

## **BAB III**

### **METODOLOGI PENELITIAN**

#### **3.1 Objek Penelitian**

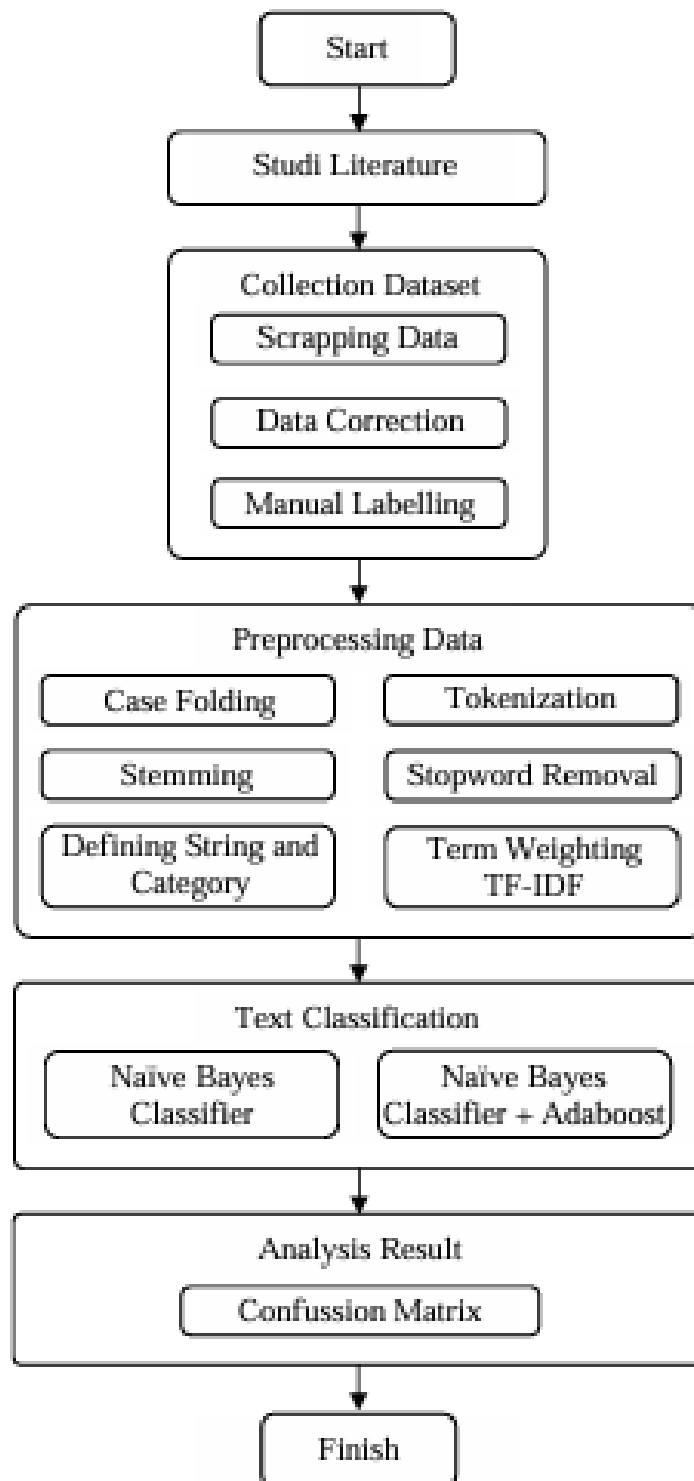
Objek penelitian ini adalah *review* ulasan aplikasi DANA yang dipergunakan untuk mengetahui tingkat kepuasan pengguna aplikasi DANA.

#### **3.2 Metode Pengumpulan Data**

Pengumpulan data yang dilakukan pada penelitian ini menggunakan teknik *Web Scrapping* dengan bahasa pemrograman *Python*. Metode tersebut merupakan proses pengambilan data dari sebuah website di dalam internet.

