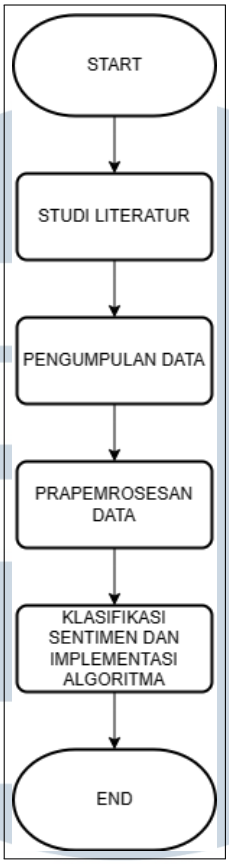
BAB 3
METODOLOGI PENELITIAN

3.1 Metode Penelitian

Metode penelitian yang dilakukan adalah melakukan studi literatur, lalu langkah kedua adalah mengumpulkan atau mengambil data dari platform *twitter*. Setelah itu, langkah ketiga adalah melakukan *preprocessing* pada data yang sudah diambil, dan langkah terakhir adalah mengklasifikasikan hasilnya ke algoritma yang digunakan untuk perbandingan hasil akurasi serta evaluasi model *machine learning*.

3.2 Alur Penelitian

Alur metode penelitian yang dilakukan dimulai dari studi literatur, pengumpulan data, proses pengolahan data / prapemrosesan data, serta klasifikasi sentimen dan implementasi algoritma seperti pada sebuah *flowchart* atau diagram alir yang ditampilkan pada Gambar 3.1.



Gambar 3.1. Diagram alir metodologi penelitian.

Flowchart atau diagram alir metode penelitian yang dilakukan seperti pada Gambar 3.1 dimulai dari pemahaman mengenai teori dan literatur jurnal penelitian sebelumnnya dan yang ada di publikasi, mengumpulkan data dan menyaring serta membersihkan data yang didapat dari platform *twitter*, membersihkan dan menyaring data serta menampilkan visualisasi data atau yang bisa disebut *text mining*, lalu memproses data tersebut menjadi lebih bersih dan tersaring dalam *text preprocessing*, dan terakhir adalah mengklasifikasikan sentimen serta implementasi kedua algoritma pada data hasil *preprocessing* tersebut.

3.2.1 Studi Literatur

Studi literatur adalah proses mencari dan mempelajari teori dan literatur yang berasal dari jurnal, tesis, maupun penelitian lain yang sudah dipublikasi dan akan digunakan dalam penelitian ini. Literatur yang dibutuhkan adalah definisi dan penjelasan mengenai platform media sosial khususnya *Twitter*, analisis sentimen, dan metode klasifikasi algoritma *Naive Bayes* dan *Support Vector Machine*.

3.2.2 Pengumpulan Data

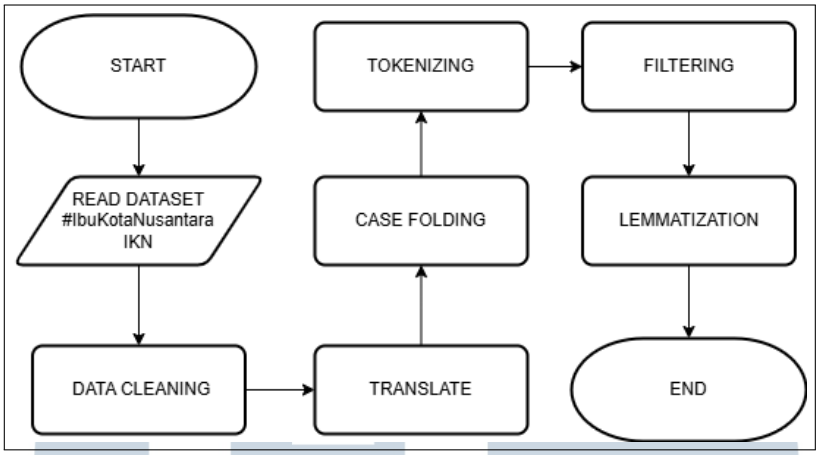
Pengumpulan data dilakukan dengan menggunakan *tools* yang bernama *tweet harvest* yang dibuat oleh Helmi Satria yang dapat *crawling data* di *twitter* mencapai kurang lebih sekitar 2000 data pada situs *twitter.com* dan data dicari mulai dari tahun 2019 serta berbahasa Indonesia. Alasan mengapa mengambil data menggunakan *tools* ini adalah karena *tweet harvest* merupakan salah satu *tools* yang mudah digunakan untuk *crawling data* dengan memasukkan *auth_token* akun *twitter* pribadi untuk melakukan *crawling*, serta mengapa mengambil data dari tahun 2019 adalah karena rencana pemindahan ibukota diresmikan pada tanggal 29 April 2019 oleh Presiden Joko Widodo [1].

