



ANALISIS SENTIMEN ULASAN APLIKASI SHAZAM DI GOOGLE PLAY STORE MENGGUNAKAN SUPPORT VECTOR MACHINE

Andri Wijaya^{1*}, Meilinda², Mutia Maharani³

^{1,2,3} Program Studi Sistem Informasi Fakultas Sains dan Teknologi Universitas Katolik Misi Charitas
Jl. Bangau No 60, 9 Ilir, Kec. Ilir Tim. II, Kota Palembang, Sumatera Selatan, 30114,
telp. 0711 378171

e-mail: ¹andri_wijaya@ukmc.ac.id, ²meilindachen05@gmail.com, ³mutiamhrni29@gmail.com

Abstrak

Shazam adalah aplikasi pengenalan musik yang populer di Google Play Store. Shazam memungkinkan pengguna untuk menemukan lagu baru dan mengidentifikasi lagu yang sedang diputarkan di sekitar mereka. Selain itu, penyediaan lirik lagu, video musik, dan rekomendasi lagu. Ulasan pengguna memberikan gambaran bagaimana pengguna merasakan aplikasi tersebut, baik itu positif atau negatif. Dalam penelitian ini, dilakukannya analisis sentimen pada ulasan pengguna Shazam di Google Play Store. Para peneliti menggunakan model Support Vector Machine (SVM) untuk mengklasifikasikan ulasan pengguna ke dalam dua sentimen, yaitu positif dan negatif. Hasil menunjukkan bahwa SVM memiliki akurasi 84% dalam memprediksi sentimen ulasan Shazam. Kemudian, dapat ditunjukkan bahwa Shazam mendapatkan lima kali lebih banyak respon positif daripada respon negatif. Maka dari itu, SVM lebih baik dalam memprediksi ulasan positif daripada ulasan negatif. Penelitian ini dapat membantu pihak-pihak yang berkepentingan untuk memahami bagaimana pengguna merasakan aplikasi tersebut secara keseluruhan.

Kata kunci: analisis sentimen, Shazam, ulasan pengguna, Support Vector Machine.

Abstract

Shazam is a popular music recognition app on the Google Play Store. Shazam allows users to discover new songs and identify songs that are playing around them. In addition, it provides song lyrics, music videos, and song recommendations. User reviews give an idea of how users perceive the application, be it positive or negative. In this study, sentiment analysis was conducted on Shazam user reviews on the Google Play Store. The researchers used a Support Vector Machine (SVM) model to classify user reviews into two sentiments, namely positive and negative. Results show that SVM has 84% accuracy in predicting the sentiment of Shazam reviews. Furthermore, it can be shown that Shazam gets five times more positive responses than negative responses. Therefore, SVM is better at predicting positive reviews than negative reviews. This research can help interested parties to understand how users perceive the app as a whole.

Keywords: sentiment analysis, Shazam, user reviews, Support Vector Machine.

1. PENDAHULUAN

Era sekarang yang semakin maju dan berkembang akan teknologi sangat membantu aktivitas manusia dari segala aspek. Salah satu aspek yang telah mengalami transformasi signifikan adalah cara untuk menikmati dan berinteraksi dengan musik. Sekarang, tidak hanya mendengarkan musik, tetapi juga berinteraksi dengan musik dalam cara yang belum pernah dibayangkan sebelumnya. Aplikasi musik seperti Shazam memungkinkan mendapati interaksi baru tersebut. Dengan kemampuannya untuk mengidentifikasi lagu hanya dalam beberapa detik, Shazam telah membuka dunia baru bagi pecinta musik di seluruh dunia. Tidak hanya itu, Shazam juga menawarkan fitur lain seperti lirik lagu, video musik, dan rekomendasi lagu, menjadikannya lebih dari sekadar alat pengenalan lagu. Shazam telah mendapatkan banyak ulasan dari pengguna di Google Play Store. Ulasan ini memberikan gambaran tentang bagaimana pengguna merasakan aplikasi tersebut, baik itu positif atau negatif.

