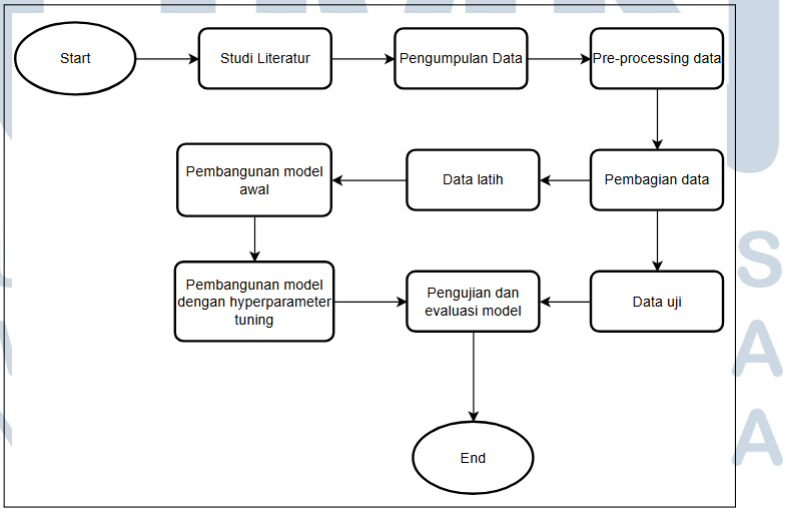
BAB 3
METODOLOGI PENELITIAN

3.1 Metodologi Penelitian

Metodologi penelitian yang digunakan dalam tugas akhir ini meliputi beberapa tahapan sistematis untuk mencapai tujuan penelitian. Proses dimulai dengan studi literatur untuk memahami konteks dan algoritma yang digunakan, dilanjutkan dengan pengumpulan serta *pre-process* data. Setelah itu, data dibagi menjadi data latih dan data uji, lalu dilakukan pembangunan model prediksi menggunakan algoritma RF dan XGB. Penelitian ini dilakukan dalam dua iterasi utama. Iterasi pertama menggunakan seluruh fitur yang tersedia dalam dataset, sementara iterasi kedua menggunakan model yang sama namun dengan implementasi *hyperparameter tuning* menggunakan *gridsearchcv* untuk meningkatkan akurasi. Setelah itu, dilakukan evaluasi performa model menggunakan *confusion matrix, classification report*, serta analisis *feature importance*. Sebagai eksperimen tambahan, dilakukan iterasi ketiga dengan hanya menggunakan 7 fitur terpenting dari masing-masing model untuk mengamati perubahan performa serta potensi efisiensi waktu komputasi.

3.2 Alur penelitian



Gambar 3.1. Diagram alur penelitian

Gambar 3.1 merepresentasikan alur metodologi dan langkah yang dilakukan dalam penelitian ini dimulai dari studi literatur dari teori yang digunakan dan berbagai media pustaka yang telah dipublikasi sebelumnya, pengumpulan dan pembersihan data, serta pembagian data tersebut menjadi data uji dan data latih untuk model RF dan XGB dan terakhir menguji dan mengevaluasi hasil penelitian dengan menggunakan *confusion matrix*.

3.3 Studi Literatur

Studi literatur dilakukan agar mendapatkan informasi mengenai cara kerja algoritma RF dan XGB serta pengimplementasian algoritma tersebut. Pada tahap ini pengumpulan informasi dari jurnal, karya ilmiah dan penelitian lainya yang berhubungan dengan objektif dari penelitian ini akan dilakukan. Tabel 3.1 berikut berisikan studi literatur yang menjadi acuan dalam penelitian ini.

Tabel 3.1. Studi literatur

Sumber	Komparasi Model	Hasil utama	Keterangan
Ahmed et al. (2021) 15	LR, KNN, NB, RF, XGB, AdaBoost	RF (86.64%), XGB (81.63%)	RF dan XGB unggul dalam memprediksi tipe kecelakaan di New Zealand diikuti AdaBoost 73.15%
Chen and Chen (2020) 16	LR, CART, RF	RF (73.38%)	RF memberikan akurasi tertinggi dalam prediksi tingkat keparahan kecelakaan di Taiwan diikuti LR 73.07% dan CART 72.6%
Ferrouhi dan Bouabdallaoui (2024) 20	RF, XGB, AdaBoost, Bagging-LSVM, Stacking	Mean squared error RF (0,0084), XGB (0,0089)	RF dan XGB termasuk ensemble dengan performa terbaik jika dibandingan model stacking
Givari et al. (2022) 21	SVM, RF, XGB	XGB (82%), RF (71%)	Pada perbandingan persetujuan pengajuan kreditdiikuti SVM 69%

