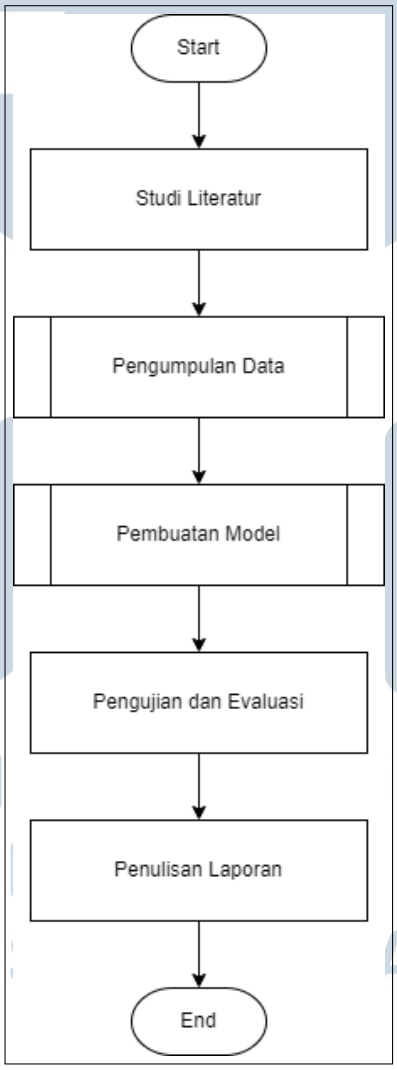
BAB 3
METODOLOGI PENELITIAN

3.1 Gambaran Umum

Gambar 3.1 merupakan gambaran umum dari metodologi penelitian yang dilakukan. Penelitian ini dimulai dengan melakukan studi literatur terlebih dahulu. Setelah melakukan studi literatur, tahap selanjutnya adalah mengumpulkan data dari Twitter. Langkah selanjutnya adalah melakukan pembuatan model. Setelah pembuatan model, akan dilakukan pengujian dan evaluasi. Tahap terakhir dari penelitian ini adalah penulisan laporan.



Gambar 3.1. Flowchart utama

3.2 Studi Literatur

Pada tahap ini dilakukan pembelajaran tentang teori literatur yang berkaitan dengan topik penelitian. Beberapa literatur yang dipelajari adalah *text mining*, analisis sentimen, *data scraping*, *remove duplicate*, *foreign word translation*, *text preprocessing*, *labeling*, *CountVectorizer*, *Term Frequency-Inverse Document Frequency (TF-IDF)*, *Synthetic Minority Oversampling Technique (SMOTE)*, *Random Forest Classifier*, *hyperparameter tuning*, dan *confusion matrix*.

3.3 Pengumpulan Data

Pada tahap ini dilakukan pengumpulan data menggunakan Snscrape. Gambar 3.2 menunjukkan langkah-langkah yang dilakukan dalam tahap pengumpulan data. Langkah pertama dalam tahap ini adalah melakukan *data scraping*. Data yang diambil adalah *tweet* Bahasa Indonesia yang mengandung kata kunci ”chatgpt” sejak tanggal 30 November 2022 sampai 17 April 2023. Setelah melakukan *data scraping*, langkah selanjutnya adalah melakukan *remove duplicate*. Langkah terakhir dari tahap ini adalah melakukan *translation*.

1. *Data scraping*

   Langkah pertama yang dilakukan dalam *data scraping* adalah *install* Snscrape dan *import* snscrape.modules.twitter. Setelah itu, siapkan *query* untuk mencari *tweet*. Dalam penelitian ini, *query* yang dimasukkan adalah ’chatgpt lang:id since:2022-11-30 until:2023-04-17’. Langkah selanjutnya adalah membuat *list* kosong untuk menampung *tweet*. Kemudian, jalankan TwitterSearchScraper dengan memasukkan *query* yang sudah disiapkan sebelumnya. Setelah itu, masukkan data yang didapatkan ke dalam list kosong yang sudah dibuat sebelumnya, lalu simpan data ke dalam file CSV.

