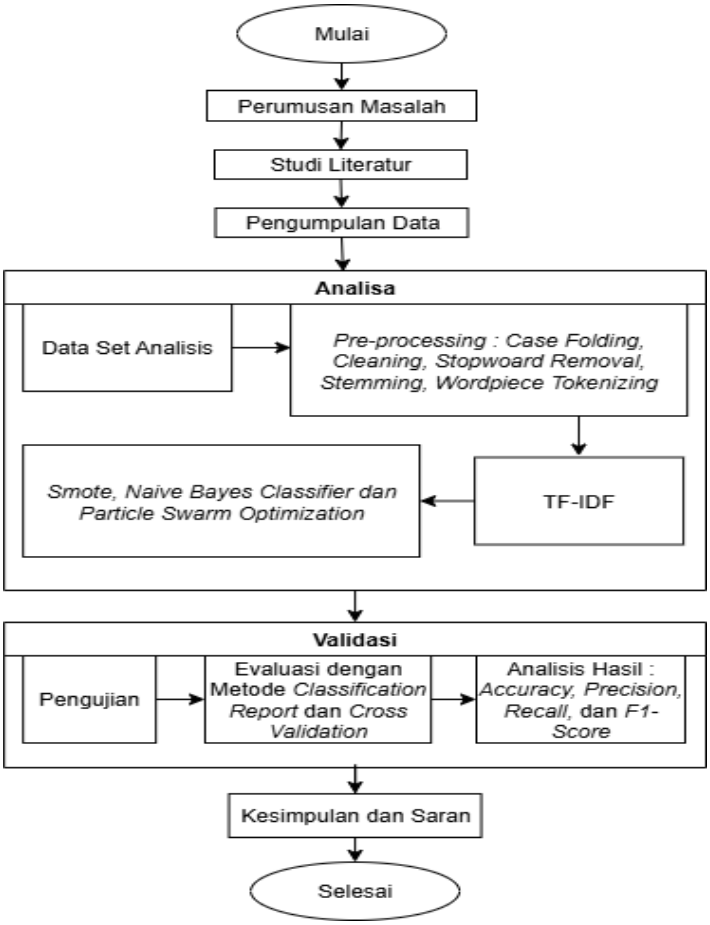
BAB III
METODOLOGI PENELITIAN

3.1 Metode Penelitian

Metode Penelitian merupakan pedoman atau tahapan-tahapan dalam melakukan penelitian. Metodologi penelitian ini dilakukan bertujuan untuk mendapatkan hasil yang sesuai dengan yang diinginkan. Penelitian yang dilakukan mengacu pada alur penelitian pada pendekatan *supervised learning* (Yadav, N., Kudale, O., Rao, A., Gupta, S., & Shitole, 2021). Proses-proses tersebut akan diuraikan menjadi alur metode penelitian yang ditunjukkan pada gambar 3.1.



Gambar 3. 1 Alur Metode Penelitian

Uraian dari gambar 3.1. dapat dilihat pada penjelasan berikut ini:

3.1.1 Perumusan Masalah

Perumusan masalah merupakan langkah awal dalam proses penelitian yang bertujuan untuk mengidentifikasi dan memahami permasalahan utama yang menjadi fokus penelitian. Perumusan masalah yang akan dilakukan dalam penelitian ini adalah bagaimana klasifikasi sentimen masyarakat terhadap pelayanan BPJS Kesehatan dapat dilakukan secara efektif, apakah opini yang disampaikan bersifat positif, netral, atau negatif. Tujuannya adalah untuk memperoleh wawasan yang komprehensif terkait persepsi publik terhadap layanan tersebut dan menemukan faktor-faktor yang memengaruhi pembentukan opini. Selain itu, penelitian ini juga bertujuan untuk menguji performa model klasifikasi yang digunakan yaitu *Naive Bayes Classifier* (NBC) yang dioptimasi menggunakan *Particle Swarm Optimization* (PSO) dalam menghasilkan tingkat performa terbaik melalui pendekatan berbasis data teks mengenai Pelayanan BPJS Kesehatan.

3.1.2 Studi Literatur

Studi literatur merupakan cara yang digunakan untuk menghimpun data atau sunber yang berhubungan dengan judul yang dipakai dalam penelitian ini. Studi literatur dapat diperoleh dengan berbagai sumber antara lain:

1. Buku/*e-Book* yang digunakan untuk membahas tentang *text mining* khususnya algoritma *Naive Bayes Classifier* (NBC) dan *Particle Swarm Optimization* (PSO).

2. Jurnal yang mengandung tentang klasifikasi dan analisis sentimen menggunakan algoritma *Naive Bayes Classifier* (NBC) dan *Particle Swarm Optimization* (PSO).