3.4 Pengumpulan data

Pada tahap ini akan dilakukan pengumpumpulan data untuk diteliti. Data yang digunakan pada penelitian ini diambil dari website kaggle berupa dataset kecelakaan lalu lintas (https://www.kaggle.com/datasets/oktayrdeki/traffic-accidents) yang diakses dan diunduh pada 1 Januari 2025. Dataset ini berisi informasi rinci mengenai kecelakaan lalu lintas dari berbagai wilayah dengan rentang waktu dari tahun 2013 hingga 2025. Informasi yang tersedia mencakup tanggal kejadian, kondisi cuaca, kondisi pencahayaan, jenis kecelakaan, jumlah korban, serta keterlibatan kendaraan. Dataset memiliki 24 kolom yaitu:

1. *crash_date*: Tanggal terjadinya kecelakaan.
2. *traffic_control_device*: Jenis perangkat pengendali lalu lintas yang terlibat.
3. *weather_condition*: Kondisi cuaca saat kecelakaan terjadi.
4. *lighting_condition*: Kondisi pencahayaan saat kecelakaan terjadi.
5. *first_crash_type*: Jenis awal dari kecelakaan.
6. *trafficway_type*: Jenis jalan yang terlibat dalam kecelakaan.
7. *alignment*: Kelurusan atau kelengkungan jalan di lokasi kecelakaan.
8. *roadway_surface_cond*: Kondisi permukaan jalan saat kecelakaan.
9. *road_defect*: Kerusakan yang terdapat pada permukaan jalan.
10. *crash_type*: Jenis keseluruhan dari kecelakaan.
11. *intersection_related_i*: Apakah kecelakaan berhubungan dengan persimpangan.
12. *damage*: Tingkat kerusakan yang diakibatkan oleh kecelakaan.
13. *prim_contributory_cause*: Penyebab utama yang berkontribusi pada kecelakaan.
14. *num_units*: Jumlah kendaraan yang terlibat dalam kecelakaan.
15. *most_severe_injury*: Cedera paling parah yang terjadi dalam kecelakaan.
16. *injuries_total*: Total jumlah cedera yang dilaporkan.
17. *injuries_fatal*: Jumlah cedera yang menyebabkan kematian.
18. *injuries_incapacitating*: Jumlah cedera yang menyebabkan ketidakmampuan fisik.
19. *injuries_non_incapacitating*: Jumlah cedera yang tidak menyebabkan ketidakmampuan fisik.
20. *injuries_reported_not_evident*: Jumlah cedera yang dilaporkan tetapi tidak terlihat.

21. *injuries_no_indication*: Jumlah kasus tanpa indikasi cedera.
22. *crash_hour*: Jam terjadinya kecelakaan.
23. *crash_day of week*: Hari terjadinya kecelakaan.
24. *crash_month*: Bulan terjadinya kecelakaan.

3.5 Pre-processing Data

Akan dilakukan *data preprocessing* untuk memastikan bahwa data yang digunakan dalam proses pelatihan model berada dalam kondisi bersih dan siap digunakan. Langkah-langkah *data preprocessing* yang dilakukan adalah sebagai berikut:

3.5.1 Handling Missing Values

Langkah pertama adalah memeriksa nilai *null* pada setiap kolom dalam dataset. Data yang memiliki nilai *null* akan dihapus untuk menjaga kualitas data dan memastikan model menerima data yang lengkap dan konsisten.

3.5.2 Duplicate Detection and Removal

Dataset diperiksa untuk mendeteksi *duplicate entries*. Jika ditemukan data yang sama, baris tersebut akan dihapus agar tidak menyebabkan bias dalam proses pelatihan model.

3.5.3 Removal of Unusable Label

Pada tahap ini dilakukan proses pembersihan data berdasarkan nilai pada kolom *prim_ontributory_cause*. Ditemukan beberapa baris yang memiliki label "*UNABLE TO DETERMINE*", yang tidak memberikan informasi jelas mengenai penyebab kecelakaan. Langkah ini dilakukan untuk memastikan bahwa hanya data dengan label yang *valid* dan dapat diinterpretasikan oleh model yang digunakan dalam proses pelatihan dan pengujian.

