

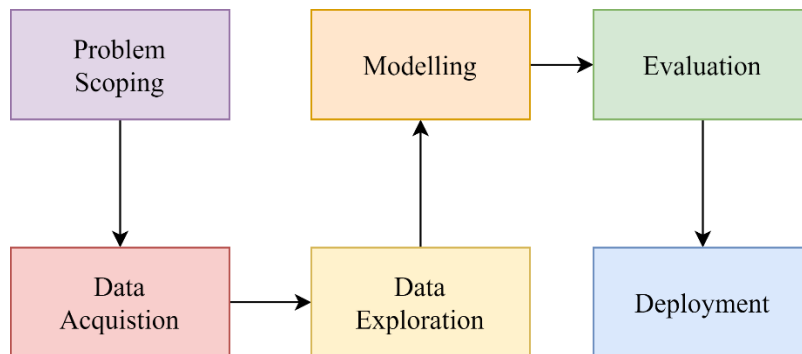
BAB III METODE PENELITIAN

3.1 Jenis Penelitian

Penelitian ini menggunakan jenis penelitian *Design and Development (D&D)* dengan pendekatan *AI Project Cycle*. Pendekatan ini terdiri dari enam tahapan: *problem scoping, data acquisition, data exploration, modelling, evaluation, dan deployment*. Metodologi ini bertujuan untuk merancang, mengembangkan, menguji, dan mengimplementasikan model klasifikasi level triase menggunakan arsitektur CNN satu dimensi, serta mengintegrasikan model tersebut ke dalam aplikasi web.

3.2 Desain Penelitian

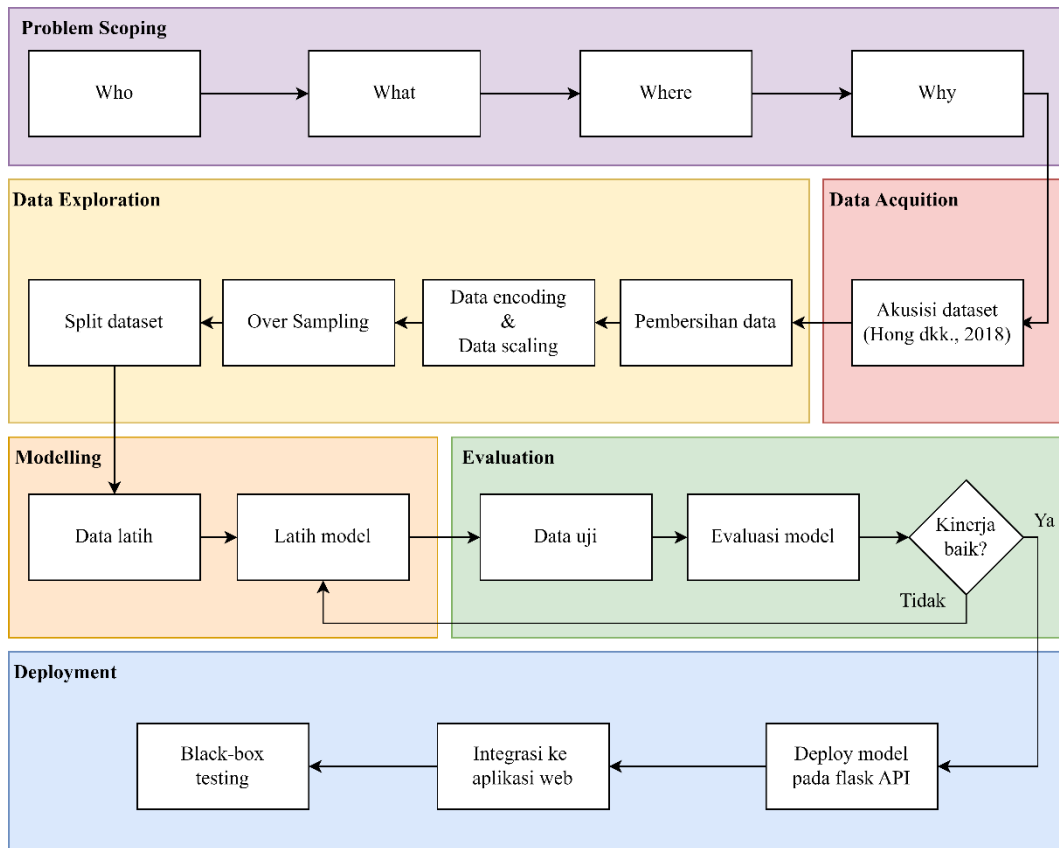
Desain penelitian yang akan digunakan pada penelitian ini berdasarkan pendekatan *AI Project Cycle* di mana prosedurnya mengikuti enam tahapan yang ada. Tahapannya terdiri dari *problem scoping, data acquisition, data exploration, modelling, evaluation, dan deployment* (Herdiana dkk., 2023; Prasetyo & Taufik Ridwan, 2023). Gambar 3.1 merupakan diagram dari *AI Project Cycle*.



Gambar 3.1 *AI Project Cycle*.

3.3 Prosedur Penelitian

Penelitian ini dilakukan melalui beberapa tahapan yang digambarkan pada Gambar 3.2 prosedur penelitian.



Gambar 3.2 Prosedur Penelitian.

3.3.1 Problem Scoping

Pada *problem scoping* terdapat metode yang disebut *The 4Ws Canvas*, ini membantu dalam menentukan batasan masalah dengan cara mengajukan pertanyaan berikut:

- Who*: Siapa pihak yang terpengaruh masalah tersebut.
- What*: Mengacu pada masalah apa yang dihadapi.
- Where*: Di mana situasi atau lokasi permasalahan tersebut terjadi.
- Why*: Kenapa masalah tersebut perlu diatasi.

