

## BAB V

### SIMPULAN DAN SARAN

#### 5.1 Simpulan

Berdasarkan hasil yang dipaparkan dapat disimpulkan beberapa poin sebagai berikut :

- a. Program telah dirancang dengan baik dan secara sistem dapat secara otomatis mengidentifikasi dan mengklasifikasikan sentimen dari ulasan pengguna.
- b. Penerapan kombinasi teknik pembobotan kata TF-IDF dan *Count Vectorizer* pada algoritma Naïve Bayes dapat meningkatkan akurasi 80% pada model yang diusulkan, dengan hasil baik diperoleh menggunakan teknik TF-IDF dan *Naive Bayes*.
- c. Hasil perbandingan sentimen pada kedua aplikasi tidak terdapat perbedaan yang signifikan yang didapatkan namun bila dibandingkan, jumlah sentimen positif pada aplikasi shopee lebih baik dibandingkan aplikasi tokopedia.

#### 5.2 Saran

Berdasarkan pembahasan yang telah dipaparkan terdapat beberapa saran yang dapat diterapkan untuk penelitian selanjutnya yaitu:

- a. Penggunaan kombinasi teknik pembobotan dan algoritma yang berbeda seperti SVM, atau *Deep Learning*, Atau *Random Forest*, dsb, dan menambahkan *feature* yang lain
- b. Menggunakan sistem penandaan otomatis yang dioptimalkan khusus untuk bahasa Indonesia, seperti model yang disiapkan untuk memahami gaya bahasa umum. Langkah ini dimaksudkan untuk memastikan bahwa label yang dihasilkan konsisten dan akurat.

- c. Pertimbangkan penggunaan teknik untuk mengurangi data yang dapat membantu menyempurnakan pemahaman terhadap dataset yang luas, mempercepat proses analisis, dan meningkatkan efisiensi.
- d. Pertimbangkan untuk dapat menggunakan dataset dari berbagai sumber, tidak hanya dari *Naive Bayes*, bisa menggunakan data Twitter Api, dsb