3.5.4 Feature Filtering

Kolom *crash_ hour*, *crash_day_of_week* dan *crash_month* dihapus karena informasi yang terkandung di dalamnya seperti hari, bulan, tahun, dan waktu kejadian dianggap redundan dan berpotensi menyebabkan bias dalam proses prediksi karena informasi ini terdapat pada kolom *crash_date*. Faktor musim dan kondisi cuaca yang relevan telah direpresentasikan secara lebih spesifik pada kolom lain seperti *roadway surface condition*, *weather condition* dan *crash month*.

3.5.5 Categorization of Primary Contributory Cause

Kolom *prim_contributory_cause*, yang memiliki 39 kemungkinan nilai, akan diklasifikasikan menjadi dua kategori utama: *Human Error* (HE) dan *Non-Human Error* (Non-HE). Pendekatan ini bertujuan untuk membedakan kecelakaan yang disebabkan oleh kesalahan manusia dengan faktor eksternal yang berada di luar kendali pengemudi. Dengan melakukan kategorisasi ini, analisis dapat lebih terfokus pada dampak faktor lingkungan terhadap jenis kecelakaan lalu lintas. Penentuan kategori ini dilakukan secara manual berdasarkan interpretasi terhadap deskripsi dari masing-masing nilai pada kolom *prim_contributory_cause*. Nilai-nilai yang secara langsung berkaitan dengan perilaku atau kesalahan pengemudi, seperti *"FAILING TO YIELD RIGHT-OF-WAY "*, *"DISREGARDING TRAFFIC SIGNALS"*, dan *"EXCEEDING SAFE SPEED FOR CONDITIONS"*, dikategorikan sebagai Human Error. Sementara itu, nilai-nilai yang berhubungan dengan kondisi lingkungan, infrastruktur, atau hal-hal di luar kendali manusia, seperti *"WEATHER"*, *"OBSTRUCTED CROSSWALKS"*, dan *"ROAD ENGINEERING/SURFACE/MARKING DEFECTS"*, dikategorikan sebagai Non-Human Error. Tabel 3.2 berikut menunjukan seluruh kategori yang terdapat pada kolom *prim_contributory_cause*.

Tabel 3.2. Kategorisasi awal

Primary Contributory Cause	HE / NON-HE
UNABLE TO DETERMINE	-
FAILING TO YIELD RIGHT-OF-WAY	HE
FOLLOWING TOO CLOSELY	HE
DISREGARDING TRAFFIC SIGNALS	HE
IMPROPER TURNING/NO SIGNAL	HE

FAILING TO REDUCE SPEED TO AVOID CRASH	HE
IMPROPER OVERTAKING/PASSING	HE
DISREGARDING STOP SIGN	HE
IMPROPER LANE USAGE	HE
NOT APPLICABLE	NON-HE
DRIVING SKILLS/KNOWLEDGE/EXPERIENCE	HE
WEATHER	NON-HE
IMPROPER BACKING	HE
OPERATING VEHICLE IN ERRATIC MANNER	HE
VISION OBSCURED	NON-HE
DISTRACTION - FROM INSIDE VEHICLE	HE
DRIVING ON WRONG SIDE/WRONG WAY	HE
DISREGARDING OTHER TRAFFIC SIGNS	HE
EQUIPMENT - VEHICLE CONDITION	NON-HE
UNDER THE INFLUENCE OF DRUGS	HE
PHYSICAL CONDITION OF DRIVER	HE
DISTRACTION - FROM OUTSIDE VEHICLE	HE
EXCEEDING SAFE SPEED FOR CONDITIONS	HE
TURNING RIGHT ON RED	HE
EXCEEDING AUTHORIZED SPEED LIMIT	HE
DISREGARDING ROAD MARKINGS	HE
ROAD CONSTRUCTION/MAINTENANCE	NON-HE
EVASIVE ACTION	NON-HE
CELL PHONE USE OTHER THAN TEXTING	HE
ROAD ENGINEERING/SURFACE/MARKING DEFECTS	NON-HE
HAD BEEN DRINKING	HE
DISREGARDING YIELD SIGN	HE
DISTRACTION - OTHER ELECTRONIC DEVICE	HE
RELATED TO BUS STOP	NON-HE
TEXTING	HE
ANIMAL	NON-HE
OBSTRUCTED CROSSWALKS	NON-HE
BICYCLE ADVANCING LEGALLY ON RED LIGHT	NON-HE
PASSING STOPPED SCHOOL BUS	HE

MOTORCYCLE ADVANCING LEGALLY ON RED LIGHT	NON-HE

Pada kolom ini pelabelan yang masuk ke kategori non-HE terdapat 12 sedangkan 27 kategori yang masuk ke HE.

