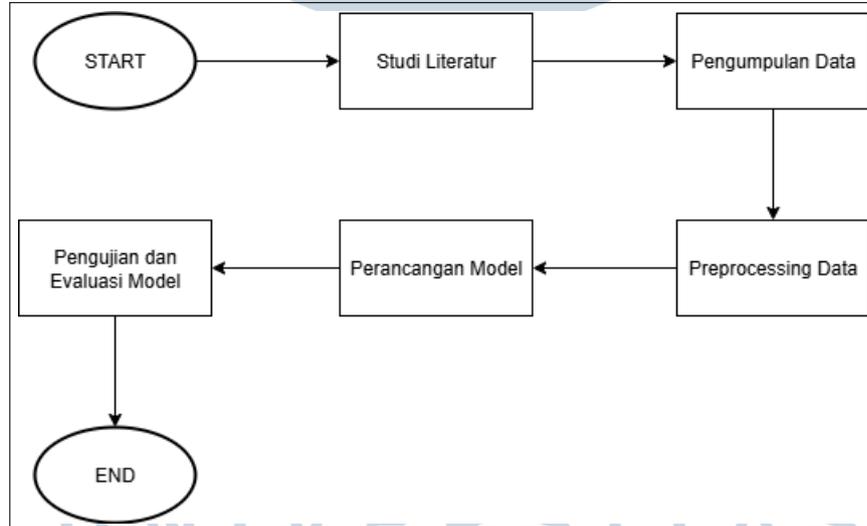


## BAB 3 METODOLOGI PENELITIAN

Bab ini membahas metodologi penelitian yang digunakan untuk menganalisis ujaran kebencian (*hate speech*) di media sosial X. Penelitian ini menggabungkan model bahasa BERT sebagai pembentuk representasi teks dengan XGBoost sebagai algoritma klasifikasi. Proses penelitian dilakukan secara sistematis, dimulai dari studi literatur, pengumpulan dan preprocessing data, perancangan sistem, implementasi model, pengujian, evaluasi, hingga pelaporan hasil.

Alur metodologi ditunjukkan pada Gambar 3.1. Data yang digunakan berasal dari dataset Okky Ibrohim [44], yang berisi tweet berbahasa Indonesia dan telah dilabeli oleh 30 anotator. Setelah melalui tahap preprocessing, data digunakan dalam perancangan dan implementasi sistem klasifikasi. Evaluasi dilakukan untuk menilai performa model, dan seluruh tahapan didokumentasikan dalam bentuk laporan penelitian



Gambar 3.1. Alur metodologi penelitian

### 3.1 Studi Literatur

Studi literatur dilakukan sebagai landasan untuk memahami secara komprehensif topik penelitian yang berkaitan dengan ujaran kebencian (*hate speech*) dan penerapan algoritma *machine learning* dalam mendeteksinya.

Penelitian ini meninjau berbagai jurnal ilmiah dan artikel, baik dari sumber nasional maupun internasional, dengan rentang waktu publikasi antara tahun 2016 hingga 2024. Fokus utama kajian tertuju pada fenomena *hate speech* yang terjadi di platform media sosial X, serta pendekatan-pendekatan teoritis dan teknis yang telah dikembangkan untuk deteksi ujaran kebencian. Kajian ini mencakup teori dasar mengenai deteksi *hate speech*, tahapan *text processing*, serta penerapan model *machine learning hybrid* BERT dan XGBoost, serta metode evaluasi kinerja model menggunakan Confusion Matrix.

