompokan selama proses pelatihan model. Sebelum dilakukan analisis, seluruh citra X-ray terlebih dahulu melalui proses penyaringan (*screening*) kualitas, di mana citra dengan kualitas rendah atau yang tidak dapat dibaca dieliminasi agar tidak memengaruhi performa pelatihan model secara negatif. Dataset ini terdiri dari satu variabel dependen bernama "penyakit", yang memiliki empat kelas kategori, yaitu "COVID-19", "Normal", "Pneumonia", dan "Tuberkulosis".

| COVID-19 | Normal | Pneumonia | Tuberkulosis |
|---|---|---|---|
|  |  |  |  |

Gambar 3.1 Sampel Dataset Sinar-X bagian Paru-paru [67]

## 3.2 Metode Penelitian

Metode kuantitatif merupakan pendekatan penelitian yang memanfaatkan data dalam bentuk angka atau variabel kuantitatif untuk mengumpulkan, menganalisis, dan menyimpulkan informasi [68]. Metode ini mengacu pada penggunaan pendekatan statistik dan matematis untuk mengukur dan menggambarkan fenomena atau hubungan antara variabel-variabel dalam suatu penelitian [69]. Sementara itu, beberapa *framework* yang diusulkan untuk digunakan dalam pengembangan model yang akan dibuat:

Tabel 3.1 Tabel Perbandingan Framework

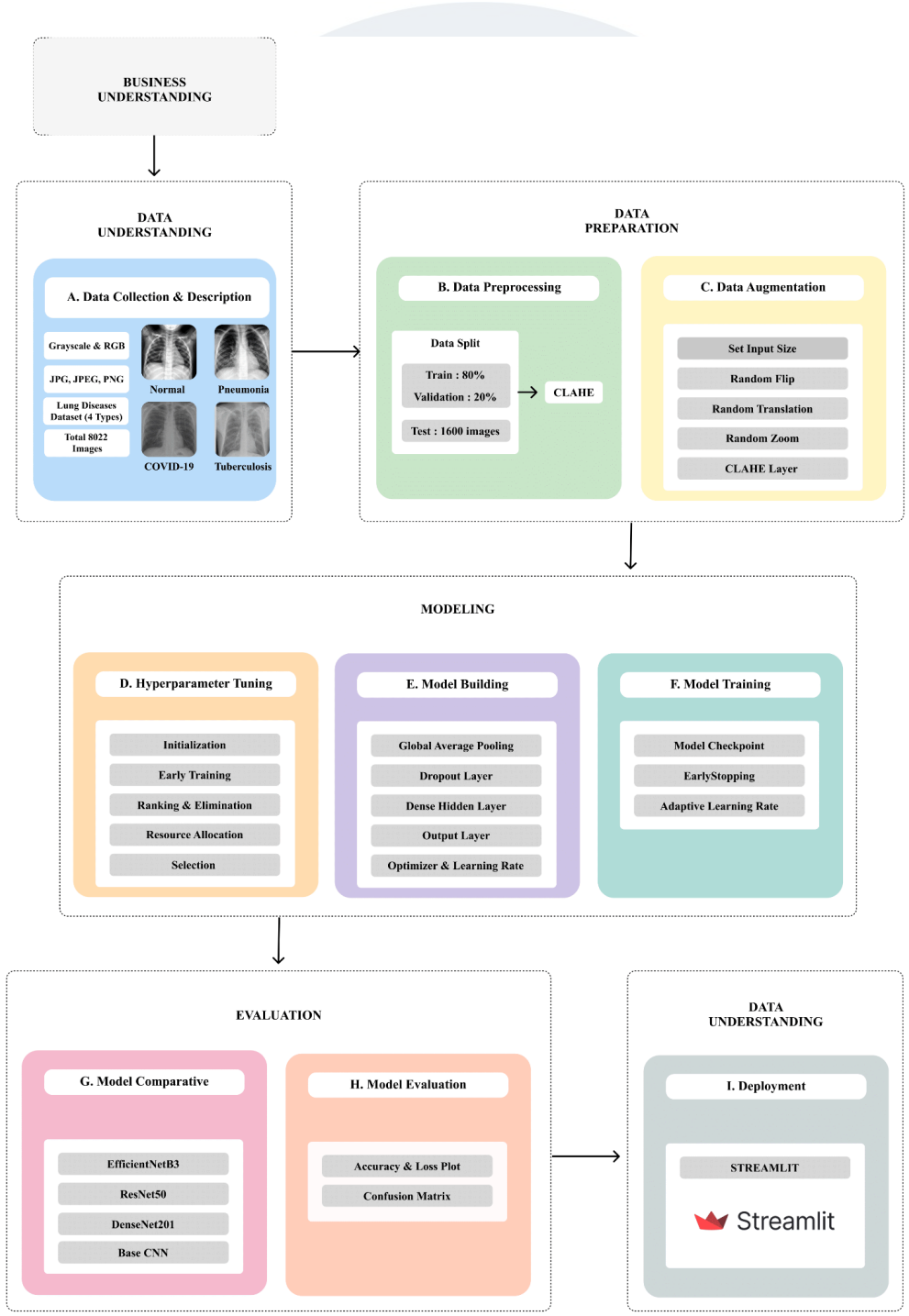
| **Model Data Mining** | **SEMMA** | **CRISP-DM** | **KDD** |
|---|---|---|---|
| **Jumlah Fase** | 5 | 6 | 6 |
| **Fase** | *1. Sample*<br>*2. Explore*<br>*3. Modify*<br>*4. Model*<br>*5. Assesment* | *1. Business Understanding*<br>*2. Data Understanding*<br>*3. Data Preparation*<br>*4. Modeling*<br>*5. Evaluation*<br>*6. Deployment* | *1. Pre-KDD*<br>*2. Selection*<br>*3. Transformation*<br>*4. Data Mining*<br>*5. Interpretation*<br>*6. Post-KDD* |
| **Kelebihan** | 1.Sederhana dan mudah dipahami.<br>2.Fokus pada pemahaman data dan iteratif dalam pendekatan pemodelan.<br>3.Cocok untuk proyek kecil hingga menengah. | 1.Struktur yang terorganisir dengan tahapan-tahapan yang jelas.<br>2.Memiliki fase evaluasi yang kuat.<br>3.Fleksibel dan dapat disesuaikan dengan kebutuhan proyek tertentu. | 1.Mencakup seluruh proses penemuan pengetahuan dari data.<br>2.Mengintegrasikan konsep dari berbagai bidang, termasuk statistika, *machine learning*, dan database. |
| **Kekurangan** | 1.Kurangnya panduan formal, lebih bersifat umum.<br>2.Kurang efektif | 1.Terlalu rinci untuk proyek kecil.<br>2.Proses yang panjang dan kompleks, | 1.Terkadang dianggap sebagai pendekatan yang terlalu |

