YAYASAN AKRAB PEKANBARU
Akrab Juara : Jurnal Ilmu-ilmu Sosial
Volume 10 Nomor 4 Edisi November 2025 (1164-1177)

9

APPLICATION OF THE BERT MODEL IN MEASURING USER PERCEPTION OF THE MAGIC INVESTMENT APPLICATION ON THE GOOGLE PLAY STORE

Tabina Fasya Benedicta, Ade Setiawan, Luthfi Indriyani, Astrid Noviriandini, Sandra Dewi Saraswati
Universitas Bina Sarana Informatika
(Naskah diterima: 1 October 2025, disetujui: 28 October 2025)

Abstract

Investment is one of the most effective ways to achieve long-term financial gains. Nowadays, numerous digital platforms offer investment services, including the Ajaib application. The growing public interest in investing has been driven by influencers and online advertisements, yet it has also led to the rise of fraudulent schemes and fake investment platforms. Therefore, evaluating user satisfaction through sentiment analysis of application reviews becomes essential. This study aims to analyze user sentiments toward the Ajaib investment application based on reviews collected from the Google Play Store. The dataset consists of Indonesian-language reviews from the period 2019–2024, processed using Google Colab and the BERT (Bidirectional Encoder Representations from Transformers) algorithm. The classification results yielded 1,393 reviews, comprising 696 positive and 697 negative sentiments, indicating that negative opinions were slightly more dominant. The model achieved an accuracy of 85%, F1-score of 85%, recall of 85%, and precision of 87%, demonstrating that the BERT algorithm performs effectively in sentiment analysis for investment-related applications.
Keywords: *Web-Based Point of Sale System, Software Architecture, CV Digital Printing*

Abstrak

Investasi merupakan salah satu cara paling efektif untuk meraih keuntungan finansial jangka panjang. Saat ini, banyak platform digital yang menawarkan layanan investasi, termasuk aplikasi Ajaib. Meningkatnya minat masyarakat untuk berinvestasi didorong oleh influencer dan iklan daring, namun juga menyebabkan maraknya skema penipuan dan platform investasi palsu. Oleh karena itu, mengevaluasi kepuasan pengguna melalui analisis sentimen ulasan aplikasi menjadi penting. Penelitian ini bertujuan untuk menganalisis sentimen pengguna terhadap aplikasi investasi Ajaib berdasarkan ulasan yang dikumpulkan dari Google Play Store. Dataset ini terdiri dari ulasan berbahasa Indonesia dari periode 2019–2024, yang diolah menggunakan Google Colab dan algoritma BERT (Bidirectional Encoder Representations from Transformers). Hasil klasifikasi menghasilkan 1.393 ulasan, yang terdiri dari 696 sentimen positif dan 697 sentimen negatif, yang menunjukkan bahwa opini negatif sedikit lebih dominan. Model ini mencapai akurasi 85%, skor F1 85%, recall 85%, dan presisi 87%, yang menunjukkan bahwa algoritma BERT bekerja efektif dalam analisis sentimen untuk aplikasi terkait investasi.
Kata Kunci: Sistem Point of Sale Berbasis Web, Arsitektur Perangkat Lunak, Percetakan Digital CV

I. INTRODUCTION

Nowadays, people have changed their financial management behavior, one of which is investing. Many people want to invest because of the potential for long-term profits. Besides long-term gains, investing also requires minimal activity, resulting in passive income, which is why many people are interested in investing. In this study, the author was interested in analyzing sentiment regarding the Ajaib app to determine satisfaction based on reviews on the Google Play Store.

The growing public interest in investing has led to numerous influencers and advertisements promoting investment opportunities, leading to an increase in fraud and fraudulent investment cases.

"Ajaib is an investment app that makes it easy for users to invest in mutual funds and stocks online. Furthermore, Ajaib also provides an investment learning platform for both beginners and experienced users" (Ajaib, 2023).

The problem arises from measuring and analyzing satisfaction levels in reviews using the BERT algorithm, which can separate data from public trust in investing in the app, which contains negative and positive reviews, aimed at assessing the emotional state of app users.

In research conducted by Alifa and al Faraby, BERT is a deep learning model that has achieved outstanding results in various natural language translation tasks. Because it has six layers of transformers layered on top of each encoder and decoder, the training process is very complex, the configuration is very difficult, the training time is very long, and the cost is very high. However, Google's BERT pre-trained model is available as open source and can be used without first building the model. BERT processing begins with the word and embedding representations from the embedding layer. To create new intermediate representations, each layer uses telescoping attention calculations on the word representations from the previous layer (Kurniawan et al., 2022).

Sentiment analysis in the Ajaib application can help companies identify strategic investors and increase investor confidence. By analyzing customer opinions and reviews, companies can determine the level of customer satisfaction. Sentiment analysis using the BERT algorithm can play a role.

It is crucial to improve user service in the investment sector. Currently, many investors are unsure where to invest and lack confidence due to the abundance of news reporting investment fraud. This situation has given rise to a crucial role for social media, particularly the Google Play Store, as a source of information regarding opinions on the Ajaib app. Ajaib app review data on the Google Play Store is unstructured and uses much informal language, necessitating a process for analyzing the review data.

In a study conducted by Tarisa Putri Cahyani and Danang Teguh Qoyyimi, M.Sc., Ph.D., using a similar case, they conducted sentiment analysis on the Halodoc app on the Google Play Store using the Bidirectional Encoder Representations from Transformers (BERT) algorithm and the Long Short Term Memory Network (LS-TM) algorithm. Three data labeling methods were used: automatic data labeling using a Valence Aware Dictionary and Sentiment Reasoner (VADER) lexicon, manual data labeling using user ratings, and manual labeling by researchers. Research results showed that the BERT algorithm, labeled through user ratings, was the most effective for implementation (Cahyani & Qoyyimi, 2023).

Another study, using the same case, was conducted by Ekka Pujo Ariesanto Akhmad in Sentiment Analysis of DLU Ferry App Reviews on the Google Play Store. This study used 1,575 reviews, consisting of positive, neutral, and negative sentiments. This study measured the performance of the Bidirectional Encoder Representations from Transformers (BERT) method in sentiment classification using a pretrained IndoBERT-base model with fine-tuning techniques. The test results obtained an accuracy of 86% (Akhmad, 2023). Another study also discussed the BERT model, entitled "Sentiment Analysis of Customer Reviews of the Ruang Guru App Using the BERT Method," conducted by Raden Mas Rizqi Wahyu Panca Kusuma Atmaja and Wiyli Yustanti. This model obtained an F1 score of 98.9% with a 70:30 training and test data ratio. The model was then evaluated, yielding an accuracy of 99%, a precision of 64.13%, and a recall of 60.51%. The sentiment percentage criterion was weighted at 99% and was declared valid based on averaging over 10 epochs with consistent improvement (Atmaja & Yustanti, 2021).

This demonstrates the BERT algorithm's high accuracy in processing text, possessing a well-structured understanding of language meaning, thus addressing some language processing challenges. The results of this study are expected to instill confidence in investors regarding investment platforms and assist companies in improving their services. Therefore, the researchers intend to conduct a study entitled: "Sentiment Analysis of the Ajaib Application to Determine Satisfaction Based on Opinions Using the BERT Algorithm."

