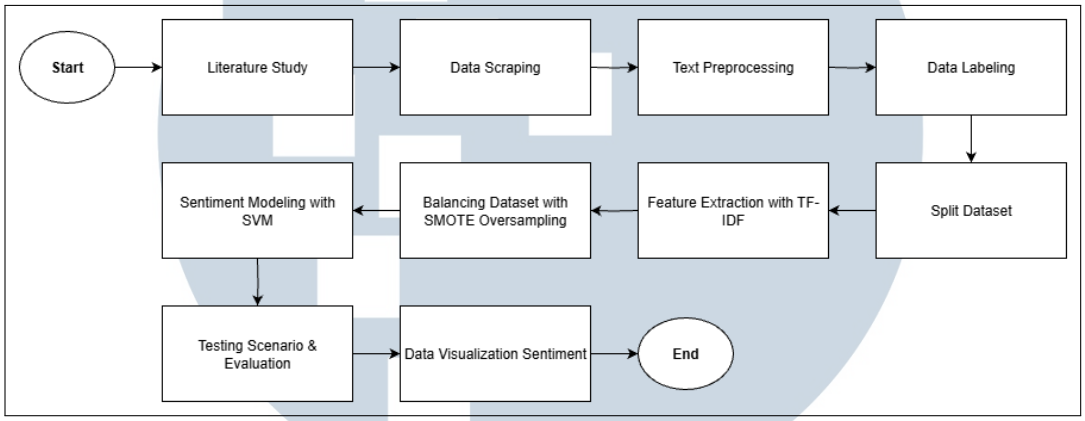
BAB 3
METODOLOGI PENELITIAN

Penelitian ini mengikuti alur yang terstruktur dan sistematis untuk memastikan setiap tahapan proses berjalan secara optimal. Berikut merupakan tahapan-tahapan dalam proses penelitian yang dilakukan.



Gambar 3.1. Alur Metodologi Penelitian

Gambar 3.1 menunjukkan alur proses penelitian yang dilaksanakan secara sistematis. Tahap pertama adalah *Literature Study* untuk memperoleh dasar teori yang relevan mengenai analisis sentimen, media sosial, serta algoritma SVM. Selanjutnya dilakukan *Data Scraping*, yaitu proses pengumpulan data dari platform media sosial X menggunakan *Tweet-Harvest*. Data yang diperoleh kemudian diproses melalui tahap *Text Preprocessing* untuk membersihkan dan menyiapkan teks mentah menjadi data yang dapat dianalisis.

Langkah berikutnya adalah *Data Labeling*, yaitu pemberian label sentimen terhadap setiap *tweet* secara manual ke dalam kategori positif, netral, atau negatif. Setelah pelabelan, dilakukan *Feature Extraction using TF-IDF* untuk mengubah data teks menjadi bentuk numerik. Berikutnya dilakukan opsi untuk *balancing dataset* untuk menguji nantinya apakah model akan menunjukkan hasil yang lebih baik jika menggunakan data yang seimbang.

Tahapan berikutnya adalah *Sentiment Modeling with Support Vector Machine (SVM)*, yaitu proses pembangunan model klasifikasi untuk menganalisis sentimen berdasarkan fitur-fitur yang telah dipilih. Proses ini mencakup pelatihan, pengujian, serta *tuning model*. Seluruh rangkaian proses ini bertujuan untuk menghasilkan sistem analisis sentimen yang akurat dan dapat digunakan sebagai

acuan penelitian lebih lanjut.