Berikut merupakan jawaban dari *problem scoping*, *who*: tenaga kesehatan triase; *what*: *overcrowded* yang meningkatkan beban kerja petugas; *where*: terjadi di IGD; *why*: untuk mendukung petugas kesehatan dalam memberikan label triase pasien, diperlukan aplikasi pendukung keputusan sehingga petugas dapat membuat keputusan dengan baik ketika terjadi *overcrowded*.

3.3.2 Data Acquisition

Tahap berikutnya *data acquisition*, *dataset* yang digunakan pada penelitian ini diambil dari penelitian yang telah dilakukan oleh Hong dkk. *Dataset* yang tersedia berisi informasi triase pasien IGD yang mencakup berbagai atribut medis yang relevan untuk klasifikasi level triase seperti umur, jenis kelamin, tanda vital yang dicatat saat proses triase, tersedia atau tidaknya perangkat penyuplai oksigen saat triase, keluhan utama pasien. Terdapat 560.486 pasien yang diperoleh ketika mengunjungi IGD. Proses pengumpulan *dataset* ini dilakukan dibawah lisensi *Open Access*, di mana data disertakan dalam publikasi artikel ilmiah (Hong dkk., 2018).

Berikutnya proses pembersihan dilakukan dengan menghapus baris yang memiliki nilai kosong, tujuannya agar data lebih bersih dan siap untuk digunakan. Selanjutnya, tipe data kategorial diubah menjadi format yang dapat digunakan dalam penanganan data yang tidak seimbang melalui *one hot-encoding*. *One hot-encoding* membuat kolom baru untuk setiap kategori unik dengan nilai biner (0 atau 1). Langkah berikutnya adalah *scaling* data menggunakan *robust scaler* untuk memastikan semua fitur memiliki standarisasi. Untuk menangani data yang tidak seimbang, digunakan teknik *over-sampling* yang menyintesis contoh baru dari kelas minoritas sehingga distribusi kelas lebih seimbang. Terakhir, data dibagi menjadi data latih dan data uji dengan rasio tertentu, seperti 90% untuk latih dan 10% untuk uji, untuk memastikan model dapat diuji dengan data yang tidak terlihat selama pelatihan. Tahapan ini memastikan data yang digunakan bersih dan siap digunakan untuk melatih model klasifikasi.

3.3.3 Data Exploration

Data yang telah diperoleh dapat dikategorikan. kategori pertama yaitu demografi yang terdiri dari dua atribut mencakup *age* (umur) dan *gender* (jenis kelamin). Kategori kedua adalah tanda vital yang terdiri dari tujuh atribut di antaranya *triage_vital_sbp* (tekanan darah sistolik), *triage_vital_dbp* (tekanan darah diastolik), *triage_vital_rr* (laju pernapasan), *triage_vital_hr* (denyut jantung), *triage_vital_temp* (suhu tubuh), *triage_vital_o2* (saturasi oksigen), dan *triage_vital_temp* (ketersediaan perangkat pendukung oksigen). Kategori keempat adalah keluhan utama yang terdiri dari 200 keluhan yang paling sering muncul dan

kategori terakhir merupakan *ESI* (label triase) dengan standar *ESI*. Tabel 3.1, menunjukkan ringkasan dari *dataset* yang diperoleh.

Tabel 3.1
Informasi Nama Atribut dan Tipe Data dari *Dataset*

Kategori	Nama atribut	Tipe data
Demografi	<i>age</i>	Numerik
	<i>gender</i>	Kategorial
Tanda vital triase	<i>triage_vital_sbp</i>	Numerik
	<i>triage_vital_dbp</i>	Numerik
	<i>triage_vital_rr</i>	Numerik
	<i>triage_vital_hr</i>	Numerik
	<i>triage_vital_temp</i>	Numerik
	<i>triage_vital_o2</i>	Numerik
	<i>triage_vital_o2_device</i>	Numerik
Keluhan utama	Dimulai dari <i>cc_abdominalcramping</i> hingga <i>cc_wristpain</i>	Kategorial
Label triase	<i>esi</i>	Kategorial

Pada keluhan utama karena terdapat lebih dari 1000 nilai unik untuk keluhan utama pasien, hanya 200 nilai yang paling sering muncul yang dipertahankan sebagai kategori unik. Nilai-nilai ini mencakup lebih dari 90% dari semua kunjungan pasien. Sementara itu, semua nilai keluhan lainnya digabungkan ke dalam kategori "Lainnya". Pendekatan ini menyederhanakan data dengan fokus pada keluhan yang paling umum.

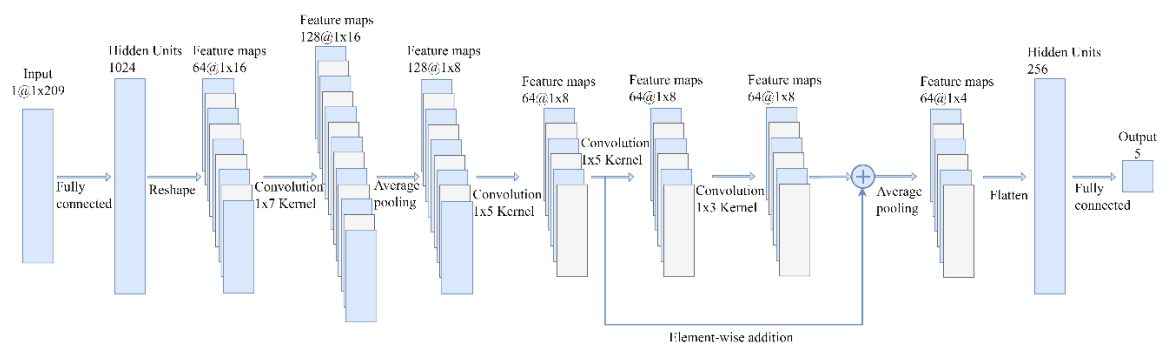
3.3.4 Modelling

Arsitektur model CNN satu dimensi yang digunakan dalam penelitian ini dirancang untuk menangani data *tabular* dalam proses klasifikasi level triase dengan teknik *soft reorder*. Model dimulai dengan lapisan *input* (*Input Layer*) yang menerima data dengan bentuk (*None, 209*), di mana 209 adalah jumlah fitur dalam *dataset*. Lapisan ini diikuti oleh lapisan *dense* dengan 1024 unit, yang berfungsi mengekstrak data *tabular* sehingga memiliki korelasi lokal.

