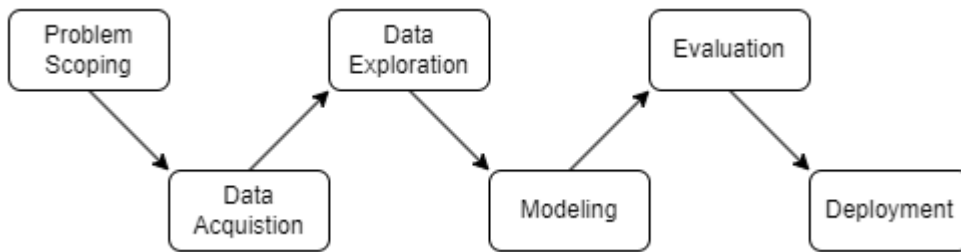


BAB III METODE PENELITIAN

3.1. Metode Penelitian

Metode penelitian yang akan digunakan dalam penelitian skripsi kali ini, yaitu dengan menerapkan konsep *AI Project Cycle*. Konsep metodologi direpresentasikan pada gambar berikut.



Gambar 3.1. *AI Project Cycle*

Berdasarkan gambar di atas, konsep penelitian ini memiliki enam tahapan, diantaranya adalah *problem scoping*, *data acquisition*, *data exploration*, *modeling*, *evaluation*, dan *deployment*. Alur pelaksanaan penelitian dijelaskan sebagai berikut:

1. *Problem Scoping*

Proses identifikasi atau pemetaan batasan masalah penting untuk menjelaskan dan memfokuskan tujuan atau target, sehingga memudahkan pencarian solusi. Untuk membantu proses ini, digunakan metode 4W, yang meliputi: *Who* (Siapa yang terlibat dalam masalah), *What* (Apa saja masalah dan faktor-faktor yang mendukungnya), *Where* (Lokasi atau situasi tempat masalah tersebut diuji), dan *Why* (Alasan perlunya menyelesaikan masalah tersebut dan manfaat yang akan diperoleh) (Azimah & Wardani, 2022). Secara spesifik, dalam konteks penelitian ini: *Who* melibatkan peneliti, *What* berkaitan dengan pengembangan aplikasi web diagnosis, *Where* berlangsung di perangkat lokal, dan *Why* ditujukan untuk memberikan hasil yang lebih efisien dalam proses diagnosis.

2. *Data Acquisition*

Proses pengumpulan data merupakan langkah awal penting dalam pengembangan proyek AI, dimana data tersebut digunakan sebagai bahan

dasar yang akan diolah dan dianalisis untuk mengidentifikasi masalah dan menghasilkan solusi terbaik. Terdapat berbagai metode untuk memperoleh sumber data ini, antara lain menggunakan alat seperti kamera, mikrofon, dan sensor; melalui observasi dengan survei, dan penelitian; memanfaatkan open data dari sumber seperti BPS, Kaggle, dan berbagai API (misalnya REST API, Twitter API, YouTube API); serta teknik web *scraping* atau web *crawling* (Azimah & Wardani, 2022). Pada penelitian ini, dataset yang digunakan berasal dari beberapa sumber yang tersedia di Kaggle.

3. *Data Exploration*

Exploratory Data Analysis (EDA), yang diperkenalkan oleh John Tukey, merupakan proses yang digunakan untuk memahami isi, komponen, dan karakteristik *dataset* agar dapat mengidentifikasi pola-pola yang ada. Tujuan dari EDA adalah untuk mendorong ahli statistik dalam mengeksplorasi data dan merumuskan hipotesis. Dalam praktiknya, EDA melibatkan penggunaan berbagai metode seperti *Summary Descriptive Statistics*, yang meliputi analisis frekuensi, *modus*, *mean*, *median*, rentang, varians, dan standar deviasi dari data. Selain itu, metode *Visualization* memungkinkan penyajian data dalam format grafis, termasuk *Bar Chart*, *Histogram*, *Box Plot*, *Scatter Plot*, *Star Plot*, *Chernoff Plot*, dan *Maps*. Metode lainnya, seperti *Clustering* dan *Anomaly Detection*, memanfaatkan elemen visual untuk mempermudah pemahaman terhadap tren dan memudahkan identifikasi outlier dalam *dataset* (Azimah & Wardani, 2022). Dalam penelitian ini, eksplorasi data yang dilakukan melibatkan augmentasi, pembagian data menjadi tiga bagian (*train*, *val*, *test*), dan visualisasi untuk memudahkan dalam memahami distribusi data.