II. RESEARCH METHODS

In the research stage regarding sentiment analysis of the magic application to determine satisfaction based on opinions, the diagram can be described as follows:

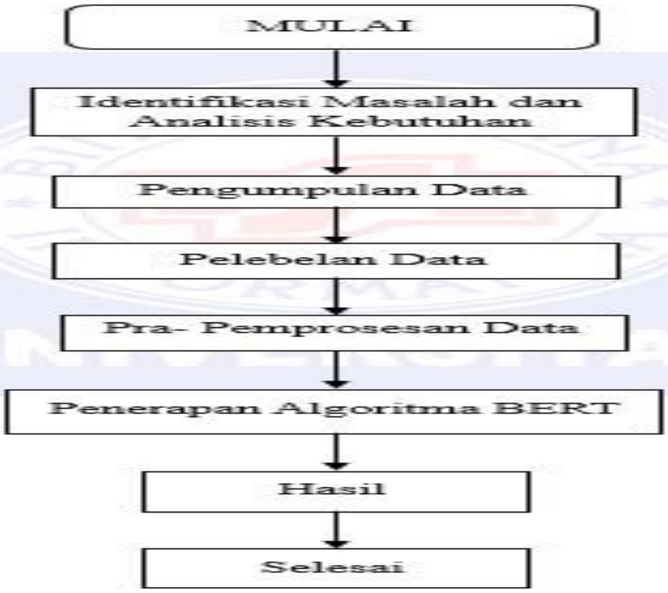


Figure 1. Research Stages

1. Problem Identification and Needs Analysis:

This study examines the problems encountered during the implementation of the BERT algorithm for sentiment analysis on the Ajaib app and its data grouping process.

2. Data Collection

The data collection process used a scraping technique by collecting Ajaib app user reviews from the Google Play Store. The review data collected consisted of 1,393 ungrouped reviews.

3. Data Labeling

Data labeling is the process of dividing reviews into positive and negative categories. Labels are assigned to the positive and negative categories in the data.

4. Data Pre-Processing

Pre-processing is the pre-processing of the collected data to maximize the results of the algorithm application, thus facilitating the subsequent steps. There are several stages in data processing, as follows:

a. Data Cleaning

Data cleaning is the process of removing incomplete and inconsistent data from the dataset. The primary goal of data cleansing is to ensure the quality of the data used for analysis.

b. Tokenization

Tokenization is the process of breaking down words, phrases, or specific symbols.

c. Stopwords

Stopwords are the process of removing frequently occurring text that doesn't add meaning to the text.

d. Steaming

Steamming is the process of converting different words but having the same meaning into text.

5. Application of the BERT Algorithm

In this study, the author used the BERT algorithm to solve the research problem. The BERT algorithm was created as a pre-trained model, so it didn't require a large amount of training data because the BERT algorithm was trained on a large amount of data.

6. Results

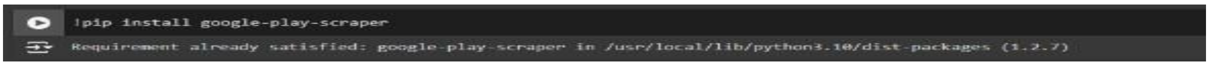
This process ensures that the BERT algorithm can produce accurate data using the method used. Several tests have shown how well the BERT method predicts sentiment analysis. Using the BERT algorithm ensures its accuracy, resulting in good predictions for the Ajaib app's sentiment analysis results to determine satisfaction based on opinions.

III. RESEARCH RESULTS

1. Data Collection

Data collected using the following scraping technique is the data collection method:

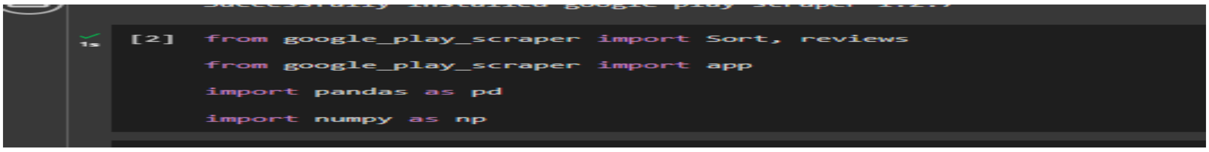
a. Installing



```
!pip install google-play-scraper
Requirement already satisfied: google-play-scraper in /usr/local/lib/python3.10/dist-packages (1.2.7)
```

Figure 1. Installation Script

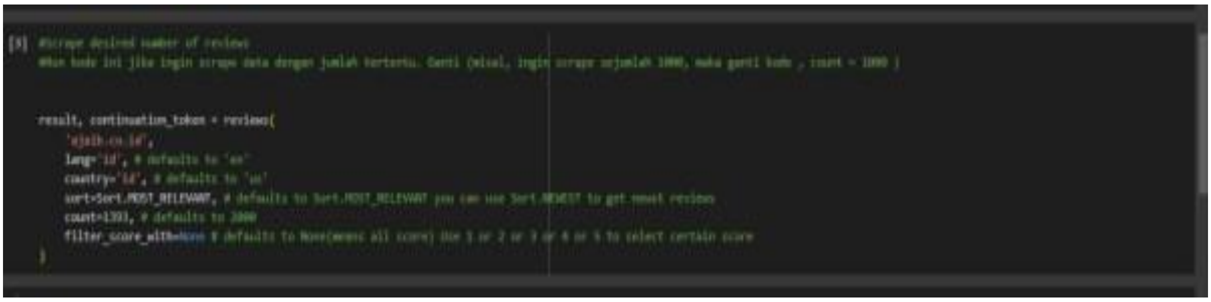
b. Import Google Scraper, Pandas, and Numpy.



```
Successfully installed google-play-scraper-1.2.7
[2] from google_play_scraper import Sort, reviews
    from google_play_scraper import app
    import pandas as pd
    import numpy as np
```

Figure 2 Import Script

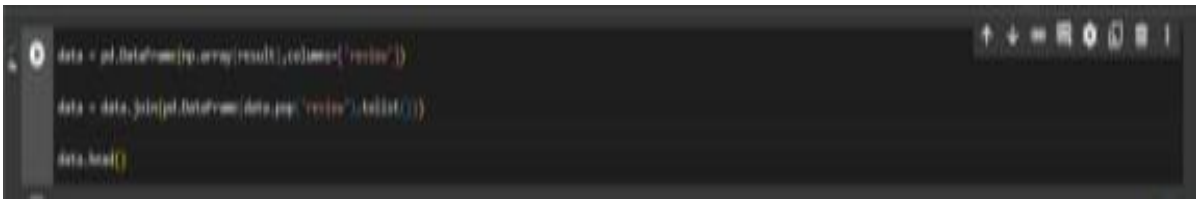
c. Request the review data you wish to retrieve.



```
[3] #scrape desired number of reviews
    #Run kode ini jika ingin scrape data dengan jumlah tertentu. Ganti (misal, ingin scrape sejumlah 1000, maka ganti kode , count = 1000 )


    result, continuation_token = reviews(
        'ajaib.co.id',
        lang='id', # defaults to 'en'
        country='id', # defaults to 'us'
        sort=Sort.MOST_RELEVANT, # defaults to Sort.MOST_RELEVANT you can use Sort.NEWEST to get newst reviews
        count=1393, # defaults to 100
        filter_score_with=None # defaults to None(means all score) Use 1 or 2 or 3 or 4 or 5 to select certain score
    )
```

