CHAPTER 3

RESEARCH METHODOLOGY

3.1 Introduction

This chapter explains the research methodology used to analyze sentiment related to the "free meal" program promoted by Prabowo and Gibran. Also public reactions to the program through social media, especially through the X application or Twitter. This methodology includes the process of data collection, data pre-processing, data modelling, to classification using machine learning techniques to identify sentiment patterns (positive, negative, or neutral). This study aims to generate meaningful insights from social media data related to public sentiment towards the program.

3.2 Research Framework

This research framework includes the following steps:

1. Problem Definition and Literature Review
2. Data Collection: Retrieve data from Twitter using specific keywords.
3. Data Pre-processing: Cleaning and preparing data for further analysis.
4. Feature Extraction: Applying stemming and vectorization techniques.
5. Sentiment Classification: Using machine learning models (KNN, Naive Bayes, and SVM).
6. Model Evaluation: Compares model performance using evaluation matrices.

The details of the research framework for this study are shown in Figure 3.1.

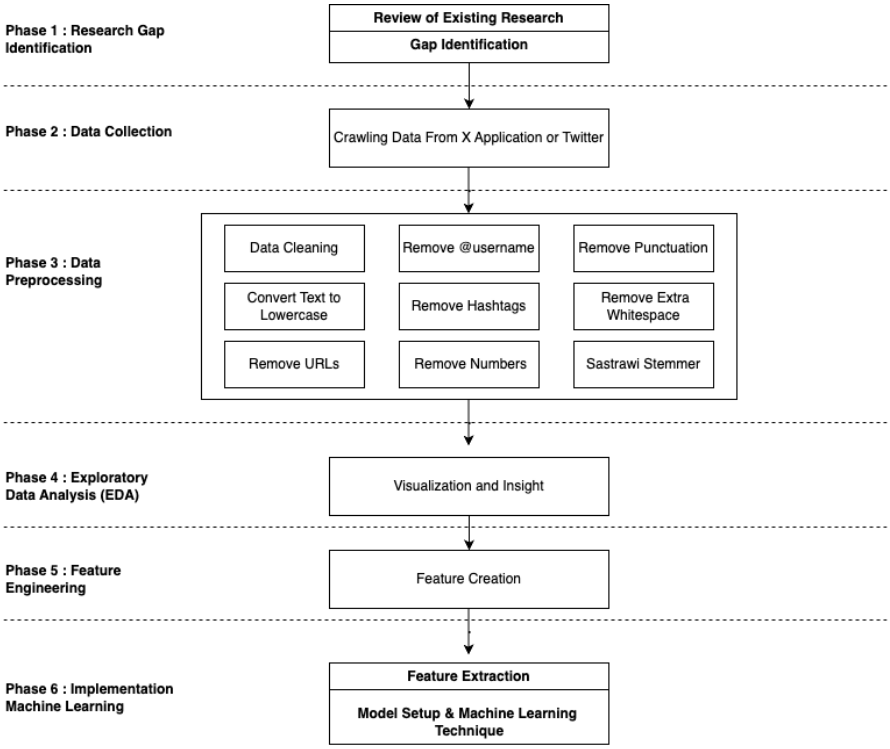

Figure 3.1 Research Framework For Sentiment Analysis

3.3 Problem Formulation

The main objective of this study is to use a sentiment analysis approach to public reactions on social media with machine learning technique classification, thus providing valuable data for further government policies. However, to ensure accurate and reliable analysis, several problems need to be solved.

a. Identifying public sentiment regarding the "free meal" program.

b. Comparing the performance of KNN, Naive Bayes, and SVM algorithms in sentiment classification based on Twitter data.

3.4 Data Collection

Data was collected from the Twitter platform using Crawling Data Technique. The keywords used for crawling data are:

a) "Free meal"

b) "Free school meal program"

c) "Prabowo Gibran"

The data collected covers the time span from 2023 to 2025, as well as data prior to 2023 to provide historical context. Information taken includes:

a) Text tweet

b) Posting date

c) Username

d) Number of retweets and likes

The following dataset was obtained by crawling data process on application X or twitter. The data obtained is related to tweets about Prabowo and Gibran's free meal program. This data collected from 2023 until January 2025.

```
from google.colab import drive
drive.mount('/content/drive')
```

```
Mounted at /content/drive
```

```
import pandas as pd
import numpy as np
import re
from Sastrawi.Stemmer.StemmerFactory import StemmerFactory

data = pd.read_csv('/content/drive/My Drive/content/dataset/final_dataset.csv')
data
```

Figure 3.2 Load The Dataset

The total dataset that has been collected is 2916 rows of data with 15 columns as shown in the Figure 3.3