| Model Data Mining | SEMMA | CRISP-DM | KDD |
|---|---|---|---|
| | untuk proyek besar dan kompleks.<br>3.Tidak memberikan petunjuk khusus tentang manajemen proyek. | akan sulit diterapkan sepenuhnya. | teoritis.<br>2.Fase-fase tertentu memerlukan keterampilan dan pengetahuan yang mendalam. |

Dalam penelitian ini, *framework* yang digunakan adalah CRISP-DM. CRISP-DM adalah model proses yang menggambarkan siklus hidup dalam proses *data mining*. Dibandingkan dengan model lain seperti SEMMA dan KDD *(Knowledge Discovery in Databases)*, CRISP-DM lebih cocok digunakan dalam proyek klasifikasi citra X-ray seperti ini karena Keunggulan utama CRISP-DM, yang membedakannya dari SEMMA dan KDD, adalah adanya langkah *deployment* yang terintegrasi secara eksplisit dalam *framework*. Pada SEMMA (*Sample, Explore, Modify, Model, Assess*), proses hanya berakhir pada tahap evaluasi tanpa panduan lebih lanjut terkait penerapan hasil model ke dalam sistem nyata. Sementara pada KDD, proses *deployment* tidak dijelaskan secara spesifik dan cenderung terlalu teoritis untuk implementasi pada aplikasi nyata, khususnya dalam proyek-proyek berskala menengah hingga besar. Langkah *deployment* pada CRISP-DM sangat relevan dan selaras dengan tujuan utama penelitian ini, yaitu "Penerapan CLAHE dan Convolutional Neural Network dalam Deteksi Penyakit Paru-paru pada Citra Radiografi". Pada tahap deployment, hasil model yang telah dibangun dan dievaluasi akan diimplementasikan ke dalam workflow nyata sebagai sistem deteksi penyakit paru berbasis citra X-ray [70].