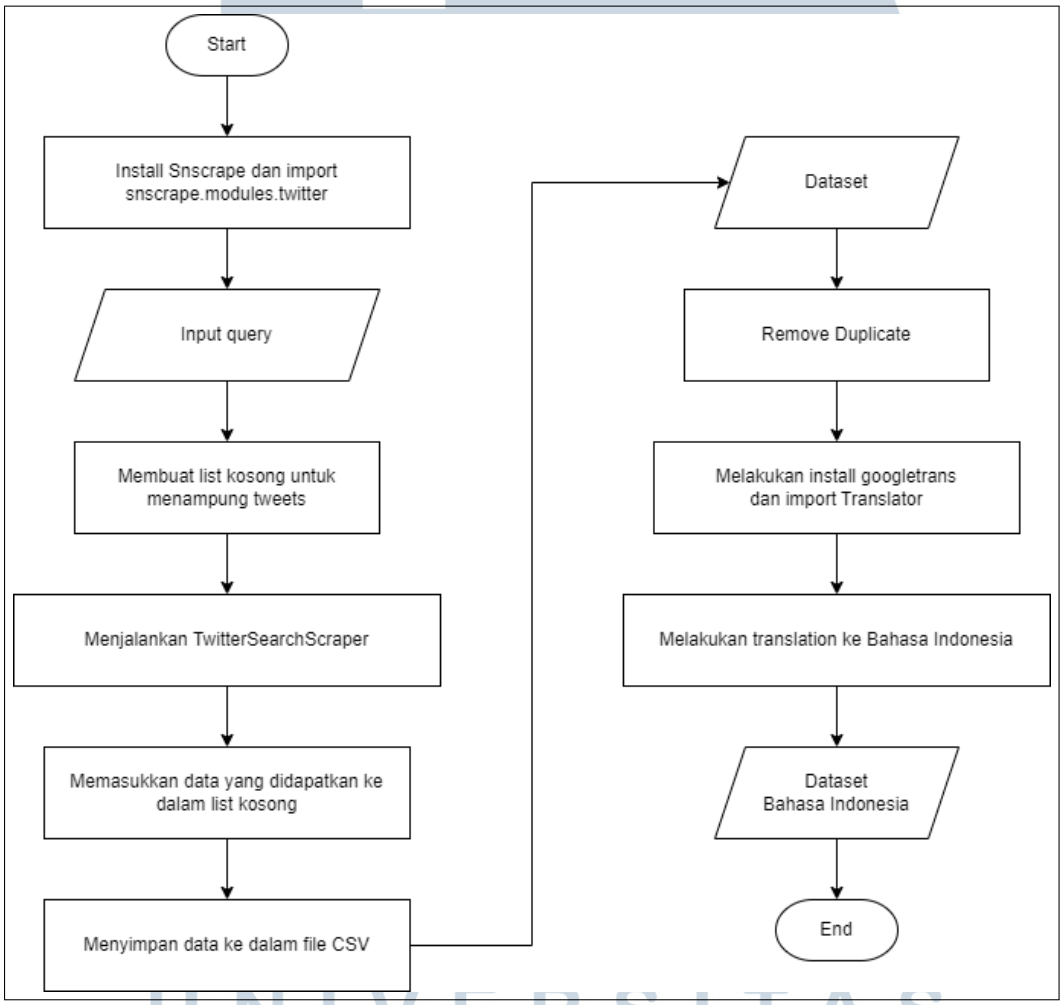
2. *Remove duplicate*

   Setelah mendapatkan data, langkah selanjutnya adalah melakukan *remove duplicate* untuk membuang data yang sama. Tahap ini diperlukan agar tidak terjadi pengulangan data yang bisa mempengaruhi hasil perhitungan.

3. *Translation*

   Translation dilakukan untuk mengubah data ke Bahasa Indonesia. Ketika melakukan *data scraping*, akan ada *tweet* yang masih tercampur dengan

bahasa asing. Untuk itu, diperlukan *translation* agar bisa mengubah *tweet* ke Bahasa Indonesia. Langkah pertama yang dilakukan dalam *translation* adalah install *googletrans* dan import *Translator*. Setelah itu jalankan *Translator* dengan memasukkan kode bahasa tujuan untuk melakukan penerjemahan ke Bahasa Indonesia. Jika sudah, dataset Bahasa Indonesia siap digunakan untuk tahap *text preprocessing*.

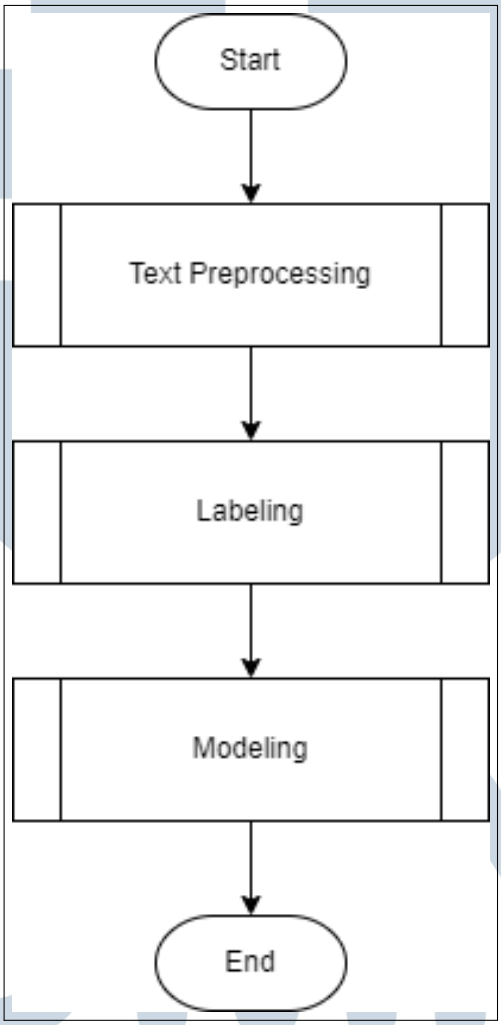


Gambar 3.2. Flowchart pengumpulan data

3.4 Pembuatan Model

Pada tahap ini dilakukan pembuatan model, mulai dari tahap *text preprocessing*, *labeling*, sampai penerapan *Random Forest Classifier*. Gambar 3.3 menunjukkan langkah-langkah yang dilakukan dalam tahap pembuatan model. Setelah mendapatkan data dari *Twitter*, data tersebut akan dibersihkan terlebih