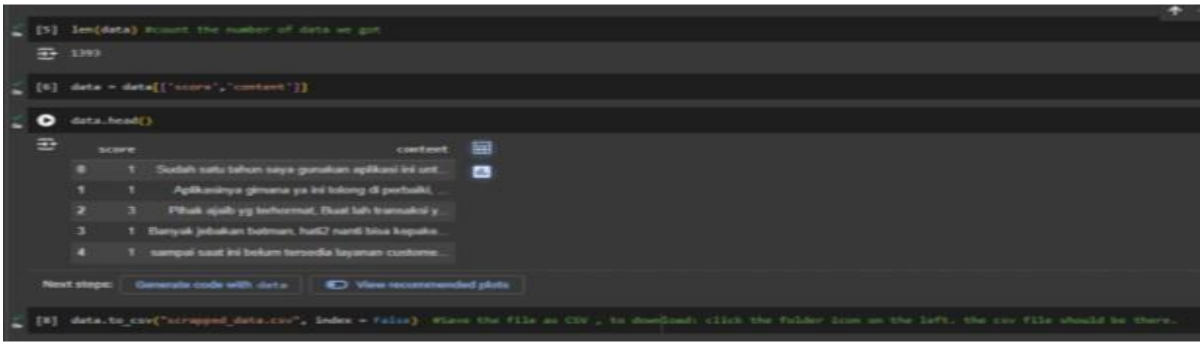
Figure 3 Scraping Script

d. Creating a data frame from the data.



```
data = pd.DataFrame(np.array(result),columns=['review'])

data = data.join(pd.DataFrame(data.pop('review').tolist()))

data.head()
```

e. Displays the amount of data obtained and then saves it to a CSV file.



```
[5] len(data) #count the number of data we got
    1393

[6] data = data[['score','content']]

    data.head()
       score                                       content
    0      1  Sudah satu tahun saya gunakan aplikasi ini unt...
    1      1      Aplikasinya gimana ya ini tolong di perbaiki, ...
    2      3     Pihak ajaib yg terhormat, Buat lah transaksi y...
    3      1  Banyak jebakan batman, hal2 nanti bisa kepake...
    4      1   sampai saat ini belum tersedia layanan custome...

[8] data.to_csv("scrapped_data.csv", index = False)  #Save the file as CSV , to download: click the folder icon on the left. the csv file should be there.
```

Figure 5 Script Displays the Number

Data Table 1 Scarping Result Data

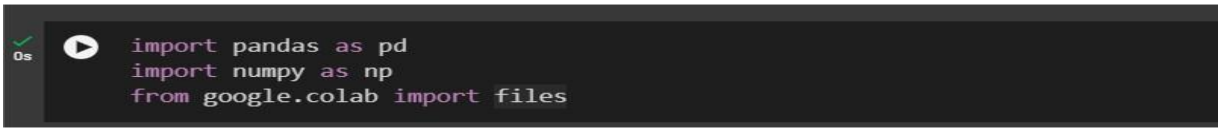
sentiment	text
5	So far so good because I'm just a beginner
1	Oh my gosh, this is the first time I've played an investment app. I swear, processing just one order takes so long, even during working hours.
5	Very easy to understand
5	thank you for helping me/us to invest
5	This app is really top notch
1	Please, make the app magical, don't change it too much. It's just frustrating. I've created a waiting list and it's gone. What can I do? A version from 2023 or earlier would be better. The chat forum is also getting more and more hidden. What needs to be improved is the server.
1	Data verification takes a really long time, it's been more than 6 months...

5	good, complete and easy to understand code leveral andrxm t9
1	I've been investing in Ajaib stocks for over a year, but I still don't know the customer service. The portfolio was closed just to request investor data. It's inaccessible due to difficulties uploading an ID photo. It's strange that we've already topped up and bought/sold shares, but we're being asked to upload an ID photo again. Meanwhile, uploading the ID photo keeps failing. What should I do?
1	I topped up my balance, why didn't it come in full? Only 50% came in, but in the full history, there was no response in the chat. It's a loss. I haven't invested yet, just deposited and it's already like this.

2. Data Labeling

Data labeling to separate positive and negative reviews, here is how to label data:

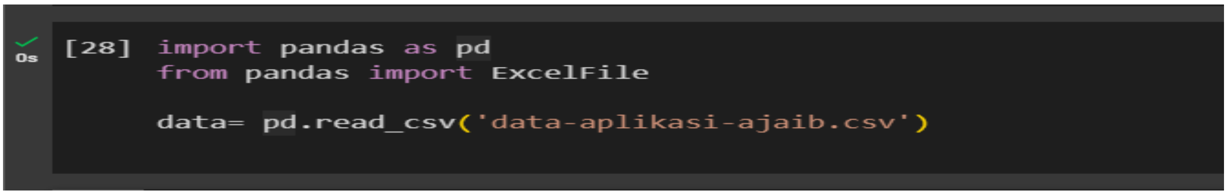
a. Importing into pandas, numpy and google colab.



```
import pandas as pd
import numpy as np
from google.colab import files
```

Figure 6. Labeling Import Script

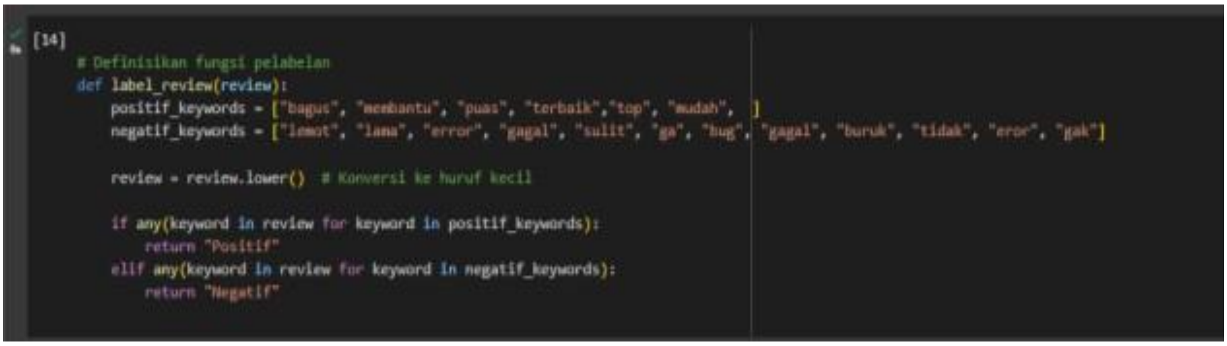
b. Reading CSV data.



```
import pandas as pd
from pandas import ExcelFile

data= pd.read_csv('data-aplikasi-ajaib.csv')
```

Figure 7 Script to read data

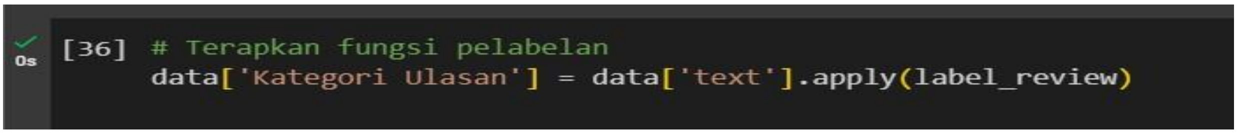
c. Apply labeling



```
# Definisikan fungsi pelabelan
def label_review(review):
    positif_keywords = ["bagus", "membantu", "puas", "terbaik","top", "mudah",  ]
    negatif_keywords = ["lemot", "lama", "error", "gagal", "sulit", "ga", "bug", "gagal", "buruk", "tidak", "eror", "gak"]

    review = review.lower()  # Konversi ke huruf kecil

    if any(keyword in review for keyword in positif_keywords):
        return "Positif"
    elif any(keyword in review for keyword in negatif_keywords):
        return "Negatif"
```