Dengan mempertimbangkan keunggulan CRISP-DM yang selaras dengan kebutuhan penelitian ini, tahapan penelitian disusun berdasarkan

alur kerja dalam CRISP-DM, yang meliputi langkah-langkah berikut:




Gambar 3.2 Alur Penelitian

### 3.2.1 Data Collection & Description

Langkah awal dalam proses pengembangan model adalah pengumpulan data. Data yang relevan dan representatif sangat penting untuk memastikan kualitas model yang dihasilkan.

1. Pengumpulan data dilakukan melalui sumber sekunder yaitu *Kaggle*, dataset berisi citra radiografi paru-paru.
2. Mengidentifikasi dataset yang sesuai dengan penelitian, termasuk citra paru-paru yang mencakup kategori penyakit yang ditargetkan (Pneumonia, COVID-19, Tuberkulosis, Normal).
3. Dataset terdiri dari 8.022 citra yang mencakup empat jenis penyakit paru.
4. Citra tersedia dalam format JPG, JPEG, dan PNG, dan terdiri dari dua mode warna yaitu *grayscale* dan RGB.

### 3.2.2 Data Preprocessing

Setelah data dikumpulkan, tahap selanjutnya adalah pra-pemrosesan untuk menyiapkan data agar siap digunakan dalam proses pelatihan. Langkah ini bertujuan untuk meningkatkan kualitas data dan menjaga konsistensi *input*.

1. Melakukan pembagian dataset menjadi data latih dan data validasi dengan rasio 80:20.
2. Menggunakan teknik pengurangan *noise* CLAHE untuk meningkatkan kualitas citra.

### 3.2.3 Data Augmentation

Untuk memperkaya variasi data dan mengurangi risiko *overfitting*, dilakukan proses augmentasi citra. Teknik augmentasi ini

mensimulasikan kondisi citra yang berbeda tanpa perlu menambah data baru.

1. Menentukan dimensi *input* citra sebagai masukan berwarna *RGB* dengan ukuran 224×224 piksel.
2. Melakukan pembalikan citra secara horizontal secara acak.
3. Menerapkan translasi acak hingga 20% secara horizontal dan vertikal.
4. Melakukan pembesaran atau pengecilan citra secara acak hingga 20% dan menambahkan *layer* CLAHE kedalam *data augmentation*.

### 3.2.4 Hyperparameter Tuning

Agar model dapat mencapai performa optimal, dilakukan penyetelan *hyperparameter*. Proses ini bertujuan untuk menemukan kombinasi parameter terbaik melalui pendekatan sistematis dan adaptif.

1. Menyusun arsitektur model dengan berbagai *pretrained model* dasar dan menentukan *hyperparameter* yang akan di-*tuning*.
2. Mengatur rentang dan nilai-nilai potensial untuk setiap *hyperparameter*.
3. Menginisialisasi algoritma *tuning* dengan parameter seperti *max_epochs, factor,* dan *objective* untuk memaksimalkan akurasi validasi.
4. Menjalankan *tuning hyperparameter* secara bertahap dalam beberapa *round*, dengan mengevaluasi kombinasi *hyperparameter* pada jumlah *epoch* yang rendah, kemudian melanjutkan kombinasi terbaik ke tahap pelatihan yang lebih mendalam.

5. Menyeleksi model dengan performa terbaik dari hasil *tuning* untuk digunakan dalam pelatihan dan evaluasi akhir.

### 3.2.5 Model Building

Setelah hyperparameter ditentukan, langkah berikutnya adalah membangun model klasifikasi. Arsitektur yang digunakan disesuaikan dengan kebutuhan klasifikasi dan hasil *tuning* sebelumnya.

1. Melakukan penyetelan parameter yang telah terpilih melalui proses *hyperparameter tuning* pada arsitektur model untuk meningkatkan kinerja model.
2. Klasifikasi penyakit fokus pada Pneumonia, COVID-19, Tuberkulosis dan Normal sebagai kategori utama.

### 3.2.6 Model Training

Model yang telah dibangun kemudian dilatih menggunakan data yang telah diproses. Selama pelatihan, digunakan *callback* untuk memaksimalkan efisiensi dan hasil model.