## DAFTAR PUSTAKA

- Agustina, N., Citra, D. H., Purnama, W., Nisa, C., & Kurnia, A. R. (2022). Implementasi Algoritma *Naive Bayes* untuk Analisis Sentimen Ulasan Shopee pada Google Play Store. *MALCOM: Indonesian Journal of Machine Learning and Computer Science*, 2(1), 47–54. <https://doi.org/10.57152/malcom.v2i1.195>
- Ahmed, T., Mukta, S. F., Al Mahmud, T., Hasan, S. Al, & Gulzar Hussain, M. (2022). Bangla Text Emotion Classification using LR, MNB and MLP with TF-IDF & CountVectorizer. *ICSEC 2022 - International Computer Science and Engineering Conference 2022, December*, 275–280. <https://doi.org/10.1109/ICSEC56337.2022.10049341>
- Alhaq, Z., Mustopa, A., Mulyatun, S., & Santoso, J. D. (2021). Penerapan Metode Support Vector Machine Untuk Analisis Sentimen Pengguna Twitter. *Journal of Information System Management (JOISM)*, 3(2), 44–49. <https://doi.org/10.24076/joism.2021v3i2.558>
- Bengesi, S., Oladunni, T., Olusegun, R., & Audu, H. (2023). A *Machine Learning*-Sentiment Analysis on Monkeypox Outbreak: An Extensive Dataset to Show the Polarity of Public Opinion From Twitter Tweets. *IEEE Access*, 11(January), 11811–11826. <https://doi.org/10.1109/ACCESS.2023.3242290>
- Bintang Zulfikar Ramadhan, Ibnu Riza, & Iqbal Maulana. (2023). Analisis Sentimen Ulasan Pada Aplikasi *E-commerce* Shopee Dengan Menggunakan Algoritma *Naive Bayes*. *Jurnal Teknik Mesin, Elektro Dan Ilmu Komputer*, 3(3), 23–39. <https://doi.org/10.55606/teknik.v3i3.2411>
- Birjali, M., Kasri, M., & Beni-Hssane, A. (2021). A comprehensive survey on sentiment analysis: Approaches, challenges and trends. *Knowledge-Based Systems*, 226, 107134. <https://doi.org/10.1016/j.knosys.2021.107134>
- Cahyono, N., & Anggista Oktavia Praneswara. (2023). Analisis Sentimen Ulasan Aplikasi TikTok Shop Seller Center di Google Playstore Menggunakan Algoritma *Naive Bayes*. *Indonesian Journal of Computer Science*, 12(6), 3925–3940. <https://doi.org/10.33022/ijcs.v12i6.3473>
- Chandel, S., Clement, C. B., Serrato, G., & Sundaresan, N. (2022). *Training and Evaluating a Jupyter Notebook Data Science Assistant*. 1–8. <http://arxiv.org/abs/2201.12901>
- Darmawan, G., Alam, S., & Sulistyono, M. I. (2023). Analisis Sentimen Berdasarkan Ulasan Pengguna Aplikasi Mypertamina Pada Google Playstore Menggunakan Metode *Naive Bayes*. *STORAGE – Jurnal Ilmiah Teknik Dan Ilmu Komputer*, 2(3), 100–108.
- Demircan, M., Seller, A., Abut, F., & Akay, M. F. (2021). Developing Turkish sentiment analysis models using *machine learning* and *e-commerce* data. *International Journal of Cognitive Computing in Engineering*, 2(July), 202–207. <https://doi.org/10.1016/j.ijcce.2021.11.003>
- Dikananda, F., Rinaldi Dikananda, A., & Anwar, S. (2024). Analisis Sentimen Terhadap Marketplace Menggunakan Algoritma *Naive Bayes*. *JATI (Jurnal Mahasiswa Teknik Informatika)*, 8(4), 8219–8225. <https://doi.org/10.36040/jati.v8i4.10965>
- Ernianti Hasibuan, & Elmo Allistair Heriyanto. (2022). Analisis Sentimen Pada Ulasan