3.1.1. Pengumpulan Data

Pada tahapan ini dilakukan pengumpulan data dengan cara melakukan proses *crawling* menggunakan *Tweet Harvest* berfungsi sebagai *Twitter Crawler*, yaitu alat yang dapat mengumpulkan *tweet* berdasarkan kata kunci, rentang waktu, atau parameter lainnya dengan cara memiliki API Key dan *Access Token* agar bisa mengakses data dari X kemudian data diambil dengan kata kunci "pelayanan bpjs" dengan waktu pengambilan data selama tahun 2024.

3.1.3 *Preprocessing*

Sebelum melakukan proses analisis sentimen, data yang telah dikumpul melalui proses *Crawling data Twitter* selanjutnya dilakukan pembersihan data. Tujuan dari *pre-pocessing* adalah untuk membersihkan dan mengubah teks mentah menjadi bentuk yang lebih terstruktur dan siap digunakan dalam tahap selanjutnya. Tahapan *preprocessing* yang dilakukan pada tahapan ini adalah sebagai berikut:

a. *Case folding*

Case folding merupakan proses mengubah semua huruf dalam teks menjadi huruf kecil agar tidak ada perbedaan antara huruf besar dan kecil dalam analisis.

b. *Cleaning*

Cleaning merupakan proses menghilangkan karakter khusus, tanda baca, atau simbol yang tidak relevan atau mengganggu dalam analisis.

c. *Stopword Removal*

Stopword removal merupakan tahapan penghapusan kata-kata umum yang tidak memberikan kontribusi signifikan dalam pemahaman teks, seperti "dan", "atau", "yang", dll.

d. *Stemming*

Stemming adalah proses mengubah variasi bentuk kata ke bentuk kata bakunya semisal menulis menjadi tulis atau pengaturan menjadi atur.

e. *Wordpiece tokenizing*

Wordpiece tokenizing merupakan teknik tokenisasi teks dalam pemrosesan bahasa alami yang sering digunakan, misalnya, untuk membuat *wordcloud*. Metode ini membagi teks menjadi unit-unit kecil yang disebut *wordpieces*, yang dapat berupa kata utuh maupun potongan kata. *Wordpiece tokenizing* berguna dalam berbagai aplikasi, seperti pemodelan bahasa, penerjemahan mesin, dan tugas pemrosesan bahasa alami lainnya. Salah satu keunggulannya adalah kemampuannya menangani kata-kata yang jarang muncul atau kata-kata yang tidak ada dalam kamus.

3.1.4 Labelisasi Data

Setelah data dibersihkan, Langkah berikutnya adalah melakukan labelisasi, yaitu memberikan label sentimen pada setiap data teks berdasarkan kategori tertentu, seperti positif (1), netral (0), dan negatif (-1). Proses ini dilakukan menggunakan *TextBlob,* yang merupakan pustaka pemrosesan bahasa alami untuk menentukan label kelas sentimen secara otomatis (Shahputri & Yamasari, 2023). Meskipun pelabelan dilakukan secara otomatis, pengecekan manual pada sebagian

data tetap disarankan untuk menghindari kesalahan klasifikasi yang dapat terjadi karena konteks bahasa yang ambigu atau kata-kata idiomatik yang tidak dipahami oleh model terjemahan dan *TextBlob*.

3.1.5 Data Transformasi

Transformasi data merupakan proses mengubah data yang telah dipilih ke dalam format yang dapat digunakan oleh model analisis sentimen. Proses ini melibatkan konversi teks menjadi representasi numerik atau fitur yang dapat diproses oleh algoritma. Transformasi ini sangat penting karena algoritma pembelajaran mesin dan metode statistik umumnya tidak dapat langsung mengolah data teks mentah tanpa adanya transformasi yang sesuai.

Tahap transformasi mencakup teknik ekstraksi fitur yang bertujuan untuk mengonversi data teks ke dalam bentuk yang lebih sesuai untuk analisis sentimen. Salah satu metode yang digunakan adalah TF-IDF (*Terms Frequency-Inverse Document Frequence*), yaitu teknik yang menghitung frekuensi kemunculan suatu kata dalam sebuah dokumen (*Term Frequency*) dan mengoreksinya dengan *Inverse Document Frequency*, yang mengukur seberapa umum atau langka kata tersebut dalam keseluruhan dataset. Dengan cara ini, kata-kata umum seperti “dan” atau “dari” yang sering muncul dalam banyak dokumen tidak akan dianggap terlalu penting dalam analisis.

3.1.6 Penyeimbang Data Kelas Sentimen (SMOTE)

Metode SMOTE *(Synthetic Minority Oversampling Technique)* digunakan untuk mengatasi ketidakseimbangan kelas dengan menambah sampel baru pada

kelas minoritas secara sintetis. Penambahan ini dilakukan agar distribusi jumlah data antar kelas menjadi lebih seimbang dan model dapat mengenali pola dari seluruh kelas secara lebih optimal..