Selanjutnya, bentuk data disesuaikan melalui lapisan *reshape*, yang mempersiapkan data untuk diproses oleh lapisan konvolusi. Lapisan konvolusi

pertama (*Conv1D*) memiliki 128 filter dengan ukuran *kernel* yang sesuai, menghasilkan *output* dengan bentuk (*None, 16, 128*). Untuk meningkatkan stabilitas dan kecepatan pelatihan, digunakan lapisan normalisasi *batch* (*BatchNormalization*) setelah setiap lapisan konvolusi. *Output* dari lapisan ini kemudian dikurangi dimensinya menjadi (*None, 8, 128*) menggunakan lapisan *average pooling* (*AveragePooling1D*).

Arsitektur ini juga mencakup beberapa lapisan konvolusi tambahan (*Conv1D*), masing-masing diikuti oleh lapisan *batch normalization* untuk menjaga kestabilan pelatihan. Setelah dua lapisan konvolusi dan normalisasi tambahan, data melewati proses penjumlahan residu (*Add*) yang membantu dalam menjaga informasi fitur penting dari lapisan sebelumnya. Model juga memiliki beberapa lapisan konvolusi tambahan dengan *batch normalization*, diikuti oleh penjumlahan residu (yang ditunjukkan oleh (*element-wise addition*)) untuk menjaga informasi yang sudah di ekstrak. Gambar 3.3 merupakan diagram arsitektur model klasifikasi level triase.



Gambar 3.3 Arsitektur Model Klasifikasi Level Triase

Selanjutnya, *output* dari lapisan residu diproses oleh lapisan *average pooling* tambahan yang mengurangi dimensinya menjadi (4, 64), lalu diratakan melalui lapisan *flatten* menjadi vektor satu dimensi sepanjang 256. Vektor ini diproses oleh lapisan *dense* dengan 256 unit, diikuti oleh lapisan *dropout* untuk mencegah *overfitting*. Akhirnya, lapisan *dense* terakhir memiliki 5 unit untuk menghasilkan probabilitas kelas. Setelah model terbentuk maka model akan dilatih menggunakan data latih yang telah disiapkan.

3.3.5 Evaluation

Masuk ke tahap evaluasi, model diuji menggunakan data tes untuk melihat kinerja dari model yang sudah dilatih. Kinerja model akan dievaluasi melalui lima

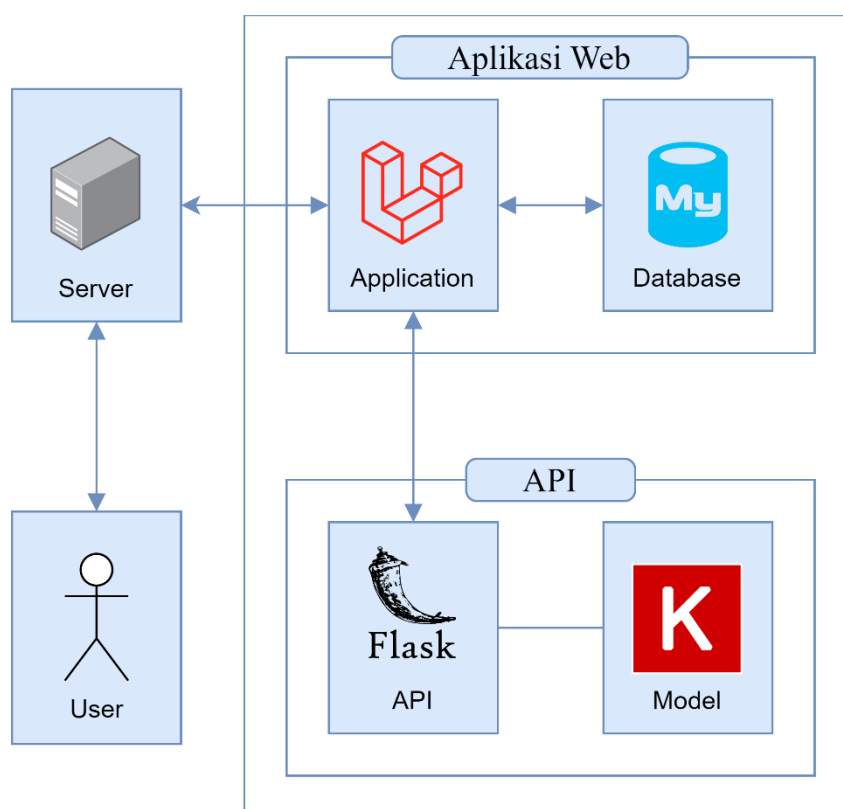
metrik kinerja seperti *confusion matrix*, *akurasi*, *precision*, *recall*, dan *f1-score*. *Confusion matrix* ialah tabel yang dapat melihat berapa banyak prediksi benar dan salah. akurasi merupakan rasio prediksi yang benar (baik positif atau negatif) untuk melihat gambaran umum dari kinerja model. *Precision* untuk melihat seberapa baik kinerja model dalam membuat prediksi yang benar (untuk kelas positif) dari seluruh prediksi positif. *Recall* untuk melihat menggambarkan kinerja seberapa baik model dalam mengidentifikasi kelas positif dengan benar. Jadi *precision* dan *recall* merupakan gambaran tentang seberapa baik model dapat memprediksi setiap level dengan benar. *F1-score* memperlihatkan keseimbangan antara *precision* dan *recall* untuk setiap level triase. Terdapat beberapa model lain yang dijadikan perbandingan seperti model *neural network*, *XGBoost*, dan *logistic regression*. Detail lain dari metrik evaluasi dijelaskan pada teknik analisis data.

