

## BAB V

### SIMPULAN DAN SARAN

#### 5.1 Simpulan

Berdasarkan hasil penelitian ini diperoleh kesimpulan sebagai berikut, yang sekaligus menjawab poin-poin tujuan dan manfaat penelitian:

1. Penelitian berhasil mengembangkan dan mengimplementasikan aplikasi *web sentiment analysis* berbasis *Flask framework* yang mampu menganalisis ulasan pelanggan Tokopedia secara otomatis. Sistem yang dibangun telah terintegrasi dengan fitur *automated web scraping* menggunakan Selenium, *real-time sentiment classification* dengan akurasi 85-95%, dan *interactive dashboard* yang dilengkapi *role-based access control* untuk memisahkan akses antara *admin* dan *stakeholder business*, sehingga memastikan keamanan dan efisiensi pengelolaan data.
2. Sistem berhasil mengidentifikasi dan menganalisis sentimen pelanggan terhadap produk-produk di Tokopedia melalui *automated data processing* yang menghasilkan visualisasi interaktif komprehensif. Implementasi *word cloud*, *sentiment distribution charts*, dan *keyword analysis* yang dapat diakses melalui *web dashboard* telah terbukti memberikan *business intelligence yang actionable* dan memudahkan interpretasi data sentimen secara visual dan intuitif.
3. Penelitian berhasil menyediakan rekomendasi strategis berbasis data melalui *stakeholder dashboard* yang menampilkan *insights* komprehensif dengan dukungan fitur *automated reporting* dalam format PDF dan *admin-curated analysis*. Sistem ini terbukti membantu pengambilan keputusan dalam peningkatan kualitas produk dan optimalisasi *customer experience* secara *data-driven*, dengan mereduksi waktu pemrosesan dari 2-3 jam manual menjadi 5-10 menit otomatis untuk 1000+ ulasan.

4. Sistem terintegrasi yang dikembangkan berhasil mendemonstrasikan koneksi antara *automated sentiment analysis* dengan *business actionable insights* melalui *web-based solution* yang dapat digunakan secara *real-time*. Implementasi ini memungkinkan *strategic decision making* yang lebih efektif dalam bidang pemasaran, *product development*, dan *customer service optimization* dengan dukungan data sentimen yang akurat dan terkini.
5. Penelitian memberikan kontribusi akademis signifikan dalam penerapan *web-based sentiment analysis* untuk *e-commerce* Indonesia dengan mendemonstrasikan integrasi teknologi modern (*Flask, scikit-learn, Selenium*) dalam menghasilkan *end-to-end business intelligence solution* yang *scalable* dan *user-friendly*. Kontribusi ini mencakup pengembangan framework yang dapat diadaptasi untuk platform *e-commerce* lainnya di Indonesia, serta metodologi yang dapat menjadi referensi untuk penelitian sejenis di masa mendatang.

## 5.2 Saran

1. Mengembangkan *RESTful API* untuk meningkatkan *scalability* dan memungkinkan integrasi dengan sistem lain.
2. Mengembangkan *multi-platform support* untuk menganalisis ulasan dari platform lain seperti Shopee dan Bukalapak.
3. Menambahkan *advanced NLP techniques* seperti BERT yang dioptimalkan untuk *Indonesian language processing* untuk meningkatkan akurasi analisis sentimen.
4. Mengembangkan *automated alert system* untuk memberikan notifikasi proaktif ketika terjadi perubahan sentimen yang signifikan
5. Berikan bobot berdasarkan panjang teks atau tingkat keyakinan model untuk meningkatkan *confidence score*.

