BAB III
METODOLOGI PENELITIAN DAN PERANCANGAN SISTEM

3.1 Metodologi Penelitian

Penelitian dilakukan dalam beberapa tahap, yaitu telaah literatur, analisis kebutuhan, desain sistem, pemograman sistem, *testing* and *debug*, serta konsultasi dan penulisan laporan. Setiap tahap penelitian dapat diuraikan sebagai berikut:

1. Telaah literatur dilakukan dengan melakukan studi dari berbagai referensi seperti jurnal atau buku tentang metode-metode klasifikasi *user feedback* yaitu seperti metode N-Gram, TF-IDF, Algoritma Multinomial Naïve Bayes.
2. Analisa kebutuhan dilakukan bersamaan dengan telaah literatur. Kebutuhan sistem dirancang dan dianalisa berdasarkan informasi yang didapat selama telaah literatur.
3. Desain sistem dilakukan setelah analisa kebutuhan agar rancangan aplikasi terstruktur dan sesuai kebutuhan. Alur logika program digambarkan kedalam bentuk *flowchart*.
4. Tahap pemograman sistem merupakan penulisan kode, *training* dan *testing dataset* mengimplementasikan algoritma Multinomial Naïve Bayes dengan TF-IDF dan N-gram. *Dataset* yang digunakan dalam penelitian ini merupakan komentar dari aplikasi.
5. Testing dan debug dilakukan untuk menguji dan validasi apakah sistem berjalan dengan sesuai dengan kebutuhan, serta memperbaiki bagian yang fungsionalitasnya tidak sesuai. *Dataset* pada *testing* akan diuji

berdasarkan dari hasil *training*. Selanjutnya menghitung evaluasi performa akan menampilkan tabel *confusion matrix* dan nilai *accuracy*, *precision*, *recall*, *F-measure* yang akan memberikan kesimpulan mengenai metode yang dipilih.

6. Konsultasi dengan pembimbing dilakukan agar penelitian terarah dan terdapat saran untuk menjadi masukan kepada peneliti.

3.2 Gambaran Umum Sistem

Sistem yang dibangun merupakan implementasi algoritma Multinomial Naïve Bayes, TF-IDF dan N-Gram. Sistem ini berfungsi untuk menalisa sentimen dari kolom *review* yang berasal dari komentar pengguna, Analisa dilakukan dengan mengklasifikasikan komentar menjadi tiga yaitu label positif, negatif, dan netral.

Pengguna dapat mengambil *dataset* dari kolom *review* dengan format xlsx, lalu pengguna dapat melakukan *import dataset* ke dalam sistem. Sistem akan melakukan klasifikasi dengan melakukan *text preprocessing* komentar yaitu dengan mengubah komentar menjadi huruf kecil, menghapus tanda baca dan angka, Tahap *text preprocessing* di mana kalimat dibersihkan dengan menghapus tanda baca dan angka. Setelah itu kalimat akan dipisah menjadi token yang disimpan dalam bentuk *array* dan mengubah kalimat menjadi kata dasar supaya pada *Bag of Words* (BOW) tidak memiliki makna kata yang sama. Setelah itu, hasil *text preprocessing* disimpan di dalam *corpus*, yang nantinya akan dibagi menjadi dua yaitu *train* dan *test*, pengguna menentukan berapa banyak data yang digunakan sebagai data *train* dan data *test*.

Setelah data telah terbagi, data tersebut akan *vectorization* yang secara umum merupakan proses mengubah data *corpus* menjadi angka. Sistem ini menggunakan dua jenis *vectorizer* yaitu N-Gram *Vectorizer* dan TF-IDF *Vectorizer*. Di mana N-gram *Vectorizer* berfungsi untuk mengubah kalimat menjadi token dan menghitung masing-masing kosakata dalam dokumen untuk membuat vektor dari tiap kalimat dan disajikan kedalam bentuk matriks. Selanjutnya TF-IDF *Vectorizer* akan menghitung bobot frekuensi dari tiap kata dalam dokumen dan mengubah data N-gram ke bobot TF-IDF. Hasil TF-IDF merupakan bobot dari tiap kata akan disimpan sebagai model.