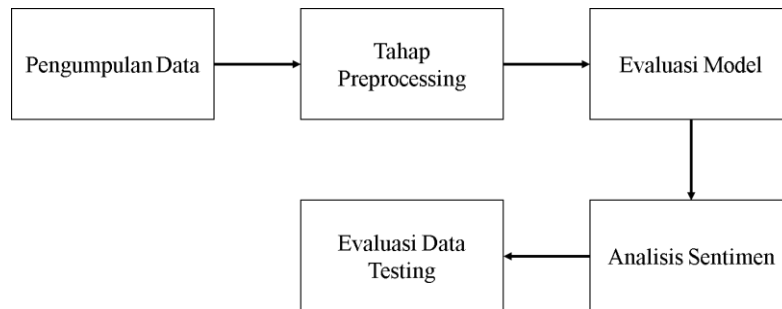
Untuk memfasilitasi pihak-pihak yang berkepentingan dalam pengumpulan data mengenai kepuasan dari pengguna suatu aplikasi tertentu, analisis sentimen diperlukan dalam mengklasifikasikan data ulasan pengguna yang didasarkan dari nilai rating dan komentar berupa teks di situs layanan Google Play Store. Analisis sentimen, yang istilah lain juga dapat dikenal sebagai *opinion mining*, merupakan suatu spesialisasi yang mempertimbangkan pendapat, perasaan, evaluasi, penilaian, sikap, dan emosi seseorang terhadap entitas seperti produk, layanan, organisasi, individu, masalah, peristiwa, topik, dan atribut mereka [1]. Analisis sentimen membantu organisasi memahami sikap dan respons individu atau kelompok pada topik yang dibahas dalam dokumen secara keseluruhan [2].

Sehubungan dengan hal-hal yang sudah disebutkan, penelitian akan dilakukannya klasifikasi ulasan pengguna Shazam di Google Play Store ke dalam dua sentimen, yakni positif dan negatif. Dalam pengklasifikasian ini, metode SVM akan digunakan untuk membantu proses klasifikasi. SVM adalah teknik pembelajaran mesin yang digunakan untuk memetakan input ke output yang diinginkan. SVM tergunakan secara optimal dalam menyelesaikan masalah klasifikasi dan regresi [3]. SVM menggunakan fungsi-fungsi linier di dalam ruang fitur sebagai ruang hipotesis. SVM diberi instruksi dalam memanfaatkan algoritma pembelajaran yang mengambil inspirasi dari teori optimasi dengan mempergunakan *learning bias* yang diambil referensinya dari teori pembelajaran statistik [4].