3.1 Literature Study

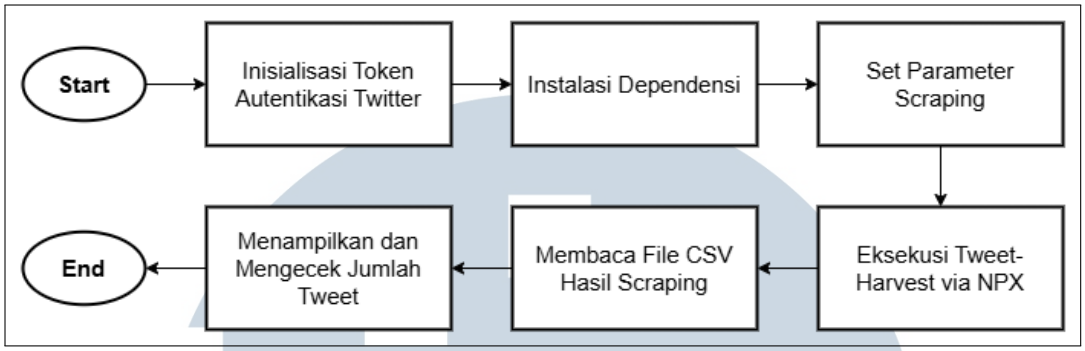
Studi literatur dilakukan sebagai landasan untuk memahami secara komprehensif topik penelitian yang berkaitan dengan analisis sentimen terhadap tagar #KaburAjaDulu dan penerapan algoritma *machine learning* dalam mengklasifikasikannya. Penelitian ini meninjau berbagai jurnal ilmiah dan artikel, baik dari sumber nasional maupun internasional, dengan rentang waktu publikasi antara tahun 2016 hingga 2024.

Fokus utama kajian ini adalah persepsi publik terhadap isu yang diangkat melalui tagar #KaburAjaDulu di media sosial X, yang merepresentasikan fenomena migrasi ke luar negeri sebagai respons terhadap kondisi sosial dan ekonomi di Indonesia. Kajian ini menyoroti bagaimana opini publik terbentuk dan tersebar secara daring, serta bagaimana pendekatan analisis sentimen dapat digunakan untuk memahami sikap masyarakat terhadap isu tersebut.

Literatur yang dikaji mencakup teori dasar analisis sentimen, tahapan *text preprocessing*, serta penerapan algoritma *Support Vector Machine* (SVM) dalam klasifikasi teks. Selain itu, dibahas pula teknik ekstraksi fitur menggunakan *Term Frequency-Inverse Document Frequency* (TF-IDF) dan evaluasi kinerja model melalui *Confusion Matrix* serta metrik klasifikasi seperti *accuracy*, *precision*, *recall*, dan *F1-score*. Studi literatur ini menjadi landasan teoritis yang mendukung pemodelan sentimen terhadap opini publik terkait tagar #KaburAjaDulu.

3.2 Data Scraping

Metode *data scraping* dilakukan dengan menggunakan Tweet-Harvest, sebuah *library open-source* berbasis *Node.js* yang dirancang untuk pengambilan data *tweet* secara edukatif melalui antarmuka web media sosial X (sebelumnya Twitter). Proses ini dijalankan menggunakan perintah `npx` tanpa perlu instalasi permanen. Untuk mendukung proses *scraping*, digunakan beberapa *tools* tambahan seperti *Node.js* sebagai lingkungan eksekusi dan *Pandas* untuk membaca serta mengelola data hasil *scraping*. Proses pada Gambar 3.2 dapat dilakukan pada platform seperti Google Colab yang mendukung eksekusi perintah berbasis *command line*.