3.3.6 Deployment

Setelah mengevaluasi model dan mendapatkan kinerja dari model, model dengan kinerja yang paling baik akan diintegrasikan kedalam aplikasi yang dapat diakses melalui *application programming interface* (API). *Framework* yang digunakan dalam membangun API tersebut adalah Flask yang merupakan mikro *framework*. API Flask ini akan menyediakan *endpoint* untuk menerima data *input*, melakukan *pre-processing* data, mengklasifikasikannya melalui model yang digunakan, dan mengembalikan hasilnya. API ini ditujukan untuk menjadi jembatan antar aplikasi lainnya yang ingin mengakses model. Agar pengguna dapat berinteraksi, *framework* Laravel digunakan sebagai jembatan untuk menangani antara pengguna dengan API untuk klasifikasi. Jembatan ini berupa aplikasi web, di mana seluruh interaksi pengguna terhadap API untuk mengakses model dapat diakses melalui aplikasi tersebut. Pada aplikasi tersebut Laravel akan memvalidasi *input* pengguna, melakukan *request* ke API, menerima respons dari API, dan menampilkan kembali hasilnya ke pengguna. Proses *deployment* ini melibatkan beberapa desain diagram seperti, arsitektur aplikasi, *flowchart*, *entity-relationship diagram*, *activity* dan *use case diagram*.

A. Arsitektur Aplikasi Klasifikasi Level Triase

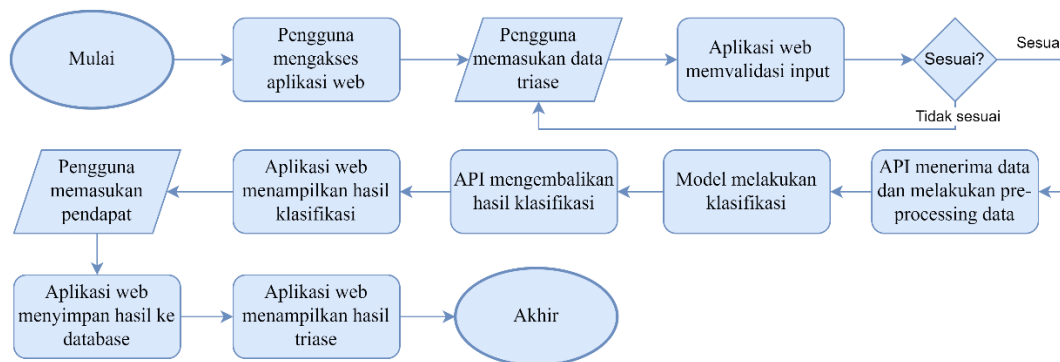
Gambar 3.4 menunjukkan diagram arsitektur aplikasi klasifikasi level triase yang terdiri dari beberapa komponen yang bekerja secara terintegrasi untuk memberikan hasil klasifikasi triase. *User* memulai interaksi melalui antarmuka yang disediakan oleh aplikasi web, yang dibangun menggunakan *framework* Laravel. Aplikasi Laravel bertanggung jawab untuk memvalidasi *input* dari *user*, mengirimkan data tersebut ke API Flask, dan menampilkan hasil klasifikasi kembali kepada *user*. API Flask berfungsi sebagai penghubung antara aplikasi web dan model CNN satu dimensi yang dikembangkan menggunakan Keras. Model ini, yang diintegrasikan pada Flask, menerima data *input* dari Laravel, memprosesnya, dan mengembalikan hasil klasifikasi. Server mengatur lalu lintas data antara *user*, aplikasi web, dan API untuk memastikan setiap permintaan diproses. *Database* yang terhubung dengan Laravel digunakan untuk menyimpan data yang diperlukan oleh aplikasi.



Gambar 3.4 Diagram Arsitektur Aplikasi Klasifikasi Level Triase

B. Flowchart Aplikasi Klasifikasi Level Triase

Flowchart dari aplikasi klasifikasi level triase yang digambarkan pada Gambar 3.5, dimulai dari pengguna yang mengakses aplikasi web, kemudian pengguna memasukkan data triase yang diperlukan. Aplikasi web kemudian memvalidasi *input* yang diberikan, jika *input* sesuai, data dikirim ke API untuk diproses lebih lanjut, jika tidak, pengguna diminta untuk memasukkan data yang valid. Setelah data diterima oleh API, proses *pre-processing* dilakukan sebelum data dimasukkan ke model klasifikasi. Model ini memproses data dan menghasilkan hasil klasifikasi yang kemudian dikirim kembali ke aplikasi web melalui API. Aplikasi web menampilkan hasil klasifikasi kepada pengguna. Selain itu, pengguna dapat memasukkan pendapat yang akan disimpan dalam *database* oleh aplikasi web. Terakhir aplikasi web akan menampilkan hasil triase.