4. *Modeling*

Proses implementasi algoritma ke dalam bahasa pemrograman sebagai bagian dari pelatihan model pembelajaran mesin (*training phase*). Tujuannya adalah untuk mengembangkan kemampuan model dalam mengambil keputusan atau membuat prediksi yang akurat (Sanjaya & Kurniawan, 2024). Dalam penelitian ini, diterapkan metode transfer learning menggunakan model EfficientNet-B0.

5. *Evaluation*

Tahap evaluasi adalah tahapan yang dilakukan setelah model pembelajaran mesin selesai dilatih. Selama tahap ini, model diuji untuk mengukur akurasi dari data yang telah diproses selama pelatihan. Jika evaluasi menunjukkan bahwa model belum mencapai kinerja yang diharapkan, proses pelatihan akan diulang untuk meningkatkan efektivitas model tersebut. Setelah model dinilai berfungsi dengan baik, proses selanjutnya adalah melanjutkan ke tahap implementasi atau penerapan (Sanjaya & Kurniawan, 2024). Secara spesifik, dalam penelitian ini model diuji menggunakan metrik *precision*, *recall*, dan *f1-score* untuk mengukur akurasi dari data yang telah diproses selama pelatihan. Pelatihan dilakukan sebanyak dua kali untuk memastikan bahwa model mencapai kinerja yang diharapkan.

6. *Deployment*

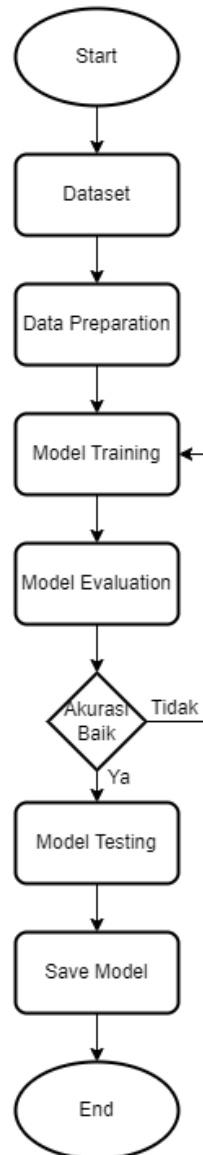
Proses implementasi AI dalam sebuah aplikasi atau sistem bertujuan untuk memenuhi kebutuhan produk yang dibuat, sehingga diharapkan dapat memudahkan pengguna dalam mengakses dan menggunakan website tersebut (Azimah & Wardani, 2022). Penelitian ini mengintegrasikan AI ke dalam aplikasi web menggunakan *framework* Flask, yang mendukung integrasi mulus dan pengelolaan *backend* yang efektif.

3.2. Perancangan Sistem

Pada perancangan sistem ini akan dijelaskan mengenai alur perancangan model, *use case diagram* aplikasi, *activity diagram* aplikasi.

3.2.1. Flowchart Perancangan Model

Flowchart ini dirancang untuk menggambarkan secara visual tahapan perancangan model secara umum, mulai dari awal hingga akhir.



Gambar 3.2. *Flowchart* Perancangan Model

1. *Dataset*

Kumpulan data yang diorganisir dan digunakan untuk tujuan analisis, penelitian, atau pengembangan algoritma. Ini dapat mencakup berbagai jenis data, seperti gambar, teks, atau angka, dan seringkali digunakan dalam konteks ilmiah, penelitian, atau pengembangan teknologi. *Dataset* dapat bersifat publik atau internal, tergantung pada tujuannya.

2. *Data Preparation*

Serangkaian langkah untuk menyiapkan data sebelum dilakukan analisis. Ini melibatkan pembersihan data dengan mengatasi nilai yang hilang atau outlier, transformasi data untuk memenuhi kebutuhan analisis, reduksi dimensi jika perlu, pemilihan data yang relevan, penggabungan data dari berbagai sumber, perubahan format agar sesuai, manajemen duplikasi, penanganan ketidakseimbangan kelas, dan pelabelan data jika diperlukan. Tujuannya adalah memastikan bahwa data yang digunakan untuk analisis memiliki kualitas yang baik, sesuai dengan kebutuhan, dan dapat menghasilkan hasil yang akurat dan dapat diandalkan.