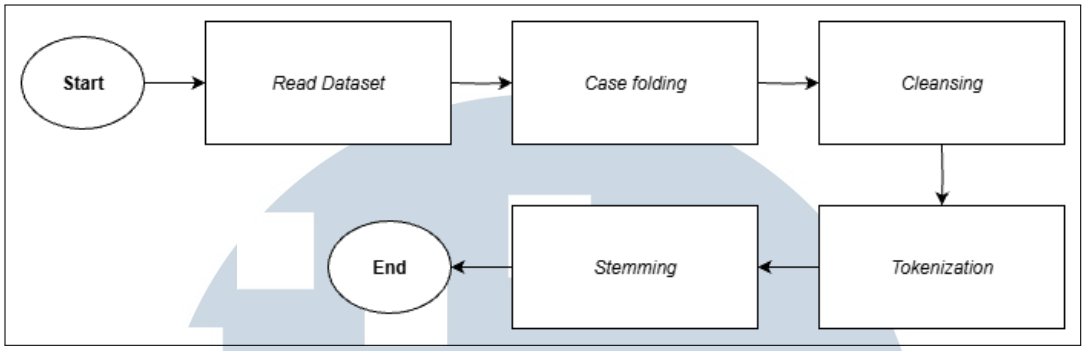


Gambar 3.2. Alur *Data Scraping*

Gambar 3.2 Menunjukan data dikumpulkan secara manual melalui empat kali percobaan selama 1–2 hari dalam periode antara 1 Januari hingga 30 April 2025 dengan menggunakan kata kunci ”#KaburAjaDulu” atau ”kabur aja dulu”. Proses pengambilan difokuskan pada *tweet* berbahasa Indonesia yang relevan dengan topik penelitian. Hasilnya, sebanyak 4000 *tweet* berhasil diperoleh dan disimpan dalam format `CSV` untuk selanjutnya digunakan dalam tahap *preprocessing* dan analisis lebih lanjut.

3.3 Text Preprocessing

Text preprocessing data merupakan tahap penting dalam analisis sentimen untuk mengubah data teks mentah menjadi bentuk yang lebih bersih dan terstruktur. Dalam penelitian ini, data yang digunakan berasal dari *platform* media sosial X yang cenderung mengandung banyak elemen tidak relevan seperti *mention*, tautan, dan simbol-simbol khusus. Oleh karena itu, dilakukan pembersihan atau *cleansing* terhadap elemen-elemen tersebut agar tidak memengaruhi hasil analisis. Namun, tagar seperti ”#KaburAjaDulu” tetap dipertahankan karena merupakan objek utama yang menjadi fokus penelitian ini.



Gambar 3.3. Alur *Text Preprocessing*

Pada Gambar 3.3 merupakan langkah-langkah *text preprocessing* yang meliputi:

1. *Case Folding*: Mengubah seluruh teks menjadi huruf kecil.
2. *Cleansing*: Menghapus karakter non-alfabet, tanda baca, URL, *mention*, *hashtag*, angka, dan *emoji*.
3. *Tokenizing*: Memecah kalimat menjadi kata-kata (*token*).
4. *Stemming*: Mengubah kata menjadi bentuk dasarnya menggunakan *library* seperti *Sastrawi*.

Tahapan *cleansing* diawali dengan menghapus karakter non-alfabet seperti tanda baca, simbol, URL (contoh: https://x.com/...), dan *mention* pengguna (contoh: `@username`). Penghapusan dilakukan secara selektif agar tidak menghilangkan kata kunci yang relevan, seperti tagar utama. Hal ini penting karena tagar tersebut tidak hanya menjadi alat pencarian, tetapi juga konteks dari opini yang disampaikan oleh pengguna.

Setelah tahap *cleansing*, dilakukan proses *case folding* untuk menyamakan seluruh teks menjadi huruf kecil guna menghindari redundansi kata akibat perbedaan kapitalisasi. Selanjutnya dilakukan *tokenization* untuk memisahkan kalimat menjadi kata-kata, *stopword removal* untuk menghilangkan kata-kata tidak bermakna, dan *stemming* untuk mengubah kata menjadi bentuk dasarnya menggunakan *library* seperti *Sastrawi*. Semua tahapan ini memastikan data siap digunakan dalam tahap *labeling* dan klasifikasi sentimen dengan hasil yang lebih akurat dan konsisten.

3.4 Data Labeling

Pelabelan data sentimen dalam penelitian ini dilakukan secara otomatis menggunakan pendekatan *lexicon-based* dengan daftar kata sentimen dalam Bahasa Indonesia. Setiap unggahan dianalisis berdasarkan kemunculan kata-kata bernilai positif, negatif, atau netral untuk menentukan kelas sentimennya. Metode ini dipilih karena efisien dan telah banyak digunakan dalam penelitian analisis sentimen pada teks berbahasa Indonesia (Wicaksono & Purwarianti, 2017).

Untuk memverifikasi ketepatan pelabelan otomatis tersebut, peneliti melakukan validasi manual terhadap 20 data yang dipilih secara acak dari korpus. Proses validasi ini bertujuan mengecek kesesuaian antara label yang dihasilkan oleh sistem dengan penilaian manusia berdasarkan konteks kalimat. Hasil pengecekan menunjukkan tingkat kesesuaian yang cukup tinggi, sehingga pendekatan *lexicon-based* dianggap cukup representatif untuk digunakan sebagai label pada proses pelatihan model klasifikasi.