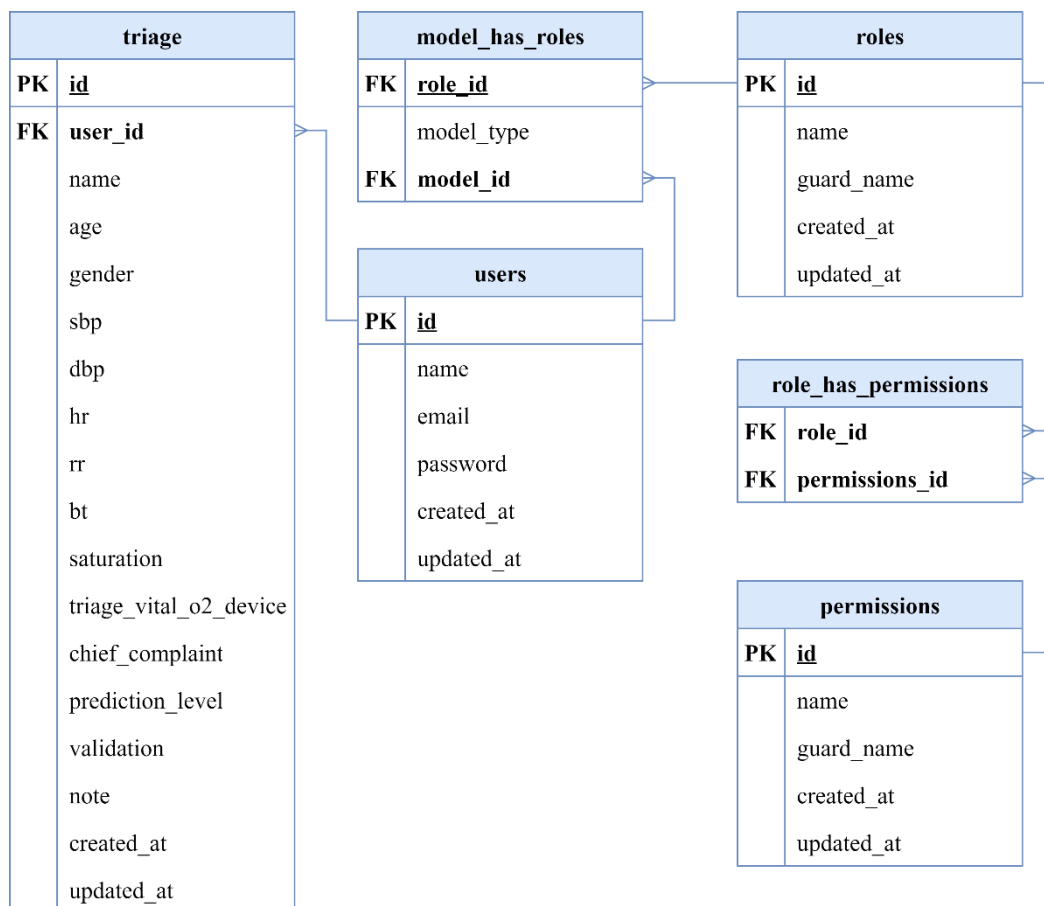


Gambar 3.5 *Flowchart* Aplikasi Klasifikasi Level Triase

C. Entity Relationship Diagram (ERD)

Kemudian untuk *entity relationship diagram* (ERD) dari aplikasi klasifikasi level triase yang diperlihatkan pada Gambar 3.6, diagram tersebut menggambarkan hubungan antar tabel dalam basis data yang digunakan oleh aplikasi. Diagram ini terdiri dari beberapa tabel seperti *triage*, *users*, *roles*, *permissions*, *model_has_roles*, dan *role_has_permissions*. Tabel *triage* menyimpan data triase pasien, termasuk informasi seperti nama, usia, jenis kelamin, tekanan darah (*sbp* dan *dbp*), detak jantung (*hr*), frekuensi pernapasan (*rr*), suhu tubuh (*bt*), saturasi oksigen (*saturation*), kebutuhan alat bantu pernapasan (*triage_vital_o2_device*), keluhan utama (*chief_complaint*), level prediksi (*prediction_level*), validasi (*validation*), catatan (*note*), serta waktu pembuatan dan pembaruan data (*cretaed_at*

dan *updated_at*). Entitas *users* menyimpan informasi pengguna aplikasi, seperti nama, email, *password*, dan *timestamp*.



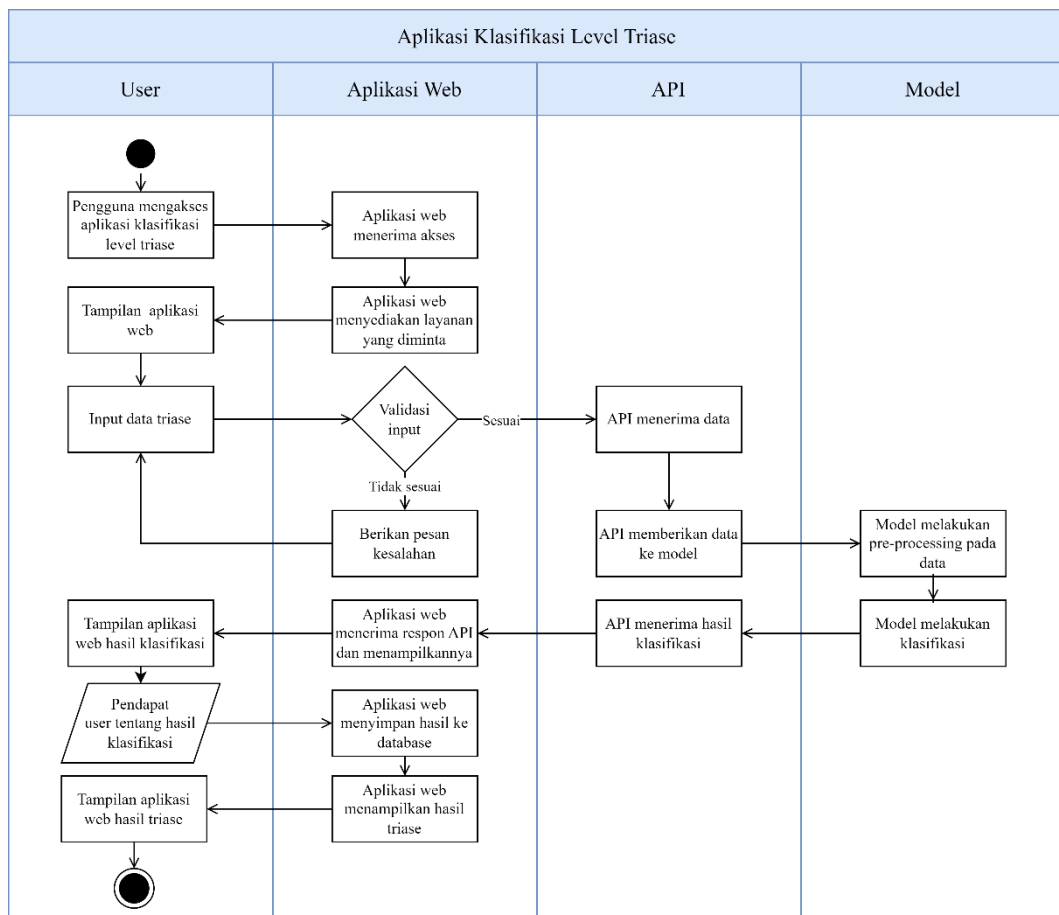
Gambar 3.6 ERD Aplikasi Klasifikasi Level Triase