## DAFTAR PUSTAKA

- Acero, A., & Ronaldo, I. (2024). *DeskDealFinder: Application for Online Shopping Optimization on Desktop Products with Web Scraping*. <https://doi.org/10.20944/preprints202407.2032.v1>
- Acero, I. (2024). *DeskDealFinder: Application for Online Shopping Optimization on Desktop Products with Web Scraping*. <https://doi.org/10.20944/preprints202407.2032.v1>
- Aditya, C. S. K., Wicaksono, G. W., Wicaksono, G. W., Heryawan, H. A. S., & Heryawan, H. A. S. (2023). Sentiment Analysis of the 2024 Presidential Candidates Using SMOTE and Long Short Term Memory. *Jurnal Informatika Universitas Pamulang*, 8(2), 279–286. <https://doi.org/10.32493/informatika.v8i2.32210>
- Anreaja, L. J., Harefa, N. N., Galih, J., Negara, P., Nathan, V., Pribyantara, H., & Prasetyo, A. B. (2022a). *JISA (Jurnal Informatika dan Sains) Naive Bayes and Support Vector Machine Algorithm for Sentiment Analysis Opensea Mobile Application Users in Indonesia*.
- Anreaja, L. J., Harefa, N. N., Galih, J., Negara, P., Nathan, V., Pribyantara, H., & Prasetyo, A. B. (2022b). *JISA (Jurnal Informatika dan Sains) Naive Bayes and Support Vector Machine Algorithm for Sentiment Analysis Opensea Mobile Application Users in Indonesia*.
- Barrett, P. E., Shopbell, P. L., Britton, M. C., Ebert, R., Barrett, P., Hunter, J., Miller, J. T., Hsu, J.-C., & Greenfield, P. (2005). *matplotlib-A Portable Python Plotting Package* (Vol. 347). <https://www.researchgate.net/publication/234238535>
- Chamekh, A., Mahfoudh, M., & Forestier, G. (2022a). *Sentiment Analysis Based on Deep Learning in E-Commerce*. 498–507. [https://doi.org/10.1007/978-3-031-10986-7\\_40i](https://doi.org/10.1007/978-3-031-10986-7_40i)
- Chamekh, A., Mahfoudh, M., & Forestier, G. (2022b). *Sentiment Analysis Based on Deep Learning in E-Commerce*. 498–507. [https://doi.org/10.1007/978-3-031-10986-7\\_40i](https://doi.org/10.1007/978-3-031-10986-7_40i)
- Christanto, H. J., Sondra, S., Widodo, A., Dewi, C., Singgalen, Y. A., Riantama, D., Dayarana, A., & Silalahi, K. (2024). SENTIMENT ANALYSIS FOR TIKTOK SHOP'S CLOSURE IN INDONESIA USING NAIVE BAYES MODELS AND NLP. *Journal of Theoretical and Applied Information Technology*, 15(7). [www.jatit.org](http://www.jatit.org)
- Deng, Y., Wang, H., & Wu, Y. (2024a). Public opinion and Chinese exports: evidence from Twitter sentiment analysis. *Journal of the Asia Pacific Economy*. <https://doi.org/10.1080/13547860.2024.2403512>
- Deng, Y., Wang, H., & Wu, Y. (2024b). Public opinion and Chinese exports: evidence from Twitter sentiment analysis. *Journal of the Asia Pacific Economy*. <https://doi.org/10.1080/13547860.2024.2403512>
- Gusniawan Pradana, E. (n.d.). IMPLEMENTASI WEB CRAWLER UNTUK MENCARI HARGA BARANG TERMURAH DARI BERBAGAI SITUS E-COMMERCE INDONESIA. In *Teknologipintar.org* (Vol. 2, Issue 9).
- Habib Kusuma, I., & Cahyono, N. (2023). *Analisis Sentimen Masyarakat Terhadap Penggunaan E-Commerce Menggunakan Algoritma K-Nearest Neighbor*. 8(3).
- Hadasik, B. (2024). *Reduction of information asymmetry in e-commerce: the web scraping approach*. Wydawnictwo Uniwersytetu Ekonomicznego w Katowicach. <https://doi.org/10.22367/9788378758907>