Pelabelan diawali dengan tokenisasi setiap *tweet* yang telah dipra-proses, kemudian token-token tersebut dicocokkan dengan entri dalam kamus sentimen SentiStrength-ID. Jika kata ditemukan, skor sentimen diambil sesuai nilai yang tersedia. Jika tidak ditemukan, kata diabaikan atau dianggap netral. *Library* ini juga mendukung pengenalan terhadap kata negasi seperti "tidak" dan *booster* seperti "sangat" untuk menyesuaikan kekuatan sentimen dari kata-kata yang mengikutinya.

Selanjutnya, dilakukan penghitungan skor total sentimen untuk masing-masing *tweet*. Skor positif dan negatif dijumlahkan dengan memperhatikan pengaruh dari kata negasi (seperti "tidak") dan *booster* (seperti "sangat") yang dapat mengubah kekuatan atau arah sentimen. Untuk tagar #KaburAjaDulu, skor dapat ditambahkan secara manual jika dianggap memiliki muatan sentimen tertentu berdasarkan konteks. Setelah skor akhir ditentukan, *tweet* akan diklasifikasikan sebagai "positif" jika skor total > 0, "negatif" jika < 0, dan "netral" jika nilainya sama dengan 0.

3.5 Split Dataset

Split dataset adalah proses membagi kumpulan data menjadi beberapa bagian untuk keperluan pelatihan (*training*), validasi (*validation*), dan pengujian (*testing*) model *machine learning*. Tujuan dari pembagian ini adalah agar model dapat belajar dari sebagian data, lalu diuji pada data yang belum pernah dilihat

sebelumnya untuk menilai performanya secara objektif.

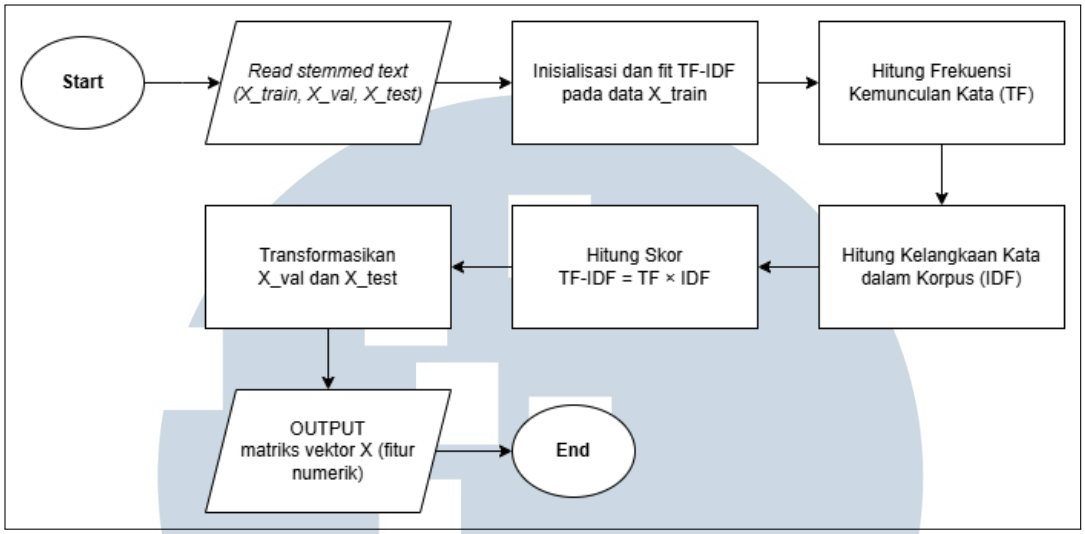
Biasanya, data dibagi ke dalam tiga *subset*, yaitu:

1. Data *training*: digunakan untuk melatih model agar mengenali pola dari data.
2. Data *validation*: digunakan untuk mengatur parameter model atau melakukan *tuning* agar tidak *overfitting* atau *underfitting*.
3. Data *testing*: digunakan untuk mengevaluasi kinerja akhir model setelah pelatihan dan *tuning* selesai.