Tabel *roles* dan *permissions* mengelola peran dan izin dalam aplikasi, memungkinkan kontrol akses yang terstruktur. Hubungan antara *roles* dan *permissions* dikelola melalui entitas *role_has_permissions*, sedangkan hubungan antara *users* dan *roles* dikelola melalui entitas *model_has_roles* yang mencakup tipe model dan ID model.

D. Activity Diagram

Gambar 3.7 menunjukkan *activity diagram* aplikasi klasifikasi level triase, gambar tersebut menggambarkan alur kerja dari interaksi pengguna hingga hasil klasifikasi triase ditampilkan. terdapat empat komponen seperti Pengguna, Aplikasi Web, API, dan Model. Proses dimulai ketika pengguna mengakses aplikasi

klasifikasi triase, yang kemudian menyediakan antarmuka untuk memasukkan data triase. Setelah data dimasukkan, aplikasi web melakukan validasi *input*, jika data tidak sesuai, pesan kesalahan diberikan kepada pengguna. Jika data sesuai, data dikirim ke API yang kemudian meneruskan data tersebut ke Model untuk dilakukan *pre-processing* dan klasifikasi. Model memproses data dan mengembalikan hasil klasifikasi ke API. Kemudian API menerima hasil klasifikasi dan mengirimkannya kembali ke aplikasi web. Berikutnya aplikasi menampilkan hasil klasifikasi kepada pengguna. Setelah itu pengguna memberikan validasi kesesuaian level klasifikasi dan dapat menambahkan catatan yang disimpan oleh aplikasi web. Terakhir aplikasi web akan menampilkan hasil triase.

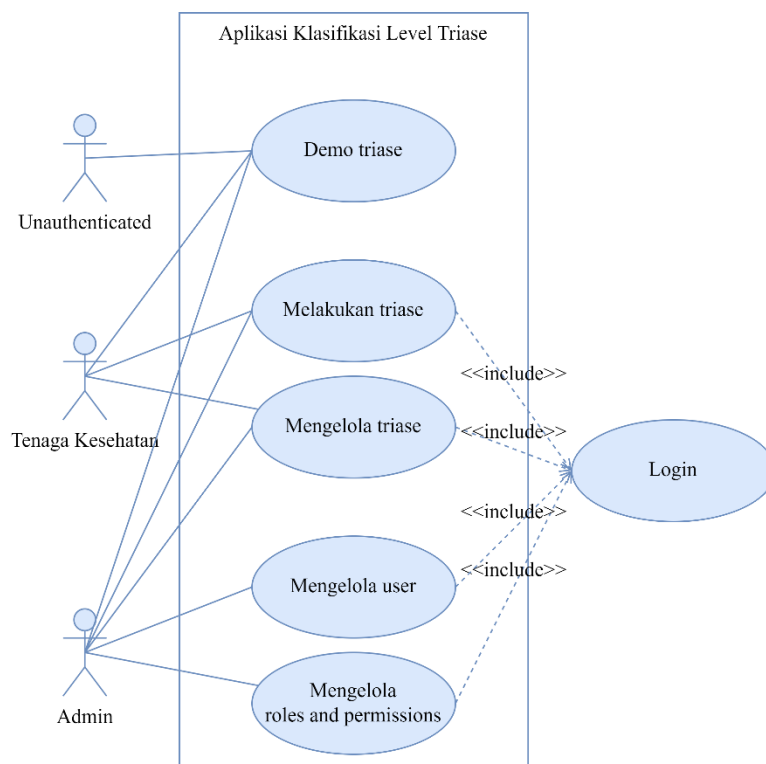


Gambar 3.7 Activity Diagram Aplikasi Klasifikasi Level Triase

E. Use case Diagram

Terakhir ada Gambar 3.8 menunjukkan *use case diagram* untuk aplikasi klasifikasi level triase, yang menggambarkan berbagai interaksi antara aktor (pengguna) dan fungsi-fungsi utama dalam aplikasi web. Gambar diagram tersebut

menggambarkan dua jenis aktor yang terdiri dari Tenaga Kesehatan dan Admin, masing-masing dengan akses ke fitur yang berbeda berdasarkan peran mereka. Tenaga Kesehatan dapat melakukan triase untuk pasien, serta mengelola triase yang mencakup melihat, mengedit, dan menghapus data triase. Admin memiliki hak akses yang lebih luas, termasuk kemampuan untuk mengelola pengguna dan mengelola peran and izin, memastikan bahwa setiap pengguna memiliki akses yang tepat sesuai dengan peran mereka. Setiap fungsi utama dalam aplikasi membutuhkan proses *login*, yang memastikan bahwa hanya pengguna yang ter autentikasi dapat mengakses fitur pada *dashboard* aplikasi tersebut.