Figure 8 Labeling Script

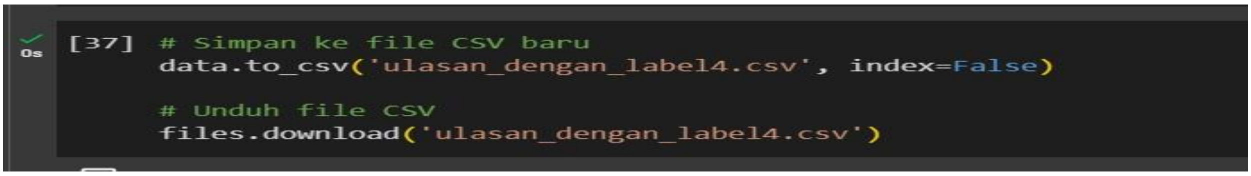
d. Applying the labeling function.



```
# Terapkan fungsi pelabelan
data['Kategori Ulasan'] = data['text'].apply(label_review)
```

Figure 9. Labeling Function Script

e. Saving to CSV.



```
[37]  # Simpan ke file CSV baru
      data.to_csv('ulasan_dengan_label4.csv', index=False)

      # Unduh file CSV
      files.download('ulasan_dengan_label4.csv')
```

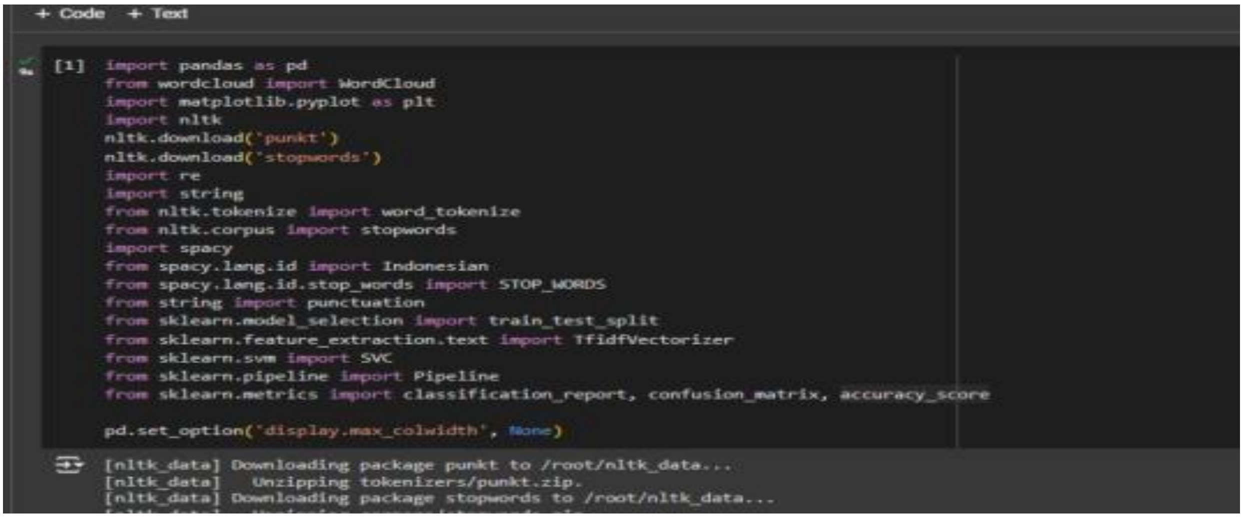
Figure 10: CSV Saving Script

Table 2: Labeling Results

Sentiment	text
Positive	So far so good because I'm just a beginner
Negative	Oh my gosh, this is my first time using an investment app, and I swear, processing just one order takes so long, even during working hours.
Positive	Very easy to understand
Positive	thank you for helping me/us to invest
Positive	This app is really top notch
Negative	Please, stop changing the app so much. It's frustrating. I've created a waiting list and it's gone. What can I do? A version from 2023 or earlier would be better. The chat forum is also getting more and more hidden. What needs to be improved is the server upgrade.
Negative	Data verification takes a really long time, it's been more than 6 months........
Positive	good, complete and easy to understand code leveral andrxmt9
Negative	I've been investing in Ajaib stocks for over a year, but I still don't know the customer service. The portfolio was closed just to request investor data. It's inaccessible due to difficulties uploading an ID photo. It's strange that we've already topped up and bought/sold shares, but we're being asked to upload an ID photo again. The ID photo upload keeps failing. What should I do?
Negative	I topped up my balance, why didn't it come in full? Only 50% came in, but in the full history, there was no response in the chat. It's a loss. I haven't invested yet, just deposited and it's already like this.

3. Data processing

a. Import the required libraries.

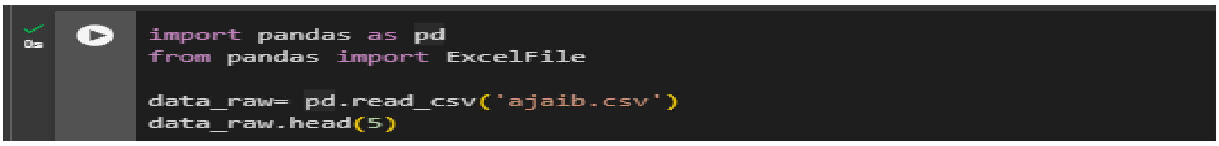


```
[1]  import pandas as pd
     from wordcloud import WordCloud
     import matplotlib.pyplot as plt
     import nltk
     nltk.download('punkt')
     nltk.download('stopwords')
     import re
     import string
     from nltk.tokenize import word_tokenize
     from nltk.corpus import stopwords
     import spacy
     from spacy.lang.id import Indonesian
     from spacy.lang.id.stop_words import STOP_WORDS
     from string import punctuation
     from sklearn.model_selection import train_test_split
     from sklearn.feature_extraction.text import TfidfVectorizer
     from sklearn.svm import SVC
     from sklearn.pipeline import Pipeline
     from sklearn.metrics import classification_report, confusion_matrix, accuracy_score

     pd.set_option('display.max_colwidth', None)
```

```
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   Unzipping tokenizers/punkt.zip.
[nltk_data] Downloading package stopwords to /root/nltk_data...
```

Figure 11: Import Data Processing Script

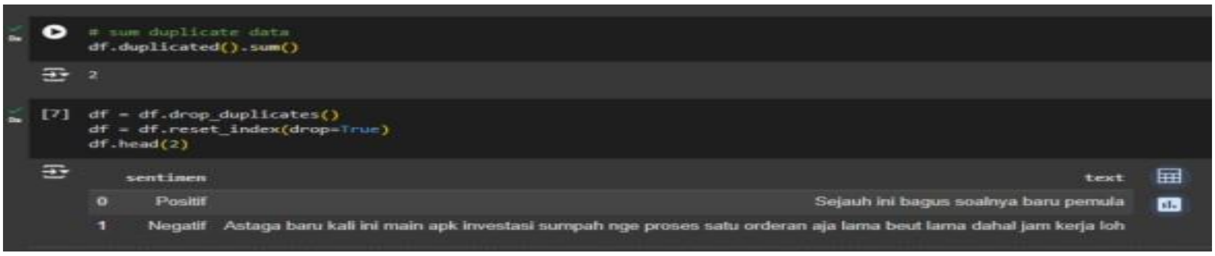
b. Entering Data



```
import pandas as pd
from pandas import ExcelFile

data_raw= pd.read_csv('ajaib.csv')
data_raw.head(5)
```