2	1866464297657975123	Wed Dec 11 21:21:34 +0000 2024	0	@StevanFirman15 @prabowo Yakin pak makan grati...	1866956515414094156			
3	1866754957552259364	Wed Dec 11 18:42:05 +0000 2024	0	@03__nakula Makanya program Prabowo makan sian...	1866916381872230896	NaN	03__nakula	
4	1866877557489926164	Wed Dec 11 16:07:49 +0000 2024	0	Prancis Dukung Program Makan Bergizi Gratis Pr...	1866877557489926164	NaN	NaN	
...	...	...	...	...	...	...	...	
2911	1851865142977310806	Thu Oct 31 05:53:50 +0000 2024	0	saltingan banget dia kenapa sihhh dari dulu ga...	1851865142977310806	NaN	NaN	
2912	1851865037457231952	Thu Oct 31 05:53:25 +0000 2024	2	hi apa masi pd kenal ak ABIS GANTI AVAA	1851865037457231952	NaN	NaN	
2913	1851806115144798370	Thu Oct 31 05:53:08 +0000 2024	0	@abu_waras @prabowo Banyak yang belum sadar ba...	1851864964790919221	NaN	abu_waras	
2914	1851834028288057434	Thu Oct 31 05:52:57 +0000 2024	0	@jaeminmna @bulanrpw Kamu udah makan? Udah say...	1851864921203462351	NaN	jaeminmna	
2915	1851864884654326214	Thu Oct 31 05:52:49 +0000 2024	0	Keluarga Besar Lapas Kelas IIB Brebes Siap Men...	1851864884654326214	https://pbs.twimg.com/media/GbMlOEwakAI_aR7.jpg	NaN	

2916 rows × 15 columns

Figure 3.3 The Dataset Preview

3.5 Data Pre-Processing

Initial analysis needs to be completed before moving on to further pre-processing. Data merging procedures are required to unify all the raw data into a single data frame once we have a good understanding of the features available in the data set. Several data processing and data transformation procedures will be used on the data set in an attempt to further unify the disorganized raw data. Table 3.1 lists every detail of the data pre-processing that was used.

Table 3.1 Data Pre-processing Methods

Data Pre-Processing	Purpose
Preliminary Analysis	To evaluate the provided dataset and obtain insightful knowledge for the modelling phase that follows
Data Cleaning	Find the missing value and eliminate the rows that do not have it
Data Visualization	A pie chart illustrating the trend of each variable for sentiment analysis free meal program.

3.5.1 Preliminary Analysis

Preliminary analysis is an important step in any data analysis because it helps to become familiar with the data set, understand its structure, format, and the types of variables it contains. Preliminary investigations can identify problems that must be corrected for a reliable analysis, such as missing values, outliers, or contradictions.

In this initial analysis process there are 2 stages that will be carried out, namely:

a. Identify common patterns in raw data.
b. Evaluate data distribution by time and keywords.

3.5.2 Data Cleaning

Data cleaning is an important process in sentiment analysis, especially to ensure that the data used is clean, relevant, and can be processed well by the model. Here are the data cleaning steps carried out on the Twitter tweet dataset about the Prabowo-Gibran free meal program:

1. **Initialize Sastrawi Stemmer**

   Sastrawi is used to perform stemming, which is changing affixed words into basic forms (root words). For example, "makannya" becomes "makan". Using Stemming helps simplify word variations so that the model can more easily recognize patterns in the data.

2. **Convert Text to Lowercase**

   All letters in the text are converted to lowercase. Makes the analysis more consistent because uppercase and lowercase are treated the same. For example, "PRABOWO" and "prabowo" are considered the same.

3. **Remove URLs**

   Removes links (URLs) from text such as "https://...". Links do not provide relevant sentiment information and can interfere with analysis.

4. **Remove @username**

   Removes mentions or tags such as "@user". Mentions are usually not relevant for sentiment analysis because they only point to a specific account.

5. **Remove Hashtags**

   Removing hashtags such as "#prabowo" or "#makangratis". Hashtags can be removed because they often do not contain the context needed in sentiment analysis, although there are certain cases where hashtags are analyzed separately.

6. **Remove Numbers**

   Removes numbers from text. Numbers usually have no meaning in the context of sentiment, unless specifically relevant (can be processed separately if important).

7. **Remove Punctuation**

   Removes punctuation such as ".", ",", "?", etc. Punctuation does not contribute directly to sentiment analysis.

8. **Remove Extra Whitespace**

   Removes excess whitespace in text. Makes text neater and easier to read.

9. **Apply Stemming**

   Uses the Sastrawi stemmer to convert words to their basic form. Reduces variations in words that have the same meaning.