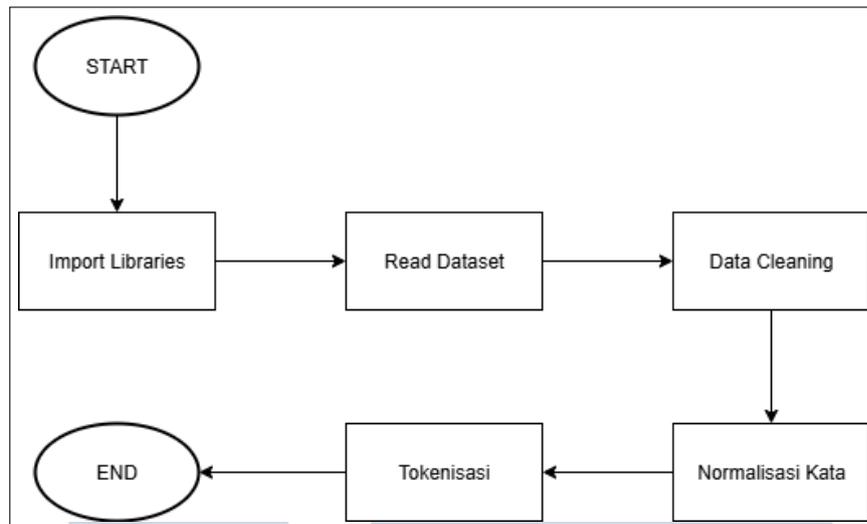
### 3.2 Pengumpulan Data

Pengumpulan data dalam penelitian ini dilakukan dengan menggabungkan dua sumber utama, yaitu dataset yang telah tersedia dari penelitian sebelumnya oleh Ibrohim dan Budi [44], serta data hasil *scraping* menggunakan *Twitter Search API*. Proses *scraping* dilakukan dengan menggunakan kata kunci dan frasa yang umum digunakan dalam ujaran kebencian dan bahasa kasar, yang telah diidentifikasi berdasarkan studi literatur terdahulu.

Data hasil *crawling* kemudian dikurasi dan dianotasi dalam dua tahap, yakni klasifikasi terhadap *hate speech* dan *abusive language*, serta anotasi lanjutan yang mencakup identifikasi target, kategori, dan tingkat intensitas ujaran kebencian. Untuk menjaga kualitas dan objektivitas data, proses anotasi dilakukan menggunakan pendekatan *crowdsourcing* melalui sistem berbasis web yang dirancang khusus. Sebanyak 30 anotator dengan latar belakang yang beragam dilibatkan dalam proses ini guna meningkatkan validitas dan keandalan hasil anotasi.

### 3.3 Preprocessing Data

Dataset yang digunakan telah melalui proses pelabelan oleh penyedia data berdasarkan anotasi manual menggunakan 30 anotator. Setelah pelabelan, data memasuki tahap preprocessing untuk memastikan kualitas dan kebersihan data sebelum digunakan sebagai input model XGBoost. Proses preprocessing ini mencakup pembersihan teks dari elemen tidak relevan, normalisasi kata, serta validasi ulang terhadap label yang diberikan. Seluruh tahapan preprocessing ditampilkan pada Gambar 3.2 dan dijelaskan secara detail berikut ini:



Gambar 3.2. Alur Preprocessing Dataset

Tahapan preprocessing pada Gambar 3.2 dapat dijelaskan lebih mendetail sebagai berikut:

Tahapan-tahapan *pre-processing* yang ditunjukkan pada Gambar 3.2 dapat dijelaskan sebagai berikut:

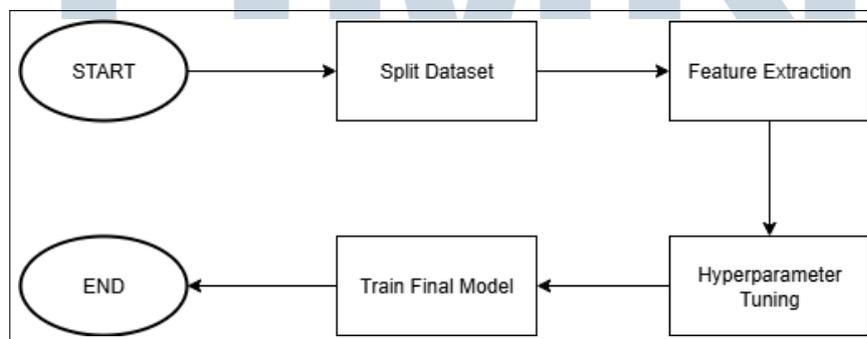
- **Import Libraries:** Tahapan awal berupa pemanggilan seluruh pustaka Python yang dibutuhkan dalam proses analisis data, mencakup pustaka untuk manipulasi data (seperti *pandas*), pemrosesan teks (seperti *re*, serta algoritma pemodelan (seperti *xgboost*) dan libraries lainnya.
- **Read Dataset:** Membaca data tweet yang telah diberi label oleh penyedia dataset. Dataset dimuat ke dalam struktur data yang sesuai, seperti *DataFrame*, agar dapat diproses lebih lanjut.
- **Data Cleaning:** Melakukan pembersihan terhadap konten teks tweet dengan menghapus URL, mention (@), hashtag (#), angka, emoji, simbol, karakter non-ASCII, serta elemen-elemen lain yang tidak relevan terhadap analisis ujaran kebencian.
- **Normalisasi Kata:** Mengonversi kata-kata tidak baku, seperti slang, singkatan, atau kesalahan penulisan, ke bentuk baku menggunakan kamus normalisasi. Proses ini juga mencakup *case folding* (mengubah semua huruf menjadi huruf kecil).