- Aplikasi Amazon Shopping Di Google Play Store Menggunakan *Naive Bayes Classifier*. *Jurnal Teknik Dan Science*, 1(3), 13–24. <https://doi.org/10.56127/jts.v1i3.434>
- Fahrezi, A., Salam, F. N., Ibrahim, G. M., Syaiful, R. R., & Saifudin, A. (2022). Pengujian Black Box Testing pada Aplikasi Inventori Barang Berbasis Web di PT. AINO Indonesia. *LOGIC: Jurnal Ilmu Komputer Dan Pendidikan*, 1(1), 1–5. <https://journal.mediapublikasi.id/index.php/logic>
- Fitriani, Y., Utami, S., & Junadi, B. (2022). Perancangan Sistem Informasi Human Capital Management Berbasis Website. *Journal of Information System, Applied, Management, Accounting and Research*, 6(4), 792–803. <https://doi.org/10.52362/jisamar.v6i4.919>
- Fransiska, S., & Irham Gufroni, A. (2020). Sentiment Analysis Provider by.U on Google Play Store Reviews with TF-IDF and Support Vector Machine (SVM) Method. *Scientific Journal of Informatics*, 7(2), 2407–7658. <http://journal.unnes.ac.id/nju/index.php/sji>
- Ghimire, D. (2020). Comparative study on Python web frameworks: Flask and Django. *Metropolia University of Applied Sciences*, May, 13–33. <https://urn.fi/URN:NBN:fi:amk-2020052513398>
- Guntara, R. G. (2023). Visualisasi Data Laporan Penjualan Toko Online Melalui Pendekatan Data Science Menggunakan Google Colab. *Jurnal Ilmiah Multidisiplin*, 2(6), 2091–2100.
- Gur, I., Nachum, O., Miao, Y., Safdari, M., Huang, A., Chowdhery, A., Narang, S., Fiedel, N., & Faust, A. (2023). Understanding HTML with Large Language Models. *Findings of the Association for Computational Linguistics: EMNLP 2023*, 2803–2821. <https://doi.org/10.18653/v1/2023.findings-emnlp.185>
- Handika. (2024). *Pemanfaatan Python dan Google Colab Dalam Pembelajaran Statistika Deskriptif*. 379–389.
- Huang, H., Zavareh, A. A., & Mustafa, M. B. (2023). Sentiment Analysis in *E-commerce* Platforms: A Review of Current Techniques and Future Directions. *IEEE Access*, 11(August), 90367–90382. <https://doi.org/10.1109/ACCESS.2023.3307308>
- Jannah, N. Z. B., & Kusnawi, K. (2024). Comparison of Naïve Bayes and SVM in Sentiment Analysis of Product Reviews on Marketplaces. *Sinkron*, 8(2), 727–733. <https://doi.org/10.33395/sinkron.v8i2.13559>
- Juanita, S., Adiyarta, K., & Syafrullah, M. (2022). Sentiment analysis on E-Marketplace User Opinions Using Lexicon-Based and Naïve Bayes Model. *International Conference on Electrical Engineering, Computer Science and Informatics (EECSI), 2022-Octob*(October 2022), 379–382. <https://doi.org/10.23919/EECSI56542.2022.9946537>
- Kristiyanti, D. A., Putri, D. A., Indrayuni, E., Nurhadi, A., & Umam, A. H. (2020). E-Wallet Sentiment Analysis Using Naïve Bayes and Support Vector Machine Algorithm. *Journal of Physics: Conference Series*, 1641(1). <https://doi.org/10.1088/1742-6596/1641/1/012079>
- Kusuma, I. H., & Cahyono, N. (2023). Analisis Sentimen Masyarakat Terhadap Penggunaan *E-commerce* Menggunakan Algoritma K-Nearest Neighbor. *Jurnal Informatika: Jurnal*

- Pengembangan IT*, 8(3), 302–307. <https://doi.org/10.30591/jpit.v8i3.5734>
- Li, Z., Li, R., & Jin, G. (2020). Sentiment analysis of danmaku videos based on naïve bayes and sentiment dictionary. *IEEE Access*, 8, 75073–75084. <https://doi.org/10.1109/ACCESS.2020.2986582>
- Marchionini, G. (2023). Information and data sciences: Context, units of analysis, meaning, and human impact. *Data and Information Management*, 7(1), 100031. <https://doi.org/10.1016/j.dim.2023.100031>
- Musfiroh, D., Khaira, U., Utomo, P. E. P., & Suratno, T. (2021). Analisis Sentimen terhadap Perkuliahan Daring di Indonesia dari Twitter Dataset Menggunakan InSet Lexicon. *MALCOM: Indonesian Journal of Machine learning and Computer Science*, 1(1), 24–33. <https://doi.org/10.57152/malcom.v1i1.20>
- Nur'aeni, N. N., Ainulyaqin, M., & Edy, S. (2024). Dampak Fenomena *E-commerce* Pada Tingkat Penjualan Di Pasar Tradisional Ditinjau Dari Psikologi Dan Ekonomi Islam. *Jurnal Ilmiah Ekonomi Islam*, 10(1), 270. <https://doi.org/10.29040/jiei.v10i1.12146>
- Oktaviani, V., Warsito, B., Yasin, H., Santoso, R., & Suparti. (2021). Sentiment analysis of *e-commerce* application in Traveloka data review on Google Play site using Naïve Bayes classifier and association method. *Journal of Physics: Conference Series*, 1943(1). <https://doi.org/10.1088/1742-6596/1943/1/012147>
- Pölsterl, S. (2020). Scikit-survival: A library for time-to-event analysis built on top of scikit-learn. *Journal of Machine Learning Research*, 21, 1–6.
- Pratmanto, D., Rousyati, R., Wati, F. F., Widodo, A. E., Suleman, S., & Wijianto, R. (2020). App Review Sentiment Analysis Shopee Application in Google Play Store Using *Naive Bayes* Algorithm. *Journal of Physics: Conference Series*, 1641(1). <https://doi.org/10.1088/1742-6596/1641/1/012043>
- Putri, D. A., Kristiyanti, D. A., Indrayuni, E., Nurhadi, A., & Hadinata, D. R. (2020). Comparison of *Naive Bayes* Algorithm and Support Vector Machine using PSO Feature Selection for Sentiment Analysis on E-Wallet Review. *Journal of Physics: Conference Series*, 1641(1), 0–6. <https://doi.org/10.1088/1742-6596/1641/1/012085>
- Rahman, A. S. F. N. A. Y. (2024). Analisis sentimen aplikasi shopee, tokopedia, lazada dan blibli menggunakan leksikon dan random forest. 12(3).
- Ranjan, S., & Mishra, S. (2020). *Comparative Sentiment Analysis of App Reviews*.
- Raza, G. M., Butt, Z. S., Latif, S., & Wahid, A. (2021). Sentiment Analysis on COVID Tweets: An Experimental Analysis on the Impact of *Count Vectorizer* and TF-IDF on Sentiment Predictions using Deep Learning Models. *2021 International Conference on Digital Futures and Transformative Technologies, ICoDT2 2021*. <https://doi.org/10.1109/ICoDT252288.2021.9441508>
- Rukhmana, T. (2021). Jurnal Edu Research Indonesian Institute For Corporate Learning And Studies (IICLS) Page 25. *Jurnal Edu Research : Indonesian Institute For Corporate Learning And Studies (IICLS)*, 2(2), 28–33.
- Rusdy, I., Canbulat, I., Zhang, C., Wei, C., & McQuillan, A. (2024). The development and implementation of design *Flowchart* for probabilistic rock slope stability assessments: a review. *Geoenvironmental Disasters*, 11(1). <https://doi.org/10.1186/s40677-024->

