

DAFTAR PUSTAKA

- [1] H. Gunawan and A. K. H. Saputro, "Pemanfaatan Aplikasi Mobile Untuk Mempercepat Pencarian Tempat Indekos Berbasis Android," *Jurnal Muara Sains, Teknologi, Kedokteran dan Ilmu Kesehatan*, vol. 1, no. 2, pp. 85–96, 2018, doi: 10.24912/jmstkik.v1i2.1454.
- [2] "Data DataReportal Keadaan Digital Indonesia 2025." Accessed: Feb. 25, 2025. [Online]. Available: <https://datareportal.com/reports/digital-2025-indonesia>
- [3] A. C. Saputra and A. S. Saragih, "KLASIFIKASI RATING APLIKASI ANDROID DI GOOGLE PLAY STORE MENGGUNAKAN ALGORITMA GRADIENT BOOST Agus Sehatman Saragih," *Oktober*, vol. 6, no. 1, pp. 18–29, 2022, [Online]. Available: <https://e-journal.upr.ac.id/index.php/JT/article/view/7891>
- [4] C. M. Dachi and S. S. Berutu, "Klasifikasi Sentimen Opini Pada Aplikasi Guitar Tuner," pp. 1444–1459, [Online]. Available: <https://ojs.stmik-banjarbaru.ac.id/index.php/jutisi/article/view/2041>
- [5] A. Ariq and F. G. Tuner, "ANALISIS SENTIMEN POSITIF APLIKASI GUITARTUNA DAN FENDER GUITAR TUNER DENGAN MENGGUNAKAN METODE ALGORITMA NAÏVE BAYES DAN SUPPORT VECTOR MACHINE," 2024. [Online]. Available: <https://repository.upnvj.ac.id/28184/2/AWAL.pdf>
- [6] C. Wulandari, L. Sunardi, and Hasbiana, "KLIK: Kajian Ilmiah Informatika dan Komputer Analisis Sentimen Aplikasi Spotify Pada Ulasan Pengguna di Google Play Store Menggunakan Metode Support Vector Machine," *Media Online*, vol. 4, no. 5, pp. 2588–2595, 2024, doi: 10.30865/klik.v4i5.1762.
- [7] W. Pondra Lesmana and A. Wijaya, "Analisis Sentimen Menggunakan Metode Naive Bayes dan Support Vector Machine pada Ulasan Aplikasi Joox Music," *Jurnal Ilmiah Sain dan Teknologi*, vol. 1, no. 2, pp. 110–120, 2023, doi: 10.572349/scientica.v1i2.237.
- [8] I. R. Ainunnisa and S. Sulastri, "Analisis Sentimen Aplikasi Tiktok dengan Metode Support Vector Machine (SVM), Logistic Regression dan Naïve Bayes," *Jurnal Teknologi Sistem Informasi dan Aplikasi*, vol. 6, no. 3, pp. 423–430, 2023, doi: 10.32493/jtsi.v6i3.31076.
- [9] S. Sinaulan, M. Kaunang, and S. Sunarmi, "Pengaruh Musik Terhadap Kesenangan 'Tulung Kerja' Pada Acara Pesta Pernikahan Di Desa Sion Kecamatan Tomposo Baru Kabupaten Minahasa Selatan," *Kompetensi*, vol. 3, no. 6, pp. 2366–2374, 2023, doi: 10.53682/kompetensi.v3i6.6273.
- [10] C. R. Pramudhita and U. Utomo, "Hubungan Kebiasaan Mendengarkan Musik, Pemanfaatannya Ketika Belajar Matematika Terhadap Hasil Belajar Mata Pelajaran Matematika Siswa Kelas Xi Di Smk Pika Semarang," *Jurnal Seni Musik*, vol. 8, no. 2, p. 153, 2019, [Online]. Available: <https://ejournal.upi.edu/index.php/Interlude/article/view/46582>
- [11] T. Walasary, "Survey Paper tentang Analisis Sentimen," *KONSTELASI: Konvergensi Teknologi dan Sistem Informasi*, vol. 2, no. 1, pp. 201–206, 2022, doi: 10.24002/konstelasi.v2i1.5378.