Gambar 3.8 Use Case Diagram Aplikasi Klasifikasi Level Triase

F. Fungsionalitas Aplikasi Klasifikasi Level Triase

Tabel 3.2 merupakan rancangan *endpoint* API Flask yang akan digunakan pada penelitian ini. Di mana terdiri dua *endpoint* utama, yaitu */ping* dan */predict*. *Endpoint /ping* menggunakan metode GET dan tidak memerlukan *body request*. Tujuannya adalah untuk mengecek respons server dengan mengembalikan pesan "OK". Sementara itu, *endpoint /predict* menggunakan metode POST dan

memerlukan data triase sebagai *body request*. Tugas dari *endpoint* ini adalah melakukan klasifikasi dan mengembalikan level klasifikasi yang sesuai.

Tabel 3.2
Rancangan *Endpoint* API Flask

<i>Endpoint</i>	<i>Method</i>	<i>Request Body</i>	<i>Task</i>	<i>Response</i>
<i>/ping</i>	<i>GET</i>	-	Mengecek respons server	OK
<i>/predict</i>	<i>POST</i>	Data triase	Melakukan klasifikasi	Level Klasifikasi

Sedangkan untuk Tabel 3.3 menggambarkan berbagai halaman dan fungsionalitas yang ada dalam aplikasi web. Pada Halaman *Login*, pengguna dapat memasukkan *username* dan *password* untuk masuk, dengan validasi kredensial dan menampilkan pesan eror jika ada kesalahan. Halaman Lupa *Password* memungkinkan pengguna menerima email reset *password*, dengan pesan konfirmasi setelah email dikirim. Halaman *List* Triase menampilkan data triase dalam tabel, dengan tombol aksi untuk melihat detail, mengubah, menghapus, dan menambah data triase. Pada Halaman Tambah Triase, terdapat tiga tahap: memasukkan informasi pasien, tanda vital dan keluhan utama, serta *review* dan keputusan akhir, dan pengguna dapat menyimpan hasil triase. Halaman Ubah Triase memungkinkan perubahan informasi triase kecuali hasil klasifikasi model. Halaman *List* Pengguna menampilkan daftar pengguna, dengan opsi untuk menambah, mengubah, mengubah *password*, dan menghapus pengguna. Terakhir, Halaman Peran dan Izin memungkinkan penambahan, perubahan, dan penghapusan peran serta izin melalui *pop-up* konfirmasi.

Tabel 3.3
Fungsionalitas Aplikasi Klasifikasi Level Triase

Halaman	Fungsionalitas	Deskripsi Fungsionalitas
Halaman <i>Login</i>	<i>Login</i>	Pengguna dapat memasukkan <i>username</i> dan <i>password</i> untuk masuk ke halaman triase.
	Validasi Pengguna	Memeriksa keabsahan kredensial yang dimasukkan pengguna.
	Pesan Eror <i>Login</i>	Menampilkan pesan eror jika kredensial salah.

Halaman Lupa <i>Password</i>	Mengirim Email Reset <i>Password</i>	Pengguna dapat memasukkan email untuk menerima tautan reset <i>password</i> .
	Pesan Konfirmasi	Menampilkan pesan konfirmasi bahwa email reset telah dikirim.
Halaman <i>List</i> Triase	Menampilkan Data Triase	Menampilkan data triase dalam bentuk tabel dengan kolom No, Nama, Umur, Jenis Kelamin, Level Triase, Keluhan Utama, Tanggal & Waktu, Aksi.
	Tombol Aksi: Detail	Mengarahkan pengguna ke halaman detail triase.
	Tombol Aksi: Ubah	Mengarahkan pengguna ke halaman ubah triase.
	Tombol Aksi: Hapus Tombol Triase	Menghapus data triase yang dipilih. Mengarahkan pengguna ke halaman tambah triase.
Halaman Tambah Triase	Tahap 1 Informasi Pasien	Pengguna memasukkan informasi pasien: nama, umur, jenis kelamin.
	Tahap 2 Tanda Vital dan Keluhan Utama	Pengguna memasukkan tanda vital, status oksigen, dan keluhan utama pasien.
	Tahap 3 <i>Review</i> dan Keputusan Akhir	Menampilkan ringkasan informasi yang telah dimasukkan dan hasil klasifikasi.
	Simpan Hasil Triase	Pengguna dapat menyimpan hasil triase dan kembali ke halaman <i>list</i> triase.
Halaman Ubah Triase	Mengubah Informasi Triase	Pengguna dapat mengubah informasi triase kecuali hasil klasifikasi model.
Halaman <i>List</i> Pengguna	Menampilkan Data Pengguna	Menampilkan daftar pengguna yang telah ditambahkan dalam bentuk tabel.
	Tambah Pengguna	Pengguna dapat menambah pengguna baru dengan mengisi nama lengkap, email, peran, <i>password</i> , dan konfirmasi <i>password</i> .
	Ubah Informasi Pengguna	Pengguna dapat mengubah nama lengkap, email, dan peran pengguna yang sudah ada.
	Ubah <i>Password</i> Pengguna Hapus Pengguna	Pengguna dapat mengubah <i>password</i> pengguna yang sudah ada melalui <i>pop-up</i> . Menghapus pengguna yang dipilih melalui <i>pop-up</i> konfirmasi.
Halaman Peran dan Izin	Menambah Peran dan Izin	Pengguna dapat menambah peran dan izin baru melalui <i>pop-up</i> .
	Mengubah Peran dan Izin	Pengguna dapat mengubah peran dan izin yang sudah ada melalui <i>pop-up</i> .