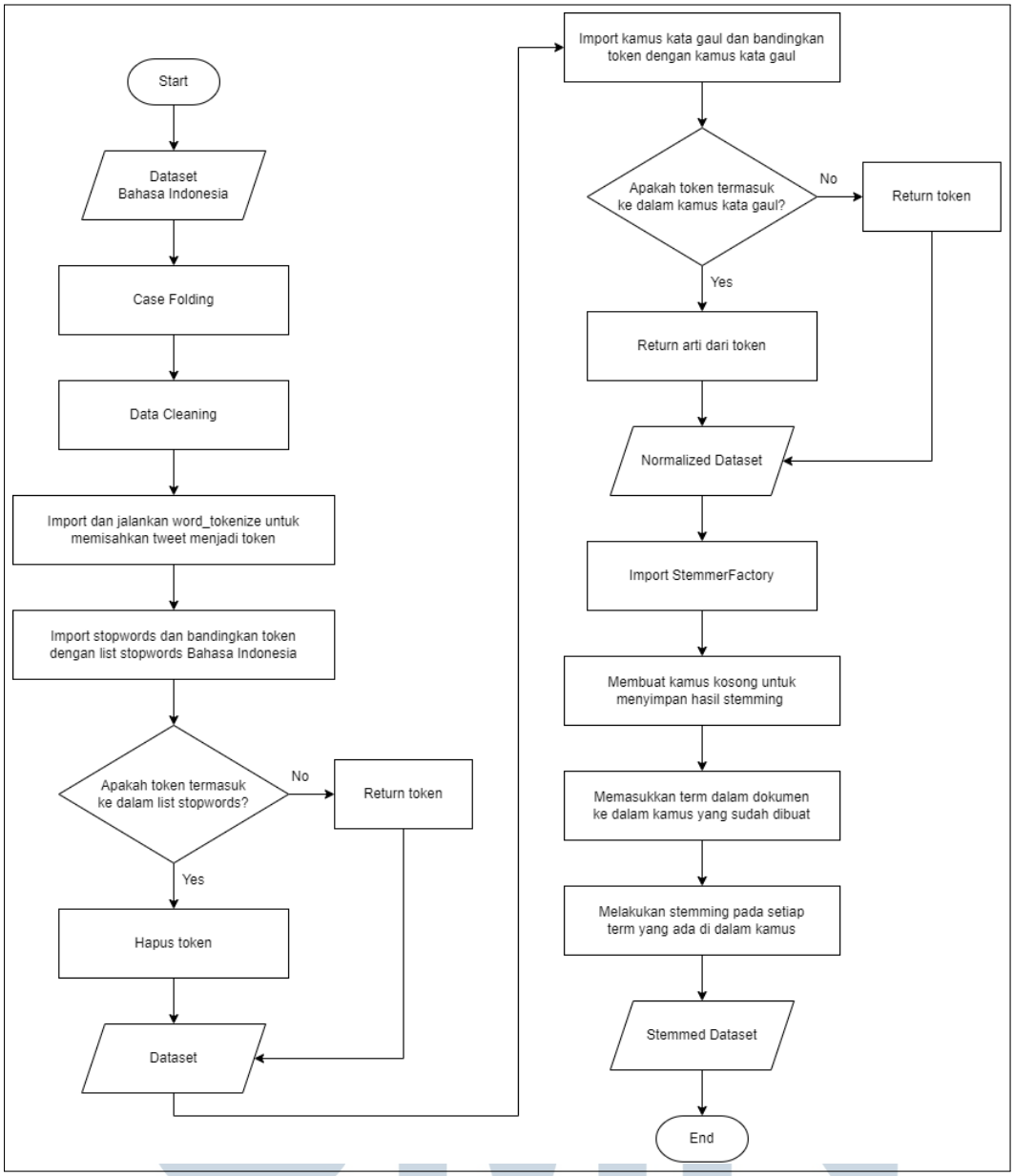
dahulu melalui *text preprocessing*. Jika sudah selesai melakukan *text preprocessing*, langkah selanjutnya adalah melakukan *labeling*. Bagian ini dilakukan untuk menentukan sentimen positif, negatif, atau netral. Setelah *labeling*, langkah selanjutnya adalah membuat *modeling* menggunakan *Random Forest Classifier*.