- [12] M. R. Fahlevvi, "Analisis Sentimen Terhadap Ulasan Aplikasi Pejabat Pengelola Informasi Dan Dokumentasi Kementerian Dalam Negeri Republik Indonesia Di Google Playstore Menggunakan Metode Support Vector Machine," *Jurnal Teknologi dan Komunikasi Pemerintahan*, vol. 4, no. 1, pp. 1–13, 2022, doi: 10.33701/jtkp.v4i1.2701.
- [13] R. N. Handayani, "Optimasi Algoritma Support Vector Machine untuk Analisis Sentimen pada Ulasan Produk Tokopedia Menggunakan PSO," *Media Informatika*, vol. 20, no. 2, pp. 97–108, 2021, doi: 10.37595/mediainfo.v20i2.59.
- [14] F. Alghifari and D. Juardi, "Penerapan Data Mining Pada Penjualan Makanan Dan Minuman Menggunakan Metode Algoritma Naïve Bayes," *Jurnal Ilmiah Informatika*, vol. 9, no. 02, pp. 75–81, 2021, doi: 10.33884/jif.v9i02.3755.
- [15] R. S. Putra and R. Dijaya, "Analisis Sentimen Pada Ulasan Aplikasi Ferizy Menggunakan Metode Support Vector Machine," *Archive.Umsida.Ac.Id*, pp. 1–10, 2023, [Online]. Available: <https://archive.umsida.ac.id/index.php/archive/preprint/view/3441>
- [16] R. R. Salam, M. F. Jamil, Y. Ibrahim, R. Rahmadden, S. Soni, and H. Herianto, "Analisis Sentimen Terhadap Bantuan Langsung Tunai (BLT) Bahan Bakar Minyak (BBM) Menggunakan Support Vector Machine," *MALCOM: Indonesian Journal of Machine Learning and Computer Science*, vol. 3, no. 1, pp. 27–35, 2023, doi: 10.57152/malcom.v3i1.590.
- [17] A. A. Syam, G. H. M, A. Salim, D. F. Surianto, and M. F. B, "Analisis teknik preprocessing pada sentimen masyarakat terkait konflik israel-palestina menggunakan support vector machine," vol. 9, no. 3, pp. 1464–1472, 2024, [Online]. Available: <https://jurnal.stkipgritlungagung.ac.id/index.php/jipi/article/view/5527>
- [18] S. Shevira, I. M. A. D. Suarjaya, and P. W. Buana, "Pengaruh Kombinasi dan Urutan Pre-Processing pada Tweets Bahasa Indonesia," *JITTER : Jurnal Ilmiah Teknologi dan Komputer*, vol. 3, no. 2, p. 1074, 2022, doi: 10.24843/jtrti.2022.v03.i02.p06.
- [19] M. Q. Alqowiy and E. Utami, "Optimasi Parameter Support Vector Machine menggunakan Algoritma Genetika untuk Deteksi Cyberbullying Parameter Optimisation of Support Vector Machine using Genetic Algorithm for Cyberbullying Detection," vol. 14, pp. 332–341, 2025, [Online]. Available: <https://sistemasi.ftik.unisi.ac.id/index.php/stmsi/article/download/4870/896>
- [20] D. Darwis, E. S. Pratiwi, and A. F. O. Pasaribu, "Penerapan Algoritma Svm Untuk Analisis Sentimen Pada Data Twitter Komisi Pemberantasan Korupsi Republik Indonesia," *EduTic - Scientific Journal of Informatics Education*, vol. 7, no. 1, pp. 1–11, 2020, doi: 10.21107/edutic.v7i1.8779.
- [21] T. R. Salsabilla and N. Pratiwi, "Jurnal Teknik Informatika dan Komputer Penerapan Support Vector Machine Untuk Analisis Sentimen pada X (Twitter) Mengenai Obat Penyebab Gagal Ginjal Akut pada Anak," pp. 67–74, 2024, [Online]. Available: <https://journal.uhamka.ac.id/index.php/jutikom/article/view/16892>
- [22] A. C. T. Angel, V. H. Pranatawijaya, and W. Widiatry, "Analisis Sentimen dan Emosi dari Ulasan Google Maps Untuk Layanan Rumah Sakit di Palangka Raya Menggunakan Machine Learning," *KONSTELASI: Konvergensi Teknologi dan Sistem Informasi*, vol. 4, no. 1, pp. 35–49, 2024, doi: 10.24002/konstelasi.v4i1.8924.