- Hanji, G., Menon, A. B., Student, A. R., Patil, A., & Nandeni, B. (2023). Harnessing Flask for Web Scraping and Sentiment Analysis: A Comprehensive Application for News and E-Commerce Reviews. In *International Journal of Computer Applications* (Vol. 185, Issue 16).
- Hariguna, T., Maulana Baihaqi, W., & Nurwanti, A. (2019). Sentiment Analysis of Product Reviews as A Customer Recommendation Using the Naive Bayes Classifier Algorithm. *International Journal of Informatics and Information Systems*, 2(2), 48–55.
- Karimi, Z. (2021). *Confusion Matrix*. <https://www.researchgate.net/publication/355096788>
- Khder, M. A. (2021a). Web scraping or web crawling: State of art, techniques, approaches and application. *International Journal of Advances in Soft Computing and Its Applications*, 13(3), 144–168. <https://doi.org/10.15849/ijasca.211128.11>
- Khder, M. A. (2021b). Web scraping or web crawling: State of art, techniques, approaches and application. *International Journal of Advances in Soft Computing and Its Applications*, 13(3), 144–168. <https://doi.org/10.15849/ijasca.211128.11>
- Lasatira, G. S., Hartomo, K. D., & Sembiring, I. (2023). Analisis Sentimen Terhadap Pengaruh Minat Belanja Berdasarkan Komentar di Marketplace Menggunakan Metode Recurrent Neural Network (RNN). *Jurnal Sistem Informasi Bisnis*, 13(2), 112–119. <https://doi.org/10.21456/vol13iss2pp112-119>
- Mantik, J., Saputri, Y. R., & Februariyanti, H. (2022). SENTIMENT ANALYSIS ON SHOPEE E-COMMERCE USING THE NAÏVE BAYES CLASSIFIER ALGORITHM. In *Jurnal Mantik* (Vol. 6, Issue 2).
- Mckinney, W. (n.d.). *pandas: a Foundational Python Library for Data Analysis and Statistics*. <https://www.researchgate.net/publication/265194455>
- Muliawan, J., & Dazki, E. (2023a). SENTIMENT ANALYSIS OF INDONESIA'S CAPITAL CITY RELOCATION USING THREE ALGORITHMS: NAÏVE BAYES, KNN, AND RANDOM FOREST. *Jurnal Teknik Informatika (JUTIF)*, 4(5), 1227–1236. <https://doi.org/10.52436/1.jutif.2023.4.5.347>
- Muliawan, J., & Dazki, E. (2023b). SENTIMENT ANALYSIS OF INDONESIA'S CAPITAL CITY RELOCATION USING THREE ALGORITHMS: NAÏVE BAYES, KNN, AND RANDOM FOREST. *Jurnal Teknik Informatika (JUTIF)*, 4(5), 1227–1236. <https://doi.org/10.52436/1.jutif.2023.4.5.347>
- Mustakim, H., & Priyanta, S. (2022). Aspect-Based Sentiment Analysis of KAI Access Reviews Using NBC and SVM. *IJCCS (Indonesian Journal of Computing and Cybernetics Systems)*, 16(2), 113. <https://doi.org/10.22146/ijccs.68903>
- 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>
- Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Müller, A., Nothman, J., Louppe, G., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, É. (2018). *Scikit-learn: Machine Learning in Python*. <http://arxiv.org/abs/1201.0490>
- Pramudita, D., Akbar, Y., & Wahyudi, T. (2024a). Analisis Sentimen Terhadap Program Kartu Indonesia Pintar Kuliah pada Media Sosial X Menggunakan Algoritma Naive Bayes.