10. **Apply Preprocessing**

    Combines all the above steps into one preprocessing pipeline that is applied to the entire dataset. Ensures all data is processed in a uniform manner.

11. **Translate Data to Minimize English Words**

    Translates English words to Indonesian using a library such as the Google Translate API. Standardizes the language so that all text is in one language (Bahasa Indonesia) to facilitate sentiment analysis.

In the Figure 3.4, it explains the flow of the data cleaning process with a literary stemmer to the process of minimizing words in English.

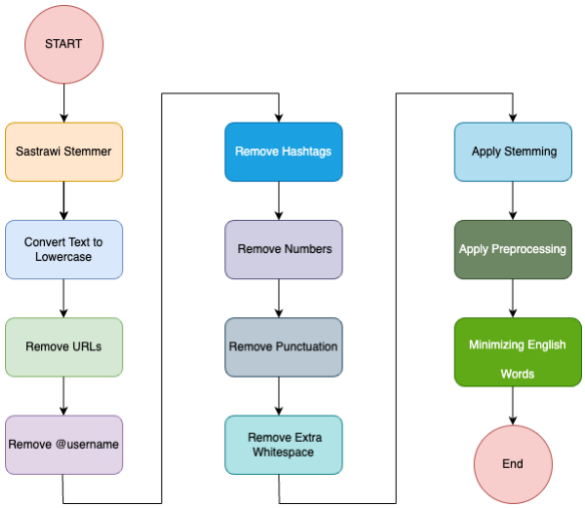

Figure 3.4 Flow Data Cleaning and Preparation

To identify missing values and remove rows and columns without values, data cleaning is done in this section. Figure 3.5 shows that in data pre-processing, several things are done such as converting text to lower case, removing URLs, removing @username, removing hashtags, removing numbers, removing punctuation and removing extra whitespace. Then apply all the pre-processing processes with the syntax `data['full_text'] = data['full_text'].apply(preprocess_text)`.

```
# Initialize Sastrawi stemmer
factory = StemmerFactory()
stemmer = factory.create_stemmer()

# Preprocessing function for tweets
def preprocess_text(text):
    text = text.lower()  # Convert text to lowercase
    text = re.sub(r"http\S+|www\S+|https\S+", '', text, flags=re.MULTILINE)  # Remove URLs
    text = re.sub(r'@\w+', '', text)  # Remove @username
    text = re.sub(r'#\w+', '', text)  # Remove hashtags
    text = re.sub(r'\d+', '', text)  # Remove numbers
    text = re.sub(r'[^\w\s]', '', text)  # Remove punctuation
    text = re.sub(r'\s+', ' ', text).strip()  # Remove extra whitespace
    text = stemmer.stem(text)  # Apply stemming
    return text

# Apply preprocessing
data['full_text'] = data['full_text'].apply(preprocess_text)
```

Figure 3.5 Data Cleaning Process

```
import matplotlib.pyplot as plt

# Count the number of duplicates
tweet_bot = len(data.loc[data['full_text'].duplicated() == True])
# Count the number of non-duplicates
tweet_normal = len(data.loc[~data['full_text'].duplicated()])
labels = 'Bot', 'Normal'
sizes = np.array([tweet_bot, tweet_normal])
colors = ['lightskyblue', 'pink']
explode= (0, 0.5)
def absolute_value(val):
    a  = np.round(val/100.*sizes.sum(), 0)

    a=  str(round(val,2))+"%"+'\n'+str(a) +" data"
    return a

plt.pie(sizes, labels=labels, colors=colors,
        autopct=absolute_value, explode=explode,shadow=True)

plt.axis('equal')
plt.title("Data Proportion")
plt.legend()
plt.show()
```

Figure 3.6 Process Cleaning Data and Create Graphs based on Data

Figure 3.7 shows that from the data that was previously collected from the 2023 – January 2025 datasets, a data cleaning process was carried out to obtain a data proportion of

which around 85.91% or 2505 data were normal data while 14.09% or 411 data were BOT data.

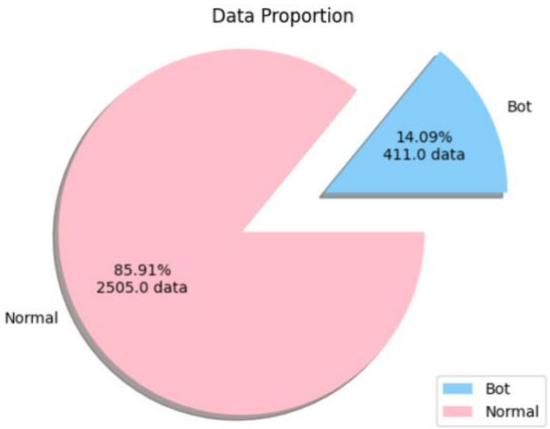

Figure 3.7 Data Proportion

3.6 Data Modeling

The cleaned data is converted into numerical format using vectorization techniques such as Term Frequency-Inverse Document Frequency (TF-IDF). This representation is used as input for the machine learning model.

