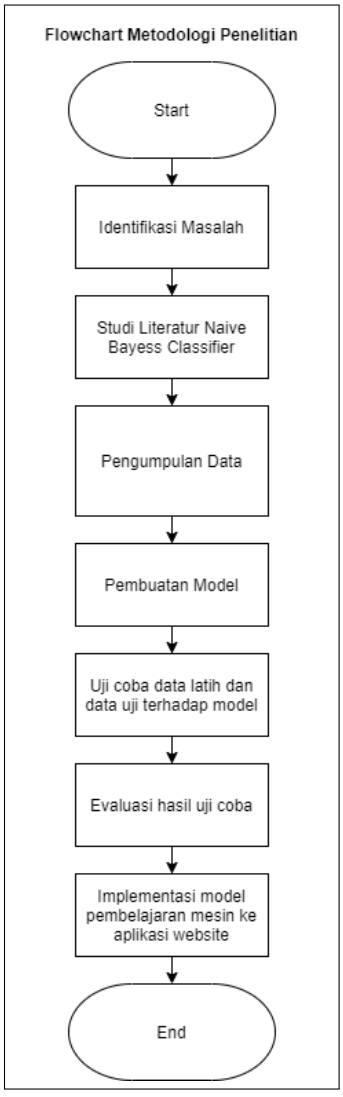
BAB 3
METODE PENELITIAN

Gambar 3.1 menunjukkan secara rinci alur tahapan yang diterapkan dalam penelitian ini, mulai dari langkah awal hingga proses akhir. Diagram tersebut bertujuan untuk memberikan pemahaman yang jelas mengenai prosedur yang diikuti dalam penelitian ini, serta bagaimana setiap tahap saling terkait dalam mencapai tujuan akhir penelitian.



Gambar 3.1. *Research Methodology Flowchart*

3.1 Identifikasi Masalah

Pada tahap ini, dilakukan wawancara dengan M. Niknik, yang menjabat sebagai dosen sekaligus koordinator penelitian pada proyek U-Tapis. Selain sebagai dosen, beliau juga merupakan seorang ahli bahasa yang sering terlibat dalam kasus-kasus linguistik forensik. Dalam wawancaranya, M. Niknik menyampaikan bahwa sistem pelaporan *harassment* di Indonesia telah mengalami kemajuan, di mana pelapor dapat mengakses website untuk melaporkan kasus tersebut, dan kemudian kasus tersebut akan diputuskan apakah akan ditindaklanjuti atau tidak. Namun, meskipun sistem ini dirancang untuk mempercepat proses, ternyata sistem tersebut justru membebani profesionalitas kerja ahli bahasa. Hal ini disebabkan oleh adanya batasan waktu tiga jam untuk memutuskan apakah kasus tersebut dapat dilanjutkan ke proses hukum atau tidak. Dalam rentang waktu tiga jam tersebut, pihak kepolisian akan menghubungi ahli bahasa untuk meminta pendapat mengenai apakah suatu pernyataan termasuk dalam kategori *harassment*. Kendala lainnya adalah terkadang ahli bahasa harus dihubungi di luar jam kerja yang wajar. Oleh karena itu, perlu adanya perancangan sistem yang dapat membantu memutuskan secara otomatis apakah suatu kalimat termasuk dalam kategori *harassment* atau tidak, guna mendukung kelancaran proses hukum dan mengurangi beban kerja ahli bahasa.

3.2 Studi Literatur

Pada tahap studi literatur, dikumpulkan landasan ilmu praktikal serta teoritis dalam upaya mendukung pengembangan model NLP untuk melakukan deteksi *harassment* hujatan. Literatur yang digunakan diantaranya pengertian *Harassment*, pengertian hinaan, pencemaran nama baik, NLP, *supervised learning*, *text preprocessing*, dan algoritma *Naive Bayes Classifier*.