*MALCOM: Indonesian Journal of Machine Learning and Computer Science*, 4(4), 1420–1430. <https://doi.org/10.57152/malcom.v4i4.1565>

Pramudita, D., Akbar, Y., & Wahyudi, T. (2024b). Analisis Sentimen Terhadap Program Kartu Indonesia Pintar Kuliah pada Media Sosial X Menggunakan Algoritma Naive Bayes. *MALCOM: Indonesian Journal of Machine Learning and Computer Science*, 4(4), 1420–1430. <https://doi.org/10.57152/malcom.v4i4.1565>

Qaiser, S., & Ali, R. (2018). Text Mining: Use of TF-IDF to Examine the Relevance of Words to Documents. *International Journal of Computer Applications*, 181(1), 25–29. <https://doi.org/10.5120/ijca2018917395>

Qeis, M. I. (2015). *APLIKASI WORDCLOUD SEBAGAI ALAT BANTU ANALISIS WACANA*. <https://www.researchgate.net/publication/316736417>

Sandar Kyaw, K., Tepsongkroh, P., Thongkamkaew, C., & Sasha, F. (n.d.). *Asia Social Issues Business Intelligent Framework Using Sentiment Analysis for Smart Digital Marketing in the E-Commerce Era*. <https://so06.tci-thaijo.org/index.php/asi>

Sandar Kyaw, K., Tepsongkroh, P., Thongkamkaew, C., & Sasha, F. (2022). *Asia Social Issues Business Intelligent Framework Using Sentiment Analysis for Smart Digital Marketing in the E-Commerce Era*. <https://so06.tci-thaijo.org/index.php/asi>

Sofiyan, U., & Abbas, Q. (2024). *NLP-Driven Sentiment Analysis: Exploring Causal Reasoning in E-Commerce Feedback*. <https://doi.org/10.13140/RG.2.2.29158.02883>

Yusliani, N., Primartha, R., & Diana, M. (2019). Multiprocessing Stemming: A Case Study of Indonesian Stemming. *International Journal of Computer Applications*, 182(40), 15–19. <https://doi.org/10.5120/ijca2019918476>

## DAFTAR RIWAYAT HIDUP



### DATA PRIBADI

NIM : 20210700009  
Nama Lengkap : Amos Dwi Prasetyo  
Tempat, Tanggal Lahir : Tangerang, 09 September 2003  
Agama : Kristen Protestan  
Alamat : Jl. Imam Bonjol, Gg. Bansin RT 001/007, Kel. Sukajadi,  
Tangerang  
No. Telephone : 0895-6107-56021

### DATA PENDIDIKAN

1. Tahun 2009 - 2015 : SD Shalom BK-3  
2. Tahun 2015 - 2018 : SMP Strada St. Maria 2  
3. Tahun 2018 - 2021 : SMA Strada St. Thomas Aquino  
4. Tahun 2021 - Sekarang : Universitas Buddhi Dharma  
Fakultas : Sains Dan Teknologi,  
Program Studi : Sistem Informasi

### PENGALAMAN PEKERJAAN

1. 2023 - 2024 : PT. Lippo Karawaci Tbk  
2. 2025 - Sekarang : PT. Data Lake Indonesia

Tangerang, 06 Agustus 2025

Amos Dwi Prasetyo

## LAMPIRAN

### Lampiran 1. Requirement Elicitation 1

No	Analisis Kebutuhan Sistem	Keterangan
	Saya Ingin Sistem dapat :	
1	Tampilannya user friendly (mudah dimengerti)	
2	Menu pilihan mudah di klik	
3	Hasil output dalam bentuk yang diinginkan user (excel / word)	
4		
5		
6		
7		
8		
9		
10		
11		

Tangerang, 10 Mei 2025

Pembimbing



Andi Leo, S.Kom., M.Kom.  
NIDN: 0405106204

Responden



Caroline F.