- **Tokenisasi:** Memecah teks tweet menjadi unit-unit kata atau *token*, yang merupakan bentuk representasi awal dari teks sebelum dilakukan ekstraksi fitur. Proses tokenisasi dilakukan menggunakan *BERT tokenizer* untuk menyesuaikan format input dengan kebutuhan model BERT, termasuk penambahan token khusus seperti [CLS] dan [SEP].

Tahapan *preprocessing* dilakukan secara sistematis untuk memastikan data bersih, dan konsisten. Penggunaan *BERT tokenizer* memungkinkan tokenisasi kontekstual yang mempertahankan struktur dan konten semantik teks, sehingga mendukung pembentukan representasi vektor yang optimal bagi model klasifikasi XGBoost.

### 3.4 Perancangan Model

Perancangan sistem dilakukan untuk membangun model klasifikasi yang mampu mendeteksi ujaran kebencian pada platform X secara otomatis. Penelitian ini menggunakan algoritma XGBoost (Extreme Gradient Boosting), yaitu metode ensemble learning berbasis pohon keputusan yang efektif untuk klasifikasi. Input model berupa representasi vektor hasil ekstraksi embedding dari BERT, yang diperoleh melalui proses tokenisasi menggunakan BERT tokenizer dan agregasi dengan mean pooling. Seluruh tahapan mulai dari preprocessing data, ekstraksi fitur, hingga pelatihan model dirancang secara sistematis untuk menghasilkan klasifikasi yang akurat. Alur lengkap perancangan sistem divisualisasikan dalam Gambar 3.3.



Gambar 3.3. Alur Perancangan Model

Langkah-langkah perancangan model yang ditunjukkan pada Gambar 3.3 dapat dijelaskan sebagai berikut:

- **Split Dataset:** Dataset dibagi menjadi data latih, data validasi dan data uji. Pembagian ini dilakukan untuk memisahkan data yang digunakan dalam pelatihan model dan data yang digunakan untuk evaluasi agar proses pengujian lebih objektif.
- **Feature Extraction:** Representasi fitur dari data dihasilkan pada tahap ini. Dalam konteks model BERT, setiap teks akan diubah menjadi representasi vektor berbasis embedding yang mencerminkan konteks semantik dari kalimat. Dalam memperoleh representasi satu vektor per kalimat, digunakan teknik *mean pooling* terhadap semua token, dengan mengecualikan token khusus seperti [CLS] dan [SEP]. Hasilnya adalah vektor berdimensi tinggi yang merepresentasikan konteks keseluruhan kalimat, dan digunakan sebagai fitur input pada tahap klasifikasi selanjutnya.
- **Hyperparameter Tuning:** Proses ini bertujuan untuk mencari kombinasi parameter terbaik dari algoritma XGBoost guna meningkatkan performa model. Parameter seperti *learning rate*, *max depth*, *n estimators*, *subsample*, dan *colsample bytree* disesuaikan secara manual berdasarkan hasil evaluasi terhadap data latih.
- **Train Final Model:** Model dilatih ulang menggunakan data latih dengan konfigurasi hyperparameter yang telah disesuaikan secara manual berdasarkan eksplorasi dari inisialisasi awal. Model ini merupakan versi final yang digunakan dalam tahap evaluasi maupun prediksi.

### 3.4.1 Split Dataset

Pembagian dataset dilakukan untuk memisahkan data ke dalam tiga subset, yaitu data latih (*training set*), data validasi (*validation set*), dan data uji (*testing set*) dengan rasio 70:15:15. Pembagian ini bertujuan untuk memastikan proses pelatihan model berlangsung optimal serta evaluasi performa model dilakukan secara objektif.

Data latih digunakan untuk melatih model, data validasi digunakan untuk menyetel parameter dan menghindari *overfitting*, sedangkan data uji digunakan untuk mengukur performa akhir model terhadap data yang belum pernah dilihat sebelumnya.

Pembagian dataset dilakukan secara acak dengan mempertimbangkan distribusi kelas melalui metode *stratified split*, sehingga proporsi kelas tetap

seimbang pada ketiga subset data. Implementasi teknis pembagian ini dilakukan menggunakan fungsi `train_test_split` dari pustaka `scikit-learn` secara bertahap, dengan parameter `stratify=y` untuk menjaga keseimbangan label.

### 3.4.2 Feature Extraction

Proses ekstraksi fitur dalam penelitian ini dilakukan dengan memanfaatkan model BERT (Bidirectional Encoder Representations from Transformers) sebagai *feature extractor*. BERT memiliki kemampuan memahami konteks kata dalam kalimat secara mendalam melalui mekanisme *self-attention*, sehingga dapat menghasilkan representasi teks yang lebih kaya secara semantik dibandingkan metode konvensional seperti *TF-IDF*.