Dalam konteks menganalisis sentimen ulasan suatu aplikasi di Google Play Store dengan implementasi algoritma SVM, terdapat beberapa penelitian relevan yang telah dilakukan sebelumnya. Penelitian-penelitian tersebut salah satunya yang diteliti oleh Friska Aditia Indriyani, Ahmad Fauzi, Sutan Faisal. Pada penelitian ini, para peneliti melakukan penerapan algoritma Naïve Bayes serta Support Vector Machine untuk menganalisis sentimen aplikasi TikTok dengan memperoleh hasil akurasi 79% untuk Naïve Bayes dan 84% untuk SVM yang dimana lebih tinggi dibandingkan dengan metode Naïve Bayes [5]. Lalu, hasil akurasi terbaik dihasilkan oleh Support Vector Machine sebesar 92% pada penelitian oleh Rismi Nurlaely, Dwi Sartika Simatupang, Kamdan, Ivana Lucia Kharisma [6]. Menyusul penelitian lain, yang menggunakan Decision Tree dan Support Vector Machine dalam menganalisis sentimen twitter pada program MBKM, hasilnya masing-masing 72,86% dan 84,76% [7]. Kemudian ada penelitian [8] yang menerapkan algoritma Naïve Bayes, Random Forest, dan Support Vector Machine. Sesuai penelitiannya, dihasilkan hasil akurasi dari Naïve Bayes adalah 94,16%, Random Forest menghasilkan 97,16% serta Support Vector Machine sebesar 96,01% dalam mengklasifikasikan sentimen review aplikasi Ruangguru sebagai positif atau negatif. Selain itu, penelitian lainnya [9], menyebutkan bahwa hasil akurasi dari Support Vector Machine sebesar 98%, bersamaan dengan penelitian [10], terhitung bahwa akurasi mencapai 74,20%. Penelitian yang menganalisis sentimen tentang fenomena PHK dengan Naïve Bayes dan Support Vector Machine [11], menyimpulkan hasil akurasi masing-masing 74,1% dan 84%. Adapun penelitian lain yang meneliti sentimen data ulasan pengguna aplikasi TIX ID. Penelitian ini menggunakan algoritma Support Vector Machine dan hasil akurasi yang diperoleh sebesar 94% [12]. Penelitian-penelitian diatas yang memaparkan mengenai tingkat akurasi Support Vector Machine yang tinggi, juga didukung oleh penelitian ini yang mengungkapkan bahwa hasil penelitiannya menghasilkan akurasi sebesar 98,98% [13]. Dalam penelitian-penelitian terkait sebelumnya yang telah dibahas, para peneliti merasa terdorong untuk menggunakan algoritma SVM untuk menganalisis sentimen ulasan, terutama pada aplikasi Shazam. Secara lebih lengkapnya, para peneliti akan menyusun penelitian yang akan diberi judul “Analisis Sentimen Ulasan Aplikasi Shazam di Google Play Store Menggunakan Support Vector Machine”. Digunakannya SVM agar hasil yang didapati dapat dipaparkan secara lebih terperinci dan informasi dapat diterima oleh publik secara jelas dan akurat. Maka dari itu, para peneliti mengharapkan bahwa untuk kemudian hari, penelitian ini dapat berguna dan bermanfaat khususnya dalam memperhatikan sisi popularitas aplikasi, pelayanan aplikasi yang berkualitas, serta dapat melakukan evaluasi yang lebih baik dan membenahi kekurangan yang ada.

2. METODE PENELITIAN

Pada penelitian ini akan dilakukannya analisis ulasan pengguna pada aplikasi Shazam di Google Play Store dengan menggunakan algoritma SVM, yang dilalui beberapa tahapan penelitian. Tahapan-tahapan dapat dibaca pada gambar dibawah, yakni Gambar 1.



Gambar 1. Rangkaian Penelitian

2.1 Pengumpulan Data

Pada penelitian tahap ini akan dilakukan pengambilan data primer. Data primer merupakan perolehan data yang bersumber (orang / organisasi) secara langsung yang digunakan untuk kepentingan studi yang dilakukan dengan wawancara ataupun observasi [14]. Data akan diambil sebanyak 1000 ulasan pengguna khususnya kategori “paling relevan”. Alasan utamanya adalah kategori tersebut menampilkan data ulasan yang paling berkaitan dengan aplikasi serta detil ulasan pengguna tentang tanggapan positif atau negatif dari aplikasi Shazam di situs Google Play Store yang diimplementasikan dengan bahasa pemrograman python untuk melakukan *scraping* di suatu web.

Web scraping didefinisikan sebagai perolehan dokumen semi-terstruktur dari internet [15]. *Web scraping* dapat dicapai dengan menuliskan program otomatis yang memperoleh data dari server web (HTML atau file tipe lain), lalu dilakukannya parsing data untuk mengesktrak informasi yang dibutuhkan [16]. Setelah *scraping* dilakukan, data tersebut akan disimpan dan dijadikan file yang memiliki tipe .csv.

2.2 Preprocessing

Pada penelitian tahap ini akan dilakukan proses pembersihan data dari yang sudah dikumpulkan sebelumnya. Data-data tersebut akan disusun dan dirangkai menjadi lebih dapat digunakan untuk tahap setelah ini. *Preprocessing* mencakup beberapa tahap, seperti: *cleaning* dan *case folding* (diubahnya semua huruf menjadi huruf kecil), lalu *tokenizing* (suatu kalimat akan dipisah untuk setiap kata).

Kemudian dilakukannya *stopword removal* dan *Stemming*. *Stopword removal* adalah cara untuk membuang kata-kata yang sekiranya tidak memiliki pengaruh pada kalimat yang ada [17]. Serta dilakukannya proses *stemming*. *Stemming* adalah cara dalam mengubah kata-kata yang ada menjadi bentuk kata yang paling dasar. Pembersihan data atau *data cleaning* merupakan teknik yang memiliki tujuan untuk menghilangkan data yang tidak terlalu penting atau tidak lengkap (data yang hilang ataupun tidak valid) [18].