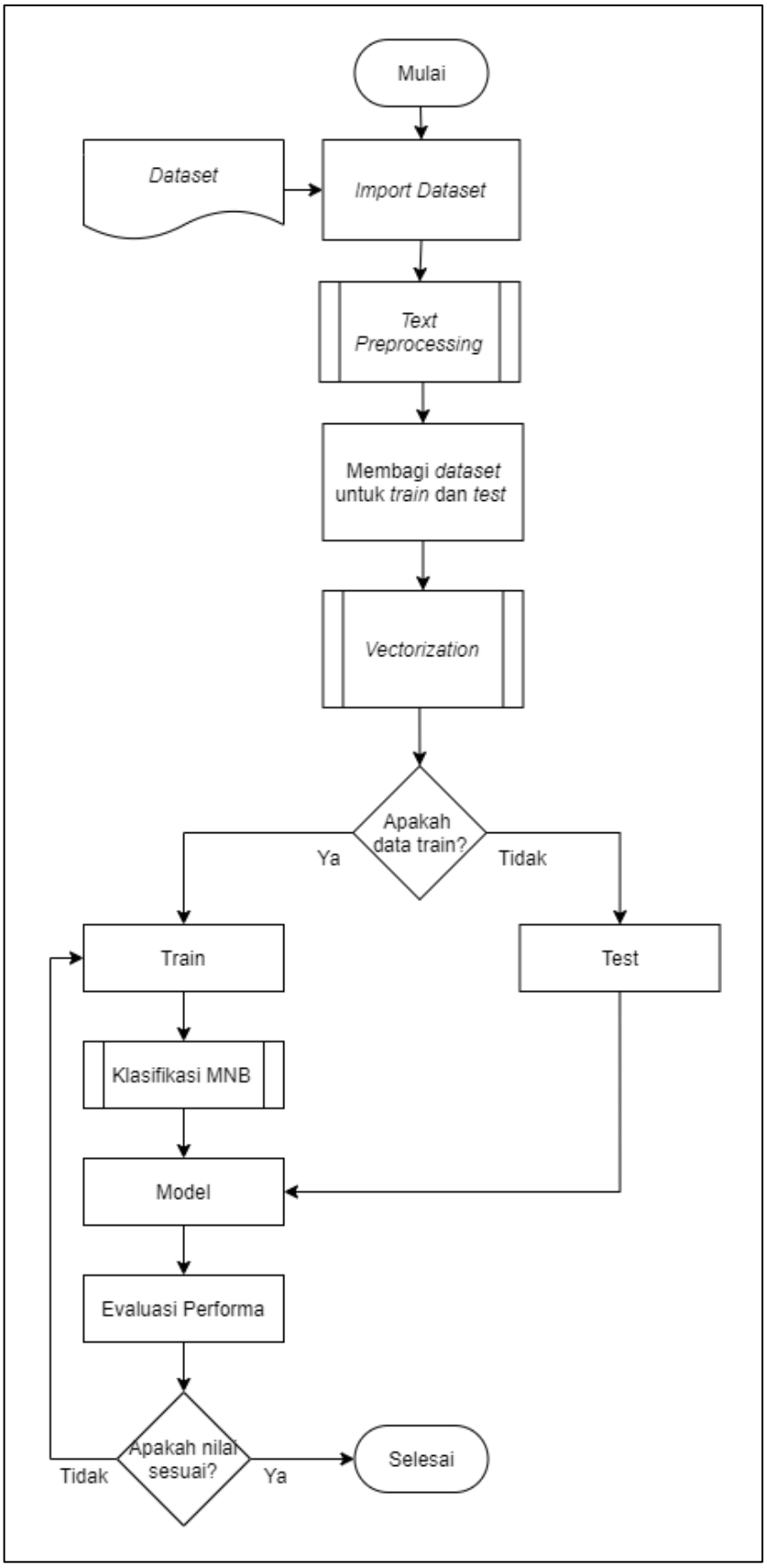
Setelah data disimpan sebagai model, *classifier* ini akan melakukan klasifikasi dengan menggunakan perhitungan Multinomial Naïve Bayes untuk training data dan komentar. Hasil dari perhitungan Multinomial Naïve Bayes akan muncul *confusion matrix* dan akan dihitung *accuracy*, *precision*, *recall* dan *F1-Score*.