3. *Model Training*

Proses di mana model *machine learning* "mempelajari" pola dari data pelatihan untuk membuat prediksi atau keputusan di masa depan. Ini melibatkan pemilihan model, fitur, dan fungsi kerugian, serta proses iteratif untuk mengoptimalkan kinerja model pada data pelatihan dan memastikan bahwa model dapat menggeneralisasi dengan baik pada data baru yang belum pernah dilihat sebelumnya.

4. *Model Evaluation*

Proses evaluasi kinerja model pada data yang telah dilihat selama pelatihan dan mungkin juga data yang belum pernah dilihat sebelumnya. Ini mencakup penggunaan metrik evaluasi seperti akurasi, *precision*, dan *recall* untuk mengukur seberapa baik model dapat menghasilkan prediksi atau keputusan pada data tertentu. Model *evaluation* seringkali melibatkan tahap validasi dan tuning untuk meningkatkan kinerja model.

5. *Model Testing*

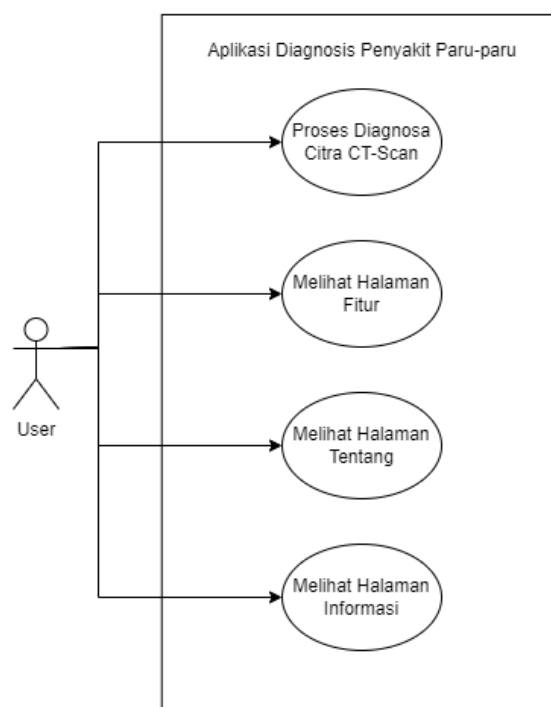
Tahap pengujian model menggunakan *dataset* yang tidak pernah dilihat selama pelatihan. Data uji digunakan untuk mengevaluasi sejauh mana model dapat menggeneralisasi pada situasi dunia nyata yang baru. Ini bertujuan untuk memberikan pemahaman lebih lanjut tentang kinerja model di luar data pelatihan.

6. Save Model

Tahap penyimpanan model dilakukan setelah model yang telah dilatih mencapai kinerja yang diinginkan.

3.2.2. Use Case Diagram Aplikasi

Use case diagram digunakan untuk menggambarkan interaksi antara aktor dan aplikasi. Diagram ini membantu para pengembang memahami kebutuhan fungsional aplikasi dan menunjukkan bagaimana pengguna berinteraksi dengan aplikasi tersebut.