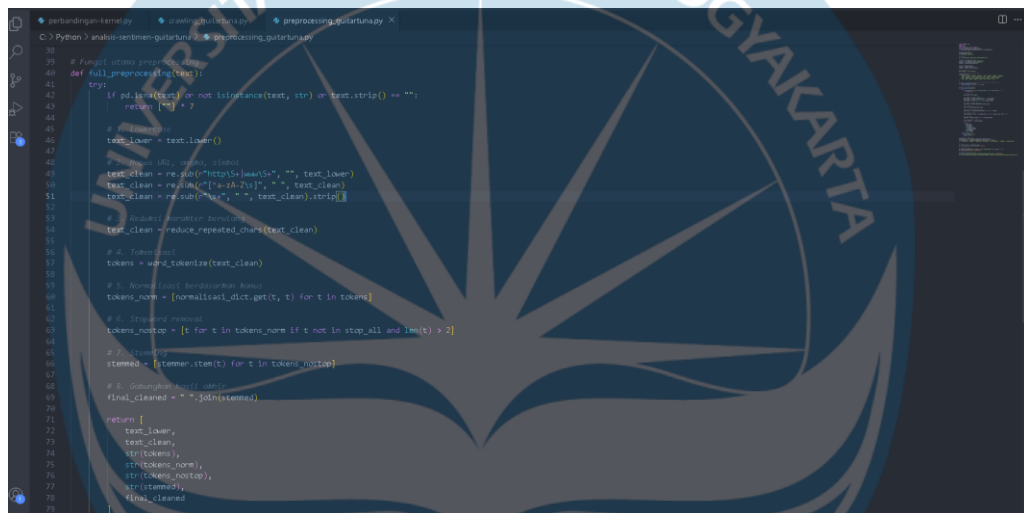
- [23] O. I. Gifari, Muh. Adha, F. Freddy, and F. F. S. Durrand, "Analisis Sentimen Review Film Menggunakan TF-IDF dan Support Vector Machine," *Journal of Information Technology*, vol. 2, no. 1, pp. 36–40, 2022, doi: 10.46229/jifotech.v2i1.330.
- [24] C. Prakoso and A. Hermawan, "Perbandingan Model Machine Learning dalam Analisis Sentimen Ulasan Pengunjung Keraton Yogyakarta pada Google Maps," *Kajian Ilmiah Informatika dan Komputer*, vol. 4, no. 3, pp. 1292–1302, 2023, doi: 10.30865/klik.v4i3.1419.
- [25] G. R. Ditami, E. F. Ripanti, and H. Sujaini, "Implementasi Support Vector Machine untuk Analisis Sentimen Terhadap Pengaruh Program Promosi Event Belanja pada Marketplace," *Jurnal Edukasi dan Penelitian Informatika (JEPIN)*, vol. 8, no. 3, p. 508, 2022, doi: 10.26418/jp.v8i3.56478.
- [26] P. P. O. Mahawardana, I. A. P. F. Imawati, and I. W. Dika, "Analisis Sentimen Berdasarkan Opini dari Media Sosial Twitter terhadap 'Figure Pemimpin' Menggunakan Python," *Jurnal Manajemen dan Teknologi Informasi*, vol. 12, no. 2, pp. 50–56, 2022, [Online]. Available: <https://ojs.mahadewa.ac.id/index.php/jmti/article/view/2111>
- [27] Y. Riadi Silitonga, "Sistem Pendeteksi Berita Hoax di Media Sosial dengan Teknik Data Mining Scikit Learn," *Jurnal Ilmu Komputer*, vol. 4, p. 173, 2019, [Online]. Available: https://digilib.esaunggul.ac.id/UEU-Journal-11_1244/20004/practicum-module
- [28] R. W. Abdullah and R. Kusumastuti, "ANALISIS PENGOLAHAN EKSTRAKSI FITUR CITRA UNTUK KLASIFIKASI JENIS APEL MENGGUNAKAN SCIKIT-LEARN DENGAN ALGORITMA K- IMAGE FEATURE EXTRACTION PROCESSING ANALYSIS FOR APPLE TYPE CLASSIFICATION USING SCIKIT-LEARN WITH THE K-NEAREST NEIGHBOR," vol. 12, no. 1, pp. 165–174, 2025, doi: 10.25126/jtiik.2025129149.
- [29] R. Ramlan, N. Satyahadewi, and W. Andani, "Analisis Sentimen Pengguna Twitter Menggunakan Support Vector Machine Pada Kasus Kenaikan Harga BBM," *Jambura Journal of Mathematics*, vol. 5, no. 2, pp. 431–445, 2023, doi: 10.34312/jjom.v5i2.20860.
- [30] S. Ernawati and R. Wati, "Evaluasi Performa Kernel SVM dalam Analisis Sentimen Review Aplikasi ChatGPT Menggunakan Hyperparameter dan VADER Lexicon," *Jurnal Buana Informatika*, vol. 15, no. 01, pp. 40–49, 2024, doi: 10.24002/jbi.v15i1.7925.
- [31] F. Putrawansyah, "Penerapan Metode Support Vector Machine Terhadap Klasifikasi Jenis Jambu Biji," *JIKO (Jurnal Informatika dan Komputer)*, vol. 8, no. 1, p. 193, 2024, doi: 10.26798/jiko.v8i1.988.