Gambar 3.3. Flowchart pembuatan model

3.4.1 Text Preprocessing

Text preprocessing dilakukan untuk menyiapkan data sebelum digunakan. Gambar 3.4 menunjukkan langkah-langkah yang dilakukan pada tahap *text preprocessing*. Setelah mendapatkan dataset, langkah pertama adalah melakukan *case folding*. Sesudah itu, dilakukan *cleaning data*. Jika sudah selesai membersihkan data, langkah selanjutnya adalah melakukan *tokenization*, *remove stopwords*, *normalization*, dan *stemming*.



Gambar 3.4. Flowchart text preprocessing

1. *Case folding*

   Case folding dilakukan untuk mengubah semua huruf menjadi huruf kecil. Tujuan dari tahap ini adalah untuk mencegah *case sensitivity*. Contoh dari *case folding* adalah teks ”ChatGPT Bakal Menenggelamkan Google” akan diubah menjadi ”chatgpt bakal menenggelamkan google”.

2. *Data cleaning*

   Data cleaning dilakukan untuk menghapus kata atau karakter yang tidak

diperlukan. Beberapa hal yang dilakukan pada tahap ini antara lain adalah menghapus *mention username*, *hashtag*, *retweet*, *url*, tanda baca, *special characters*, angka, dan *multiple spaces*.

3. *Tokenization*

   Tokenization dilakukan untuk memisahkan *tweet* menjadi kata terpisah. Contohnya pada saat melakukan *tokenization* pada teks ”chatgpt bakal menenggelamkan google”, *output*-nya akan menjadi ”chatgpt”, ”bakal”, ”menenggelamkan”, ”google”. Langkah pertama yang dilakukan dalam *tokenization* adalah menyiapkan dataset yang sudah dibersihkan pada saat *data cleaning*. Sesudah itu, *import word_tokenize* dari *nltk.tokenize*. Selanjutnya, jalankan *word_tokenize* untuk memisahkan *tweet* menjadi token.

4. *Remove stopwords*

   Remove stopwords dilakukan untuk membuang kata umum yang tidak memiliki makna. Beberapa cohtoh dari *stopwords* adalah dan, yang, dan serta. Langkah pertama yang dilakukan dalam *remove stopwords* adalah melakukan *import stopwords* dari *nltk.corpus*. Selanjutnya, jalankan *stopwords* Bahasa Indonesia untuk membandingkan token dengan *list stopwords*. Jika token termasuk ke dalam *list stopwords*, token akan dihapus dari dataset, sedangkan jika token tidak termasuk ke dalam *list stopwords*, maka token akan tetap dipakai dalam dataset.

5. *Normalization*

   Normalization dilakukan untuk menghapus kata gaul. Contoh dari kata gaul adalah trims, mantul, dan gaje. Langkah pertama yang dilakukan dalam *normalization* adalah *import* kamus kata gaul (combined_slang_words.txt) [59]. Selanjutnya, bandingkan token dengan kamus kata gaul. Jika token termasuk ke dalam kamus kata gaul, nilai yang akan dikembalikan ke dataset adalah arti dari token tersebut, sedangkan jika token tidak termasuk ke dalam kamus kata gaul, maka token akan tetap dipakai dalam dataset.

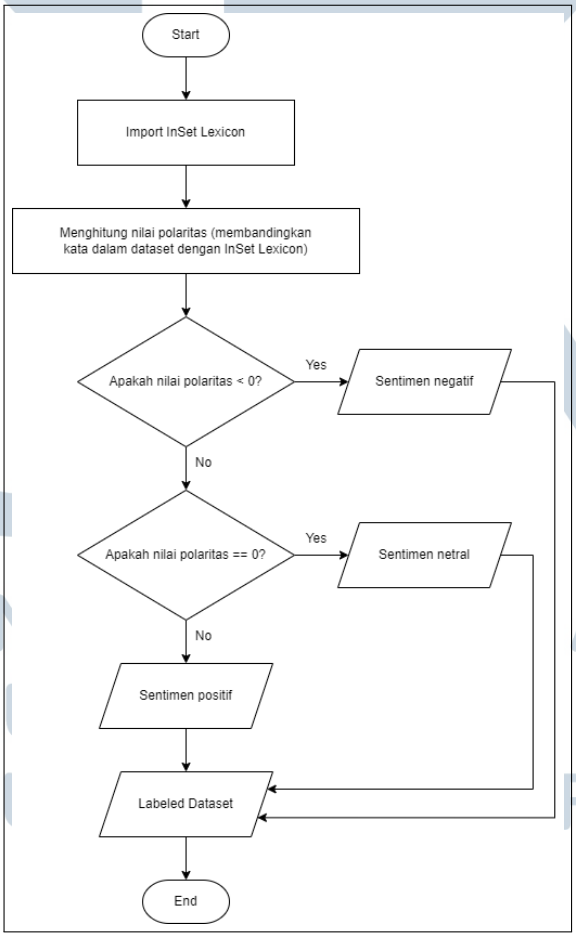
6. *Stemming*

   Stemming dilakukan untuk mengubah kata menjadi bentuk dasarnya. Sebagai contoh, kata ’mendengar’, ’mendengarkan’ dan ’terdengar’ akan diubah menjadi ’dengar’. Langkah pertama yang dilakukan dalam *stemming* adalah melakukan *install* Sastrawi dan *import* StemmerFactory. Sesudah itu, buat

kamus kosong untuk menyimpan hasil *stemming*. Langkah selanjutnya adalah memasukkan *term* dalam dokumen ke dalam kamus yang sudah dibuat. Setelah itu, lakukan *stemming* pada setiap *term* yang ada di dalam kamus. Selanjutnya, masukkan hasil *stemming* ke dalam dataset.