Mahasiswa



Amos Dwi Prasetyo  
NIM : 20210700009

Lampiran 2. Requirement Elicitation 2

No	Analisis Kebutuhan Sistem	Keterangan
	Saya Ingin Sistem dapat :	
1	memahami perasaan (positif, negatif & netral) dan review	
2	Membedakan negasi ganda, memahami sehingga hasilnya ok	
3	mengategorikan Ulasan-ulasan dan menampilkan —	
4	kedalam Report visual.	
5	memahami campuran bahasa.	
6		
7		
8		
9		
10		
11		

Tangerang, 10 Juni 2025

Pembimbing



Andi Leo, S.Kom., M.Kom.  
NIDN: 0405106204

Responden



Mahasiswa



Amos Dwi Prasetyo  
NIM : 20210700009

Lampiran 3. Requirement Elicitation 3

No	Analisis Kebutuhan Sistem	Keterangan
	Saya Ingin Sistem dapat :	
1	Tampilan Dashboard interaktif	
2	<del>log</del> insight Stakeholder lebih detail	
3	Proses loading cepat	
4		
5		
6		
7		
8		
9		
10		
11		

Tangerang, 3 Juni 2025

Pembimbing



Andi Leo, S.Kom., M.Kom.  
NIDN: 0405106204

Responden

  

**PT. FIRST MEDIA Tbk**

Aina Yoga

Mahasiswa



Amos Dwi Prasetyo  
NIM : 20210700009

Lampiran 4. Requirement Elicitation 4

No	Analisis Kebutuhan Sistem	Keterangan
	Saya Ingin Sistem dapat :	
1	Dapat login dengan mudah	
2	Tampilan admin user-Friendly	
3	Tampilan dashboard stakeholder interaktif	
4		
5		
6		
7		
8		
9		
10		
11		

Tangerang, 7 Mei 2025

Pembimbing

Responden

Mahasiswa

  
 Andi Leo, S.Kom., M.Kom.  
 NIDN: 0405106204

**PT. PRIMA WIRA UTAMA**  
  
 Figo Yuda

  
 Amos Dwi Prasetyo  
 NIM : 20210700009



Lampiran 5. Requirement Elicitation 5

No	Analisis Kebutuhan Sistem	Keterangan
	Saya Ingin Sistem dapat :	
1	Tampilan UI Menarik	
2	User Friendly	
3	Output mudah dipahami	
4	Menu Lengkap	
5	Ada fitur membership	
6		
7		
8		
9		
10		
11		

Tangerang, 22 Mei 2025

Pembimbing



Andi Leo, S.Kom., M.Kom.  
NIDN: 0405106204

Responden



Heri Pramono

Mahasiswa



Amos Dwi Prasetyo  
NIM : 20210700009

## Lampiran 6. KARTU BIMBINGAN TA/SKRIPSI



**UNIVERSITAS BUDDHI DHARMA**

Jl. Imam Bonjol No. 41 Karawaci Ilir, Tangerang

021 5517853 / 021 5586822 admin@buddhidharma.ac.id

### KARTU BIMBINGAN TA/SKRIPSI

NIM : 20210700009  
Nama Mahasiswa : AMOS DWI PRASETYO  
Fakultas : Sains dan Teknologi  
Program Studi : Sistem Informasi  
Jenjang : Strata Satu  
Tahun Akademik/Semester : 2024/2025 Genap  
Dosen Pembimbing : Andi Leo, S.Kom.,M.Kom.  
Judul Skripsi : IMPLEMENTASI APLIKASI WEB SENTIMENT ANALYSIS  
BERBASIS FLASK PADA ULASAN PELANGGAN TOKOPEDIA  
UNTUK OPTIMALISASI STRATEGI E-BUSINESS

Tanggal	Catatan	Paraf
2025-02-28	DISKUSI PENGAJUAN 3 ALTERNATIF JUDUL	
2025-03-07	PENGAJUAN DAN PENGESAHAN BAB 1	
2025-03-14	PENULISAN DAN PENGESAHAN BAB 2	
2025-03-21	PENGARAHAN DAN KERANGKA PENULISAN BAB 3	
2025-04-11	- FINALISASI BAB 3 DAN KERANGKA PENULISAN BAB 4 - SIAPKAN RE DAN DESAIN QUESTIONARE	
2025-04-25	- REVISI QUESTIONARE - PENDALAMAN BAB 4	
2025-05-02	- PENDALAMAN BAB 4 - EVALUASI PROGRESS RE DAN QUESTIONARE	
2025-05-24	- PENDALAMAN DESAIN OUTPUT UNTUK MASING2 TINGKATAN MANAJEMEN - PENULISAN BAB 5	
2025-05-30	EVALUASI AKHIR TENTANG KESIAPAN DAN KESANGGUPAN MHS UNTUK MAJU SIDANG PERIODE SEMESTER GENAP 2024/2025	

