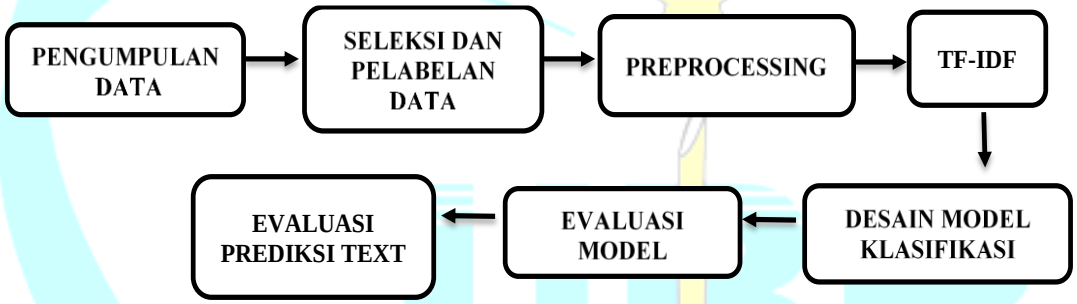
BAB III
METODE PENELITIAN

Penelitian ini bertujuan untuk mengklasifikasikan dan memprediksi ulasan pengguna aplikasi Tokopedia ke dalam beberapa kategori menggunakan algoritma Naive Bayes. Tahapan yang dilakukan dalam penelitian ini terdiri dari data scraping, data labeling, preprocessing, TF-IDF, training models, dan model performance evaluation. Tahapan tersebut dijelaskan secara rinci sebagai berikut:



Gambar 1.1 Tahapan Penelitian

1.1 Pengumpulan Data

Data ulasan pengguna diperoleh dari aplikasi Tokopedia dalam kurun waktu lima bulan terakhir terhitung sejak 16 Oktober 2024 hingga 27 Maret 2025. menggunakan teknik web scraping. Ulasan yang dikumpulkan merupakan tanggapan pengguna terhadap aplikasi yang tersedia di platform Google Play Store (Nugraha & Gustian, 2024).Setelah proses pengumpulan data, dilakukan pemilihan ulasan yang relevan dan kontekstual (Ruslan et al., 2023). Selain itu, data yang terpilih diklasifikasikan ke dalam beberapa label tertentu. Semua data disimpan dalam format teks dan disusun menjadi dataset yang digunakan pada tahap analisis selanjutnya.

1.2 Seleksi dan Pelabelan Data

Data ulasan pengguna diperoleh dari aplikasi Tokopedia menggunakan teknik web scraping, dengan total data ulasan yang diambil sebanyak 1.500 data. Ulasan yang dikumpulkan merupakan respon pengguna terhadap aplikasi yang tersedia pada platform Google Play Store. Setelah proses pengumpulan data dilakukan, dilakukan pemilihan ulasan yang relevan dan kontekstual. Selain itu, data yang terpilih dikelompokkan menjadi 4 kategori, yaitu: Fitur Aplikasi, Layanan, Promosi, dan Pembayaran. Seluruh data disimpan dalam format teks dan disusun menjadi dataset yang digunakan pada tahap analisis selanjutnya. Setelah data

ulasan berhasil dikumpulkan, dilakukan proses pelabelan manual menjadi empat kategori utama. Dari total 1.500 ulasan, sebanyak 1.244 ulasan memenuhi kualifikasi, sedangkan sisanya tidak relevan atau tidak memiliki konteks yang jelas. Hasil pelabelan menunjukkan bahwa kategori Fitur Aplikasi mendominasi dengan jumlah 572 ulasan (45,98%), diikuti oleh Layanan sebanyak 394 ulasan (31,67%), Pembayaran sebanyak 79 ulasan (6,35%), dan Promo sebanyak 199 ulasan (16,00%). Distribusi ini menunjukkan bahwa sebagian besar ulasan pengguna berfokus pada pengalaman mereka dalam menggunakan fitur aplikasi Tokopedia.

Tabel 1.2 Pelabelan Data

No	*Content*	*Category*
0	sudah langganan plus, pakai vocer yg katanya b...	Promo
1	Gak jelas. Daftar baru kata nya dapat promo, u...	Promo
2	Tadinya saya dapat promo yang besar, tetapi se...	Promo
3	Untuk tokopedia hilangkan saja fitur kurir rek..	Fitur aplikasi
4	Tolong dikasih fitur ganti ekspedisi, saya mua...	Fitur aplikasi
…	…	…
1244	Pusing sama sistem kupon tokped, punya kupon t...	Fitur aplikasi

1.3 Preprocessing

Agar data teks dapat diproses oleh algoritma, maka dilakukan beberapa tahap preprocessing yaitu *lower case, cleaning, tokenizing, stopwords*, dan *stemming* sebagai berikut:

1. *Case Folding*

Case folding merupakan tahapan dalam pengolahan teks yang bertujuan untuk mengubah semua huruf kapital menjadi huruf kecil (Annisa & Ulama, 2023). Langkah ini berguna untuk menyamakan format penulisan, sehingga proses pencarian dan analisis teks menjadi lebih efektif. Hal ini penting karena tidak semua dokumen menggunakan huruf kapital secara konsisten (Indriyani et al., 2023). Pada tabel 2 kolom sebelah kiri, kita dapat melihat huruf kapital seperti "Tadinya", "Gopay", dan "Ada". Setelah *case folding*, semua huruf kapital diubah menjadi huruf kecil, seperti "tadinya", "gopay", dan "ada".