#### **3.3 Diagram Alir Penelitian**

Diagram alir penelitian bertujuan untuk menguraikan seluruh kegiatan yang akan dilakukan selama penelitian. Terdapat 5 proses utama dalam penelitian ini, yaitu Tahapan awal, *collection dataset* (Pengumpulan data), *Preprocessing data*, *Text classification*, dan *Analysis Result*. Metodologi penelitian yang digunakan dapat dilihat pada gambar 3.1 berikut ini :



Gambar 3.1 Metodologi Penelitian

### **3.3.1 Studi Literature**

Pada tahap ini melakukan studi literature untuk menentukan studi kasus serta metode yang akan digunakan saat melakukan penelitian. Studi literature dilakukan dengan cara pengumpulan jurnal, literature paper, makalah, buku, maupun situs internet sebagai sumber pustaka yang berkaitan dengan materi penulisan khususnya analisis sentiment menggunakan *Naïve Bayes Classifier* dan *Adaboost*.

### **3.3.2 Collection Dataset**

Pada tahapan *collection dataset* terbagi menjadi tiga proses yang akan dilakukan diantaranya sebagai berikut :

#### **3.3.2.1 Scrapping Data**

Proses pengumpulan data dilakukan dengan teknik *scrapping* yang bersumber dari ulasan *Google Play Store*. *Scrapping* data ulasan aplikasi DANA dari situs *Google Play Store* dilakukan dengan menggunakan *google colab* dengan bahasa pemrograman *python* dengan memanfaatkan *library google-play-scraper* dengan parameter `lang='id', country='id', sort=MOST.RELEVANT` dan `Count=10000 Filter_score_with=None`. Data yang diambil hanya kolom *Username, Rating, Tanggal* dan *Ulasan*.

#### **3.3.2.2 Data Correction**

Tahapan *Data Correction* dilakukan untuk membuang atribut yang tidak dibutuhkan meliputi penghapusan data yang terdeteksi duplikasi. Proses *data correction* dilakukan di MS.Excel dengan meng-*import* file csv hasil dari *scrapping data*.

### 3.3.2.3 Manual Labelling

Pelabelan dilakukan secara manual menggunakan *rating* ulasan sebagai acuan sentimen, yaitu *rating* 1, 2, 3 dikelompokkan menjadi sentimen negatif, sedangkan *rating* 4 dan 5 dikelompokkan menjadi sentimen positif.

### 3.3.3 Preprocessing Data

Preprocessing data dilakukan dalam beberapa proses bertujuan untuk menyiapkan data agar dapat diolah pada saat dilakukannya pemodelan. *Preprocessing data* melingkupi kegiatan memperbaiki data serta menghapus konten yang tidak diperlukan. Tahap *Text Preprocessing* bertujuan untuk mengurangi atribut yang kurang berpengaruh nantinya terhadap proses klasifikasi (Zulqornain and Adikara, 2021). *Text Preprocessing* dilakukan dengan bahasa pemrograman *Python*, berikut ini beberapa proses yang dilakukan :

#### 1) *Case Folding*

Pada tahap ini akan dilakukan pengkonversian teks ulasan kedalam format kecil (*lowercase*). Hal ini bertujuan untuk memberikan bentuk standar pada teks.

#### 2) *Tokenization*

Pada tahap ini akan dilakukan pemecahan teks dalam sebuah kalimat menjadi kata tunggal (*token*). Proses ini juga menghilangkan karakter yang dianggap tidak memiliki pengaruh terhadap pemrosesan teks.

#### 3) *Stopword Removal*

Pada tahap ini akan dilakukan penghapusan kata-kata yang terlalu umum dan tidak memiliki (atau sedikit) arti. Umumnya memiliki frekuensi kemunculan yang jumlahnya cukup banyak dibandingkan dengan kata lainnya.

#### 4) *Stemming*

Pada tahap ini akan dilakukan perubahan bentuk menjadi kata dasar dan menghilangkan kata imbuhan atau kata majemuk.

#### 5) *Difining String and Category*

*Difining string and category* dilakukan terhadap dataset baru hasil *preprocessing data* bertujuan agar mesin dapat membaca setiap kolom pada dataset dengan type yang telah ditentukan agar memudahkan pada tahap klasifikasi.