- Samsir, Irmayani, D., Edi, F., Harahap, J. M., Jupriaman, Rangkuti, R. K., Ulya, B., & Watrianthos, R. (2021). Naives Bayes Algorithm for Twitter Sentiment Analysis. *Journal of Physics: Conference Series*, 1933(1), 0–6. <https://doi.org/10.1088/1742-6596/1933/1/012019>
- Sendi Alpin Rizaldi, Syariful Alam, I. K. (2023). *ANALISIS SENTIMEN PENGGUNA APLIKASI JMO ( JAMSOSTEK MOBILE ) PADA GOOGLE PLAY STORE MENGGUNAKAN METODE NAIVE BAYES*. 2(3), 109–117.
- Sham, N. M., & Mohamed, A. (2022). Climate Change Sentiment Analysis Using Lexicon, *Machine learning* and Hybrid Approaches. *Sustainability (Switzerland)*, 14(8). <https://doi.org/10.3390/su14084723>
- Shenify, M. (2024). Sentiment analysis of Saudi *e-commerce* using naïve bayes algorithm and support vector machine. *International Journal of Data and Network Science*, 8(3), 1607–1612. <https://doi.org/10.5267/j.ijdns.2024.3.006>
- Suryani, L., & Edy, K. (2020). Pengembangan Aplikasi “Lost & Found” Berbasis Android Dengan Menggunakan Metode Term Frequency – Inverse Document Frequency (Tf-Idf) Dan Cosine Similarity. *Electro Luceat*, 6(2), 190–204. <https://doi.org/10.32531/jelekn.v6i2.232>
- Tabinda Kokab, S., Asghar, S., & Naz, S. (2022). Transformer-based deep learning models for the sentiment analysis of social media data. *Array*, 14(April), 100157. <https://doi.org/10.1016/j.array.2022.100157>
- Tangraeni, A. I., & Sitokdana, M. N. N. (2022). Analisis Sentimen Aplikasi E-Government pada Google Play Menggunakan Algoritma Naïve Bayes. *JATISI (Jurnal Teknik Informatika Dan Sistem Informasi)*, 9(2), 785–795. <https://doi.org/10.35957/jatisi.v9i2.1835>
- Tri Putra, K., Amin Hariyadi, M., & Crysdiyan, C. (2023). Perbandingan Feature Extraction Tf-Idf Dan Bow Untuk Analisis Sentimen Berbasis Svm. *Jurnal Cahaya MAndalika*, 1449.
- Turki, T., & Roy, S. S. (2022). Novel Hate Speech Detection Using Word Cloud Visualization and Ensemble Learning Coupled with *Count Vectorizer*. *Applied Sciences (Switzerland)*, 12(13). <https://doi.org/10.3390/app12136611>
- Ulhaq, D., Masa, A. P. A., & Setyadi, H. J. (2024). Analisis Perbandingan User Experience Aplikasi *E-commerce* Shopee Dan Tokopedia Menggunakan Metode User Experience Questionnaire (UEQ). *Adopsi Teknologi Dan Sistem Informasi (ATASI)*, 3(1), 37–44. <https://doi.org/10.30872/atasi.v3i1.1183>
- Ulya, S., Ridwan, A., Cholid Wahyudin, W., & Hana, F. M. (2022). Text Mining Sentimen Analisis Pengguna Aplikasi Marketplace Tokopedia Berdasar Rating dan Komentar Pada Google Play Store. *Jurnal Bisnis Digital Dan Sistem Informasi*, 3(2), 33–40. <https://ejr.umku.ac.id/index.php/BIDISFO/article/view/1799>
- V. Kevin Sitanayah Que, Ade Iriani, H. D. P. (2020). Analisis Sentimen Transportasi Online Menggunakan Support Vector Machine Berbasis Particle Swarm Optimization. *Jurnal Nasional Teknik Elektro Dan Teknologi Informasi*, 9(2), 162–170. <https://doi.org/10.22146/jnteti.v9i2.102>