Gambar 3.3. Use Case Diagram

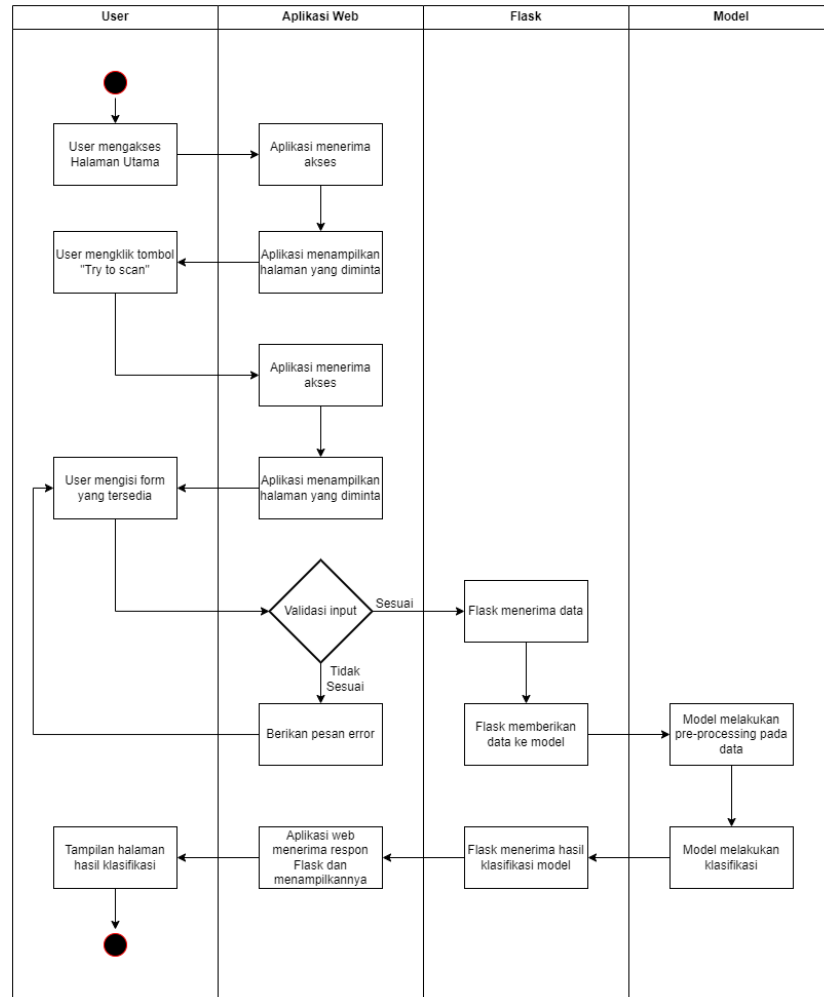
Seperti yang terlihat pada gambar, diagram ini memetakan empat kasus penggunaan utama dalam aplikasi diagnosis penyakit paru-paru: proses diagnosis citra *CT-Scan*, melihat halaman fitur, melihat halaman tentang, dan melihat halaman informasi. Setiap kasus penggunaan ini menunjukkan tugas yang dapat dilakukan pengguna dalam aplikasi, memudahkan pemahaman alur kerja keseluruhan dan interaksi antara pengguna dengan aplikasi.

3.2.3. Activity Diagram Aplikasi

Activity diagram menggambarkan berbagai interaksi pengguna dengan aplikasi web diagnosis *CT-Scan*. Diagram ini secara rinci memvisualisasikan alur kegiatan dari pengguna yang meliputi proses diagnosa citra *CT-Scan*, akses ke halaman fitur, halaman tentang, dan halaman informasi. Setiap aktivitas ini dipetakan untuk memberikan gambaran yang jelas tentang bagaimana pengguna berinteraksi dengan sistem serta bagaimana sistem merespons aksi-aksi tersebut.

1. Proses diagnosis citra *CT-Scan*

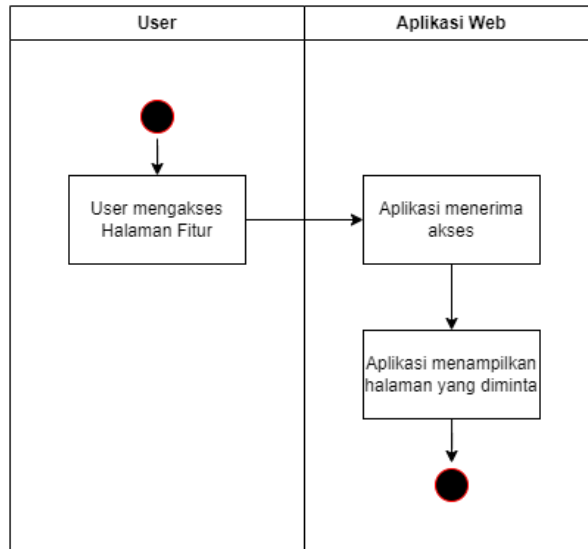
Proses dimulai ketika pengguna mengakses halaman utama dan memilih untuk memulai *scan* dengan mengklik tombol "Try to Scan". Setelah itu, pengguna diarahkan untuk mengisi form yang mencakup pengunggahan citra *CT-Scan* dan informasi tambahan yang relevan. Sistem kemudian melakukan validasi terhadap data yang dimasukkan; jika terdapat kesalahan, sistem akan memberi tahu pengguna untuk melakukan koreksi. Setelah validasi berhasil, data dikirim ke Flask yang selanjutnya meneruskannya ke model klasifikasi. Model ini menganalisis citra dan menghasilkan diagnosa yang kemudian ditampilkan kembali ke pengguna melalui aplikasi.