Setiap komentar yang telah melalui split diubah menjadi vektor embedding melalui model *pre-trained* BERT. Dalam memperoleh representasi vektor dari suatu komentar secara keseluruhan, digunakan teknik *mean pooling*, yaitu dengan menghitung rata-rata dari semua token embeddings yang dihasilkan BERT. Vektor hasil *mean pooling* ini kemudian digunakan sebagai input fitur untuk model klasifikasi XGBoost.

Pendekatan ini memungkinkan sistem untuk memanfaatkan pemahaman konteks yang dihasilkan oleh BERT tanpa perlu melatih ulang seluruh model, sehingga efisien dalam hal komputasi namun tetap memberikan performa yang baik.

### 3.4.3 Hyperparameter Tuning

*Hyperparameter* merupakan parameter yang tidak dipelajari secara langsung oleh model dari data, melainkan harus ditentukan sebelum proses pelatihan. Pada model XGBoost, beberapa *hyperparameter* penting yang memengaruhi kompleksitas model dan performa akhir antara lain kedalaman maksimum pohon (`max_depth`), laju pembelajaran (`learning_rate`), serta jumlah pohon estimasi (`n_estimators`) [35]. Pemilihan nilai parameter yang tepat dapat meningkatkan kemampuan generalisasi model serta menghindari masalah *overfitting* maupun *underfitting*.

Penyesuaian *hyperparameter* secara sistematis dikenal sebagai *hyperparameter tuning*, yaitu proses pencarian kombinasi nilai optimal dari parameter-parameter tertentu untuk memaksimalkan kinerja model. Dibandingkan penyetalan manual yang bersifat subjektif, tuning otomatis menawarkan pendekatan

yang lebih terstruktur dan reproducible, sehingga menghasilkan performa model yang lebih stabil dan konsisten [45].

Metode tuning yang digunakan dalam penelitian ini adalah *Grid Search*, yaitu pendekatan yang mengevaluasi seluruh kombinasi parameter dalam ruang pencarian yang telah ditentukan secara eksplisit. Proses ini diimplementasikan menggunakan fungsi `GridSearchCV` dari pustaka `scikit-learn`, yang secara otomatis melakukan pencarian parameter terbaik dengan pendekatan *cross-validation*. Evaluasi setiap kombinasi parameter dilakukan berdasarkan metrik F1-score sebagai skor utama [46].

Meskipun *Grid Search* cenderung lebih lambat dibandingkan pendekatan seperti *Random Search*, metode ini menjamin bahwa seluruh kombinasi dalam ruang pencarian diuji, sehingga kemungkinan menemukan konfigurasi terbaik menjadi lebih tinggi. Dalam penelitian ini, ruang pencarian parameter dibatasi pada nilai-nilai yang umum digunakan untuk XGBoost, seperti  $\text{max\_depth} = \{4, 6, 8\}$ ,  $\text{learning\_rate} = \{0.01, 0.1, 0.2\}$ , dan  $\text{n\_estimators} = \{100, 200\}$ , agar waktu komputasi tetap efisien namun hasil tuning tetap optimal.

### 3.5 Pengujian dan Evaluasi

Tahap pengujian dan evaluasi pada model penelitian digunakan untuk menilai tingkat kinerja model dalam klasifikasi tweet ke dalam kategori yang sesuai khususnya dalam konteks ujaran kebencian. Evaluasi performa model dilakukan menggunakan metode *confusion matrix*. Confusion matrix memperlihatkan jumlah prediksi yang benar maupun salah dalam tiap kategori, terdiri atas empat komponen utama: *True Positive (TP)*, *True Negative (TN)*, *False Positive (FP)*, dan *False Negative (FN)*. Berdasarkan nilai-nilai ini, sejumlah metrik performa utama dapat dihitung sebagai berikut:

- **Accuracy** mengukur proporsi prediksi yang benar dari seluruh prediksi.
- **Precision** menilai seberapa banyak dari prediksi positif yang benar-benar positif.
- **Recall** atau sensitivitas menunjukkan sejauh mana model berhasil mengenali data positif.
- **F1-Score** merupakan rata-rata harmonik dari precision dan recall, digunakan untuk memberikan keseimbangan antara keduanya terutama pada kasus data

yang tidak seimbang.

Evaluasi ini bersifat kuantitatif, bertujuan untuk mengukur sejauh mana model mampu melakukan klasifikasi dengan benar. Seluruh perhitungan metrik dilakukan secara otomatis menggunakan fungsi evaluasi bawaan dari pustaka `scikit-learn`, yang telah teruji dalam banyak penelitian *Natural Language Processing* dan klasifikasi teks. Selain itu, digunakan juga visualisasi confusion matrix untuk memberikan gambaran distribusi prediksi model secara intuitif.