3.3 Pengumpulan Data

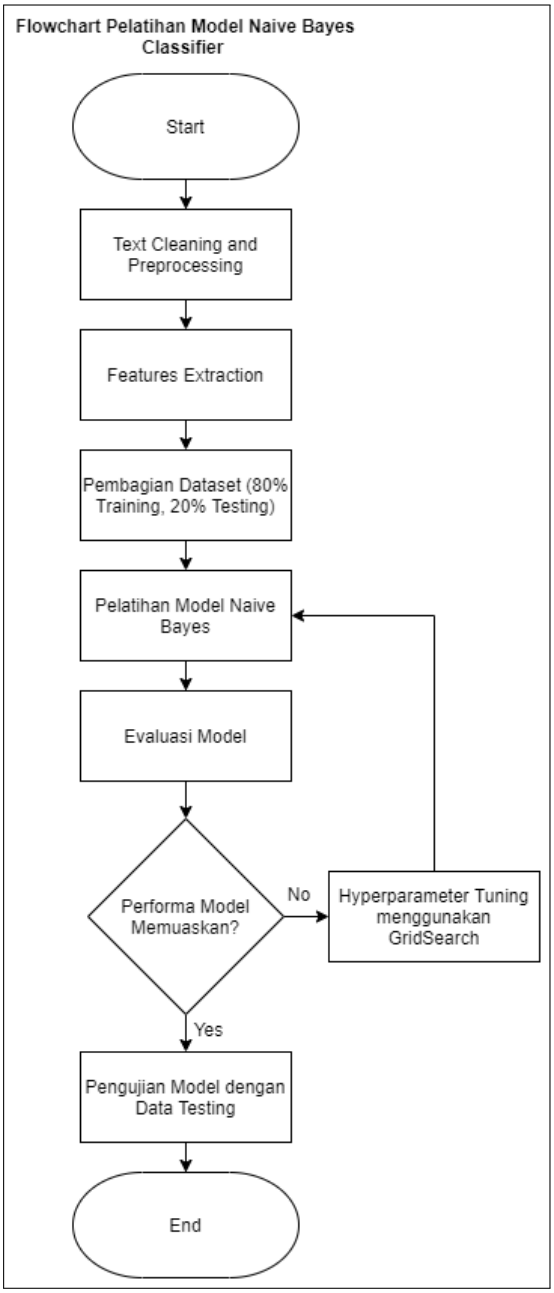
Pengumpulan data dalam penelitian ini melibatkan 18.000 kalimat yang dihasilkan oleh model bahasa ChatGPT menggunakan metode *query* yang spesifik. Kalimat-kalimat ini kemudian dikategorikan ke dalam tiga kelompok, yakni kalimat yang mengandung hinaan, pencemaran nama baik, dan yang netral (tidak mengandung *harassment*). *Labeling* data dilakukan oleh ChatGPT dengan mengacu pada aturan, peraturan, dan definisi yang telah ditentukan sebelumnya. Definisi ini

didasarkan pada literatur yang relevan, contoh-contoh kasus yang tervalidasi oleh ahli, serta Undang-Undang Informasi dan Transaksi Elektronik (UU ITE) yang berlaku di Indonesia. Pendekatan ini memastikan bahwa data yang dihasilkan sesuai dengan konteks dan batasan makna *harassment*, khususnya untuk kategori penghinaan dan pencemaran nama baik.

Penggunaan data sintetis yang dihasilkan oleh ChatGPT memiliki beberapa keunggulan yang telah diakui dalam literatur. Pertama, data ini membantu mengatasi keterbatasan data asli yang sulit diperoleh karena alasan privasi atau hukum. Kedua, fleksibilitas dalam mendesain *query* memungkinkan peneliti untuk memastikan bahwa data yang dihasilkan tetap relevan dan sesuai dengan tujuan penelitian. Penelitian seperti oleh Xu Guo dan Yiqiang Chen (2024) menunjukkan bahwa data sintetis dari generative AI dapat secara efektif menggantikan data asli untuk tugas-tugas spesifik [38]. Dalam konteks NLP, Ghanadian et al. (2024) menggunakan data sintetis yang dihasilkan oleh model besar seperti ChatGPT untuk mendeteksi ide bunuh diri, yang secara signifikan meningkatkan performa model [39].

3.4 Pembuatan Model

Gambar 3.2 merupakan alur *flowchart* perancangan model *Naive Bayes Classifier*, dari tahap pembagian dataset hingga pengujian model dengan data *testing*, proses perancangan model termasuk *hyperparameter tuning* untuk memaksimalkan model yang dirancang.