- Valero-Carreras, D., Alcaraz, J., & Landete, M. (2023). Comparing two SVM models through different metrics based on the *Confusion Matrix*. *Computers and Operations Research*, 152(December 2022), 106131. <https://doi.org/10.1016/j.cor.2022.106131>
- Wiyanto, W., Fadhilah, S., & Siswandi, A. (2022). E-Tourism Sebagai Media Wisata Kabupaten Bekasi Berbasis Website. *Journal of Practical Computer Science*, 2(1), 1–14. <https://doi.org/10.37366/jpcs.v2i1.1035>



## LAMPIRAN



**UNIVERSITAS BUDDHI DHARMA**

Jl. Imam Bonjol No. 41 Karawaci Ilir, Tangerang  
021 5517853 / 021 5586822 admin@buddhidharma.ac.id

### KARTU BIMBINGAN TA/SKRIPSI

NIM : 20201000012  
Nama Mahasiswa : STEVEN  
Fakultas : Sains dan Teknologi  
Program Studi : Teknik Informatika  
Jenjang : Strata Satu  
Tahun Akademik/Semester : 2024/2025 Ganjil  
Dosen Pembimbing : Indah Fenriana, S.Kom.,M.Kom  
Judul Skripsi : ANALISIS SENTIMEN MEMBANDINGKAN PENGGUNA APLIKASI  
: E-COMMERCE TOKOPEDIA DAN SHOPEE MENGGUNAKAN  
ALGORITMA NAIVE BAYES

Tanggal	Catatan	Paraf
2024-09-09	Pengajuan judul	X
2024-09-19	Bimbingan bab 1	X
2024-09-26	Rancangan program	X
2024-10-10	Bimbingan bab 2, revisi bab 1	X
2024-10-17	Bimbingan program	X
2024-10-24	Bimbingan bab 3, revisi bab 2	X
2024-10-31	Bimbingan program, revisi bab 3	X
2024-11-21	Bimbingan bab 4	X
2024-11-28	Bimbingan bab 5, revisi bab 4	X
2024-12-05	Bimbingan program	X
2024-12-12	Bimbingan semua paper dan program (ACC maju sidang)	X