2.3 Evaluasi Model

Pada penelitian tahap ini akan dilakukan pengumpulan data ulasan (review), dimana akan terpisah menjadi dua, yakni data untuk training dan data untuk testing. Sentimen yang berbasis Leksikon dapat dilakukan secara manual, yakni mencari kata-kata sentimen yang tergolong positif ataupun negatif [1].

SVM ialah suatu teknik pembelajaran terawasi yang membuat pemetaan fitur input-output dari *data training* secara menyeluruh. Fungsi pemetaan yang dimaksud dapat berupa fungsi klasifikasi atau regresi [19]. SVM menggunakan ruang hipotesis yang terdiri dari fungsi-fungsi linier pada *feature space*. SVM dapat dilatih menggunakan algoritma pembelajaran menurut teori optimasi dengan dilakukannya penerapan *learning bias* yang berasal dari teori pembelajaran statistik [20].

2.4 Analisis Sentimen

Pada penelitian tahap ini akan diterapkannya proses penentuan sentimen berupa teks baik positif ataupun negatif. Analisis sentimen dapat dijelaskan sebagai teknik komputasi persepsi, perasaan dan subjektivitas teks. Analisis sentimen diklasifikasikan berupa polaritas teks yang ada pada suatu dokumen atau kalimat ataupun pandangan (pendapat) secara positif ataupun negatif.

Analisis sentimen yang dilakukan pada penelitian ini akan terpecah menjadi dua label. Yang pertama: ulasan 4-5 tergolong positif. Yang kedua: ulasan 1-3 tergolong negatif. Dapat digolongkan tentunya dilakukan inialisasi pada *data training* dan *data testing*. Score kurang dari 3 dianggap negatif. Sebaliknya, score lebih dari 3 dianggap positif. Kemudian, dilanjutkan dengan memisahkan kata menjadi suatu token [21].

2.5 Evaluation Data Testing

Pada penelitian tahap ini akan dilakukannya evaluasi dari hasil *data testing*. Hasilnya dapat berupa pencarian nilai (*F1-score*, *precision*, *recall score*, serta *accuracy*). Hasil-hasil yang didapati akan melalui dengan alat uji confusion matrix yang dimana alat uji ini bertugas untuk mengetahui sebaran kebenaran data prediksi terhadap data aktual [22].

Jika dijabarkan satu-persatu dari penilaian yang dilakukan oleh confusion matrix adalah: *Accuracy* merupakan rasio prediksi suatu kebenaran (baik respon positif atau negatif). Lalu, *precision*. Ini merupakan rasio pada antara jumlah prediksi positif yang benar dengan total jumlah prediksi positif. Sedangkan, *recall* merupakan rasio positif benar dibandingkan dengan semua data positif. Kemudian, *F1-score* dapat diartikan sebagai perbandingan rata-rata *precision* dan *recall* [23]. Dikarenakan penelitian ini lebih berfokus pada tingkat akurasi, maka rumus dari akurasi adalah sesuai dengan persamaan (1).

$$Accuracy = \frac{TP+TN}{(TP+TN+FP+FN)} \times 100\% \tag{1}$$

Dengan TP yang merupakan kepanjangan dari *True Positive*, TN yang jika dipanjangkan menjadi *True Negative*, dan FP adalah *False Positive*, serta FN yakni *False Negative*.

3. HASIL DAN PEMBAHASAN

3.1 Hasil

Data dan temuan akan dibahas pada bagian hasil dengan tahapan melakukan *web scraping* terlebih dahulu, kemudian menggolongkan teks menjadi positif dan negatif yang dikenal dengan sebutan pelabelan. Terakhir, dilakukannya tahapan preprocessing.