Penerapan SMOTE dilakukan pada data latih, baik pada pengujian *train-test split* maupun *10-Fold Cross Validation*. Pembatasan ini bertujuan untuk menghindari kebocoran data *(data leakage)* yang dapat terjadi apabila proses *oversampling* diterapkan pada keseluruhan dataset. Dengan demikian, model hanya mempelajari pola dari data latih hasil oversampling tanpa terpengaruh oleh data uji.

3.1.7 *Particle Swarm Optimization* (PSO)

Optimasi yang dilakukan dalam penelitian ini adalah *Particle Swarm Optimization* (PSO). PSO merupakan algoritma metaheuristik yang meniru perilaku sosial kawanan burung untuk menemukan nilai parameter terbaik. PSO digunakan untuk mencari nilai alpha yang memaksimalkan skor *F1-macro*, sehingga menghasilkan performa model klasifikasi yang lebih optimal dibandingkan dengan penggunaan nilai default.

Pada penelitian ini, parameter α (alpha) dipilih sebagai satu-satunya hyperparameter yang dioptimasi karena α merupakan parameter utama pada *Naive Bayes Classifier* (NBC) yang berperan dalam *Laplace smoothing* untuk mengatasi masalah probabilitas nol pada data teks hasil ekstraksi TF-IDF. Selain itu, jumlah *hyperparameter* pada *Naive Bayes Classifier* (NBC) relatif terbatas, sehingga optimasi difokuskan pada α untuk menjaga efisiensi dan kestabilan proses optimasi.

Optimasi parameter *alpha* pada algoritma *Multinomial Naive Bayes Classifier* (NBC) dilakukan menggunakan *Particle Swarm Optimization* (PSO), di mana setiap partikel merepresentasikan kandidat nilai alpha yang diinisialisasi secara acak dalam rentang [0, 2] dengan kecepatan awal acak di [-0,01, 0,01]. Algoritma dijalankan hingga maksimum iterasi, dengan pembaruan posisi partikel berdasarkan *inertia weight* serta komponen kognitif dan sosial, sehingga partikel yang menghasilkan *F1-score macro* tertinggi menentukan nilai *alpha* terbaik. Untuk menjaga reproduktifitas, pembagian data latih dan uji dilakukan menggunakan *random seed = 42*, dan nilai *alpha* dibatasi agar tetap berada dalam rentang valid [0,0001, 2]. *Hyperparameter* PSO yang digunakan sebagai berikut:

1. Jumlah partikel *(swarm_size)*: 10
2. Maksimum iterasi *(max_iter)*: 20
3. *Inertia weight (w)*: 0,5
4. Koefisien kognitif (c1): 1
5. Koefisien sosial (c2): 2
6. Rentang pencarian *alpha*: [0,0001, 2]
7. *Random seed*: 42
8. Fungsi *fitness: F1-score macro*

3.1.8 Penerapan Algoritma

Penelitian ini menggunakan algoritma dengan pendekatan *supervised learning* yaitu *Naive Bayes Classifier* (NBC). Proses yang terjadi dalam algoritma ini adalah dengan memberikan model dari data latih yang sudah diberikan lebel sentimennya dan kemudian model akan belajar dari data tersebut untuk dapat

memprediksi sentimen pada data uji. Model ini mengasumsikan independensi antar fitur dan menghitung probabilitas untuk menentukan kelas sentimen berdasarkan kata-kata yang muncul dalam teks.

3.1.9 Evaluasi

Evaluasi bertujuan untuk melihat hasil kinerja dari model yang digunakan dengan hasil yang telah ada sebelumnya. Evaluasi yang digunakan dalam penelitian ini adalah *Cross Validation*, evaluasi tersebut dilakukan menggunakan teknik *Stratified K-Fold Cross Validation* dengan k=10 untuk memastikan bahwa pembagian data latih dan uji tetap mempertahankan proporsi kelas. Dengan teknik ini, data dibagi menjadi 10 bagian dan proses pelatihan serta pengujian dilakukan sebanyak 10 kali secara bergantian, sehingga hasil evaluasi menjadi lebih stabil dan menghindari *overfitting* terhadap satu set data uji. Selain *Cross Validation* penelitian ini juga menggunakan *classification report*. Dalam evaluasi tersebut menghasilkan :

1. *Accuracy* memberikan gambaran umum terhadap kinerja model.
2. *Precision* mengevaluasi seberapa tepat model dalam memprediksi suatu kelas.
3. *Recall* mengukur kemampuan model dalam menemukan semua data dari suatu kelas.
4. *F1-Score* memberikan keseimbangan antara *precision* dan *recall.*