3.4.2 Labeling

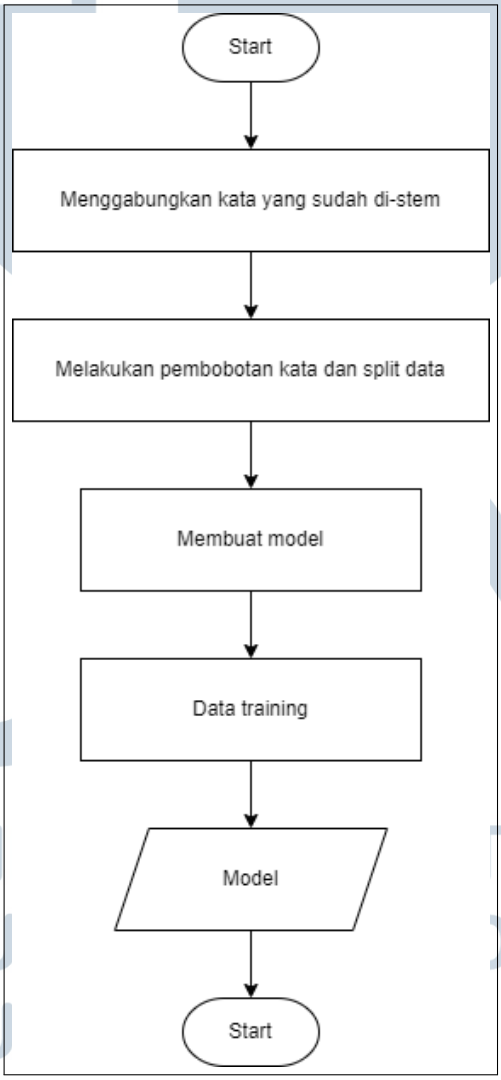
Labeling dilakukan untuk menentukan sentimen positif, negatif, atau netral. Gambar 3.5 menunjukkan langkah-langkah yang dilakukan pada tahap *labeling*. Langkah pertama adalah *import* Indonesia Sentiment Lexicon [60]. Langkah selanjutnya adalah menghitung nilai polaritas dengan cara membandingkan kata-kata dalam dataset dengan Indonesia Sentiment Lexicon untuk melihat polaritas dari masing-masing kata. Jika nilai polaritas lebih kecil dari nol, maka sentimen bernilai negatif. Jika nilai polaritas sama dengan nol, maka sentimen bernilai netral. Jika nilai polaritas lebih besar dari nol, maka sentimen bernilai positif.



Gambar 3.5. Flowchart labeling

3.4.3 Modeling

Pada tahap ini dilakukan beberapa hal untuk mengukur *performance* dari metode *Random Forest Classifier*. Gambar 3.6 menunjukkan alur dari tahap *modeling*. Setelah melakukan *labeling*, langkah selanjutnya adalah menggabungkan kembali kata-kata yang sudah dilakukan *stemming*. Sesudah itu, lakukan pembobotan kata dan juga *split* data. Langkah selanjutnya adalah membuat model. Dalam penelitian ini, model yang digunakan adalah *Random Forest Classifier* menggunakan *hyperparameter tuning*. Jika sudah membuat model, langkah selanjutnya adalah melakukan *data training*.