3.1.1 Pengumpulan Data

Penelitian ini mendapatkan data dari situs Google Play Store pada aplikasi Shazam. Data diperoleh berdasarkan kategori paling relevan. Gambar 2 menunjukkan informasi yang didapatkan berasal dari 1000 ulasan dengan teknik scraping menggunakan bahasa pemrograman python.

content	score
Padahal dulu bagus banget ni app. Tapi semenjak beberapa bulan lalu, sekitar 2-3 bulan lalu lah, ni app mulai ga sebgus dulu	1
Shazam yang sekarang sangat tidak bagus, udah ga kebaca lagi lagu yang didengar. Kalo dulu suara pelan sekali atau dari kej	1
Seluruhnya udah bagus, pencarian nya pun cepat gak lama Tapi ada yang kurang yaitu lagu yang di speed up, di slowed, dilofi,	4
[06092022] gua install ini karna d.o. exo (korean idol actor) katanya install, awal awal gua pake yah cukup worth it. Sampai h	3
Kenapa semakin di update semakin buruk sihh, gak bisa membaca lagu lagi, ayo donk diperbaiki untuk Developer nya Kecewa	2
Bagus banget, aplikasi Shazam bisa mendeteksi lagu dari sumber audio luar ataupun perangkat sendiri tanpa harus keluar ap	4
Abis diupdate jadi parah banget, masa gak bisa ketebak lagunya, padahal suara jelas Dulu suara sayup sayup aja bisa ketemu	1
SANGAT TIDAK PUAS, karna saia mencari lagu. Tapi, tidak ada hasilnya, gak setiap lagu bisa ke analisa.	1
Sudah bagus, pencarian nya juga cepat sekalieh. Hasil nya juga sesuai, tpi untuk saran jika lagu tersebut di cover ataupun tid	4
Shazam keren semua lagu favorit saya dapat di temukan di Shazam bahkan dari jarak yang cukup jauh Shazam dapat mereka	5
Dulu susah banget cari judul lagu pas lagi di Cafe, Mall atau tempat umum atau pas lagi jadi Soundtrack iklan atau film. Nah !	5
Makin jelek ni apk .. dulu gampang banget nyari lagu sekarang malah susah padahal suaranya jelas ... tolong di perbaiki	1
Saya berikan bintang lima karna selama ini apa yg saya cari, musik yg bisa dikata sukar untuk judul musiknya susah didapat ta	5
Ini yang aplikasi saya cari mantap bener gak perlu lama tinggal pake rekam suara langsung dapat lagunya the best	5
Sangat sangat buruk pencarian search lebih ketimbang AI selamat tinggal pencundang apk ronsokan	1
Luar biasa! aplikasi ini bisa mendeteksi dan mencarikan judul musik walaupun jauh dari sumber suara dan terdengar sayup-si	5
Sangat bagus dan cepat dalam mencari lagu, tapi ada 1 lagu dari playlist saya yang tidak bisa ditemukan jadi saya kasih 4 binti	4
Aplikasinya sangat membantu sekali untuk mencari musik yang sulit diketahui dengan hanya mendengar saja, dengan adanya	5
Kok sekarang ngebaca lagunya lambat banget yah bahkan seringkali ga kebaca	3
Cari judul lagu jadi lebih mudah, keren aplikasinya.	5
Bagus sekali, karena udah bisa cari judul lagu + tanpa iklan bagus banget ðŸŽ‰ðŸŽ‰ðŸŽ‰	5
Bagus juga sih aplikasi nya soal nya pas aku mau cari judul lagu yang aku mau lngsng ketemu	5

Gambar 2. Hasil Scraping

Setelah scraping dilakukan dan data diperoleh. Akan dilakukannya pelabelan data. Pelabelan data bertujuan agar dapat mengetahui ulasan 4-5 tergolong positif dan ulasan 1-3 tergolong negatif. Hasil dapat dilihat pada gambar di bawah ini, yakni Gambar 3.

```
def pelabelan(rate):  
    if rate < 3:  
        return 'negatif'  
    else:  
        return 'positif'  
  
df['Label'] = df['score'].apply(pelabelan)  
df.head()
```

	content	score	Label
0	Padahal dulu bagus banget ni app. Tapi semenja...	1	negatif
1	Shazam yang sekarang sangat tidak bagus, udah ...	1	negatif
2	Seluruhnya udah bagus, pencarian nya pun cepat...	4	positif
3	[06092022] gua install ini karna d.o. exo (kor...	3	positif
4	Kenapa semakin di update semakin buruk sih, g...	2	negatif