Gambar 3.4. Activity Diagram Proses Diagnosis Citra CT-Scan

2. Halaman fitur

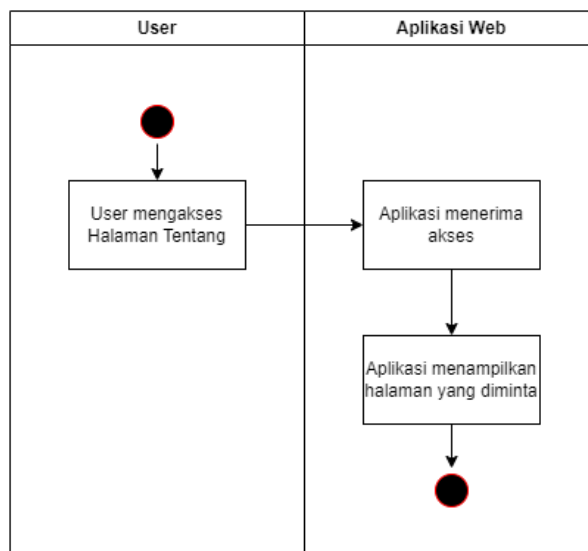
Proses melihat halaman fitur dimulai saat pengguna memilih menu fitur yang ada pada navbar. Setelah diakses, sistem secara otomatis memuat dan menampilkan halaman tersebut, yang berisi informasi detail mengenai semua fitur yang disediakan aplikasi.



Gambar 3.5. *Activity Diagram* Halaman Fitur

3. Halaman Tentang

Proses melihat halaman tentang diawali ketika pengguna memilih menu tentang yang ada pada navbar. Setelah pengguna memilih opsi tersebut, sistem akan segera memuat dan menampilkan halaman tentang yang mencakup informasi rinci mengenai latar belakang pengembangan aplikasi, tujuan dari pembuatannya, dan detail lain yang relevan.

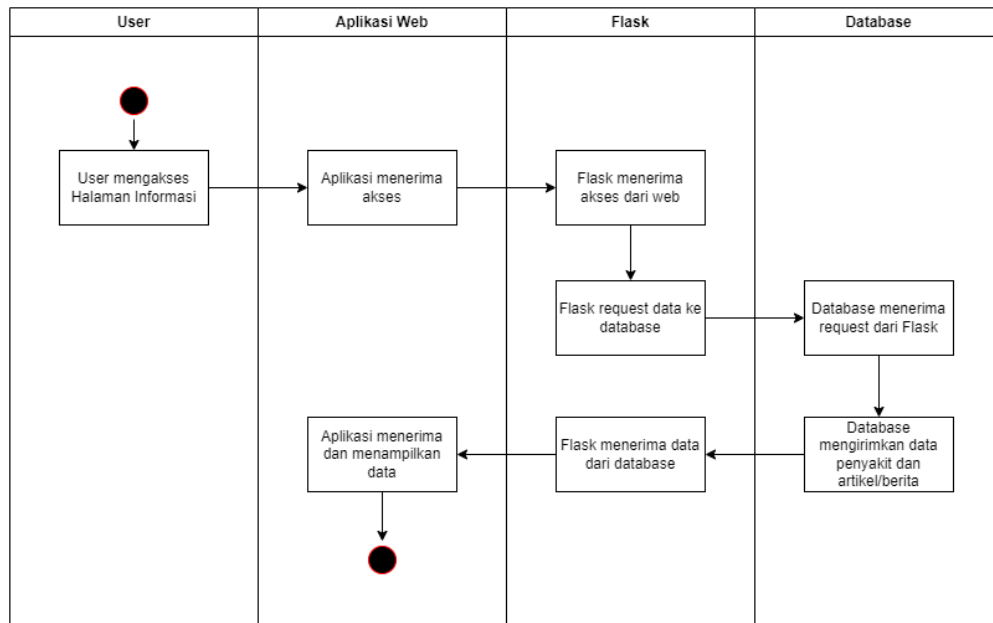


Gambar 3.6. *Activity Diagram* Halaman Tentang

4. Halaman Informasi

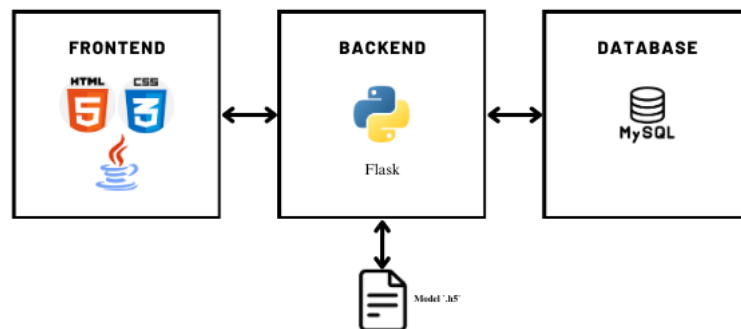
Proses melihat halaman informasi dimulai ketika pengguna memilih menu informasi yang ada pada navbar. Setelah dipilih, sistem langsung memuat dan