Figure 12 Data Entry Script

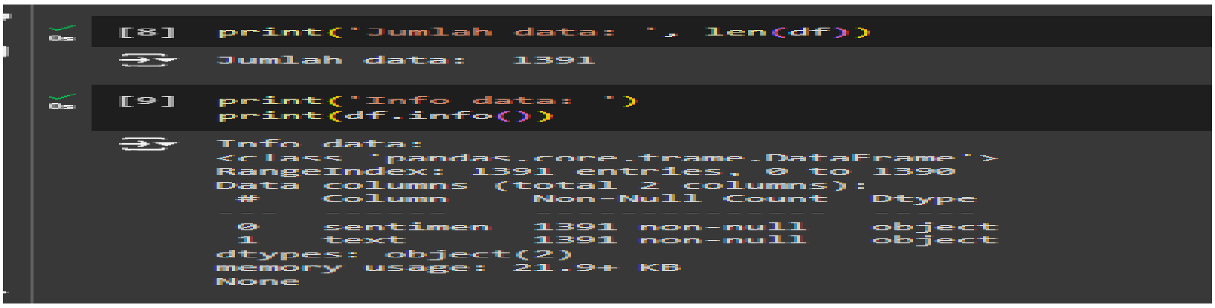
c. Removing duplicate text.



```
# sum duplicate data
df.duplicated().sum()

2

[7] df = df.drop_duplicates()
    df = df.reset_index(drop=True)
    df.head(2)

       sentimen                                                                                     text
    0   Positif                                                        Sejauh ini bagus soalnya baru pemula
    1   Negatif   Astaga baru kali ini main apk investasi sumpah nge proses satu orderan aja lama beut lama dahal jam kerja loh
```

Figure 13 Duplicate Text Script

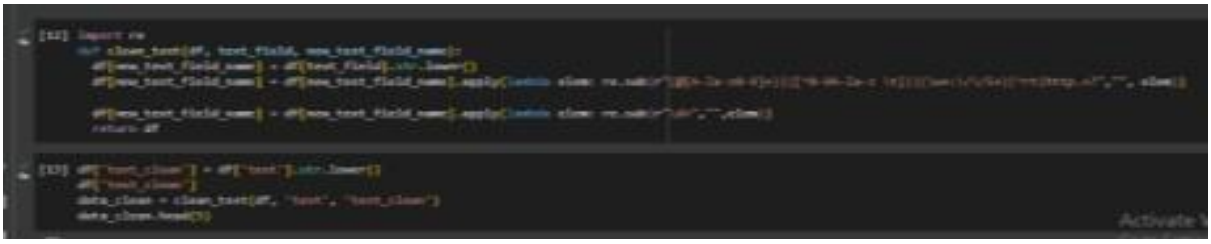
d. Exploratory data analysis.



```
[8]  print('Jumlah data: ', len(df))

     Jumlah data:  1391

[9]  print('Info data: ')
     print(df.info())

     Info data:
     <class 'pandas.core.frame.DataFrame'>
     RangeIndex: 1391 entries, 0 to 1390
     Data columns (total 2 columns):
      #   Column     Non-Null Count  Dtype
     ---  ------     --------------  -----
      0   sentimen   1391 non-null   object
      1   text       1391 non-null   object
     dtypes: object(2)
     memory usage: 21.9+ KB
     None
```

Figure 14 EDA Script

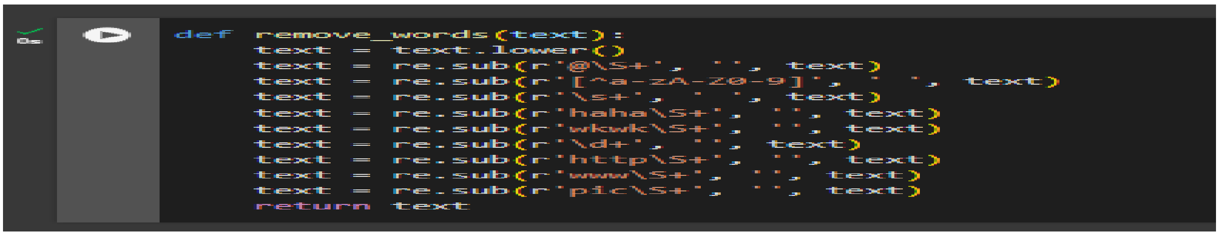
e. Cleaning up text in a data frame.



```
[12] import re
     def clean_text(df, text_field, new_text_field_name):
         df[new_text_field_name] = df[text_field].str.lower()
         df[new_text_field_name] = df[new_text_field_name].apply(lambda elem: re.sub(r"(@[A-Za-z0-9]+)|([^0-9A-Za-z \t])|(\w+:\/\/\S+)|^rt|http.+?", "", elem))

         df[new_text_field_name] = df[new_text_field_name].apply(lambda elem: re.sub(r"\d+", "", elem))
         return df

[13] df['text_clean'] = df['text'].str.lower()
     df['text_clean']
     data_clean = clean_text(df, 'text', 'text_clean')
     data_clean.head(5)
```

Figure 15 Script to Delete Data Frame Text

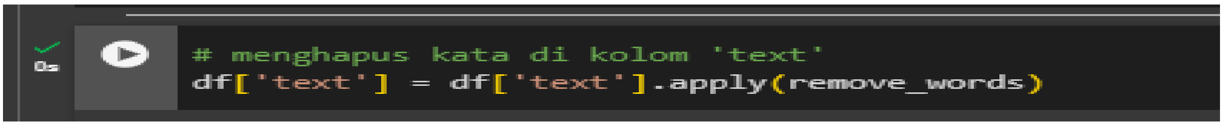
f. Clearing text of irrelevant words



```
def remove_words(text):
    text = text.lower()
    text = re.sub(r'@\S+', '', text)
    text = re.sub(r'[^a-zA-Z0-9]', ' ', text)
    text = re.sub(r'\s+', ' ', text)
    text = re.sub(r'haha\S+', '', text)
    text = re.sub(r'wkwk\S+', '', text)
    text = re.sub(r'\d+', '', text)
    text = re.sub(r'http\S+', '', text)
    text = re.sub(r'www\S+', '', text)
    text = re.sub(r'pic\S+', '', text)
    return text
```

Figure 16 Script for Deleting Text

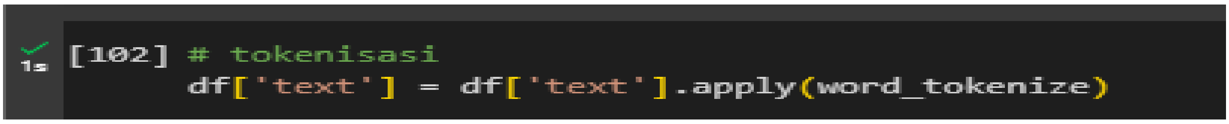
g. Deleting words in a text field.



```
# menghapus kata di kolom 'text'
df['text'] = df['text'].apply(remove_words)
```