Dalam penelitian ini, dataset dibagi ke dalam tiga skema proporsi: 60:20:20, 70:15:15, dan 80:10:10 untuk *training*, *validation*, dan *testing*. Tujuan dari variasi ini adalah untuk mengevaluasi pengaruh proporsi data latih terhadap performa klasifikasi, serta mengamati bagaimana *trade-off* antara ukuran data pelatihan dan data uji memengaruhi generalisasi model. Skema-skema tersebut merupakan praktik umum yang digunakan dalam eksperimen *machine learning* untuk membandingkan kestabilan model terhadap variasi ukuran data [25, 26, 27]. Selain itu, Sammut dan Webb mencatat bahwa skema 60:20:20 dan 70:15:15 adalah pembagian yang banyak digunakan di berbagai eksperimen pembelajaran mesin [27].

3.6 Feature Extraction using TF-IDF

Pada tahap ini, dilakukan proses ekstraksi fitur (*feature extraction*) menggunakan metode *TF-IDF* (T*erm Frequency - Inverse Document Frequency*). Tujuan dari tahapan ini adalah untuk memberikan bobot pada setiap kata serta mengubah data kategorikal berupa teks menjadi representasi numerik yang dapat digunakan dalam pemodelan *machine learning*.



Gambar 3.4. Alur *Feature Extraction using TF-IDF*

Gambar 3.4 Menunjukan metode *TF-IDF* bekerja dengan dua komponen utama. *Term Frequency* (TF) mengukur seberapa sering sebuah kata muncul dalam sebuah dokumen, kata yang muncul lebih sering akan memiliki bobot lebih tinggi dalam konteks dokumen tersebut. Sementara itu, *Inverse Document Frequency* (IDF) digunakan untuk menilai seberapa unik atau jarangnya kata tersebut di seluruh korpus dokumen; kata yang sering muncul di banyak dokumen akan diberi bobot lebih rendah karena dianggap kurang informatif.

Nilai gabungan TF dan IDF dari setiap kata digunakan untuk membentuk vektor representasi dokumen, di mana setiap elemen vektor mencerminkan bobot relatif dari suatu kata terhadap dokumen tersebut. Data yang diproses merupakan data kategorikal dari teks yang telah melalui tahap *preprocessing* (seperti *tokenisasi*, dan *stemming*), sehingga menghasilkan variabel independen berbobot X yang siap digunakan dalam tahap pemodelan selanjutnya.

3.7 SMOTE Oversampling

Setelah proses ekstraksi fitur dilakukan menggunakan metode TF-IDF, langkah selanjutnya yang direncanakan adalah penanganan data tidak seimbang (*imbalanced dataset*). Ketidakseimbangan kelas dapat menyebabkan model klasifikasi menghasilkan prediksi yang bias terhadap kelas mayoritas. Oleh karena itu, perlu dilakukan strategi khusus untuk menyeimbangkan jumlah data antarkelas sebelum masuk ke tahap pelatihan model.

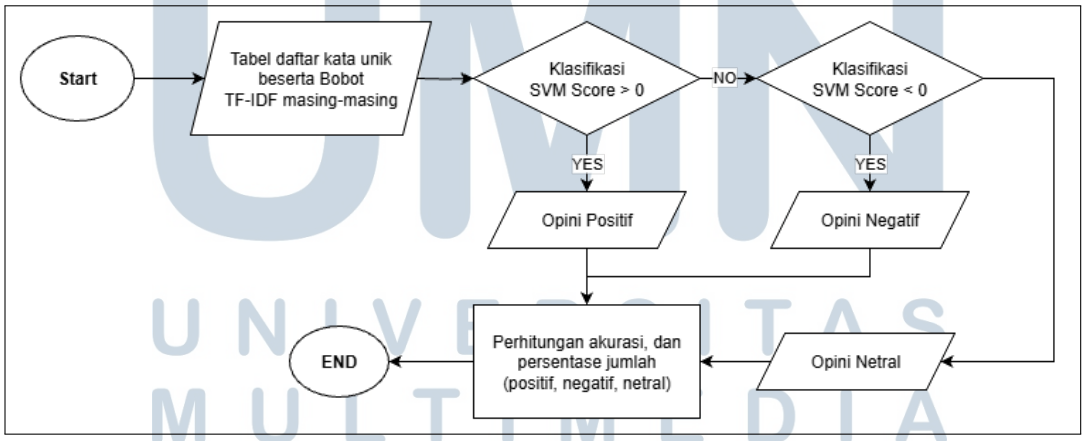
Untuk mengatasi permasalahan tersebut, dalam penelitian ini direncanakan