Mengetahui  
Ketua Program Studi



Hartana Wijaya, M.Kom

Tangerang, 06 January 2025

Pembimbing



Indah Fenriana, S.Kom.,M.Kom

## DAFTAR RIWAYAT HIDUP



### Data Pribadi

Nama Lengkap : Steven  
Tempat/Tanggal Lahir : Jakarta, 05 September 2001  
Jenis Kelamin : Laki-Laki  
Alamat : Jl. Darma Wangsa Raya  
Agama : Buddha  
Telepon : 081310536228  
Email : chousteven61@gmail.com

### Pendidikan Formal

2007-2012 : SD Galatia 3  
2012-2014 : SD Mutiara Kasih  
2014-2017 : SMP Mutiara Kasih  
2017-2020 : SMK Mutiara Kasih

### Pengalaman Organisasi :

PERFOBUMI (Multimedia, 2022 – Sekarang)  
MYGen (Multimedia, 2022 – Sekarang)

### Pengalaman Kerja :

PT.Perkasa Multindo Sejahtera (Online Marketing, November 2018 – Januari 2019)

Tangerang, 05 Februari 2025

Steven

20201000012

### Lampiran 3 Listing Program Model Pelatihan

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.naive_bayes import MultinomialNB
from sklearn.metrics import confusion_matrix, classification_report, accuracy_score
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np
data = pd.read_csv("Datakeseluruhan.csv") #data tokopedia dan shopee
df = data.dropna()
X_train, X_test, y_train, y_test = train_test_split(
    df['stemming_data'],
    df['sentiment_label'],
    test_size=0.2,
    random_state=42,
    stratify=df['sentiment_label']
)
print(f'Jumlah data latih: {len(X_train)}')
print(f'Jumlah data uji: {len(X_test)}')
# TF-IDF Vectorizer
tfidf_vectorizer = TfidfVectorizer(
    min_df=2,
    max_df=0.8,
    ngram_range=(1, 2),
    max_features=10000
)
X_train_tfidf = tfidf_vectorizer.fit_transform(X_train.fillna(""))
X_test_tfidf = tfidf_vectorizer.transform(X_test.fillna(""))

# Count Vectorizer
count_vectorizer = CountVectorizer(
    min_df=2,
    max_df=0.8,
```

```

ngram_range=(1, 2),
max_features=10000
)
X_train_count = count_vectorizer.fit_transform(X_train.fillna(""))
X_test_count = count_vectorizer.transform(X_test.fillna(""))

# TF-IDF Model
mnb_tfidf = MultinomialNB()
mnb_tfidf.fit(X_train_tfidf, y_train)
y_pred_tfidf = mnb_tfidf.predict(X_test_tfidf)

# Count Vector Model
mnb_count = MultinomialNB()
mnb_count.fit(X_train_count, y_train)
y_pred_count = mnb_count.predict(X_test_count)

# Evaluasi TF-IDF
print("TF-IDF Vectorizer Results:")
print("Accuracy:", accuracy_score(y_test, y_pred_tfidf))
print("\nClassification Report:")
print(classification_report(y_test, y_pred_tfidf))
print("\nConfusion Matrix:")
print(confusion_matrix(y_test, y_pred_tfidf))

# Evaluasi Count Vectorizer
print("Count Vectorizer Results:")
print("Accuracy:", accuracy_score(y_test, y_pred_count))
print("\nClassification Report:")
print(classification_report(y_test, y_pred_count))
print("\nConfusion Matrix:")
print(confusion_matrix(y_test, y_pred_count))
print("\n" + "="*50 + "\n")

```

```

# Perbandingan Detail
print("\nkomparasi Detail:")
print("TF-IDF Akurasi:", accuracy_score(y_test, y_pred_tfidf))
print("Count Vector Akurasi:", accuracy_score(y_test, y_pred_count))

```

#### Lampiran 4 Listing Program *Paired T-Test*