LAMPIRAN

Hasil Pembobotan TF-IDF (Sparse Matrix)

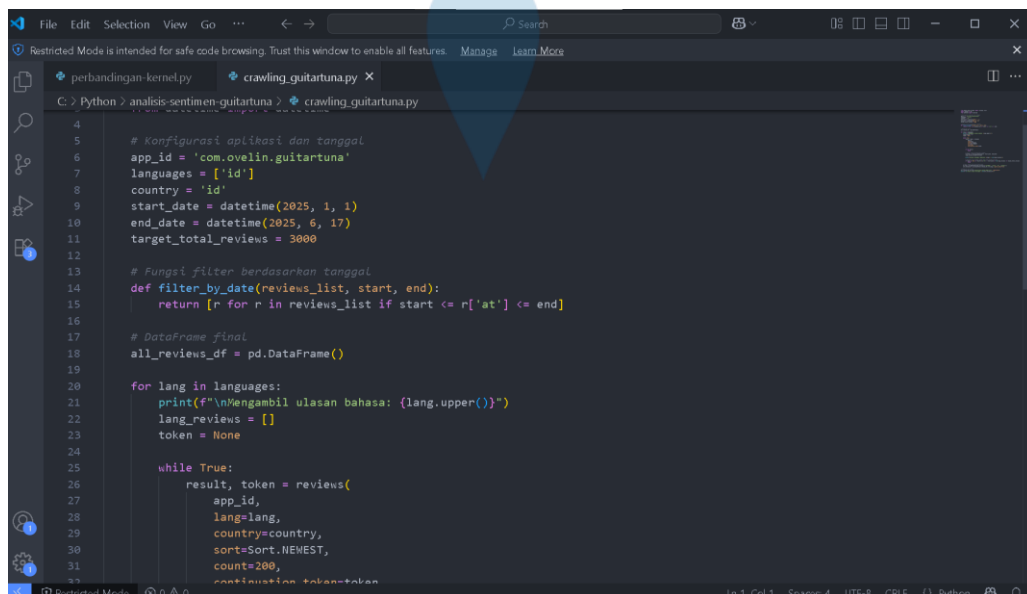
```
(0, 634) 0.2563771600220382
(0, 296) 0.2563771600220382
(0, 654) 0.25979168240309564
(0, 639) 0.6076336816502442
(0, 226) 0.6076336816502442
(0, 322) 0.25022470549603343
(1, 293) 0.7264445230587137
(1, 1) 0.6872251122579838
(2, 364) 0.7263229407992526
(2, 61) 0.6873536103554888
```

Lampiran 1 Hasil TF-IDF Sparse Matrix



```
38
39 # Fungsi untuk preprocessing
40 def full_preprocess(text):
41     try:
42         if pd.isnull(text) or not isinstance(text, str) or text.strip() == "":
43             return None
44
45         # Tokenisasi
46         text_lower = text.lower()
47
48         # Menghapus karakter khusus
49         text_clean = re.sub("[^a-zA-Z]", "", text_lower)
50         text_clean = re.sub("[^a-zA-Z]", "", text_clean)
51         text_clean = re.sub(" ", "", text_clean)
52
53         # Menghapus karakter berulang
54         text_clean = reduce_repeated_char(text_clean)
55
56         # Tokenisasi
57         tokens = word_tokenize(text_clean)
58
59         # Normalisasi berdasarkan dokumen
60         tokens_norm = [normalisasi_dok.get(t, t) for t in tokens]
61
62         # Filter stopwords
63         tokens_stop = [t for t in tokens_norm if t not in stop_all and len(t) > 2]
64
65         # Stemming
66         stemmed = [stemmer.stem(t) for t in tokens_stop]
67
68         # Gabungkan hasil akhir
69         final_cleaned = " ".join(stemmed)
70
71     except:
72         return None
73
74     return [
75         text_lower,
76         text_clean,
77         str(tokens),
78         str(tokens_norm),
79         str(tokens_stop),
80         str(stemmed),
81         final_cleaned
82     ]
```