Terdapat beberapa jurnal penelitian yang menggunakan *tweet-harvest* ini untuk melakukan pengumpulan data di platform *twitter*. Penelitian pertama menganalisis sentimen *tweet* mengenai *chatgpt* menggunakan EDA di Indonesia. Data yang dikumpulkan adalah *ChatGPT* [29]. Penelitian kedua yang menggunakan *tweet-harvest* yang menganalisis sentimen pengaruh jam kerja terhadap kesehatan mental generasi Z. Penelitian tersebut mengumpulkan data tentang generasi z atau gen z [30].

Jumlah data yang terkumpul sedikit terbatas dibanding waktu sebelum Elon Musk menjadi pemilik *twitter* yang baru karena adanya larangan baru yang diumumkan oleh Elon Musk yaitu termasuk jumlah data yang dapat diambil menjadi jauh lebih sedikit dibanding sebelumnya untuk mengurangi *data scraping* yang sering terjadi di *twitter* mengacu sumber dari https://twitter.com/elonmusk/status/1675187969420828672.

3.2.3 Proses Pengolahan Teks (*Text Preprocessing*)

Setelah mendapatkan data dari *data crawling* menggunakan *tweet-harvest*, dilakukan *preprocessing* pada data untuk membersihkan dan membuat data menjadi lebih bersih dan rapih. Alur *preprocessing* diterapkan seperti pada Gambar 3.2.



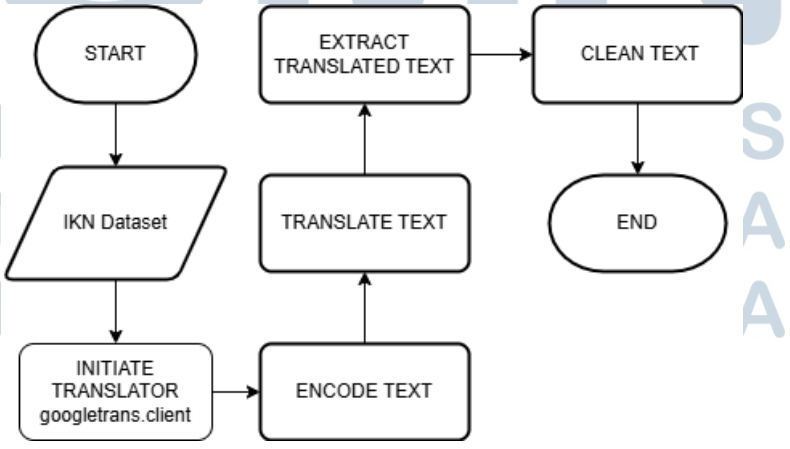
Gambar 3.2. Diagram alir penerapan *Text Preprocessing*.

1. ***Data Cleaning***

   Penerapan *preprocessing* ini dilakukan pertama kali dengan melakukan pembersihan data atau *data cleaning* menggunakan *python library re* atau *regular expression* untuk menghapus berbagai karakter selain huruf atau kata yang bisa merusak klasifikasi dan pelabelan sentimen nanti.

2. ***Translate***

   Setelah membersihkan data, dilakukan terjemahan data yang masih dalam bahasa Indonesia ke bahasa Inggris menggunakan *python library googletrans* dan proses terjemahan tersebut bisa memakan banyak waktu tergantung jumlah data. *translate* dilakukan dalam penelitian ini karena *VADER* bisa melakukan *labelling* pada data yang berbahasa Inggris karena *lexicon*-nya mengandung bahasa Inggris. Berikut adalah *flow translate* yang akan dilakukan:



Gambar 3.3. *Flow* terjemahan data menggunakan *google.trans*

Berdasarkan Gambar 3.3, dapat dijabarkan alur terjemahan mulai dari langkah pertama adalah inisialisasi objek *translator* dari *googletrans.client*, kemudian langkah kedua adalah melakukan *encoding* pada teks dalam format *ASCII* dan mengabaikan karakter yang tidak bisa di-*encode*. Pada langkah ketiga, hasil *encode* tersebut yang berupa teks akan dilakukan proses terjemahan dari bahasa Inggris ke Indonesia. Langkah keempat adalah mengekstrak atribut 'text' dari objek hasil terjemahan. Langkah terakhir adalah memastikan data hasil terjemahan tidak ada *mention*, karakter non-alphanumeric, dan *URL* dengan melakukan *cleaning text*.