menampilkan halaman tersebut yang diisi dengan berbagai artikel, berita, dan informasi penyakit paru-paru yang relevan.



Gambar 3.7. Activity Diagram Halaman Informasi

3.2.4. Skema Integrasi Sistem



Gambar 3.8. Skema Integrasi Sistem

Dalam pengembangan aplikasi diagnosis penyakit paru-paru ini, terdapat tiga komponen utama yang saling terintegrasi, yaitu *frontend*, *backend*, dan *database*. Berikut adalah penjelasan mengenai ketiga komponen tersebut.

1. *Frontend*

Dalam penelitian ini, produk dikembangkan dalam bentuk website yang berfungsi sebagai antarmuka pengguna dengan sistem. Perancangan *frontend* website ini menggunakan kombinasi dari HTML, CSS, dan *JavaScript*. HTML digunakan untuk membangun struktur dasar halaman web, CSS diterapkan untuk

mengatur tampilan dan gaya visual, sedangkan *JavaScript* diintegrasikan untuk meningkatkan interaktivitas pada halaman web tersebut. Selain itu, *bootstrap* juga diintegrasikan dalam pengembangan *frontend* untuk memudahkan responsivitas dan estetika antarmuka. *Framework* ini membantu dalam mengimplementasikan desain yang responsif dan modern.

2. *Backend*

Backend merupakan komponen penting dalam arsitektur aplikasi, berfungsi sebagai penghubung antara *frontend*, model, dan *database*. Dalam penelitian ini, Flask digunakan sebagai *framework* utama untuk membangun *backend* karena kesederhanaannya, fleksibilitasnya, dan dukungannya yang luas untuk integrasi dengan pustaka Python lainnya, termasuk TensorFlow dan Keras.

Backend Flask diinisialisasi untuk menangani berbagai permintaan dari *frontend*, bertindak sebagai server yang menerima, memproses, dan mengirimkan *respons*. Flask mengatur berbagai rute untuk menghubungkan URL dengan fungsi tertentu, seperti mengunggah gambar *CT-Scan*, melakukan diagnosis penyakit, dan mengakses hasil diagnosis. Model yang telah dilatih dimuat ke dalam aplikasi Flask saat server dijalankan. Ketika pengguna mengunggah gambar *CT-Scan*, gambar tersebut diproses oleh Flask dan diteruskan ke model untuk prediksi, kemudian hasilnya dikirim kembali ke *frontend*.

Selain itu, Flask berinteraksi dengan *database* untuk menyimpan data pasien, hasil diagnosis, dan informasi relevan lainnya, serta memungkinkan pengambilan data untuk ditampilkan di *frontend*. Validasi input dilakukan untuk memastikan format dan isi data sesuai sebelum diproses lebih lanjut atau disimpan. Dengan menggunakan Flask sebagai *backend*, integrasi model dengan website menjadi efisien dan fleksibel, menyediakan solusi handal untuk pengolahan data dan komunikasi antara *frontend*, model, dan *database*.

3. *Database*

Penelitian ini menggunakan *database* untuk menyimpan data yang dibutuhkan oleh sistem. *Database* yang digunakan adalah MySQL, yang dikenal sebagai sistem manajemen basis data relasional yang andal dan efisien. MySQL digunakan untuk mengelola dan mengoperasikan data yang disimpan dalam

beberapa tabel. Bahasa SQL digunakan sebagai alat untuk menjalankan operasi dan menghubungkan antara aplikasi dengan *database*.

3.3. Skema Pengujian

Proses pengujian dirancang secara sistematis untuk mengukur kemampuan model dalam mengidentifikasi penyakit paru-paru melalui citra *CT-Scan*. Pengujian ini terdiri dari dua jenis, yaitu pengujian model di dalam aplikasi, dan pengujian aplikasi menggunakan metode blackbox.