Lampiran 2 Kode Crawling Data Ulasan



```
4
5 # Konfigurasi aplikasi dan tanggal
6 app_id = 'com.ovelin.guitartuna'
7 languages = ['id']
8 country = 'id'
9 start_date = datetime(2025, 1, 1)
10 end_date = datetime(2025, 6, 17)
11 target_total_reviews = 3000
12
13 # Fungsi filter berdasarkan tanggal
14 def filter_by_date(reviews_list, start, end):
15     return [r for r in reviews_list if start <= r['at'] <= end]
16
17 # DataFrame final
18 all_reviews_df = pd.DataFrame()
19
20 for lang in languages:
21     print(f"\nMengambil ulasan bahasa: {lang.upper()}")
22     lang_reviews = []
23     token = None
24
25     while True:
26         result, token = reviews(
27             app_id,
28             lang=lang,
29             country=country,
30             sort=Sort.NEWEST,
31             count=200,
32             continuation_token=token
33         )
```

Lampiran 3 Kode Pre-processing

```

1 # analisis_sentimen_guitaruna > labeling.py
2
3 # Baca file hasil preprocessing
4 df = pd.read_excel("guitaruna_review_preprocessing.normalized.xlsx")
5
6 # Daftar kata positif dan negatif campuran
7 positive_words = [
8     'bagus', 'baik', 'mudah', 'cepat', 'puas', 'benar', 'suka', 'senang', 'mantap', 'efektif', 'akurat', 'luar biasa', 'keren',
9     'good', 'great', 'excellent', 'easy', 'fast', 'helpful', 'love', 'nice', 'awesome', 'perfect', 'useful', 'simple', 'amazing', 'best'
10 ]
11
12 negative_words = [
13     'buruk', 'jelek', 'susah', 'psejelas', 'lambat', 'tidak', 'gagal', 'hilang', 'masalah', 'parah', 'mengecewakan', 'bohong', 'bayan', 'tidak akurat', 'fals', 'rusak', 'uang', 'berlangganan',
14     'bad', 'poor', 'difficult', 'hard', 'slow', 'error', 'fail', 'problem', 'disappointing', 'worse', 'useless', 'refund', 'regret'
15 ]
16
17 # Fungsi labeling
18 def label_sentiment(text):
19     if not isinstance(text, str) or text.strip() == "":
20         return "neutral"
21
22     tokens = text.split()
23     pos_count = sum(1 for word in tokens if word in positive_words)
24     neg_count = sum(1 for word in tokens if word in negative_words)
25
26     if pos_count > neg_count:
27         return "positive"
28     elif neg_count > pos_count:
29         return "negative"
30     else:
31         return "neutral"
32
33 # Terapkan fungsi ke kolom cleaned_final
34 df['label_sentiment'] = df['cleaned_final'].apply(label_sentiment)
35
36 # Simpan ke file baru
37 df.to_excel("guitaruna_review_labeled.xlsx", index=False)
38 print("Labeling sentiment selesai dan disimpan sebagai 'guitaruna_review_labeled.xlsx'")
39