In Figure 3.8, this is the process of creating a data model. The resulting model will be entered into the machine learning technique to get the results. The syntax used for the data model creation process is :

```
vectorizer = TfidfVectorizer(max_features=5000)

X_vectorized = vectorizer.fit_transform(X)
```

```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split, GridSearchCV
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.neighbors import KNeighborsClassifier
from sklearn.naive_bayes import MultinomialNB
from sklearn.svm import SVC
from sklearn.metrics import accuracy_score, classification_report

# Divide the data into features (X) and targets (y)
X = data['full_text']
y = data['sentiment']
```

```
# TF-IDF Vectorizer to convert text to vectors
vectorizer = TfidfVectorizer(max_features=5000)  # Adjust max_features as needed
X_vectorized = vectorizer.fit_transform(X)
```

```
# Splitting data into training and test data
X_train, X_test, y_train, y_test = train_test_split(X_vectorized, y, test_size=0.2, random_state=42)
```

Figure 3.8 Process Data Modelling

3.7 Stemming Data

Stemming is done to reduce words to their basic form. For example, "eat," "the food," and "ate" all return to "eat." This process helps unite different forms of words that have similar meanings. And in this project we will use the Sastrawi library for the data stemming process.

```
# Initialize Sastrawi stemmer
factory = StemmerFactory()
stemmer = factory.create_stemmer()
```

Figure 3.9 Initialize Sastrawi Stemmer

3.8 Classification Models and Technique

The final stage to obtain sentiment analysis results is to apply and classify the data model into machine learning techniques. The machine learning techniques that will be used are KNN, SVM and Naive Bayes

Three machine learning algorithms are used for sentiment classification:

1. K-Nearest Neighbors (KNN):Classifies tweets based on the majority sentiment of their nearest neighbors in feature space.
2. Naive Bayes: Bayes' theorem based probabilistic model suitable for text classification.
3. Support Vector Machine (SVM): A supervised learning model that separates sentiment classes using hyperplanes in high-dimensional space.

Each model will be evaluated using metrics such as accuracy, precision, recall, and F1-score to determine the best performance. Model results are evaluated using the following metrics:

a. Accuracy: Percentage of correct predictions.
b. Precision: The accuracy of positive predictions.
c. Recall: The model's ability to detect all positive data.
d. F1-Score: Harmonic mean of precision and recall.

In Figure 3.10 is the model implementation process in each machine learning technique.

```
# Hyperparameter tuning untuk KNN
knn_params = {'n_neighbors': [3, 5, 7, 9], 'weights': ['uniform', 'distance']}
knn_grid = GridSearchCV(KNeighborsClassifier(), knn_params, cv=5, scoring='accuracy')
knn_grid.fit(X_train, y_train)
knn_best_model = knn_grid.best_estimator_
```

```
# Hyperparameter tuning untuk Naive Bayes
nb_params = {'alpha': [0.1, 0.5, 1.0, 1.5, 2.0]}
nb_grid = GridSearchCV(MultinomialNB(), nb_params, cv=5, scoring='accuracy')
nb_grid.fit(X_train, y_train)
nb_best_model = nb_grid.best_estimator_
```

```
# Hyperparameter tuning untuk SVM
svm_params = {'C': [0.1, 1, 10, 100], 'kernel': ['linear', 'rbf']}
svm_grid = GridSearchCV(SVC(), svm_params, cv=5, scoring='accuracy')
svm_grid.fit(X_train, y_train)
svm_best_model = svm_grid.best_estimator_
```

```
# Predicting test data and calculating accuracy
knn_pred = knn_best_model.predict(X_test)
nb_pred = nb_best_model.predict(X_test)
svm_pred = svm_best_model.predict(X_test)

# Displays accuracy results and classification reports
print("KNN Accuracy:", accuracy_score(y_test, knn_pred))

print("\nNaive Bayes Accuracy:", accuracy_score(y_test, nb_pred))

print("\nSVM Accuracy:", accuracy_score(y_test, svm_pred))
```

Figure 3.10 Implementation Model to Machine Learning Technique

3.9 Summary

This chapter explains the research methodology in detail, from data collection to evaluation of the classification model. This process ensures that sentiment analysis of the “free meal” program is conducted systematically and data-driven.