penggunaan teknik *SMOTE* (*Synthetic Minority Over-sampling Technique*). *SMOTE* bertujuan menghasilkan data sintetis pada kelas minoritas dengan cara membuat interpolasi antara sampel yang ada dan tetangga terdekatnya. Proses *oversampling* ini akan diterapkan hanya pada data *training* agar tidak memengaruhi validitas evaluasi model. Dengan penerapan *SMOTE*, diharapkan model klasifikasi dapat dilatih dengan data yang lebih seimbang, sehingga performa model menjadi lebih optimal dan adil dalam mengenali seluruh kelas.

Selain itu, karena penelitian ini menggunakan algoritma *Support Vector Machine* (*SVM*), penting untuk memastikan bahwa data dalam kondisi seimbang. Hal ini dikarenakan *SVM* sangat sensitif terhadap distribusi data antar kelas, dan kinerja model cenderung menurun jika terjadi ketidakseimbangan. Oleh sebab itu, proses penyeimbangan data menjadi langkah krusial dalam mencapai hasil klasifikasi yang akurat dan anda.

3.8 Sentiment Modeling with Support Vector Machine (SVM)

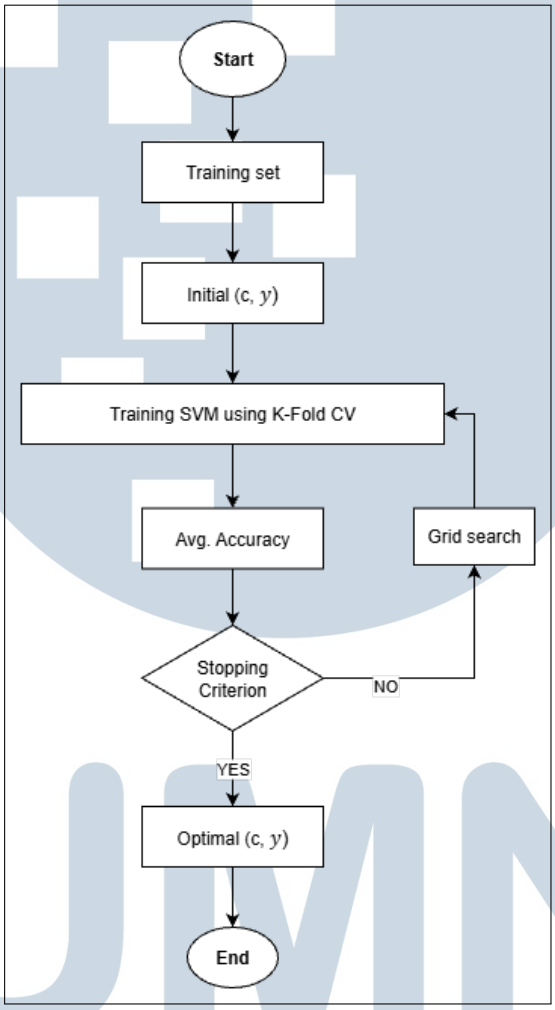
Setelah fitur teks dikonversi ke dalam bentuk vektor menggunakan *Term Frequency–Inverse Document Frequency* (TF-IDF), tahap selanjutnya adalah membangun model klasifikasi sentimen menggunakan algoritma *Support Vector Machine (SVM)*. *Flowchart SVM* digambarkan pada Gambar 3.5, untuk menunjukkan cara kerja *SVM* pada model ini.



Gambar 3.5. *Flowchart* SVM

Setelah mengetahui cara kerja SVM seperti pada Gambar 3.5. Proses dimulai dengan penentuan bobot kelas (*class weight*) untuk mengatasi masalah ketidakseimbangan distribusi antar label sentimen. Tiga skema pembobotan

diuji, yaitu tanpa bobot (*default*), bobot otomatis berdasarkan distribusi data (*balanced*), dan bobot manual yang disesuaikan berdasarkan proporsi kelas dalam data pelatihan. Pendekatan pembobotan ini bertujuan agar model tidak cenderung memprioritaskan kelas mayoritas dan tetap adil dalam memprediksi kelas minoritas.