3. ***Case Folding***

   Proses selanjutnya adalah *case folding* yang merupakan proses perubahan huruf besar yang ada pada setiap data hasil terjemahan menjadi huruf kecil (*lowercase*). Contohnya adalah terdapat kalimat "*These are the 2 units of Landed Houses*", setelah diubah menjadi "*these are the units of landed houses*".

4. ***Tokenizing***

   Proses *tokenizing* dilakukan dengan pemotongan *string* pada setiap kata yang menyusunnya dari data hasil *case folding*.Sebagai contohnya terdapat hasil *case folding* "*these are the units of landed houses*" setelah data tersebut di-*tokenized* menjadi "[these, are, the, units, of, landed, houses]". Data tersebut menjadi terpisah dan membentuk sebuah *array* yang berisi kumpulan kata dari satu baris data tersebut.

5. ***Filtering***

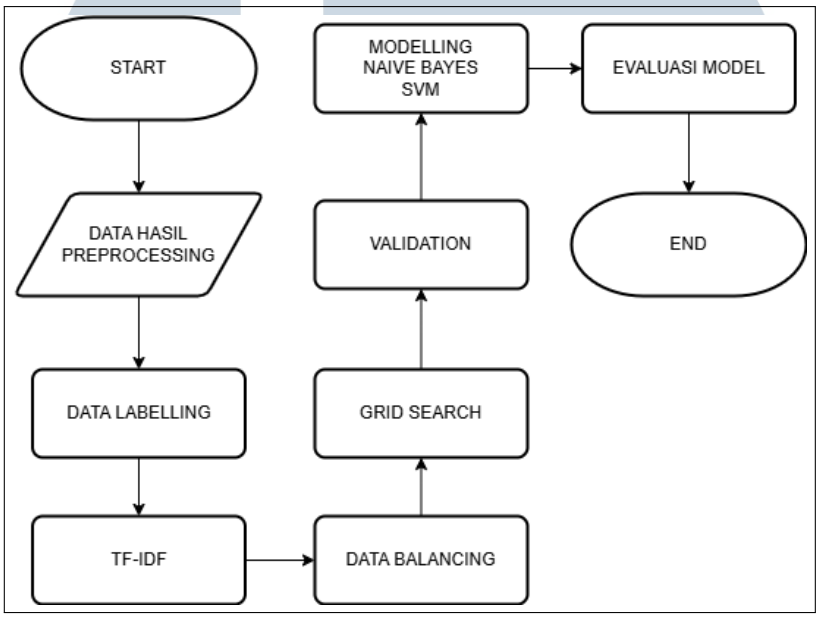
   Pada tahap *filtering* dilakukan dengan menghapus kata - kata penghubung atau pelengkap. Sebagai contoh terdapat hasil *tokenize* "[these, are, the, units, of, landed, houses, for, ministerial]", setelah dilakukan *filtering* menjadi "[units, landed, houses, ministerial]".

6. ***Lemmatization***

   Pada proses *lemmatizing*, Kata - kata yang memiliki kesamaan makna dengan kata dasarnya akan dikelompokkan, meskipun memiliki variasi bentuk atau tata bahasa yang berbeda karena adanya penambahan imbuhan yang beragam dan berubah menjadi bentuk kata dasarnya. Sebagai contoh hasil *lemmatizing* adalah "[units, landed, houses]" menjadi "unit, land, house]".

3.2.4 Klasifikasi Sentimen dan Implementasi Algoritma

Pada langkah ini, dilakukan analisis sentimen dan penerapan metode klasifikasi untuk mengevaluasi pendapat masyarakat Indonesia terkait pemindahan ibukota negara. Terdapat dua algoritma yang diimplementasikan dalam penelitian klasifikasi ini, yaitu *Naïve Bayes*, dan *Support Vector Machine*.



Gambar 3.4. *Flowchart* klasifikasi sentimen dan implementasi kedua algoritma.