3.3 Flowchart

Proses implementcasi dijelaskan dalam bentuk *Flowchart*, di mana terdiri dari: *Flowchart* utama, *Flowchart* modul *text preprocessing*, *Flowchart Vectorization*, *Flowchart* modul N-gram *Vectorizer*, *Flowchart* modul TF-IDF *Vectorizer*, dan *Flowchart* modul Multinomial Naïve Bayes.

3.3.1 Flowchart Utama

Gambar 3.1. merupakan *flowchart* utama yang menjelaskan alur implementasi secara umum. Pertama, melakukan *import* suatu *dataset* yang akan diklasifikasikan dan selanjutnya akan masuk ke tahap *text prepocessing*. Kemudian *dataset* dipisahkan untuk *training* dan *testing* sebelum lanjut ke tahap *vectorization*. Pada tahap *vectorization* yang dilakukan oleh N-Gram *Vectorizer* dan TF-IDF *Vectorizer*. Jika data merupakan data *train* maka, data akan dilatih dan diklasifikasikan dengan Multinomial Naïve Bayes dan membentuk suatu model. Selanjutnya, model akan diuji dengan data uji dan melakukan evaluasi. Jika nilai evaluasinya tercapai maka proses selesai tetapi, jika tidak sesuai akan mengulang pada pembagian data latih dan data uji.