```

# Data hasil dari perhitungan sebelumnya (asumsikan sudah ada)
classes = np.unique(y_test)
tfidf_accuracies = get_accuracy_scores(tfidf, nb_classifier, X_train, X_test, y_train, y_test)
count_accuracies = get_accuracy_scores(count, nb_classifier, X_train, X_test, y_train, y_test)
# Melakukan paired t-test
t_statistic, p_value = stats.ttest_rel(tfidf_accuracies, count_accuracies)
# Menampilkan hasil uji t
print("Hasil Paired T-Test:")
print(f't-statistic: {t_statistic:.4f}')
print(f'p-value: {p_value:.4f}')
# Kesimpulan berdasarkan nilai p-value
alpha = 0.05
if p_value < alpha:
    print("Kesimpulan: Terdapat perbedaan signifikan antara metode TF-IDF dan Count Vectorizer.")
else:
    print("Kesimpulan: Tidak terdapat perbedaan signifikan antara metode TF-IDF dan Count Vectorizer.")

```

#### Lampiran 5 Listing Program HTML Tampilan Utama

```

<!DOCTYPE html>
<html lang="id">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Web Membandingkan Sentimen</title>
    <linkrel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">
</head>
<body>
<div class="container mt-5">

```

```

<h1 class="text-center">Membandingkan Sentimen App Play Store</h1>
<div class="data-card mt-4">
  <form id="scrapeForm" class="scrapeForm" action="/compare" method="POST">
    <h4>Input Data 1</h4>
    <div class="mb-3">
      <label for="url1" class="form-label">Naive Bayes URL 1:</label>
      <input type="url" class="form-control" id="url1" name="url1" required>
    </div>
    <div class="mb-3">
      <label for="num_reviews1" class="form-label">Number of Reviews 1:</label>
      <input type="number" class="form-control" id="num_reviews1" name="num_reviews1" min="1 "
required>
      <small class="form-text text-muted">
        </small>
    </div>
    <h4>Input Data 2</h4>
    <div class="mb-3">
      <label for="url2" class="form-label">Naive Bayes URL 2:</label>
      <input type="url" class="form-control" id="url2" name="url2" required>
    </div>
    <div class="mb-3">
      <label for="num_reviews2" class="form-label">Number of Reviews 2:</label>
      <input type="number" class="form-control" id="num_reviews2" name="num_reviews2" min="1 "
required>
      <small class="form-text text-muted">
        </small>
      <button type="submit" class="btn btn-primary">Kirim</button>
    </div>
  </form>
</div>
</body>
</html>

```

## Lampiran 6 Listing Program Fungsi Scraping Dua Data

try :

```
# Mengambil ulasan untuk aplikasi pertama
reviews1, _ = reviews(app_id1, lang='en', country='us', sort=Sort.NEWEST, count=num_reviews1)
df1 = pd.DataFrame(reviews1) # Mengubah ulasan menjadi DataFrame
processed_data1 = process_reviews_data(df1) # Memproses data
logging.info(f"Memulai pengambilan ulasan untuk {url2}...")
# Mengambil ulasan untuk aplikasi kedua
reviews2, _ = reviews(app_id2, lang='en', country='us', sort=Sort.NEWEST, count=num_reviews2)
df2 = pd.DataFrame(reviews2) # Mengubah ulasan menjadi DataFrame
except NotFoundError:
    logging.error("Aplikasi tidak ditemukan. Pastikan URL yang dimasukkan benar.")
    return "Aplikasi tidak ditemukan. Pastikan URL yang dimasukkan benar."
```

## Lampiran 7 Listing Program Membaca Data Yang Sudah Dilatih

```
from flask import Flask, request, render_template
from joblib import load
import pandas as pd
from collections import Counter