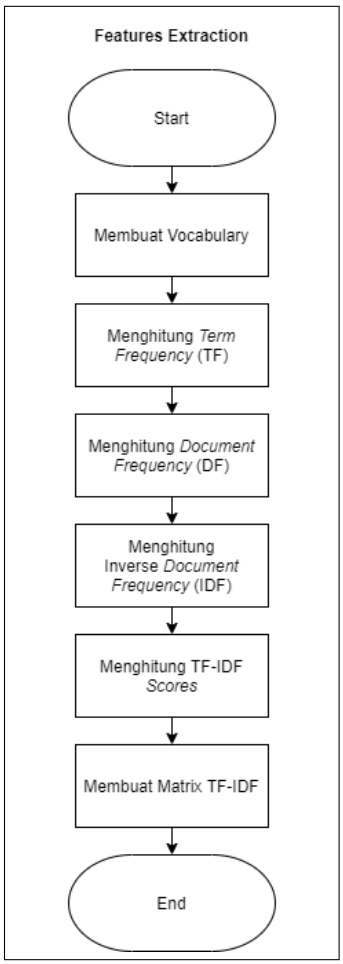
Gambar 3.2. *Model Training Flowchart*

3.4.1 *Text Cleaning and Preprocessing*

Langkah pertama dalam penelitian ini adalah *text cleaning*, yang mencakup beberapa tahapan: mengubah semua huruf menjadi huruf kecil (*lowercase*), menghapus tanda baca (*punctuation*), angka, dan spasi berlebih yang tidak relevan untuk analisis lebih lanjut. sehingga variasi kata yang tidak diperlukan dapat dihilangkan. Langkah-langkah ini memastikan teks menjadi bersih dan terstruktur,

siap untuk dianalisis oleh model pendeteksian *harassment* hujatan pada tahap berikutnya.

3.4.2 *Feature Extraction*



Gambar 3.3. *Features Extraction Flowchart*

Sesuai dengan gambar 3.3, pada tahap ini fitur teks diekstraksi menggunakan *TF-IDF Vectorization* untuk mengukur pentingnya kata dalam sebuah dokumen relatif terhadap korpus. Proses dimulai dengan membangun *vocabulary* dari teks yang telah diproses (tokenisasi). Selanjutnya, *Term Frequency* (TF) dihitung dengan menentukan frekuensi kemunculan setiap kata dalam dokumen dan menormalisasinya. Setelah itu, *Document Frequency* (DF) dicatat untuk menghitung jumlah dokumen yang mengandung kata tertentu. Dengan menggunakan DF, nilai *Inverse Document Frequency* (IDF) dihitung untuk menekankan kata-kata yang jarang muncul di korpus. Skor TF-IDF diperoleh

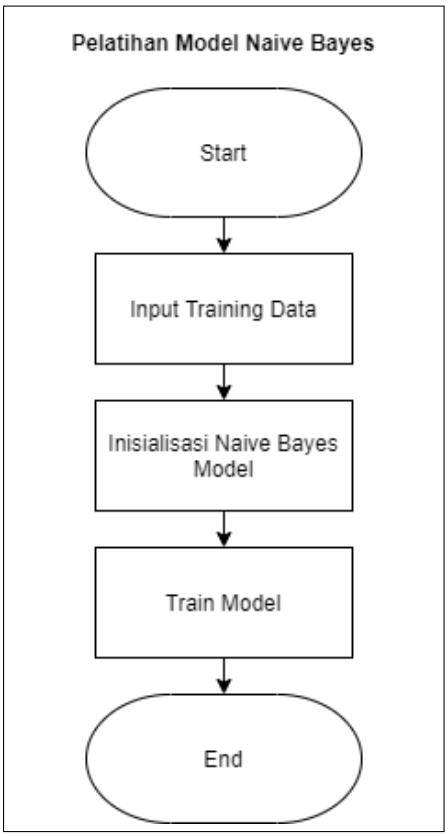
dengan mengalikan TF dan IDF, kemudian disusun menjadi matriks TF-IDF yang menjadi representasi fitur siap digunakan untuk analisis lebih lanjut.

3.4.3 Pembagian Dataset

Data yang digunakan pada penelitian ini dibagi menjadi dua bagian, yaitu *data training* dan *data testing*. Pembagian ini dilakukan dengan proporsi 80% untuk data training dan 20% untuk *data testing*. Data training digunakan untuk melatih model *Naive Bayes*, sedangkan *data testing* digunakan untuk menguji performa model yang sudah dilatih.

3.4.4 Pelatihan Model Naive Bayes

Ada beberapa varian algoritma *Naive Bayes Classifier* yang dapat digunakan untuk pembangunan model pembelajaran mesin. Pada penelitian ini, algoritma *Multinomial Naive Bayes* digunakan karena cocok untuk klasifikasi dengan fitur diskrit seperti representasi matriks TF-IDF dari dokumen.