Mengetahui  
Ketua Program Studi



Benny Daniawan, M.Kom

Tangerang, 23 June 2025

Pembimbing



Andi Leo, S.Kom.,M.Kom.

Lampiran 7. Listing Coding Scraping Tokopedia

```
for i in range(pages):
    progress = 20 + (i * 60 / pages)
    scraping_status["progress"] = int(progress)
    scraping_status["message"] = f"Scraping page {i+1} of {pages}..."

    self.log_debug(f"Processing page {i+1}")

    # Use improved parsing method with BeautifulSoup
    soup = BeautifulSoup(driver.page_source, "html.parser")

    containers = soup.findAll('article', attrs={'class': 'css-ccpe8t'})

    if not containers:
        self.log_debug("Primary selector not found, trying alternatives...")
        # Try alternative selectors
        containers = soup.findAll('article', attrs={'class': re.compile(r'css-\w+')})

    self.log_debug(f"Found {len(containers)} containers")

    page_reviews = []
    for container in containers:
        try:

            review_element = container.find('span', attrs={'data-testid': 'lblItemUlasan'})
            review = self.clean_text(review_element.text) if review_element else None

            if not review or len(review) < 10:
                continue

            # Extract product name using improved selector
            nama_produk_element = container.find('a', attrs={'class': 'styProduct'})
            if not nama_produk_element:
```

```

        # Try alternative selectors
        nama_produk_element = container.find('a', attrs={'href':
re.compile(r'/product/')}))

        nama_produk = self.clean_text(nama_produk_element.text) if
nama_produk_element else "Tidak tersedia"

        # Extract rating using improved selector
        rating_element = container.find('div', attrs={'data-testid': 'icnStarRating'})
        if rating_element:
            rating = self.clean_text(
                rating_element.get('aria-label', "").replace('Rating ', "").replace('bintang', ")
            )
        else:
            rating_element = container.find('div', attrs={'aria-label': re.compile(r'Rating \d+
bintang')})
            if rating_element:
                rating_text = rating_element.get('aria-label', "")
                numbers = re.findall(r'\d+', rating_text)
                rating = numbers[0] if numbers else "5"
            else:
                rating = "5" # Default

        review_data = {
            'Ulasan': review,
            'Nama_Produk': nama_produk,
            'Rating': rating
        }
        page_reviews.append(review_data)

    except Exception as e:
        self.log_debug(f"Error processing container: {e}")
        continue

```

Lampiran 8. Listing Coding Sentiment Analysis Tokopedia

```

def basic_analyze_sentiment(self, csv_path, progress_callback=None):
    """Fallback basic sentiment analysis"""
    global analysis_status

    try:
        if progress_callback:
            progress_callback(10, "Loading data...")

        analysis_status["message"] = "Loading data..."
        df = pd.read_csv(csv_path)
        print(f'✅ Data loaded: {len(df)} rows')

        if progress_callback:
            progress_callback(20, "Preprocessing...")

        analysis_status["message"] = "Preprocessing text..."
        df['Ulasan_Clean'] = df['Ulasan'].apply(self.normalize_text)

        # Create labels from ratings
        df['Label'] = df['Rating'].apply(lambda x: 1 if int(float(str(x).replace('bintang',
        ").strip())) > 3 else 0)

        if progress_callback:
            progress_callback(40, "Training model...")

        analysis_status["message"] = "Training model..."

        # Vectorize
        self.vectorizer = TfidfVectorizer(max_features=1000, decode_error='replace')
        X = self.vectorizer.fit_transform(df['Ulasan_Clean'])
        y = df['Label']

        # Train model
        X_train, X_test, y_train, y_test = train_test_split(