Figure 17 Script to Delete Text Columns

h. Tokenization



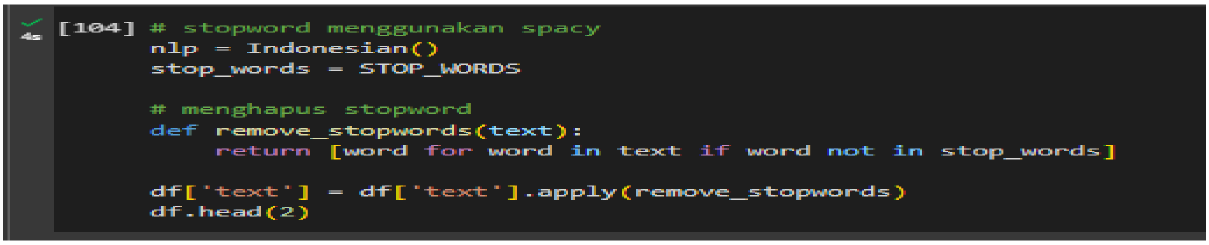
```
[102] # tokenisasi
      df['text'] = df['text'].apply(word_tokenize)
```

Figure 18 Tokenization Script

Table 3 Tokenization Results

sesudah	Sebelum
sejauh,ini,bagus,soalnya,baru, pemula	sejauh ini bagus soalnya baru pemula
astaga,baru,kali,ini,main,apk, investasi,	astaga baru kali ini main apk investasi
sumpah,nge,proses,satu,orderan,aja,lama,beut,lama ,dahal,jam,kerja,loh	Sumpah nge proses satu orderan aja lama beut lama dahal jam kerja loh
sangat,mudah,difahami	sangat mudah difahami
terima,kasih,ajaib,sudah,membantu, saya,kami,untuk, berinvestasi	terima kasih ajaib sudah membantu saya kami untuk berinvestasi
emang,markotop,sih,inii,app	emang markotop sih inii app

i. Stopwords.



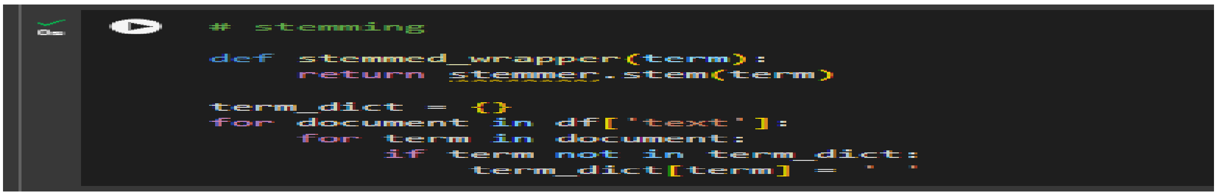
```
[104] # stopword menggunakan spacy
      nlp = Indonesian()
      stop_words = STOP_WORDS

      # menghapus stopword
      def remove_stopwords(text):
          return [word for word in text if word not in stop_words]

      df['text'] = df['text'].apply(remove_stopwords)
      df.head(2)
```

Figure 19 Stopword Script

Table 4 Stopword Results

Sesudah	Sebelum
bagus,pemula	Sejauh ini bagus soalnya baru pemula
astaga,kali,main,apk,investasi,sumpah, nge,proses, orderan,aja,beut, dahal,jam, kerja,loh	astaga baru kali ini main apk investasi sumpah nge proses satu orderan aja lama beut lama dahal jam kerja loh
mudah,difahami	sangatmudah difahami
terima,kasih,ajaib,membantu,berinvestasi	terima kasih ajaib sudah membantu sayakami untuk berinvestasi
emang,markotop, sih,inii,app	emang markotop sih inii app

j. Stemming.



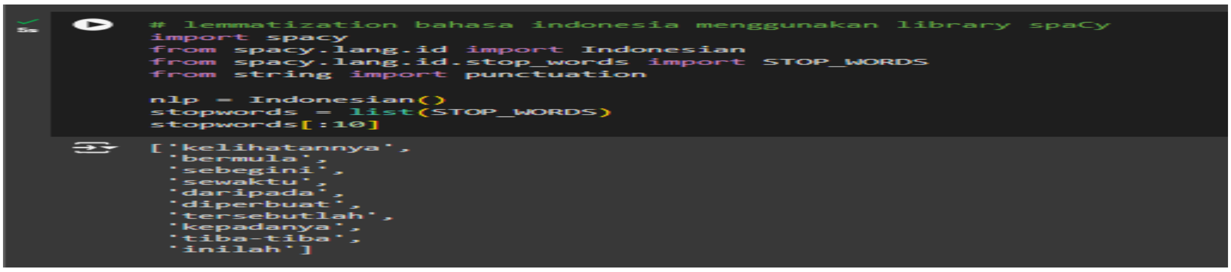
```
# stemming

def stemmed_wrapper(term):
    return stemmer.stem(term)

term_dict = {}
for document in df['text']:
    for term in document:
        if term not in term_dict:
            term_dict[term] = ' '
```

Figure 20: Stemming Script

Table 5: Stemming Results

Sesudah	Sebelum
[bagus,pemula]	sejauh ini bagus soalnya baru pemula
[astaga,kali,main,apk,investasi,sumpah,nge, proses,orderan,aja,beut,dahal,jam,kerja, loh]	astaga baru kali ini main apk investasi sumpah nge proses satu orderan aja lama beut lama dahal jam kerja loh
[mudah,difahami]	sangat mudah difahami
[terima,kasih,ajaib,membantu,berinvestasi]	terima kasih ajaib sudah membantu saya kami untuk berinvestasi
[emang,markotop,sih,inii,app]	emang markotop sih inii app

k. Lemmatization of Indonesian language.

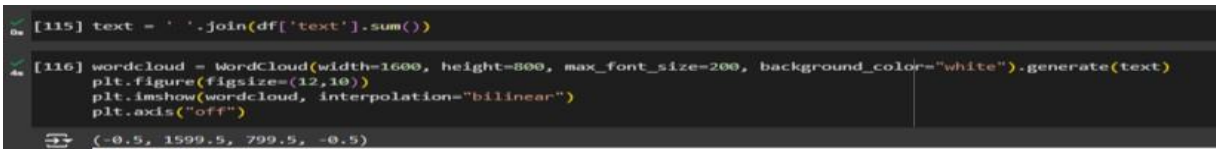


```
# lemmatization bahasa indonesia menggunakan library spaCy
import spacy
from spacy.lang.id import Indonesian
from spacy.lang.id.stop_words import STOP_WORDS
from string import punctuation

nlp = Indonesian()
stopwords = list(STOP_WORDS)
stopwords[:10]
```

```
['kelihatannya',
 'bermula',
 'sebegini',
 'sewaktu',
 'daripada',
 'diperbuat',
 'tersebutlah',
 'kepadanya',
 'tiba-tiba',
 'inilah']
```

4. Wordcloud

To visualize frequently occurring words in a text, here's how to display a wordcloud:



```
[115] text = ' '.join(df['text'].sum())

[116] wordcloud = WordCloud(width=1600, height=800, max_font_size=200, background_color="white").generate(text)
      plt.figure(figsize=(12,10))
      plt.imshow(wordcloud, interpolation="bilinear")
      plt.axis("off")
```

```
(-0.5, 1599.5, 799.5, -0.5)
```