Gambar 3.4. Pelatihan Model *Naive Bayes*

Sesuai dengan gambar 3.4, proses pelatihan model dimulai dengan memasukkan data pelatihan yang terdiri atas matriks TF-IDF sebagai fitur dan label target. Model *Multinomial Naive Bayes* kemudian diinisialisasi dengan parameter tertentu, seperti *alpha* untuk mengontrol *smoothing*. Selanjutnya, metode *fit* digunakan untuk melatih model berdasarkan data pelatihan tersebut. Model yang telah terlatih ini selanjutnya siap digunakan untuk evaluasi atau prediksi data baru.

3.4.5 Evaluasi Model

Evaluasi performa model dilakukan menggunakan beberapa metrik penting, yaitu *confusion matrix, accuracy score, precision, recall,* dan F1-score. *Confusion matrix* memberikan gambaran mengenai jumlah *true positives (TP), true negatives (TN), false positives (FP),* dan *false negatives (FN)* untuk membantu menganalisis prediksi yang benar dan salah.

Accuracy score mengukur persentase keseluruhan prediksi yang benar dan dirumuskan sebagai berikut:

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN}. \tag{3.1}$$

Precision digunakan untuk menghitung seberapa banyak prediksi positif yang benar-benar sesuai, dengan rumus:

$$\text{Precision} = \frac{TP}{TP+FP}. \tag{3.2}$$

Recall mengukur sejauh mana model mampu mendeteksi semua kasus positif yang ada, dengan rumus:

$$\text{Recall} = \frac{TP}{TP+FN}. \tag{3.3}$$

F1-score, yang merupakan rata-rata harmonis dari *precision* dan *recall*, dirumuskan sebagai berikut:

$$\text{F1-score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}. \tag{3.4}$$

Metrik-metrik ini digunakan untuk memberikan evaluasi menyeluruh terhadap performa model, terutama ketika ada ketidakseimbangan antara data positif dan negatif.

3.4.6 *Hyperparameter Tuning* menggunakan GridSearchCV

Untuk meningkatkan performa model, *tuning hyperparameter* dilakukan menggunakan `GridSearchCV`, sebuah teknik pencarian sistematis yang mengeksplorasi kombinasi parameter terbaik berdasarkan kinerja model pada *cross validation data*. Dalam penelitian ini, *hyperparameter* yang dioptimalkan adalah `alpha`, parameter *smoothing* pada model *Multinomial Naive Bayes*. Rentang nilai `alpha` yang diuji adalah $[0.1, 0.5, 1.0, 2.0, 5.0]$, dengan validasi silang sebanyak 5 lipatan (*cv=5*) untuk memastikan performa model yang baik pada data latih sekaligus generalisasi yang optimal pada data baru.

Proses *GridSearchCV* menentukan nilai `alpha` terbaik dengan mengukur akurasi model pada data validasi. Setelah mendapatkan parameter optimal, model dilatih ulang dengan parameter tersebut dan diuji pada data uji (`X_test`). Evaluasi model mencakup laporan klasifikasi (*classification report*) yang berisi metrik seperti presisi, *recall*, dan skor F_1, serta visualisasi *confusion matrix* untuk melihat distribusi prediksi yang benar dan salah di setiap kelas.

3.4.7 Pengujian Model dengan Data Testing

Setelah model dilatih, pengujian dilakukan menggunakan 20% data yang telah disisihkan untuk testing. Data testing ini digunakan untuk memprediksi label dan mengevaluasi performa model dalam situasi yang tidak terlihat selama proses pelatihan.

3.5 Evaluasi Hasil Uji Coba

Evaluasi hasil uji coba dilakukan secara manual dengan menghitung dan membandingkan nilai statistik model yang telah dilatih. Selain itu, model juga diuji menggunakan data yang berbeda dari data latih dan data uji. Data evaluasi terdiri dari 370 kalimat untuk setiap kategori (netral, hinaan, dan pencemaran nama baik). Data ini diperoleh dari LLM ChatGPT 4o.

3.6 Implementasi Model Pembelajaran Mesin ke Aplikasi *Website*

Setelah pengembangan dan pengujian algoritma, model diimplementasikan ke dalam situs web menggunakan *Flask*. *Backend* Flask menangani input pengguna dan menjalankan model pembelajaran mesin. API dan *callback* akan dirancang

untuk menerima teks dari *frontend*, memprosesnya menggunakan model, dan mengirimkan hasil prediksi kembali.