Menghapus Peran dan Izin	Pengguna dapat menghapus peran dan izin yang sudah ada melalui <i>pop-up</i> konfirmasi.
--------------------------	--

3.4 Alat dan Teknik Analisis

3.4.1 Alat Pengembangan

Pada penelitian ini terdapat beberapa alat pengembangan yang digunakan untuk membantu rancang bangun klasifikasi level triase. Alat yang digunakan terdiri dari beberapa teknologi yang berbeda seperti yang uraikan pada Tabel 3.2 alat yang digunakan pada penelitian rancang bangun aplikasi klasifikasi level triase.

Tabel 3.4
Alat Pengembangan Klasifikasi Level Triase.

Nama Teknologi	Fungsi
Pandas	<i>Library</i> Python yang menyediakan struktur data dan alat analisis data untuk memanipulasi, menganalisis, dan visualisasi data dalam bentuk tabel.
Scikit-learn	<i>Library</i> Python yang kuat dan mudah digunakan untuk <i>machine learning</i> , menyediakan berbagai alat untuk pemodelan dan pra-pemrosesan data.
Keras	Sebagai API yang dirancang untuk memudahkan pengguna dalam pengembangan kecerdasan artifisial. Keras ditulis pada bahasa pemrograman Python sebagai <i>Library</i> .
Flask	Sebagai <i>framework</i> dalam proses pembuatan API. Model akan diintegrasikan pada Flask sehingga dapat diakses oleh sumber daya lain.
Laravel	Sebagai <i>framework</i> untuk aplikasi web klasifikasi level triase.
Google Colab	Sebagai platform <i>cloud</i> untuk menulis dan menjalankan kode python, terutama untuk pengembangan kecerdasan artifisial dan analisis data. Dengan spesifikasi Intel Xeon CPU dengan 2 vCPUs (<i>virtual</i> CPUs), 13GB RAM dan 15 GB GPU.

3.4.2 Teknik Analisis Data

Untuk menghitung nilai akurasi, *precision*, *recall*, dan *F1-score* yang ditunjukkan pada Tabel 3.5. Beberapa komponen yang ada pada dari *confusion matrix* seperti M_{ii} prediksi benar sesuai dengan level aktualnya. FP (*False Positive*) prediksi salah, di mana diprediksi sebagai level tertentu padahal aktualnya bukan level tersebut.

Tabel 3.5
Confusion Matrix Model Klasifikasi Level Triase

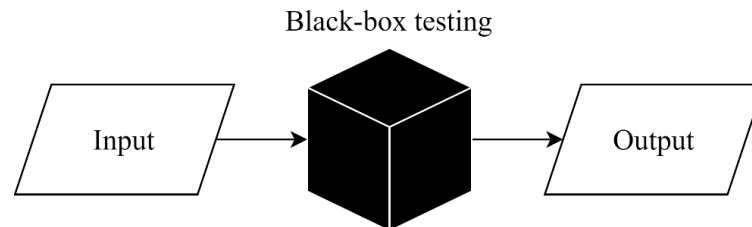
		Prediksi				
		Level 1	Level 2	Level 3	Level 4	Level 5
Aktual	Level 1	M_1	$M_{1,2}$	$M_{1,3}$	$M_{1,4}$	$M_{1,5}$
	Level 2	$M_{2,1}$	M_2	$M_{2,3}$	$M_{2,4}$	$M_{2,5}$
	Level 3	$M_{3,1}$	$M_{3,2}$	M_3	$M_{3,4}$	$M_{3,5}$
	Level 4	$M_{4,1}$	$M_{4,2}$	$M_{4,3}$	M_4	$M_{4,5}$
	Level 5	$M_{5,1}$	$M_{5,2}$	$M_{5,3}$	$M_{5,4}$	M_5

Dengan menggunakan Tabel 3.5, perhitungan dapat dilakukan untuk masing-masing level (Level 1 hingga Level 5) dan kemudian menghitung nilai akurasi dengan persamaan 1, *precision* persamaan dengan 2, *recall* persamaan 3, serta *f1-score* persamaan 4. Untuk itu model klasifikasi dengan arsitektur CNN satu dimensi akan dibandingkan dengan model *neural network*, *XGBoost*, dan *logistic regression* untuk mengevaluasi kinerja dari model-model klasifikasi dengan *dataset* yang digunakan pada penelitian ini. Hasil perhitungan tersebut akan memberikan gambaran mengenai kinerja setiap model pada masing-masing level, serta memberikan menjawab dari rumusan masalah yang sudah dipaparkan.

3.4.3 Pengujian Aplikasi

Pada rancang bangun aplikasi klasifikasi level triase, *black-box testing* digunakan sebagai metode untuk menguji fungsionalitas aplikasi. *Black-box testing* ialah metode pengujian aplikasi, di mana penguji mengevaluasi fungsionalitas aplikasi tanpa memeriksa kode sumber internal, struktur, atau implementasi program tersebut. Penguji berfokus pada *input* dan *output* aplikasi, memastikan

bahwa perangkat lunak berfungsi sesuai dengan fungsi yang ditentukan. Pendekatan ini memungkinkan pengujian yang objektif karena tidak bergantung pada pengetahuan teknis tentang bagaimana perangkat lunak dibuat (Nurfauziah & Jamaliyah, 2022). Gambar 3.9 merupakan representasi dari *black-box testing*.



Gambar 3.9 *Black-box Testing*.

Black box testing memungkinkan pengujian aplikasi dari perspektif pengguna akhir tanpa melihat atau memeriksa kode sumber internal. Pengujian ini dilakukan dengan menyimulasikan skenario penggunaan perangkat lunak, termasuk menerima *input* data triase, memproses data melalui model CNN satu dimensi, dan menghasilkan *output* klasifikasi. Dengan begitu fokus pengujian dapat memastikan bahwa semua fitur dan fungsi aplikasi yang telah diuraikan pada Tabel 3.2 dan Tabel 3.3 dapat bekerja sesuai dengan fungsi yang telah ditentukan.