Gambar 3.6. *Flowchart* SVM with *Grid Search*

Pada Gambar 3.6 langkah berikutnya adalah menentukan ruang pencarian parameter (*parameter grid*) yang akan digunakan dalam proses pencarian konfigurasi terbaik. Parameter yang disesuaikan meliputi nilai regulasi (*C*), pengaturan bobot kelas, serta batas maksimum iterasi pelatihan (*max_iter*). Nilai-nilai ini diuji menggunakan teknik pencarian grid (*Grid Search*) dengan validasi silang sebanyak lima lipatan (*5-fold cross-validation*), dan evaluasi performa dilakukan menggunakan metrik *f1-weighted*, yang sesuai untuk data tidak seimbang.

Setelah konfigurasi parameter terbaik ditemukan, model dengan parameter tersebut dipilih sebagai model akhir. Model kemudian dievaluasi pada data *validation set* dan *testing set* untuk mengetahui performa generalisasi terhadap data yang belum pernah dilihat sebelumnya. Evaluasi dilakukan berdasarkan empat metrik umum dalam klasifikasi, yaitu *precision*, *recall*, *f1-score*, dan *accuracy*. Selain itu, laporan klasifikasi dan *confusion matrix* digunakan untuk memperkuat interpretasi performa pada tiap kelas.

Melalui rangkaian tahapan ini, diperoleh model klasifikasi sentimen berbasis SVM yang optimal untuk dataset yang digunakan dalam penelitian. Tahapan ini juga memungkinkan perbandingan performa lintas skenario *preprocessing* serta mendukung pencapaian tujuan penelitian dalam menilai efektivitas model klasifikasi otomatis terhadap opini publik di media sosial.

3.9 Testing Scenario

Setelah model klasifikasi sentimen dibangun dan dikonfigurasi dengan parameter terbaik menggunakan algoritma *Support Vector Machine* (SVM), tahap berikutnya dalam metodologi adalah melakukan pengujian skenario untuk mengevaluasi performa model secara menyeluruh. Pengujian ini bertujuan untuk mengidentifikasi konfigurasi model dan praproses data yang mampu menghasilkan hasil klasifikasi terbaik.

Pengujian dilakukan dengan membandingkan performa model berdasarkan empat metrik utama, yaitu *accuracy*, *precision*, *recall*, dan *f1-score*. Keempat metrik ini dipilih karena mampu memberikan gambaran yang komprehensif mengenai kemampuan model dalam mengklasifikasikan data secara akurat dan seimbang, terutama dalam konteks data yang memiliki distribusi label yang tidak merata.

3.10 Data Visualization

Pada tahap ini, dilakukan proses visualisasi data dengan memanfaatkan dua *library* utama, yaitu *WordCloud* dan *Matplotlib*. Visualisasi menggunakan *Matplotlib* ditampilkan dalam bentuk diagram *pie* untuk menunjukkan distribusi atau proporsi jumlah data pada setiap kategori label sentimen, sehingga memudahkan dalam melihat dominasi masing-masing sentimen secara keseluruhan.

Sementara itu, visualisasi menggunakan *WordCloud* menyajikan frekuensi

kemunculan kata dalam bentuk grafis yang menarik, di mana ukuran kata dalam gambar mencerminkan seberapa sering kata tersebut muncul dalam data. Semakin besar ukurannya, semakin tinggi pula frekuensinya. *WordCloud* dibuat secara terpisah untuk masing-masing label sentimen, sehingga memudahkan dalam mengamati karakteristik kata yang sering muncul pada sentimen positif, negatif, maupun netral.

Melalui visualisasi ini, dilakukan pula analisis lebih lanjut untuk mengidentifikasi keterkaitan antara kata-kata tertentu dengan label sentimen yang menyertainya. Analisis ini berguna untuk memahami pola bahasa yang khas pada setiap kategori sentimen dan menjadi dasar dalam pengembangan model klasifikasi yang lebih akurat.