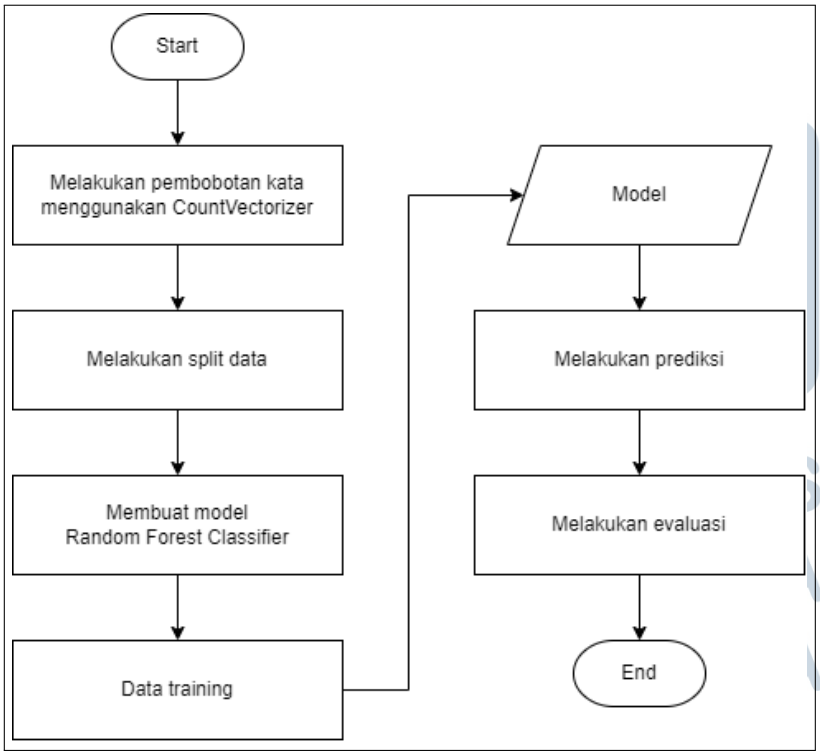


Gambar 3.6. Flowchart modeling

3.5 Pengujian dan Evaluasi

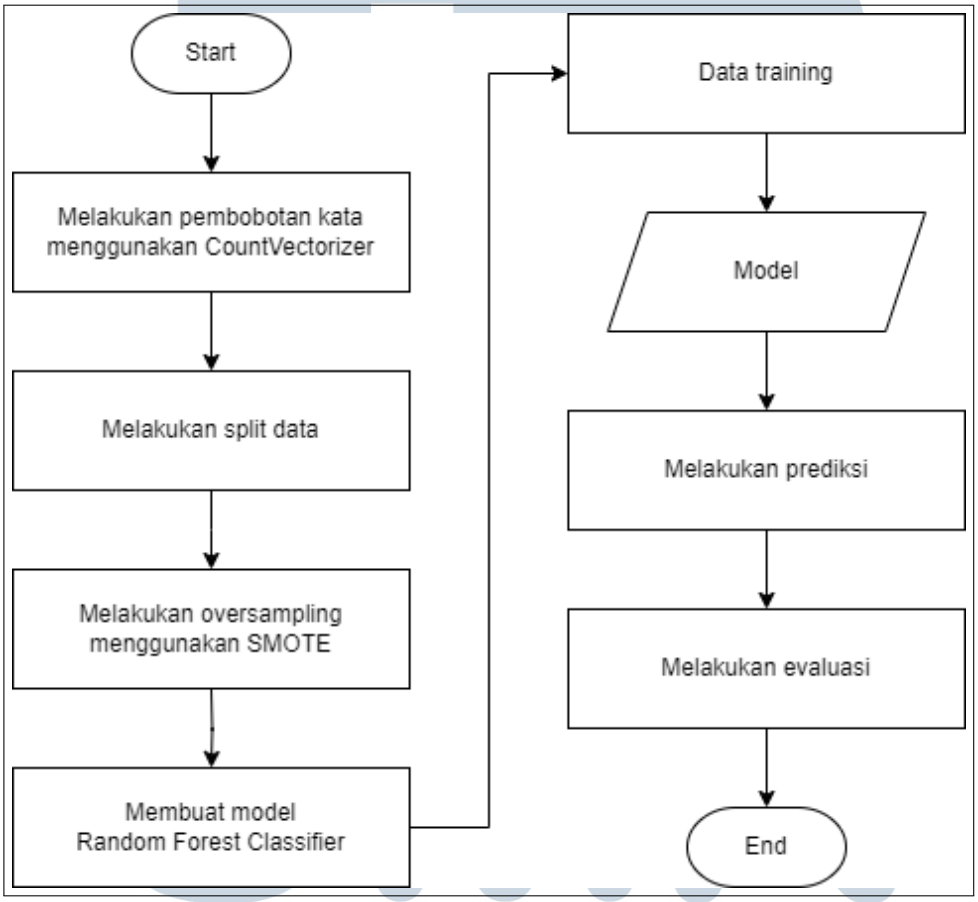
Setelah mendapatkan model, tahap selanjutnya adalah melakukan pengujian dan evaluasi. Pada tahap ini terdapat dua pengujian, yaitu membandingkan hasil *performance* dari *Random Forest Classifier* menggunakan *hyperparameter tuning* pada empat skenario berbeda, lalu membandingkan dengan model *Random Forest Classifier* tanpa *hyperparameter tuning* pada skenario yang sama. Setelah itu, akan dilakukan evaluasi menggunakan *confusion matrix* dan *classification_report*. *Confusion matrix* akan menampilkan hasil prediksi dari setiap kelas, sedangkan *classification_report* akan menampilkan akurasi, presisi, *recall*, dan *f1-score*. Empat skenario yang akan dilakukan untuk uji coba adalah sebagai berikut.