Hasil analisis sentimen dibagi menjadi tiga kategori, yakni positif, negatif, dan netral. Proses klasifikasi dan implementasi algoritma pada dataset diterapkan pada *flowchart* pada Gambar 3.4. *Data labelling* akan menggunakan *VADER Lexicon* yang merupakan *library* dari *nltk* untuk menetapkan sentimen pada tiap data.

1. ***Data Labelling***
   Proses pemberian label pada setiap baris data disebut proses labelling. Setiap data memiliki kata-kata yang dapat dihitung skor sentimennya dengan membandingkan data hasil *translate* menggunakan *python library VADER Lexicon* dari *nltk* untuk *labelling*. Data tersebut akan dikategorikan menjadi tiga sentimen yaitu positif, negatif, dan netral. Jika skor data bernilai lebih dari 0.05, data tersebut termasuk ke dalam sentimen positif. Jika skor data bernilai kurang dari -0.05, data tersebut termasuk dalam negatif, dan selain skor validasi tersebut data bersifat netral.

2. ***TF-IDF***

   Setelah data diberi label, langkah selanjutnya adalah melakukan pembobotan *TF-IDF*. Proses ini akan memberikan bobot nilai pada data yang melibatkan perhitungan statistik untuk menilai seberapa penting suatu kata dalam dokumen data. Dalam penelitian ini, penerapan ekstraksi fitur dilakukan dengan visualisasi dan *Term Frequency - Inverse Document Frequency* atau disingkat *TF-IDF* menggunakan *python library sklearn* yaitu *TF-IDF Vectorizer*.

3. ***Data Balancing***

   Setelah pemberian label dan pembobotan *TF-IDF*, langkah selanjutnya adalah memastikan bahwa data seimbang antara kelas positif, netral, dan negatif. Hal ini dapat dilakukan dengan teknik seperti *oversampling* kelas minoritas atau *undersampling* kelas mayoritas untuk menghindari bias dalam model. Untuk menerapkan *Data Balancing*, digunakan *SMOTE* dari *library imblearn* untuk melakukan *oversampling*.

4. ***Validation***

   Validasi model dilakukan menggunakan teknik *Stratified K-Fold Cross Validation* dengan *split* 10. Metode ini membagi dataset menjadi 10 bagian (*folds*) yang sama besar dan menjaga proporsi kelas yang sama di setiap *fold*. Pada setiap iterasi, model dilatih menggunakan sembilan *fold* dan diuji pada satu *fold* yang tersisa. Proses ini diulang sebanyak 10 kali sehingga setiap *fold* menjadi data uji satu kali. Akurasi rata-rata dari semua iterasi dihitung untuk mengevaluasi performa model secara keseluruhan. Teknik ini membantu memastikan bahwa model tidak *overfitting* dan memiliki kemampuan generalisasi yang baik pada data yang tidak terlihat sebelumnya.

5. ***Grid Search***

   Tahap selanjutnya adalah pemberian parameter yang tepat untuk kedua model algoritma *Naive Bayes* dan *Support Vector Machine* untuk diterapkan pada metode *GridSearchCV* untuk bisa mencari dan mendapatkan *hyperparameter* terbaik yang diterapkan di model algoritma *SVM* dan *Naive Bayes*.

6. ***Modeling Naive Bayes dan SVM***

   Data dibagi menjadi data *training* dan data *testing*. Model dilatih menggunakan data *training* sebesar 80%, kemudian diuji dengan data *testing*. Sebanyak 20% dari data *testing* digunakan sebagai bahan *testing* terhadap

model *Bernoulli Naïve Bayes* dan model *Support Vector Machine* (SVM) yang digunakan dalam penelitian ini. Setelah model dilatih, kedua model diuji menggunakan data *testing* yang sama dengan menggunakan *confusion matrix*. Terdapat beberapa skenario tambahan untuk pelatihan dan pengujian data seperti menggunakan perbandingan 90:10, 80:20, dan 70:30 data *train* dan *test*.

7. **Evaluasi Model**

   Mengevaluasi dan menganalisis kinerja kedua model yang telah dibangun. Evaluasi model bertujuan untuk memperoleh informasi dan pemahaman lebih tentang kelebihan dan kekurangan dari penerapan model *Naïve Bayes* dan *Support Vector Machine* (SVM). Evaluasi model dihasilkan dalam bentuk *confusion matrix* dan *ROC-AUC Score*.