```

```
X.toarray(), y, test_size=0.2, random_state=42, stratify=y if len(set(y)) > 1 else
None
)
```

```
self.model = GaussianNB()
self.model.fit(X_train, y_train)
```

```
if progress_callback:
```

```
    progress_callback(60, "Making predictions...")
```

```
analysis_status["message"] = "Making predictions..."
```

```
# Predict
```

```
predictions = self.model.predict(X.toarray())
```

```
# Enhanced predictions using rule-based approach
```

```
enhanced_predictions = []
```

```
for i, review in enumerate(df['Ulasan']):
```

```
    ml_pred = predictions[i]
```

```
    rule_pred = self.rule_based_prediction(review)
```

```
# Combine ML and rule-based
```

```
if rule_pred is not None:
```

```
    enhanced_predictions.append(rule_pred)
```

```
else:
```

```
    enhanced_predictions.append(ml_pred)
```

```
df['Predicted_Sentiment'] = enhanced_predictions
```

```
df['Sentiment'] = df['Predicted_Sentiment'].map({0: 'Negatif', 1: 'Positif'})
```

```
if progress_callback:
```

```
    progress_callback(80, "Calculating metrics...")
```

```
analysis_status["message"] = "Calculating metrics..."
```

```
# Calculate metrics
```

```
if len(X_test) > 0:
    y_pred = self.model.predict(X_test)
    accuracy = accuracy_score(y_test, y_pred)
    precision = precision_score(y_test, y_pred, average='weighted', zero_division=0)
    recall = recall_score(y_test, y_pred, average='weighted', zero_division=0)
    f1 = f1_score(y_test, y_pred, average='weighted', zero_division=0)
else:
    accuracy = precision = recall = f1 = 0.8 # Default values

positive_count = sum(enhanced_predictions)
negative_count = len(enhanced_predictions) - positive_count

metrics = {
    'accuracy': float(accuracy),
    'precision': float(precision),
    'recall': float(recall),
    'f1_score': float(f1),
    'total_reviews': int(len(df)),
    'positive_reviews': int(positive_count),
    'negative_reviews': int(negative_count),
    'method_used': 'basic_fallback'
}
```

Lampiran 9. Listing Coding Visualisasi & Export PDF Tokopedia

```
def _generate_wordclouds(self, df, base_filename):
    """Generate wordclouds for positive and negative sentiments"""
    try:
        # Create plots directory if it doesn't exist
        plots_dir = os.path.join(BASE_DIR, 'static', 'plots')
        os.makedirs(plots_dir, exist_ok=True)

        # Indonesian stopwords
        stopwords = {'yang', 'dan', 'di', 'dengan', 'untuk', 'tidak', 'ini', 'dari', 'dalam', 'akan',
                    'pada', 'juga',
                    'saya', 'ke', 'bisa', 'itu', 'ada', 'sih', 'aja', 'nya', 'yg', 'ya', 'ga', 'gak', 'atau', 'tapi',
                    'seperti', 'oleh', 'karena', 'adalah', 'kalau', 'dapat', 'si', 'nih', 'kok', 'banget'}

        # Function to clean text and create wordcloud
        def create_wordcloud_from_text(text, output_path, title, colormap):
            # Clean and filter text
            words = '!'.join([w for w in text.lower().split() if len(w) > 2 and w not in
                              stopwords])

            # Generate wordcloud
            if not words:
                # If no words, create empty wordcloud
                wordcloud = WordCloud(width=800, height=400, background_color='white',
                                       colormap=colormap, max_words=100).generate('tidak ada kata')
            else:
                wordcloud = WordCloud(width=800, height=400, background_color='white',
                                       colormap=colormap, max_words=100).generate(words)

            # Plot and save wordcloud
            plt.figure(figsize=(10, 5))
            plt.imshow(wordcloud, interpolation='bilinear')
            plt.axis('off')
```

```

plt.title(title)
plt.tight_layout(pad=0)

# Save wordcloud image with multiple name formats for compatibility
plt.savefig(output_path)
print(f"✅ Wordcloud saved to {output_path}")

# Clear figure to avoid memory issues
plt.close()

# Generate business reports with charts (heatmap, TF-IDF, pie charts, etc.)
try:
    print(f"📊 Generating business intelligence reports...")
    business_report_folder = os.path.join(BASE_DIR, 'static', 'business_reports')
    report_generator = BusinessReportGenerator()

    # Generate comprehensive business report
    report_data = report_generator.generate_comprehensive_report(
        df,
        business_report_folder,
        result_filename.replace('.json', '')
    )
    print(f"✅ Business reports generated successfully")

    # Generate business summary table for dashboard
    print(f"📄 Generating business summary table...")
    business_summary_table = generate_business_summary_table(df)
    results['business_summary_table'] = business_summary_table
    print(f"✅ Business summary table generated with
{len(business_summary_table)} products")

# Generate dynamic business insights
print(f"🧠 Generating dynamic business insights...")