Figure 22 Wordcloud Script

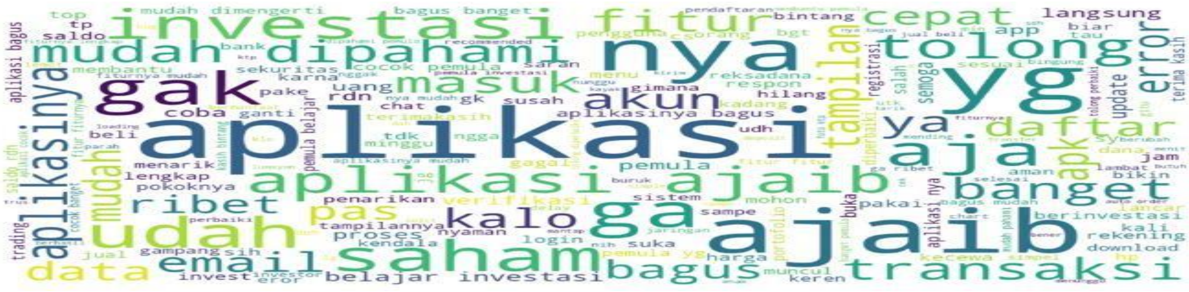
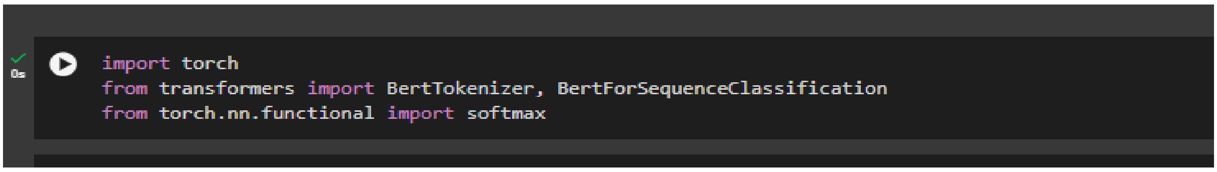


Figure 23 Wordcloud

5. BERT Implementation

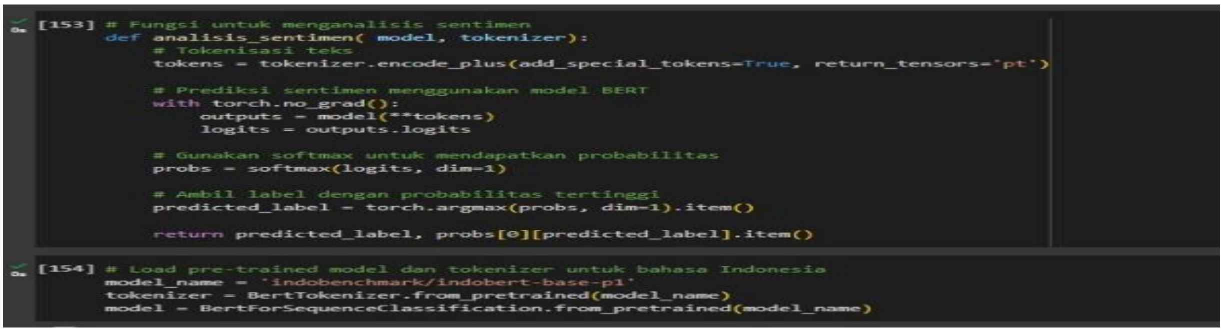
a. Import libraries and modules into BERT.



```
import torch
from transformers import BertTokenizer, BertForSequenceClassification
from torch.nn.functional import softmax
```

Figure 24 BERT Implementation Import Script

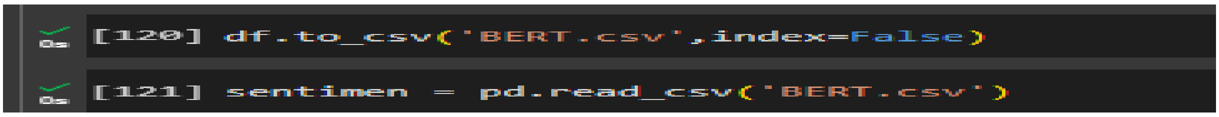
b. Implementing the BERT algorithm.



```
[153] # Fungsi untuk menganalisis sentimen
      def analisis_sentimen( model, tokenizer):
          # Tokenisasi teks
          tokens = tokenizer.encode_plus(add_special_tokens=True, return_tensors='pt')

          # Prediksi sentimen menggunakan model BERT
          with torch.no_grad():
              outputs = model(**tokens)
              logits = outputs.logits

          # Gunakan softmax untuk mendapatkan probabilitas
          probs = softmax(logits, dim=1)

          # Ambil label dengan probabilitas tertinggi
          predicted_label = torch.argmax(probs, dim=1).item()

          return predicted_label, probs[0][predicted_label].item()

[154] # Load pre-trained model dan tokenizer untuk bahasa Indonesia
      model_name = 'indobenchmark/indobert-base-p1'
      tokenizer = BertTokenizer.from_pretrained(model_name)
      model = BertForSequenceClassification.from_pretrained(model_name)
```

Figure 25 BERT Implementation Script

c. Create a CSV file



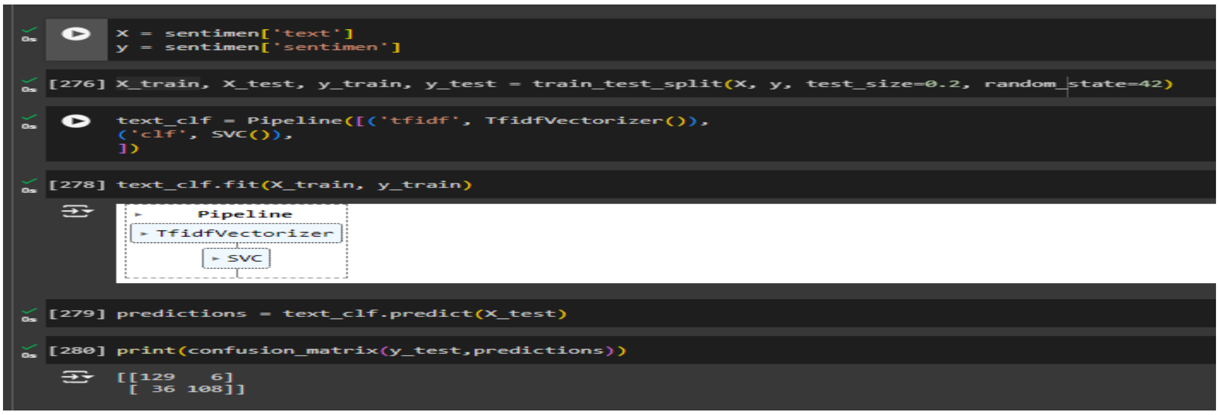
```
[120] df.to_csv('BERT.csv',index=False)

[121] sentimen = pd.read_csv('BERT.csv')
```

Figure 26. BERT CSV Creation Script

6. Confusion Matrix

The data evaluation stage involves comparing the model's predictions with the actual labels of the data being tested. The following is a confusion matrix:

a. BERT prediction evaluation.