1. *ModelCheckpoint* : Menyimpan bobot model terbaik secara otomatis berdasarkan akurasi validasi tertinggi selama pelatihan.
2. *EarlyStopping* : Menghentikan pelatihan lebih awal jika tidak ada peningkatan akurasi validasi minimal delta setelah beberapa *epoch*, serta mengembalikan bobot terbaik.
3. *ReduceLROnPlateau* : Menurunkan *learning rate* jika *loss* validasi tidak membaik dalam beberapa *epoch*, guna menjaga stabilitas dan efektivitas pelatihan.

### 3.2.7 Model Comparative

Agar dapat memilih model terbaik, dilakukan perbandingan kinerja beberapa arsitektur model *deep learning* berdasarkan metrik evaluasi yang relevan.

1. Melakukan pemilihan terhadap berbagai berbagai model *deep learning* seperti ResNet50, EfficientNetB3, BaseCNN, dan DenseNet201.
2. Performa model dianalisis berdasarkan metrik di setiap kelas (COVID-19, Normal, Pneumonia, dan Tuberkulosis), guna menilai konsistensi dan kekuatan deteksi tiap kategori.

### 3.2.8 Model Evaluation

Evaluasi menyeluruh dilakukan untuk mengukur kinerja akhir model. Penggunaan metrik seperti *confusion matrix* membantu menilai aspek prediktif model secara lebih detail.

1. Evaluasi kinerja model menggunakan metrik seperti akurasi keseluruhan (*Overall Accuracy*).
2. Melakukan analisis hasil dengan menggunakan *Confusion Matrix* untuk mengukur *accuracy*, *precision*, *recall*, dan tingkat kesalahan model.

### 3.2.9 Deployment

Langkah terakhir adalah *deployment* aplikasi agar dapat digunakan oleh pengguna akhir. Proses ini mencakup penyusunan struktur proyek hingga implementasi melalui platform berbasis *cloud*.

1. Menyusun direktori proyek secara sistematis dengan minimal terdiri dari file app.py (kode aplikasi Streamlit) dan requirements.txt (daftar dependensi), lalu menyimpan seluruh proyek ke dalam repositori GitHub agar siap untuk proses

*deployment* lintas platform.

2. Melakukan *deployment* aplikasi melalui *Streamlit Cloud* dengan langkah: login, memilih repositori GitHub yang telah diunggah, memastikan app.py terdeteksi sebagai *entry point*, serta mengatur konfigurasi dasar aplikasi untuk menjalankannya secara online.
3. Menyediakan dua mode prediksi utama dalam aplikasi: prediksi citra tunggal dan prediksi *batch* melalui unggahan file ZIP, memungkinkan fleksibilitas penggunaan oleh pengguna akhir.

## 3.3 Variabel Penelitian

### 3.3.1 Variabel Dependen

Variabel dependen didefinisikan sebagai variabel hasil yang terpengaruh dan terikat oleh variabel independen [71]. Variabel dependen yang dianalisis adalah atribut penyakit yang akan dilabeli berdasarkan variabel independen yaitu atribut penyakit yang diklasifikasikan menjadi tiga kategori yaitu, pneumonia, tuberkulosis dan COVID-19 untuk membedakan berdasarkan *features* yang dimiliki oleh citra yang ada pada dataset.

### 3.3.2 Variabel Independen

Variabel independen dapat diidentifikasi sebagai fitur-fitur yang diekstraksi dari citra tersebut. Fitur-fitur ini menjadi dasar bagi CNN untuk memahami pola-pola yang berkaitan dengan klasifikasi penyakit. Sebagai contoh, variabel independen mencakup karakteristik visual seperti bentuk, tekstur, dan pola yang ada dalam citra. CNN akan menggunakan informasi ini untuk membedakan antara pneumonia, tuberkulosis, dan COVID-19. Oleh karena itu, variabel independen

dalam hal ini adalah representasi numerik dari fitur-fitur visual pada citra yang menjadi *input* untuk algoritma CNN.

### 3.4 Teknik Pengumpulan Data

Teknik yang dilakukan dalam pengumpulan data pada penelitian ini yaitu melalui data sekunder dari *Kaggle* [67]. Dalam penelitian ini, digunakan data yang terdiri dari 8.022 citra sinar-X bagian paru-paru. Citra-citra ini dibagi menjadi data pelatihan dan data pengujian dari pasien-pasien independen. Pada data pelatihan diberi label sebagai (penyakit)-(nomor ID citra). Untuk analisis citra sinar-X, semua radiografi awalnya diskrining untuk kontrol kualitas dengan menghapus semua pemindaian berkualitas rendah atau tidak dapat dibaca. Variabel-variabel yang menggambarkan data yang dihasilkan merupakan 1 variabel dependen bernama "penyakit" yang menampilkan parameter "Pneumonia", "COVID-19", dan "Tuberkulosis".