# Memuat model yang telah disimpan
naive_bayes_model = load('naive_bayes_model.joblib')

# Menggunakan TF-IDF yang telah disimpan sebelumnya
tfidf_vectorizer = load('tfidf_vectorizer.joblib')

# Gunakan model yang telah dimuat untuk membuat prediksi
X1 = tfidf_vectorizer.transform(processed_data1['stemming_data'])
X2 = tfidf_vectorizer.transform(processed_data2['stemming_data'])
processed_data1['label'] = naive_bayes_model.predict(X1)
processed_data2['label'] = naive_bayes_model.predict(X2)
```

## Lampiran 8 Listing Program HTML Menampilkan Pie Chart 2 Data

<https://cdn.jsdelivr.net/npm/chart.js>

```
<h2>Prediction Result Pie Charts</h2>
<div class="chart-container">
  <div class="chart-wrapper">
    <canvas id="myPieChart1"></canvas>
  </div>
  <div class="chart-wrapper">
    <canvas id="myPieChart2"></canvas>
  </div>
</div>
```

## Lampiran 9 Listing Program HTML Menampilkan Kata Paling Banyak Di Bahas

```
<div class="word-stats">
  <div class="word-stats-column">
    <h2>Kata Paling Banyak Pada - App 1</h2>
    <ul>
      {% for word, count in most_common_words1 %}
      <li>
        <span>{{ word }}</span>
        <span class="word-count">{{ count }}</span>
      </li>
      {% endfor %}
    </ul>
    <h3>Example Comments</h3>
    <div class="example-comments">
      {% for word, comment in example_comments1.items() %}
      <div class="comment-item">
        <div class="comment-header">Contains keyword: <span class="keyword">{{ word
}}</span></div>
        {{ comment }}
      </div>
      {% endfor %}
    </div>
```

```

</div>
<div class="word-stats-column">
  <h2>Kata Paling Banyak Pada - App 2</h2>
  <ul>
    {% for word, count in most_common_words2 %}
    <li>
      <span>{{ word }}</span>
      <span class="word-count">{{ count }}</span>
    </li>
    {% endfor %}
  </ul>
  <h3>Example Comments</h3>
  <div class="example-comments">
    {% for word, comment in example_comments2.items() %}
    <div class="comment-item">
      <div class="comment-header">Contains keyword: <span class="keyword">{{ word
}}</span></div>
      {{ comment }}
    </div>
    {% endfor %}
  </div>
</div>
</div>
</div>
<div style="text-align: center; margin-top: 20px;">
  <a href="/" class="btn btn-primary">Kembali ke Home</a>
</div>
<script>
const chartColors = {
  backgroundColor: [
    'rgba(255, 99, 132, 0.2)',
    'rgba(54, 162, 235, 0.2)',
    'rgba(255, 206, 86, 0.2)',
    'rgba(75, 192, 192, 0.2)',

```

```

'rgba(153, 102, 255, 0.2)',
'rgba(255, 159, 64, 0.2)'
],
borderColor: [
'rgba(255, 99, 132, 1)',
'rgba(54, 162, 235, 1)',
'rgba(255, 206, 86, 1)',
'rgba(75, 192, 192, 1)',
'rgba(153, 102, 255, 1)',
'rgba(255, 159, 64, 1)'
]
};
Function createPieChart(elementId, data, chartTitle) {
const ctx = document.getElementById(elementId).getContext('2d');
const labels = Object.keys(data);
const values = Object.values(data);

return new Chart(ctx, {
type: 'pie',
data: {
labels: labels,
datasets: [{
label: chartTitle,
data: values,
backgroundColor: chartColors.backgroundColor,
borderColor: chartColors.borderColor,
borderWidth: 1
}]
},
options: {
responsive: true,
maintainAspectRatio: true, // Added to maintain aspect ratio
aspectRatio: 1, // Makes the chart a perfect circle
plugins: {

```

```
legend: {  
  position: 'top',  
  labels: {  
    font: {  
      size: 12 // Reduced font size for legend  
    }  
  }  
},  
title: {
```