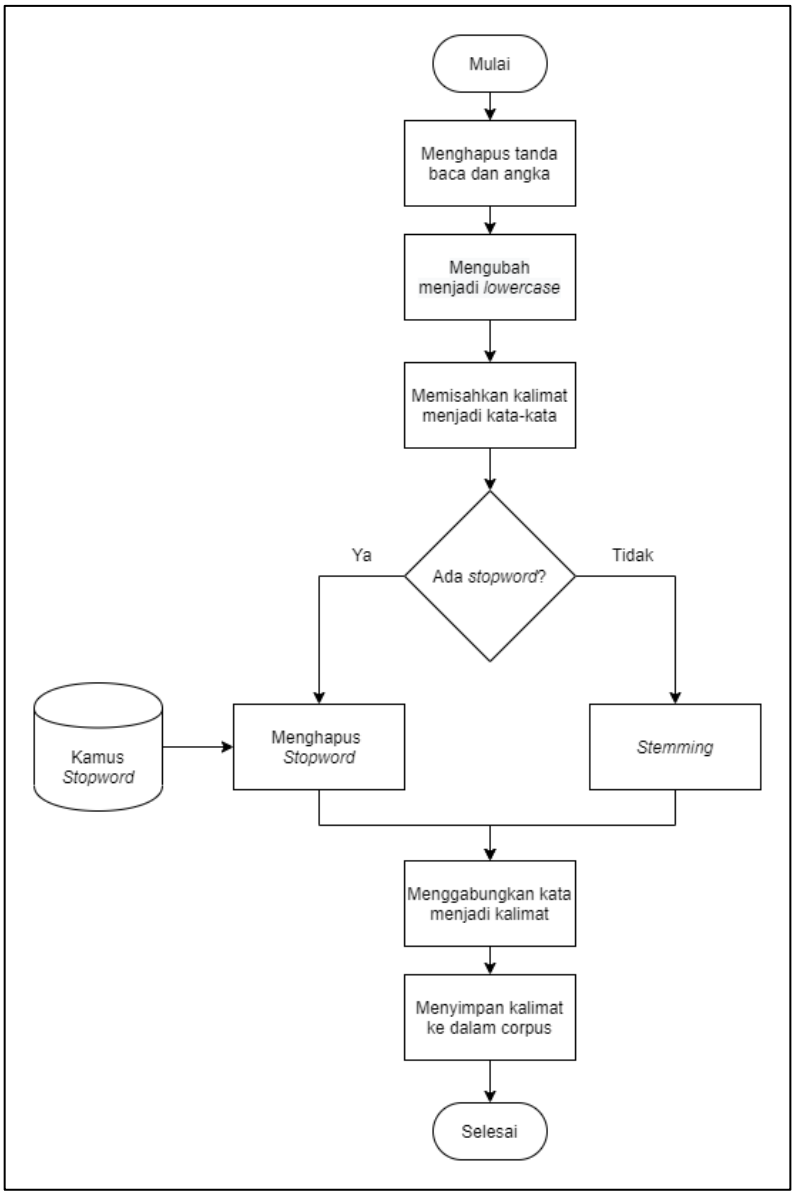
Gambar 3.1 *Flowchart* utama

3.3.2 Flowchart modul text prepocessing

Gambar 3.2 merupakan *flowchart* modul text prepocessing. Pertama, *dataset* diambil dari Endang Pamungkas (2016), terdiri dari 554 baris dan dua kolom. Kolom *review* yang berasal dari komentar pengguna dan kolom sentimen merupakan label negatif, netral dan positif berdasarkan sentimen dari komentar. Pada penelitian ini, teknik pemelajaran mesin (*supervised training*), pengisian sentimen ini dilakukan secara manual, tetapi kalau mesin telah belajar, maka pengisian kolom sentimen akan dilakukan secara otomatis.

Pada tahap *text preprocessing* kalimat tanda baca dan angka akan dihapus lalu, huruf kapital (*uppercase*) diubah menjadi huruf kecil *(lowercase)*. Kalimat akan dipisah menjadi token yang disimpan dalam bentuk *array*. Token akan dibandingkan dengan kamus *stopword*. Jika token merupakan kata *stopword* akan dihilangkan dan bila tidak ada *stopword*, maka akan langsung dilakukan proses *stemming* dengan menggunakan *library* dari Sastrawi. *Stemming* mengubah kalimat menjadi kata dasar supaya *Bag of Words* (BOW) tidak memiliki banyak kata yang mempunyai makna yang sama (seperti: ‘disukai’, ‘menyukai’, ‘kesukaan’ yang mengandung arti yang sama yaitu suka).

Untuk proses *stopword* dan *stemming* menggunakan *library* Sastrawi berdasarkan Kamus Besar Bahasa Indonesia (KBBI). Lalu setelah diubah menjadi kata dasar, semua kata akan digabungkan dan disimpan ke dalam corpus.

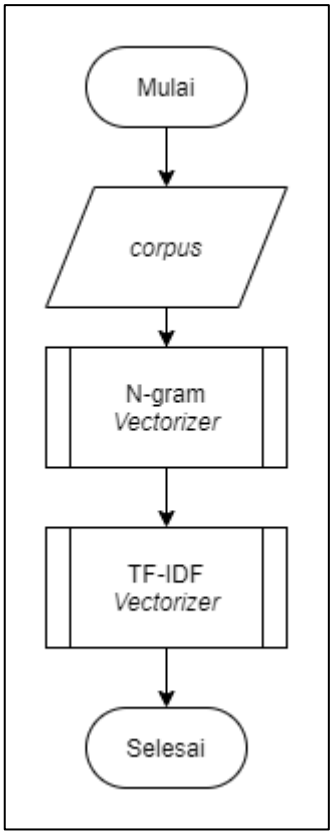


Gambar 3.2 *Flowchart* modul *text-prepocessing*

3.3.3 Flowchart modul Vectorization

Gambar 3.3 menggambarkan alur proses kerja pada modul *vectorization*. Secara umum, *Vectorization* bekerja dengan mengubah data *corpus* menjadi angka. Proses berikutnya adalah melakukan perubahan dari kalimat menjadi token dan melakukan perhitungan pada setiap kosakata dalam dokumen yang nantinya perhitungan tersebut akan digunakan untuk membuat matriks. Hasil proses N-gram *Vectorizer* berbentuk tabel matriks akan disimpan dalam dataset_transfomed.

Selanjutnya TF-IDF *Vectorizer* akan menghitung bobot frekuensi dari tiap kata dalam dokumen dan mengubah data N-gram ke bobot TF-IDF.