Gambar 3. Pelabelan

3.1.2 Tahap Preprocessing

Kemudian, ada tahapan selanjutnya merupakan *preprocessing*, yang dimana akan dilakukan pembersihan data atau *cleaning* dari data-data yang sudah dikumpulkan. Data-data tersebut akan disusun dan dirangkai menjadi lebih rapi agar dapat digunakan untuk tahap setelahnya. *Preprocessing* mencakup beberapa serangkaian tahapan yang diterapkan pada data mentah untuk menghasilkan data yang berkualitas dan akurat. Tahapan yang dimaksud adalah *cleaning*, yang memiliki fungsi dalam membuang kata-kata yang dianggap tidak memberikan pengaruh terhadap kalimat, yang dimaksud seperti tanda baca, karakter kosong, dan emoji. Setelah itu, *case folding* yang bertujuan untuk mengubah semua huruf menjadi huruf kecil. Kemudian, dilakukannya *tokenizing* agar kalimat terpotong per bagian setiap kata. Lanjut, untuk *stopword removal* agar kata-kata yang tidak berpengaruh dihilangkan. Ini dilakukan agar data dapat lebih jelas terbaca. Terakhir, dilakukannya *stemming*, agar kata-kata yang ada dapat dieja kembali menjadi kata dasar. Serangkaian tahapan ini akan menghasilkan sesuai pada gambar di bawah ini, beserta dengan *code script*.

```

lemma = WordNetLemmatizer()
stemmer = PorterStemmer()
stop_words = set(stopwords.words('indonesian'))

def CleanReview(txt):
    txt = re.sub(r'http\S+', ' ', txt)
    txt = re.sub('[^a-zA-Z]', ' ', txt)
    txt = str(txt).lower()
    txt = word_tokenize(txt)
    txt = [item for item in txt if item not in stop_words]

    txt = [lemma.lemmatize(word=w,pos='v') for w in txt]
    txt = [i for i in txt if len(i) > 2]
    txt = ' '.join(txt)
    return txt

df['CleanReview'] = df['content'].apply(CleanReview)

df.head()

```

	content	score	Label	CleanReview
0	Padahal dulu bagus banget ni app. Tapi semenja...	1	negatif	bagus banget app semenjak app seba... berkali ...
1	Shazam yang sekarang sangat tidak bagus, udah ...	1	negatif	shazam bagus udah kebaca lagu didengar kalo su...
2	Seluruhnya udah bagus, pencarian nya pun cepat...	4	positif	udah bagus pencarian nya cepat gak lagu speed ...
3	[06092022] gua install ini karna d.o. exo (kor...	3	positif	gua install karna exo korean idol actor instal...
4	Kenapa semakin di update semakin buruk sih, g...	2	negatif	update buruk sih gak membaca lagu ayo donk di...

Gambar 4. Hasil *Preprocessing*

3.2 Pembahasan

Hasil analisis data akan digambarkan jawabannya berupa dilakukannya *data training* dan *data testing*, *count vectorizer*, pengujian tingkat akurasi, pencarian jumlah prediksi yang benar dan salah untuk setiap kelas dengan *confusion matrix*, dan terakhir melihat hasil penilaian (*precision*, *recall*, *F1*) berupa laporan klasifikasi.

3.2.1 Evaluasi Model

Jika *preprocessing* sudah dilakukan semua. Maka selanjutnya adalah mempersiapkan data dalam bentuk model klasifikasi *Linear Support Vector Classification* (Linear SVC). Pertama, skrip akan dibagi datanya menjadi *data training* sebanyak 80% kemudian untuk *data testing* sebanyak 20% dengan penerapan “*train_test_split*”. Lalu, skrip menggunakan “*CountVectorizer*” berfungsi dalam mengubah teks ulasan menjadi fitur numerik yang dapat digunakan oleh model. *Code script* disertakan pada Gambar 5.

```

from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.svm import LinearSVC

x = df['CleanReview']
y = df['Label']
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.2)