```

Lampiran 4 Kode Labeling

```

1 # analisis_sentimen_guitaruna > klasifikasi_TF-IDF.py
2
3 import pandas as pd
4 from sklearn.model_selection import train_test_split
5 from sklearn.feature_extraction.text import TfidfVectorizer
6 from sklearn.svm import LinearSVC
7 from sklearn.metrics import classification_report, confusion_matrix, accuracy_score
8 import matplotlib.pyplot as plt
9
10 # 1. Baca file hasil preprocessing dan labeling
11 df = pd.read_excel("guitaruna_review_labeled.xlsx") # Ganti dengan nama file kamu
12
13 # 2. Pastikan kolom yang digunakan ada dan tidak kosong
14 df = df[df['cleaned_final'].notna() & df['label_sentiment'].notna()]
15
16 # 3. Pisahkan fitur (X) dan label (y)
17 X = df['cleaned_final']
18 y = df['label_sentiment']
19
20 # 4. TF-IDF Vectorization
21 tfidf = TfidfVectorizer(max_features=5000) # Bisa sesuaikan jumlah fitur
22 X_tfidf = tfidf.fit_transform(X)
23
24 # 5. Split data 80% (latih) dan 20% uji
25 X_train, X_test, y_train, y_test = train_test_split(
26     X_tfidf, y, test_size=0.2, random_state=42, stratify=y
27 )
28
29 # 6. Latih model SVM (linear kernel)
30 model = LinearSVC()
31 model.fit(X_train, y_train)
32
33 # 7. Prediksi dan evaluasi
34 y_pred = model.predict(X_test)
35
36 # 8. Evaluasi Hasil
37 print("Classification Report:\n")
38 print(classification_report(y_test, y_pred))
39 print("Accuracy:", accuracy_score(y_test, y_pred))
40
41 # 9. Confusion Matrix
42 cm = confusion_matrix(y_test, y_pred, labels=['positive', 'neutral', 'negative'])
43
44 plt.figure(figsize=(8, 4))
45 sns.heatmap(cm, annot=True, fmt='d', cmap='Blues',
46

```

Lampiran 5 Kode Klasifikasi TF-IDF

```

Restricted Mode is intended for safe code browsing. Trust this window to enable all features. Manage Learn More

perbandingan-kernel.py crawling_guitartuna.py preprocessing_guitartuna.py labeling.py import-matplotlib.py hasil_tf-idf.py hasil_wordcloud.py x

C:\Python> analisis-sentimen-guitartuna > hasil_wordcloud.py
1 import pandas as pd
2 from wordcloud import WordCloud
3 import matplotlib.pyplot as plt
4 from collections import Counter
5
6 # Baca file hasil preprocessing dan labeling
7 df = pd.read_excel("guitartuna_reviews_labeled.xlsx")
8
9 # Sentimen yang ingin ditampilkan
10 sentiments = ['positive', 'neutral', 'negative']
11
12 for sentiment in sentiments:
13     # Gabungkan semua teks untuk sentimen tertentu
14     text = " ".join(df[df['label_sentiment'] == sentiment]['cleaned_final'].astype(str))
15
16     if text.strip() == "":
17         print(f"❌ Tidak ada data untuk sentimen: {sentiment}")
18         continue
19
20     # Hitung frekuensi kata
21     words = text.split()
22     word_freq = Counter(words)
23     top_words = word_freq.most_common(5)
24
25     # Cetak Top 5 kata teratas
26     print(f"👉 Top 5 kata pada Sentimen '{sentiment}':")
27     for i, (word, freq) in enumerate(top_words, 1):
28         print(f"{i}. {word} ({freq}x)")
29
30     # Buat dan tampilkan WordCloud
31     wordcloud = WordCloud(width=800, height=400, background_color='white').generate(text)
32
33     plt.figure(figsize=(10, 5))
34     plt.imshow(wordcloud, interpolation='bilinear')
35     plt.title(f"WordCloud Sentimen {sentiment.capitalize()}", fontsize=10)
36     plt.axis('off')

```

Lampiran 6 Kode WordCloud Sentimen

```

perbandingan-kernel.py crawling_guitartuna.py preprocessing_guitartuna.py labeling.py import-matplotlib.py hasil_tf-idf.py hasil_wordcloud.py visualisasi-tfidf.py x

C:\Python> analisis-sentimen-guitartuna > visualisasi-tfidf.py
1 import pandas as pd
2 import matplotlib.pyplot as plt
3
4 # Baca file hasil TF-IDF
5 df_tfidf = pd.read_excel("hasil_tfidf.xlsx")
6
7 # Hitung rata-rata skor TF-IDF setiap kata
8 if 'label_sentiment' in df_tfidf.columns:
9     tfidf_features = df_tfidf.drop(columns=['label_sentiment'])
10 else:
11     tfidf_features = df_tfidf.copy()
12
13 # Hitung rata-rata skor TF-IDF setiap kata
14 mean_tfidf = tfidf_features.mean().sort_values(ascending=False)
15
16 # Ambil 20 kata dengan skor TF-IDF tertinggi
17 top_20_words = mean_tfidf.head(20)
18
19 # Visualisasi
20 plt.figure(figsize=(12, 6))
21 bars = plt.bar(top_20_words.index, top_20_words.values, color='skyblue')
22
23 # Tambahkan label nilai di atas setiap bar
24 for bar in bars:
25     yval = bar.get_height()
26     plt.text(bar.get_x() + bar.get_width()/2, yval + 0.002, f"{yval:.4f}", ha='center', va='bottom', fontsize=8)
27
28 # Tambahkan judul dan sumbu grafik
29 plt.title("20 Kata Utama dengan Rata-rata Skor TF-IDF Tertinggi", fontsize=12, fontweight='bold')
30 plt.xlabel("Kata-rata Skor TF-IDF")
31 plt.ylabel("Rata-rata Skor TF-IDF")
32 plt.xticks(rotation=45, ha='right')
33 plt.grid(axis='y', linestyle='dashed', alpha=0.7)
34 plt.tight_layout()
35
36 # Simpan sebagai gambar (untuk dimasukkan ke skripsi)
37 plt.savefig("20_kata_tertinggi_tfidf.png", dpi=300)
38 plt.show()
39