#### 6) *Term Weighting TF-IDF*

Pada tahap ini dilakukan pembobotan pada tiap kata, prosesnya yakni merubah kata menjadi bentuk *vector* yang mana setiap kata dihitung menjadi satu fitur. Term dapat berupa kata, frasa atau unit hasil indexing lainnya dalam suatu dokumen yang dapat digunakan untuk mengetahui isi atau konteks dari dokumen tersebut. setiap kata memiliki tingkat kepentingan yang berbeda dalam dokumen, maka untuk setiap kata diberikan sebuah indikator, yaitu *term weight*. Setiap kata yang kemunculannya lebih banyak dalam satu dokumen maka nilainya akan semakin tinggi. *Term Weighting TF-IDF* ini berpengaruh pada relevansi kata dalam dokumen, penurunan bobot untuk kata umum, identifikasi kata kunci, pemberian bobot yang sesuai dengan fitur, peningkatan performa model dan pengurangan noise. Pada penelitian ini proses pembobotan TF-IDF menggunakan bahasa pemrograman *Python* dengan memanfaatkan *library TFidfvectorizer*.

### **3.3.4 Text Classification**

Pada tahap ini dilanjutkan dengan membagi data menjadi data latih (*training*) dan data uji (*testing*). Hal ini dibutuhkan karena dalam algoritma klasifikasi yang digunakan merupakan model *supervised learning* menggunakan algoritma *Naïve Bayes Classifier* untuk melakukan proses *predicting sentiment*. Data digunakan untuk membuat model pembelajaran dan selanjutnya akan di evaluasi oleh data uji. Pada penelitian ini pembagian data diintegrasikan 80% untuk *data training* dan 20% untuk *data testing*, yang akan digunakan pada tahap pemodelan klasifikasi teks yaitu sebagai berikut:

#### **3.3.2.4 Klasifikasi teks menggunakan *Naïve Bayes Classifier***

Pada tahap klasifikasi algoritma yang digunakan adalah *Naïve Bayes Classifier*. Proses ini akan dilakukan langkah-langkah bagaimana algoritma NBC melakukan klasifikasi terhadap dataset yang diberikan. Klasifikasi dilakukan dengan bahasa pemrograman *Python* dengan memanfaatkan *library sklearn.naive\_bayes* untuk proses klasifikasi, selain itu menggunakan *library sklearn.metrics* untuk mengukur performa model untuk mengetahui nilai dari *Accuracy, precision, recall, dan F1-Score*.

#### **3.3.2.5 Klasifikasi teks menggunakan *Naïve Bayes Classifier* dengan *Adaboost***

Pada tahap klasifikasi algoritma yang digunakan adalah *Naïve Bayes Classifier* dengan metode *boosting* yaitu *Adaboost* dilakukan untuk bisa melihat kinerja dari algoritma yang digunakan serta metode *boosting* yang digunakan

berupa akurasi yang nantinya Adaboost ini dapat meningkatkan ketelitian dalam proses klasifikasi dan prediksi dengan cara membangkitkan kombinasi yang dipilih.

### **3.3.5 Analysis Result**

Tahapan *Analysis Result* terhadap hasil prediksi data yang telah di klasifikasikan menggunakan metode Naïve Bayes Classifier dengan membagi dataset menjadi 20% *data testing* dan 80% *data training*. Pada tahap ini juga akan dilakukan perbandingan kinerja klasifikasi *Naïve Bayes Classifier* dengan diterapkannya metode *boosting* dengan *Adaboost* dan tanpa menggunakan *Adaboost*. Hasil yang didapatkan akan ditampilkan berupa *confusion matrix*, dari nilai *confusion matrix* ini dihitung nilai *accuracy*, *precision*, *recall* dan *F1-Score*. Nantinya dari hasil tersebut dapat dianalisis bahwa hasil dari kedua metode tersebut apakah memiliki perbedaan hasil akurasi yang berbeda atau tidak.