```

```

dynamic_insights =
generate_dynamic_business_insights(business_summary_table, df)
results['dynamic_business_insights'] = dynamic_insights
print(f"✅ Dynamic business insights generated")

# Add chart paths to results for dashboard
if 'chart_paths' not in results:
    results['chart_paths'] = {}

# Update results with chart information (match what
BusinessReportGenerator actually creates)
base_chart_name = result_filename.replace('.json', "")
results['chart_paths'].update({
    'tfidf_analysis': f'business_reports/{base_chart_name}_tfidf_analysis.png',
    'sentiment_heatmap':
f'business_reports/{base_chart_name}_sentiment_heatmap.png',
    'sentiment_overall':
f'business_reports/{base_chart_name}_sentiment_overall.png',
    'sentiment_by_rating':
f'business_reports/{base_chart_name}_sentiment_by_rating.png',
    'confidence_distribution':
f'business_reports/{base_chart_name}_confidence_distribution.png',
    'top_products': f'business_reports/{base_chart_name}_top_products.png'
})
@app.route('/dashboard/<filename>/export-pdf)
@login_required(role='boss')
def export_dashboard_pdf(filename):
    """Export dashboard to professional PDF using WeasyPrint"""

    # Check WeasyPrint availability (Line 2634-2637)
    if not WEASYPRINT_AVAILABLE:
        flash('PDF export feature not available. Please install WeasyPrint')

```

```
# Load JSON data (Line 2645-2646)
```

```
with open(result_path, 'r', encoding='utf-8') as f:
```

```
    results = json.load(f)
```

```
# Prepare data for PDF template (Line 2649)
```

```
template_data = prepare_pdf_data(results, filename)
```

```
# Render PDF template (Line 2652)
```

```
html_content = render_template('dashboard_pdf_template.html', **template_data)
```

```
# Generate PDF with WeasyPrint (Line 2655-2659)
```

```
pdf = render_pdf(  
    HTML(string=html_content, base_url=request.url_root),  
    stylesheets=[],  
    presentational_hints=True  
)
```

```
# Create response (Line 2662-2667)
```

```
response = make_response(pdf)
```

```
response.headers['Content-Type'] = 'application/pdf'
```

```
response.headers['Content-Disposition'] = f'attachment; filename="{clean_filename}"'
```

The image features a large, semi-transparent watermark of the Universitas Buddhi Dharma (UBD) logo. The logo is circular with a yellow center containing a blue flame-like symbol. The text 'UNIVERSITAS BUDDHI DHARMA' is written in a circular path around the center, and 'UBD' is written at the bottom. The entire logo is set against a light pink background.