```

Lampiran 7 Kode 20 kata rata-rata TF-IDF tertinggi

```

24 )
25
26 # 5. Daftar kernel
27 kernels = ['linear', 'rbf', 'poly', 'sigmoid']
28 results = []
29
30 # 6. Loop klasifikasi & evaluasi
31 for kernel in kernels:
32     print(f"[*] Evaluasi kernel: {kernel}")
33     model = SVC(kernel=kernel)
34     model.fit(X_train, y_train)
35     y_pred = model.predict(X_test)
36
37     acc = accuracy_score(y_test, y_pred)
38     print(f"  Akurasi: {acc:4f}")
39     print(classification_report(y_test, y_pred))
40
41     # Confusion matrix
42     cm = confusion_matrix(y_test, y_pred, labels=['positive', 'neutral', 'negative'])
43     results.append({'kernel': kernel, 'accuracy': acc, 'conf_matrix': cm})
44
45 # 7. Visualisasi perbandingan akurasi
46 plt.figure(figsize=(8, 5))
47 kernel_names = [r[kernel] for r in results]
48 accuracies = [r['accuracy'] for r in results]
49
50 sns.barplot(x=kernel_names, y=accuracies, palette='viridis')
51 plt.ylim(0, 1.05)
52 plt.title("Perbandingan Akurasi Setiap Kernel SVM")
53 plt.xlabel("Jenis Kernel")
54 plt.ylabel("Akurasi")
55 plt.tight_layout()
56 plt.savefig("perbandingan_akurasi_kernel_svm.png")
57 plt.show()
58
59 # 8. Visualisasi Confusion Matrix untuk tiap kernel
60 for res in results:
61     plt.figure(figsize=(5, 4))
62     sns.heatmap(res['conf_matrix'], annot=True, fmt='d', cmap='Blues',
63                 xticklabels=['positive', 'neutral', 'negative'],
64                 yticklabels=['positive', 'neutral', 'negative'])
65     plt.title(f"Confusion Matrix (Kernel: {res['kernel']})")
66     plt.xlabel("Prediksi")
67     plt.ylabel("Actual")

```

Lampiran 8 Kode Akurasi Kernel SVM

```

1 import pandas as pd
2 import matplotlib.pyplot as plt
3
4 # Baca file hasil labeling
5 df = pd.read_excel("guitartuna_reviews_labeled.xlsx")
6
7 urutian_label_sentimen_secara_eksplisit
8 label_order = ['positive', 'neutral', 'negative']
9 sentiment_counts = df['label_sentiment'].value_counts().reindex(label_order)
10
11 # Tampilkan di terminal
12 print("Distribusi Sentimen:")
13 print(sentiment_counts)
14
15 # Simpan ke excel
16 sentiment_counts.to_frame(name='Jumlah').to_excel("distribusi_sentimen.xlsx")
17
18 # Plot distribusi dengan warna sesuai label
19 colors = ['green', 'gray', 'red'] # sesuai urutian: positive, neutral, negative
20
21 plt.figure(figsize=(6, 4))
22 sentiment_counts.plot(kind='bar', color=colors)
23 plt.title("Distribusi Sentimen Ulasan")
24 plt.xlabel("Sentimen")
25 plt.ylabel("Jumlah Ulasan")
26 plt.xticks(rotation=0)
27 plt.tight_layout()
28 plt.savefig("distribusi_sentimen.png") # Simpan ke file
29 plt.show()
30

```

Lampiran 9 Kode Distribusi Sentimen GitarTuna