### 3.5 Teknik Pengambilan Sample

Awalnya, dataset asli telah dipisahkan menjadi data pelatihan dan data validasi dengan rasio sekitar 75:25. Namun demikian, setelah dilakukan pelatihan dan pengujian model menggunakan rasio awal ini, diperoleh hasil bahwa model masih belum mencapai tingkat akurasi optimal seperti yang diharapkan. Hal ini mengindikasikan adanya ruang untuk perbaikan dalam strategi pembagian dataset. Oleh karena itu, untuk memastikan pembagian dataset yang lebih optimal dan sesuai dengan praktik terbaik yang banyak diterapkan dalam literatur, dataset tersebut kemudian digabungkan kembali dan dilakukan pembagian ulang secara manual dalam tahap coding. Pembagian ulang ini menggunakan rasio yang lebih umum diterapkan, yaitu 80:20. Rasio ini dipilih berdasarkan prinsip Pareto dan sejumlah penelitian empiris yang menunjukkan bahwa rasio tersebut secara efektif menjaga keseimbangan antara data pelatihan yang cukup untuk generalisasi model, serta data validasi yang cukup untuk evaluasi performa yang stabil [72][73].

Setelah menetapkan rasio dataset yang optimal, langkah selanjutnya adalah menentukan metode pemilihan sampel dan proses persiapan data yang akan digunakan dalam penelitian ini. Metode pemilihan sampel dalam penelitian ini melibatkan penggunaan dataset yang tersedia dari repositori *Kaggle,* yang terdiri dari 8.022 citra foto sinar-X bagian dada [67]. Sampel tersebut kemudian dikelompokkan menjadi dua bagian utama, yaitu data pelatihan dan data pengujian. Tabel 3.2 menunjukan rasio pembagian dataset yaitu 80:20 dengan data pelatihan (*Training*) mencakup 5.120 citra, sementara data validasi (*Validation*) terdiri dari 1.280 citra dan data pengujian (*Testing*) sebanyak 1.622 citra. Proses labelisasi pada data pelatihan dilakukan dengan memberikan label berdasarkan format (penyakit)-(nomor ID citra). Sebelum dilakukan analisis citra sinar-X, dilakukan *screening* pada seluruh radiografi untuk kontrol kualitas dengan menghapus semua pemindaian berkualitas rendah atau tidak dapat dibaca. Dengan teknik ini, penelitian dapat menggunakan sampel yang representatif dan bervariasi untuk melibatkan berbagai kondisi penyakit pada citra sinar-X dari bagian paru-paru.

Tabel 3.3 Tabel Pembagian Dataset

| **Class** | **Jumlah Dataset** | | |
|---|---|---|---|
| | **Training** | **Validation** | **Testing** |
| COVID-19 | 1.280 | 320 | 407 |
| Normal | 1.280 | 320 | 404 |
| Pneumonia | 1.280 | 320 | 403 |
| Tuberkulosis | 1.280 | 320 | 408 |
| Total | 5.120 | 1.280 | 1.622 |

### 3.6 Teknik Analisis Data

Analisis data pada penelitian ini dilakukan melalui pendekatan kuantitatif menggunakan algoritma CNN. Teknik analisis data akan memanfaatkan berbagai *tools* atau perangkat lunak pengolah data. Python

dipilih sebagai bahasa pemrograman utama, dengan eksekusi kode yang dijalankan melalui platform *Google Colaboratory*. Python dipilih sebagai tools untuk teknik analisis data sebab memiliki keunggulan utama dalam *machine learning* dengan ekosistemnya yang kaya dengan berbagai library seperti *scikit-learn, TensorFlow,* dan *PyTorch*, serta dukungan dari komunitas pengembang yang besar. Kemampuan Python untuk diintegrasikan dengan *framework* pemrosesan big data seperti *Apache Spark* juga membuatnya cocok untuk menangani tugas *machine learning* dengan dataset besar. Sintaksis yang mudah dipelajari dan digunakan, bersama dengan versatilitasnya yang luas di berbagai industri, menjadikan Python pilihan populer untuk pengembangan model *machine learning* [74].