1. *CountVectorizer* pada data yang tidak seimbang. Gambar 3.7 menujukkan langkah-langkah yang dilakukan pada skenario pertama, dimulai dari pembobotan kata menggunakan *CountVectorizer*, membagi data untuk *training* dan *testing*, membuat model *Random Forest Classifier*, hingga melakukan data training. Setelah mendapatkan model, langkah selanjutnya adalah melakukan prediksi. Sesudah itu akan dilakukan evaluasi.



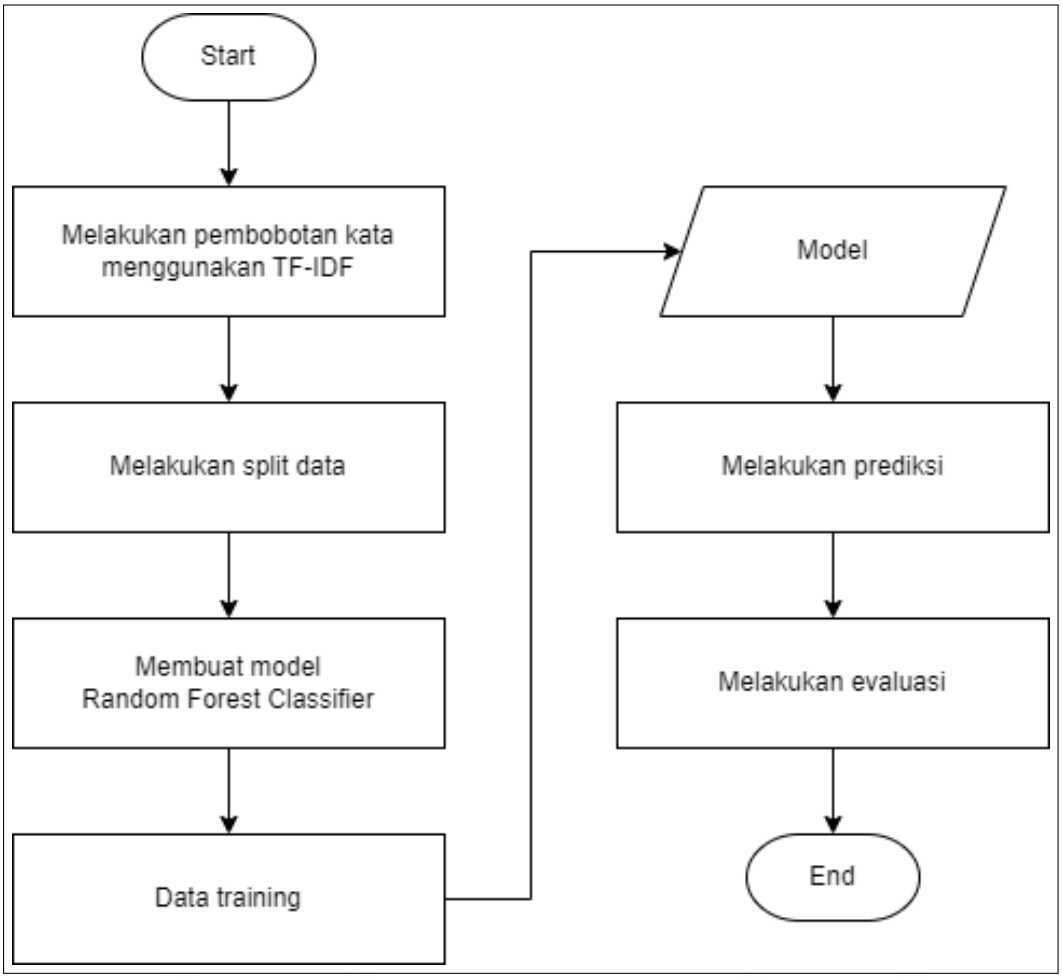
Gambar 3.7. Flowchart pembobotan kata menggunakan *CountVectorizer* pada data yang tidak seimbang

2. *CountVectorizer* pada data yang seimbang. Gambar 3.8 menujukkan langkah-langkah yang dilakukan pada skenario kedua, dimulai dari pembobotan kata menggunakan *CountVectorizer*, membagi data untuk *training* dan *testing*, melakukan *oversampling* menggunakan *SMOTE*, membuat model *Random Forest Classifier*, hingga melakukan data training. Setelah mendapatkan model, langkah selanjutnya adalah melakukan prediksi. Sesudah itu akan dilakukan evaluasi.