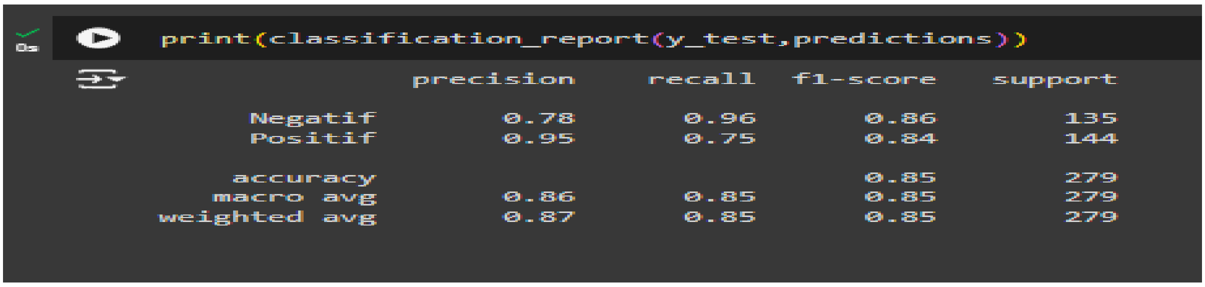


```
      X = sentimen['text']
      y = sentimen['sentimen']

[276] X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

      text_clf = Pipeline([('tfidf', TfidfVectorizer()),
      ('clf', SVC()),
      ])

[278] text_clf.fit(X_train, y_train)
```

```
[279] predictions = text_clf.predict(X_test)

[280] print(confusion_matrix(y_test,predictions))

      [[129   6]
       [ 36 108]]
```

Figure 27 BERT Prediction Script

b. Results.



```
print(classification_report(y_test,predictions))

              precision    recall  f1-score   support

     Negatif       0.78      0.96      0.86       135
     Positif       0.95      0.75      0.84       144

    accuracy                           0.85       279
   macro avg       0.86      0.85      0.85       279
weighted avg       0.87      0.85      0.85       279
```

Figure 28 BERT Result Script

IV. CONCLUSION

1. In the study, entitled "Sentiment Analysis of the Ajaib App to Determine Satisfaction Based on Opinions Using the BERT Algorithm," 1,393 data sets were used, with 696 positive reviews and 697 negative reviews. This study concluded that Ajaib app users had more negative reviews. The results showed an accuracy of 85%, an F1 Score of 85%, a recall of 85%, and a precision of 87%.
2. This study demonstrated the relationship between information and user satisfaction, enabling the company to understand user needs for future evaluation.
3. This study found a significant number of negative reviews, which could impact user trust in the Ajaib app. Therefore, the Ajaib app must improve its system in the future to garner more positive reviews and increase user trust.

REFERENCES

Akhmad, E. P. A. (2023). Analisis Sentimen Ulasan Aplikasi DLU Ferry Pada Google Play Store Menggunakan Bidirectional Encoder Representations from Transformers. *Jurnal Aplikasi Pelayaran Dan Kepelabuhanan*, *13*(2), 104–112. https://doi.org/10.30649/japk.v13i2.94 Atmaja, R. M. R. W. P. K., & Yustanti, W. (2021).

Analisis Sentimen Customer Review Aplikasi RuangGuru dengan Metode BERT (Bidirectional EncoderRepresentations from Transformers). *JEISBI (Journal of Emerging Information Systems and Business Intelligence)*, *02*(3), 55–62.

Bey, T., & Ari, I. K. (2023). Implementasi BERT pada Analisis Sentimen Ulasan Destinasi Wisata Bali. *Jurnal Elektronik Ilmu Komputer Udayana*, *12*(2), 409–420.

Braja, A. S. P., & Kodar, A. (2023). Implementasi Fine-Tuning BERT untuk Analisis Sentimen terhadap Review Aplikasi PUBG Mobile di Google Play Store. *J I M P – Jurnal Informatika Merdeka Pasuruan*, *7*(3), 120. https://doi.org/10.51213/jimp.v7i3.779

Cahyani, T. P., & Qoyyimi, D. T. (2023). *INTISARI ALGORITMA BIDIRECTIONAL ENCODER REPRESENTATIONS FROM TRANSFORMERS (BERT) PADA ANALISIS SENTIMEN Oleh TARISA PUTRI CAHYANI ABSTRACT BIDIRECTIONAL ENCODER REPRESENTATIONS FROM TRANSFORMERS (BERT) ALGORITHM ON SENTIMENT ANALYSIS By TARISA PUTRI* .

Gunawan, B., Pratiwi, H. S., & Pratama, E. E. (2018). Sistem Analisis Sentimen pada Ulasan Produk Menggunakan Metode Naive Bayes. *Jurnal Edukasi Dan Penelitian Informatika (JEPIN)*, *4*(2), 113. https://doi.org/10.26418/jp.v4i2.27526

Guntara, R. G. (2023). Visualisasi Data Laporan Penjualan Toko Online Melalui Pendekatan Data Science Menggunakan Google Colab. *Jurnal Ilmiah Multidisiplin*, *2*(6), 2091–2100.

Husin, N. (2023). Komparasi Algorit Forest, Naïve Bayes, dan Bert Untuk Multi- Class Classification Pada Artikel Cable News Network (CNN). *Jurnal Esensi Infokom : Jurnal Esensi Sistem Informasi Dan Sistem Komputer*, *7*(1), 75–84. https://doi.org/10.55886/infokom.v7i1.608

Kurniawan, B., Ari Aldino, A., & Rahman Isnain, A. (2022). Sentimen Analisis Terhadap Kebijakan Penyelenggara Sistem Elektronik (Pse) Menggunakan Algoritma Bidirectional Encoder Representations From Transformers (Bert). *Jurnal Teknologi Dan Sistem Informasi*, *3*(4), 98–106. http://jim.teknokrat.ac.id/index.php/JTSI

Kusnadi, R., Yusuf, Y., Andriantony, A., Ardian Yaputra, R., & Caintan, M. (2021). Analisis Sentimen Terhadap Game Genshin Impact Menggunakan Bert. *Rabit : Jurnal Teknologi Dan Sistem Informasi Univrab*, *6*(2), 122–129. https://doi.org/10.36341/rabit.v6i2.1765

Mahawardana, P. P. O., Imawati, I. A. P. F., & Dika, I. W. (2022). Analisis Sentimen Berdasarkan Opini dari Media Sosial Twitter terhadap "Figure Pemimpin" Menggunakan Python. *Jurnal Manajemen Dan Teknologi Informasi*, *12*(2), 50–56. https://ojs.mahadewa.ac.id/index.php/jmti/artic le/view/2111

Muflih, H. Z., Abdillah, A. R., & Hasan, F. N. (2023). Analisis Sentimen Ulasan Pengguna Aplikasi Ajaib Menggunakan Metode Naïve Bayes. *KLIK: Kajian Ilmiah Informatika Dan Komputer*, *4*(3), 1613–1621. https://doi.org/10.30865/klik.v4i3.1303 Pradipta, D., Kusrini, K., & Fatta, H. Al. (2023).

Sentiment Analysis Comments Covid-19 Variant Omicron on Social Media Instagram with Bidirectional Encoder from Transformers (BERT). *JTECS : Jurnal Sistem Telekomunikasi Elektronika Sistem Kontrol Power Sistem Dan Komputer*, *3*(1), 51. https://doi.org/10.32503/jtecs.v3i1.3219

Rizaldi, S. A. R., Alam, S., & Kurniawan, I. (2023). Analisis Sentimen Pengguna Aplikasi JMO (Jamsostek Mobile) Pada Google Play Store Menggunakan Metode Naive Bayes. *STORAGE: Jurnal Ilmiah Teknik Dan Ilmu Komputer*, *2*(3), 109–117. https://doi.org/10.55123/storage.v2i3.2334