Formulasi :

$CF(d_i) = lower(d_i)$ (1)

Keterangan:

a. Fungsi *lower*() mengubah semua huruf kapital menjadi huruf kecil.

b. Tujuannya adalah menyamakan bentuk kata, misalnya Ada dan ada menjadi sama.

Tabel 1.3 Hasil Case Folding

Text	*Case Folding*

Tadinya saya dapat promo yang besar, tetapi setelah saya top up saldo Gopay, promonya mendadak jadi kecil. Ada juga kupon lain yang tersedia tetapi tidak dapat digunakan padahal sudah memenuhi persyaratan.	tadinya saya dapat promo yang besar, tetapi setelah saya top up saldo gopay, promonya mendadak jadi kecil. ada juga kupon lain yang tersedia tetapi tidak dapat digunakan padahal sudah memenuhi persyaratan.

2. *Cleaning*

Tanda *Cleaning* merupakan langkah awal dalam pengolahan data teks yang bertujuan untuk membuang unsur-unsur yang tidak diperlukan atau dianggap mengganggu, sehingga data menjadi lebih bersih dan siap untuk dilakukan analisis lebih lanjut (Ravil et al., 2024). Pada tabel 3 kolom pertama (hasil dari lipatan huruf), teks masih mengandung tanda baca seperti koma (,), titik (.), dan penghubung lainnya. Setelah dibersihkan, tanda baca ini dihilangkan. Hasilnya adalah teks yang lebih bersih dan siap untuk diproses lebih lanjut tanpa gangguan dari simbol yang tidak perlu.

Formulasi :

$CL(d_i) = d_i - \{c \in d_i \mid c \notin \text{alfabet dan spasi}\}$ (2)

Keterangan:

a. Simbol, angka, dan karakter khusus dihapus dari teks.

b. Dapat menggunakan regex seperti [^a-zA-Z\s].

Tabel 1.4 Hasil *Cleaning*

Case Folding	*Cleaning*
tadinya saya dapat promo yang besar, tetapi setelah saya top up saldo gopay, promonya mendadak jadi kecil. ada juga kupon lain yang tersedia tetapi tidak dapat digunakan padahal sudah memenuhi persyaratan.	tadinya saya dapat promo yang besar tetapi setelah saya top up saldo gopay promonya mendadak jadi kecil ada juga kupon lain yang tersedia tetapi tidak dapat digunakan padahal sudah memenuhi persyaratan

3. *Tokenizing*

Proses ini dilakukan dengan memecah teks dari bentuk kalimat menjadi kata-kata terpisah. Tahap ini disebut tokenisasi dan bertujuan untuk memudahkan analisis dengan memecah kalimat menjadi unit kata (Prasetyo et al., 2023). Dari tabel 4, kita dapat melihat bahwa kalimat panjang dipecah menjadi 31 token yang mewakili setiap kata. Token-token ini kemudian akan digunakan dalam proses lebih lanjut seperti stemming, penghapusan stopword, atau klasifikasi teks.

Formulasi :

$T(d_i) = token(CL(CF(d_i)))$ (3)

Keterangan:

a. Fungsi token() memecah kalimat berdasarkan spasi.

b. Output berupa list kata: $T(d_i) = \{w_1, w_2, ..., w_m\}$

Tabel 1.5 Hasil *Tokenizing*

Cleaning	*Tokenizing*
tadinya saya dapat promo yang besar tetapi setelah saya top up saldo gopay promonya mendadak jadi kecil ada juga kupon lain yang tersedia tetapi tidak dapat digunakan padahal sudah memenuhi persyaratan	['tadinya', 'saya', 'dapat', 'promo', 'yang', 'besar', 'tetapi', 'setelah', 'saya', 'top', 'up', 'saldo', 'gopay', 'promonya', 'mendadak', 'jadi', 'kecil', 'ada', 'juga', 'kupon', 'lain', 'yang', 'tersedia', 'tetapi', 'tidak', 'dapat', 'digunakan', 'padahal', 'sudah', 'memenuhi', 'persyaratan']

4. *Stopwords*

Pada tahap *stop-word removal*, kata-kata umum yang tidak memiliki makna signifikan terhadap konteks teks, atau disebut *stop-word*, dihilangkan dari daftar token yang telah melalui proses stemming. Penghapusan ini bertujuan untuk meningkatkan efisiensi dalam proses pelatihan model atau pengklasifikasian data teks (Sabrani et al., 2020). Dari tabel 5 kata-kata seperti "yang", "dan", "saya", "tidak", atau "ada" sering muncul dalam teks tetapi tidak memberikan nilai informasi yang signifikan.