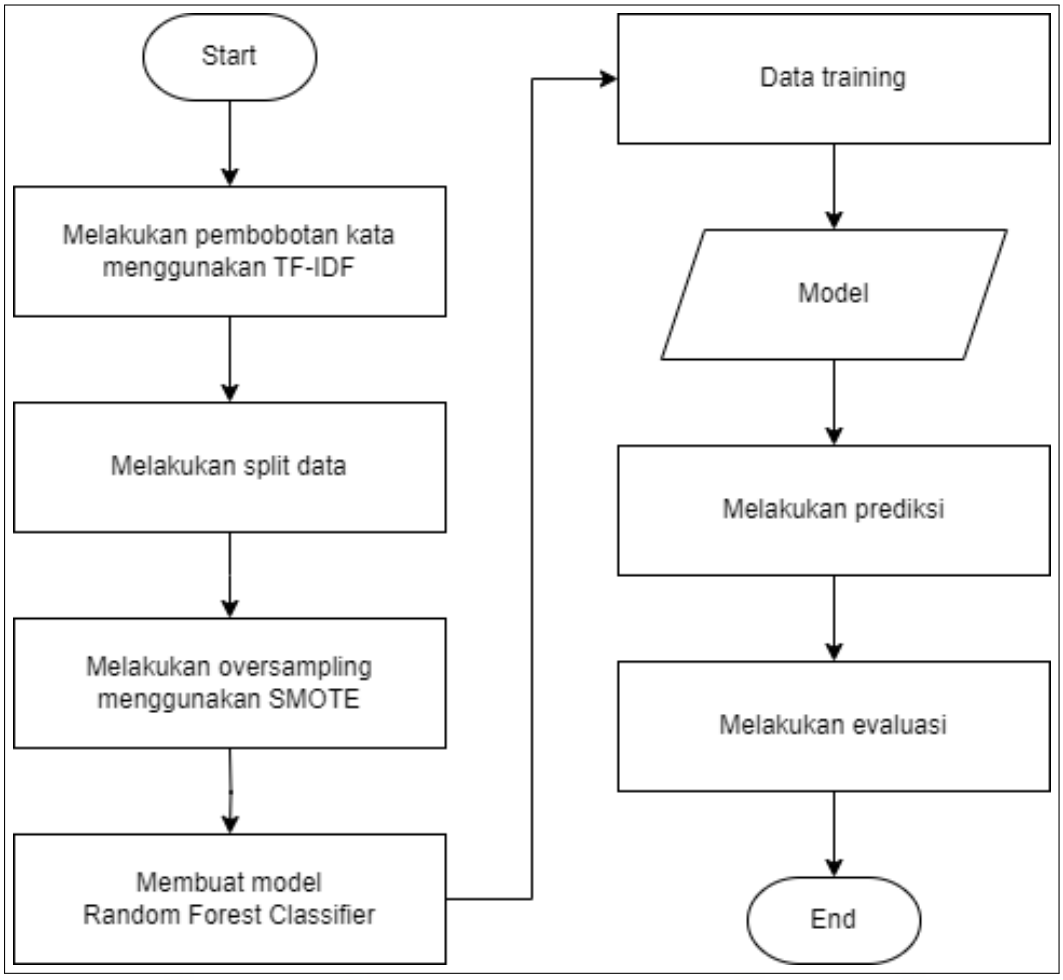
Gambar 3.8. Flowchart pembobotan kata menggunakan *CountVectorizer* pada data yang seimbang

3. *TF-IDF* pada data yang tidak seimbang. Gambar 3.9 menujukkan langkah-langkah yang dilakukan pada skenario ketiga, dimulai dari pembobotan kata menggunakan *TF-IDF*, membagi data untuk *training* dan *testing*, membuat model *Random Forest Classifier*, hingga melakukan data training. Setelah mendapatkan model, langkah selanjutnya adalah melakukan prediksi. Sesudah itu akan dilakukan evaluasi.



Gambar 3.9. Flowchart pembobotan kata menggunakan *TF-IDF* pada data yang tidak seimbang

4. *TF-IDF* pada data yang seimbang. Gambar 3.10 menujukkan langkah-langkah yang dilakukan pada skenario keempat, dimulai dari pembobotan kata menggunakan *TF-IDF*, membagi data untuk *training* dan *testing*, melakukan *oversampling* menggunakan *SMOTE*, membuat model *Random Forest Classifier*, hingga melakukan data training. Setelah mendapatkan model, langkah selanjutnya adalah melakukan prediksi. Sesudah itu akan dilakukan evaluasi.



Gambar 3.10. Flowchart pembobotan kata menggunakan *TF-IDF* pada data yang seimbang

3.6 Penulisan Laporan

Pada tahap ini dilakukan penulisan laporan dari penelitian yang sudah dilakukan. Bagian yang dikerjakan pada tahap ini antara lain adalah pendahuluan, landasan teori, metodologi penelitian, hasil dan diskusi, serta simpulan dan saran. Selain itu, pada tahap ini ditambahkan juga bagian abstrak, daftar pustaka, dan lampiran yang diperlukan, seperti formulir konsultasi skripsi dan hasil pengecekan *Turnitin*.

3.7 Spesifikasi Sistem

Spesifikasi sistem yang digunakan untuk melakukan penelitian ini adalah sebagai berikut.

1. *Device*: ASUS Vivobook S 14 Flip
2. *Processor*: Intel Core i7-12700H
3. RAM: 16 GB
4. *Software*: Visual Studio Code, Jupyter Notebook
5. Bahasa pemrograman: Python