3.5.6 Encoding Categorical Data

Proses *encoding* dilakukan untuk mengubah data kategorikal menjadi bentuk numerik agar dapat digunakan oleh model yang dibangun.

Setelah seluruh tahapan *pre-processing* dilakukan, dataset yang semula terdiri dari 209.306 baris dan 24 kolom telah direduksi menjadi 150.965 baris dan 22 kolom. Sebanyak 21 kolom digunakan sebagai fitur untuk pelatihan model, sedangkan kolom *prim_contributory_cause* dijadikan sebagai label yang telah dikategorikan ke dalam kelas HE dan non-HE.

3.6 Pembangunan Model Awal

Pada tahapan ini, model akan dibangun menggunakan bahasa pemrograman Python menggunakan *IDE Jupyter Notebook*. Data akan di dibagi menjadi data *training* dan data *testing*. Model RF dan XGB akan dilatih dengan data *training* sebesar 80% dan diuji dengan data *testing* sebesar 20%. Proses pembagian dilakukan secara acak untuk memastikan bahwa data yang digunakan dalam pelatihan dan pengujian merepresentasikan keseluruhan distribusi data secara adil.

3.7 *Hyperparameter Tuning* dan Iterasi Model Kedua

Pada tahap ini dilakukan proses *hyperparameter tuning* yang bertujuan untuk menemukan kombinasi parameter yang optimal sehingga akurasi prediksi model dapat lebih maksimal. Dalam penelitian ini, teknik *grid search* digunakan untuk melakukan pencarian kombinasi parameter terbaik. *Grid search* bekerja dengan menguji seluruh kemungkinan kombinasi parameter yang ditentukan sebelumnya, lalu memilih kombinasi yang memberikan hasil evaluasi terbaik berdasarkan data latih. *Hyperparameter* yang digunakan pada model RF berupa *n_estimators, max_depth, min_samples_split, min_samples_leaf* dan *max_features* sedangkan *hyperparameter* yang digunakan pada model XGB yaitu *n_estimators,*

max_depth, learning_rate, subsample dan *colsample_bytree*. Berikut adalah penjelasan masing-masing *hyperparameter*:

(a) *n_estimators*: Menentukan jumlah pohon yang akan dibangun dalam model.

(b) *max_depth*: Menentukan kedalaman maksimum setiap pohon.

(c) *min_samples_split* (RF): Jumlah minimum sampel yang diperlukan untuk membagi node menjadi dua cabang.

(d) *min_samples_leaf* (RF): Jumlah minimum sampel yang harus ada di setiap daun pohon setelah pemisahan.

(e) *max_features* (RF): Menentukan jumlah fitur yang dipertimbangkan saat membagi node.

(f) *learning_rate* (XGB): Mengontrol seberapa besar kontribusi setiap pohon terhadap prediksi akhir.

(g) *subsample* (XGB): Proporsi data pelatihan yang digunakan untuk membangun setiap pohon.

(h) *colsample_bytree* (XGB): Proporsi fitur yang dipilih secara acak untuk membangun setiap pohon.

3.8 Evaluasi

Evaluasi model dilakukan dengan membandingkan metrik-metrik seperti *accuracy, precision, recall, dan F1-score* serta waktu komputasi dari masing-masing model. Proses evaluasi ini dilakukan terhadap data uji (20%) yang telah dipisahkan sebelumnya dari data latih (80%), untuk mengukur kemampuan generalisasi model. Setiap model yang telah dilatih akan melakukan prediksi terhadap data uji, dan hasil prediksi ini kemudian dibandingkan dengan label sebenarnya dengan nila-nilai dari *confusion matrix* yaitu *true positive* (TP), *true negative* (TN), *false positive* (FP), dan *false negative* (FN). Evaluasi dilakukan tidak hanya pada model dasar, tetapi juga pada model hasil *hyperparameter tuning* menggunakan *gridsearchcv* sehingga kita dapat mengamati perubahan performa model dari sisi akurasi. Juga dilakukan analisis *feature importance* untuk melihat kontribusi masing-masing fitur terhadap hasil prediksi. Berdasarkan hasil tersebut, dibuat iterasi ketiga dengan hanya menggunakan fitur-fitur teratas untuk melihat dampaknya terhadap performa dan efisiensi komputasi model.