Formulasi :

$SR(d_i) = \{w_j \in T(d_i) \mid w_j \notin SW\}$ (4)

a. SW adalah himpunan stopword bahasa Indonesia.

b. Kata-kata seperti "yang", "ada", "saya", "dapat", akan dihapus.

Tabel 1.6 Hasil *Stopwords*

Tokenizing	*Stopwords*
['tadinya', 'saya', 'dapat', 'promo', 'yang', 'besar', 'tetapi', 'setelah', 'saya', 'top', 'up', 'saldo', 'gopay', 'promonya', 'mendadak', 'jadi', 'kecil', 'ada', 'juga', 'kupon', 'lain', 'yang', 'tersedia', 'tetapi', 'tidak', 'dapat', 'digunakan', 'padahal', 'sudah', 'memenuhi', 'persyaratan']	['tadinya', 'promo', 'besar', 'top', 'up', 'saldo', 'gopay', 'promonya', 'mendadak', 'jadi', 'kecil', 'kupon', 'tersedia', 'digunakan', 'padahal', 'memenuhi', 'persyaratan']

5. *Stemming*

Stemming merupakan proses untuk memetakan atau menyederhanakan sebuah kata ke bentuk dasarnya. Tujuan dari proses ini adalah untuk menghapus berbagai jenis imbuhan, seperti awalan (*prefix*), akhiran (*suffix*), maupun gabungan keduanya (*confix*), yang terdapat pada setiap (Sari & Hayuningtyas, 2019). Tujuannya adalah agar kata-kata yang memiliki makna sama tetapi berbeda bentuk (seperti “digunakan”, “menggunakan”, “penggunaan”) dianggap satu representasi: yaitu “guna”. Dengan begitu, hasil analisis akan lebih konsisten dan tidak membedakan kata hanya karena variasi bentuknya.

Formulasi :

$S(d_i) = \{stem(w_j) \mid w_j \in SR(d_i)\}$ (5)

Keterangan:

a. Fungsi stem() mengubah kata ke bentuk dasar.

b. Misalnya, "tadinya" → "tadi", "digunakan" → "guna".

Tabel 6. Hasil *Stemming*

Stopwords	*Stemming*
['tadinya', 'promo', 'besar', 'top', 'up', 'saldo', 'gopay', 'promonya', 'mendadak', 'jadi', 'kecil', 'kupon', 'tersedia', 'digunakan', 'padahal', 'memenuhi', 'persyaratan']	['tadi', 'promo', 'besar', 'top', 'up', 'saldo', 'gopay', 'promonya', 'dadak', 'jadi', 'kecil', 'kupon', 'sedia', 'guna', 'padahal', 'penuh', 'syarat']

1.4 TF-IDF

Metode *Term Frequency-Inverse Document Frequency* (TF-IDF) digunakan untuk mengekstrak fitur-fitur penting dari teks jawaban kuesioner. TF-IDF mengukur tingkat kepentingan suatu kata dalam sebuah dokumen dibandingkan dengan keseluruhan kumpulan dokumen, dengan tujuan menghasilkan akurasi yang tinggi (Gautama, 2023).

1. *Term Frequency* (TF)

Mengukur seberapa sering sebuah term (kata) muncul dalam dokumen tertentu.

Formulasi :

$TF(t, d) = f(t, d) / \max\{f(w, d) \mid w \in d\}$ (6)

Keterangan:

a. f(t, d): jumlah kemunculan term t dalam dokumen d.

b. max{f(w, d)}: frekuensi maksimum dari semua term dalam dokumen d.

c. TF bernilai tinggi jika term sering muncul dalam dokumen.

2. *Inverse Document Frequency* (IDF)

Mengukur seberapa penting **sebuah** term dalam seluruh korpus.

Formulasi :

$$IDF(t) = \log(N / df(t)) \quad (7)$$

Keterangan:

a. N: jumlah total dokumen dalam korpus.
b. df(t): jumlah dokumen yang mengandung term t.
c. IDF tinggi jika term jarang muncul dalam dokumen.

3. Gabungan TF-IDF

TF-IDF memberikan bobot tinggi untuk kata yang sering muncul di dokumen tertentu namun jarang di seluruh korpus.

Formulasi :

$$TF\text{-}IDF(t, d) = TF(t, d) \times IDF(t) \quad (8)$$

Keterangan:

a. TF-IDF merupakan hasil perkalian antara nilai TF dan IDF dari sebuah term.
b. Makin tinggi nilai TF-IDF, makin penting kata tersebut untuk dokumen tersebut.