Gambar 3.3 *Flowchart* modul *vectorization*

3.3.4 Flowchart modul N-gram Vectorizer

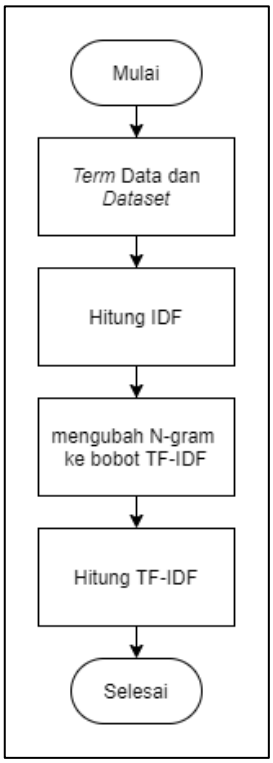
Gambar 3.4. menggambarkan kerja modul N-gram *Vectorizer*. Modul N-gram *Vectorizer* ini mengambil data corpus dan memisahkan kalimat ke bentuk token. Membuat vektor dengan menghitung tiap token sehingga setiap token memiliki bobot. Selanjutnya melakukan filter fitur yang dipakai dengan menghapus kata tidak relevan yang hanya muncul satu atau dua kali. Memastikan tidak ada kata yang duplikat di dalam *array* dan langkah terakhir dengan membuat matriks serta menyimpan ke data_transform.

Gambar 3.4 *Flowchart* N-gram *Vectorizer*

3.3.5 Flowchart modul TF-IDF Vectorizer

Gambar 3.5. merupakan *flowchart* dari TF-IDF *Vectorizer*. TF-IDF *Vectorizer* menekankan pada kata-kata yang jarang terjadi dengan memberikan bobot pada setiap kata. Bobot ditentukan melalui kombinasi frekuensi kata dalam dokumen, dan bagaimana kata tersebut ada di seluruh *corpus*. Langka pertama, mengambil data_transformed yang sudah diproses pada N-gram Vectorizer. Lalu menginisialisasi data_transformed dan menyimpan *column term* sebagai *term.*

Selanjutnya, menghitung IDF menggunakan Persamaan 2.9. Lalu, mengubah N-gram ke bobot TF-IDF menggunakan Persamaan 2.8 dengan Persamaan 2.10. Terakhir merepresentasikan ke dalam bentuk matriks dan menyimpan hasil matriks TF-IDF.

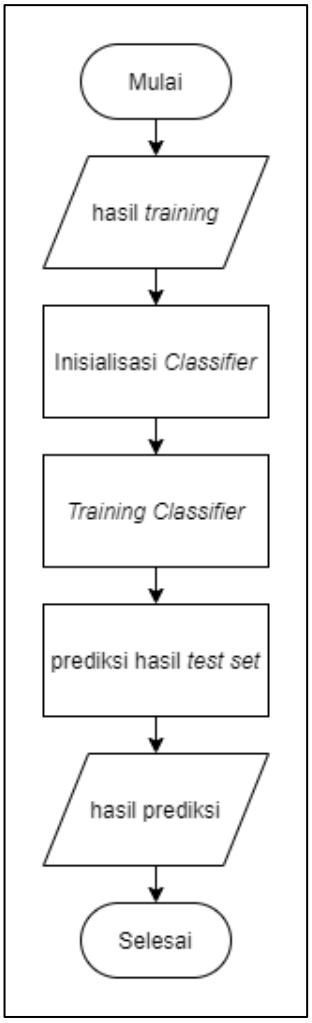


Gambar 3.5 *Flowchart* modul TF-IDF

3.3.6 Flowchart modul Multinomial Naïve Bayes

Gambar 3.7 menggambarkan proses modul Multinomial Naïve Bayes. Proses ini menggunakan library MultinomialNB dari sklearn. Pertama *classifier* harus diinisialisasi, Setelah itu, melatih X_train (ulasan) dan y_train (label). Langkah selanjutnya Multinomial Naïve Bayes akan melakukan prediksi pada X_test. Hasil prediksi X_test akan dilakukan pencocokan dengan y_test untuk mendapatkan *true positive*, *true negative*, *false positive*, *false negative* sehingga

dapat dicari tingkat akurasi, *presicion*, *recall*, dan *f1-score* dari tabel *confusion matrix*.



Gambar 3.6 *Flowchart* Multinomial Naive Bayes