BAB III

METODOLOGI PENELITIAN

3.1 Objek penelitian

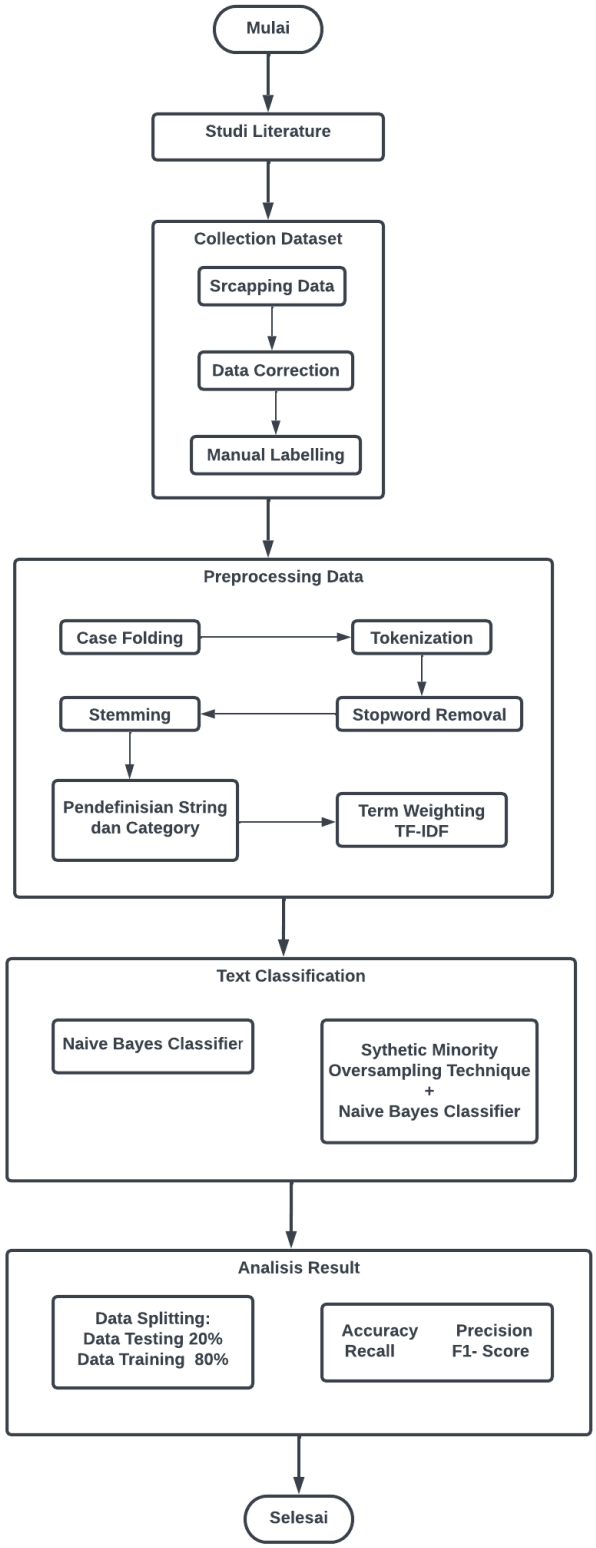
Objek pada penelitian ini adalah *review* ulasan aplikasi Peduli Lindungi yang dipergunakan dalam pelaksanaan surveilans Kesehatan oleh pemerintah Republik Indonesia dalam menangani penyebaran *corona Virus Disease* (COVID-19).

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 ini bertujuan untuk menguraikan seluruh kegiatan yang akan di lakukan selama penelitian. Terdapat lima proses utama dalam penelitian ini, yaitu Tahapan awal, *Collection dataset* (pengumpulan data), *Preprocessing data*, *Text classification*, dan Analisis *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 *sentiment analysis* menggunakan *Naïve Bayes Classifier* dan *Synthetic Minority Oversampling Technique (SMOTE).*

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 Peduli Lindungi 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 mengimport 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 pengubahan bentuk menjadi kata dasar dan menghilangkan kata imbuhan atau kata majemuk.

5) Pendefinisian *String* dan *Category*

Pendefinisian *String* dan *Category* dilakukan terhadap dataset baru hasil *preprocessing data* bertujuan agar mesin dapat membaca setiap kolom pada dataset dengan type yang telah di tentukan 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 di hitung 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 nilai nya akan semakin tinggi. Pengubahan setiap data menjadi vektor bertujuan untuk membantu algoritma klasifikasi membaca data tersebut. 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 dievaluasi 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:

3.3.4.1. 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.matrics* untuk mengukur performa model untuk mengetahui nilai dari *Accuracy, Precission, Recall, dan F1-Score.*

3.3.4.2. Klasifikasi teks menggunakan *Naïve Bayes Classifier* dengan *Synthetic Minority Oversampling Technique (SMOTE)*

Klasifikasi menggunakan Naïve Bayes Classifer dengan teknik *SMOTE* dilakukan dengan menguji coba model usulan terhadap dataset *Oversampling.* Algoritma yang digunakan yaitu Naïve Bayes Classifier dengan optimasi dataset

menggunakan Teknik *Oversampling* dengan metode *Synthetic Minority Oversampling Technique* (SMOTE). Klasifikasi dilakukan dengan bahasa pemrogaraman Python degan memanfaatkan *library imblearn.*

3.3.5. Analisis *Result*

Tahapan Analisis *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 teknik SMOTE dan tanpa menggunakan teknik SMOTE. Hasil yang didapatkan akan ditampilkan berupa *confusion matrix*, dari nilai *confusion matriks* ini dihitung nilai *Accuracy*, *Precission*, *Recall* dan *F1-Score.* Nantinya dari hasil yang didapatkan dapat dianalisis bahwa hasil dari kedua metode tersebut apakah memiliki perbedaan hasil akurasi yang berbeda atau tidak.