```

Gambar 5. *Data Training* dan *Data Testing*

```
vectorizer = CountVectorizer()
vectorizer.fit(x_train)

CountVectorizer
CountVectorizer()

x_train = vectorizer.transform(x_train)
x_test = vectorizer.transform(x_test)
```

Gambar 6. Count Vectorizer

Selanjutnya, akan melatih model Linear SVC untuk *data training* dan menguji akurasi dengan *data testing* yang menggunakan *accuracy_score*. Digunakannya nilai C yang berbeda (0.01, 0.05, 0.25, 0.5, 0.75, dan 1) untuk melatih model Linear SVC. C adalah parameter yang mengontrol *penalty* untuk kesalahan klasifikasi. Besarnya nilai C yang dihasilkan, maka akan semakin besar *penalty* untuk kesalahan klasifikasi yang dapat menghasilkan model yang lebih kompleks dan akurat. Pada output di Gambar 7., hasil menunjukkan bahwa model dengan nilai C = 0.05 memiliki akurasi tertinggi (0.84). Sedangkan, model dengan nilai C = 0.01 memiliki akurasi terendah (0.79).

```
for c in [0.01, 0.05, 0.25, 0.5, 0.75, 1]:
    svm = LinearSVC(C=c)
    svm.fit(x_train, y_train)
    print('Akurasi untuk c = %s: %s' %(c, accuracy_score(y_test, svm.predict(x_test))))

Akurasi untuk c = 0.01: 0.79
Akurasi untuk c = 0.05: 0.84
Akurasi untuk c = 0.25: 0.815
Akurasi untuk c = 0.5: 0.81
Akurasi untuk c = 0.75: 0.805
Akurasi untuk c = 1: 0.8
```

Gambar 7. Tingkat Akurasi

3.2.2 Analisis Sentimen

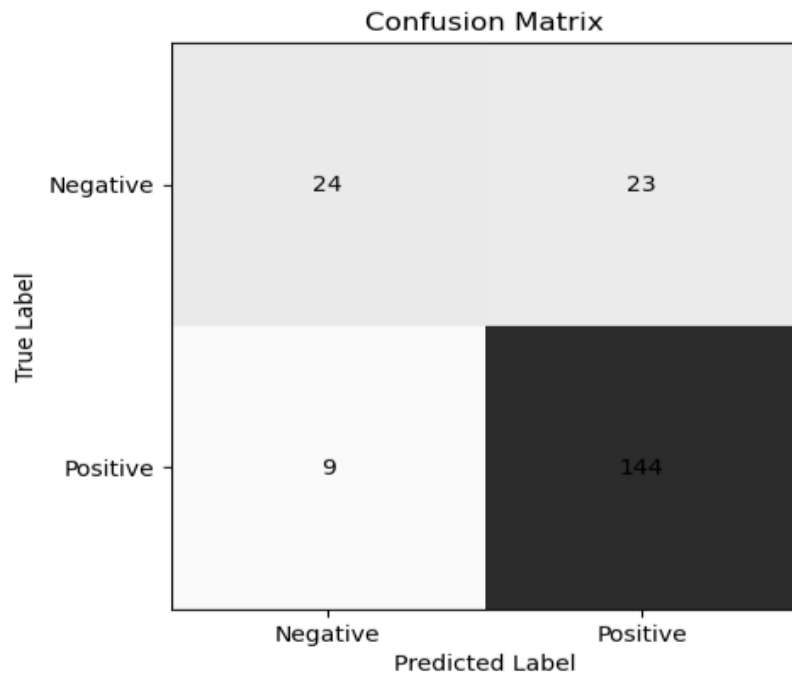
Tahap selanjutnya adalah mengetahui jumlah prediksi yang benar dan salah untuk setiap kelas dengan *confusion matrix*, tertera pada gambar dibawah, yakni Gambar 8, 9, 10 serta Tabel 1 untuk dapat dibaca lebih jelas.

```
y_pred = svm.predict(x_test)
print('Accuracy of SVM classifier on test set: {:.2f}'.format(svm.score(x_test, y_test)))

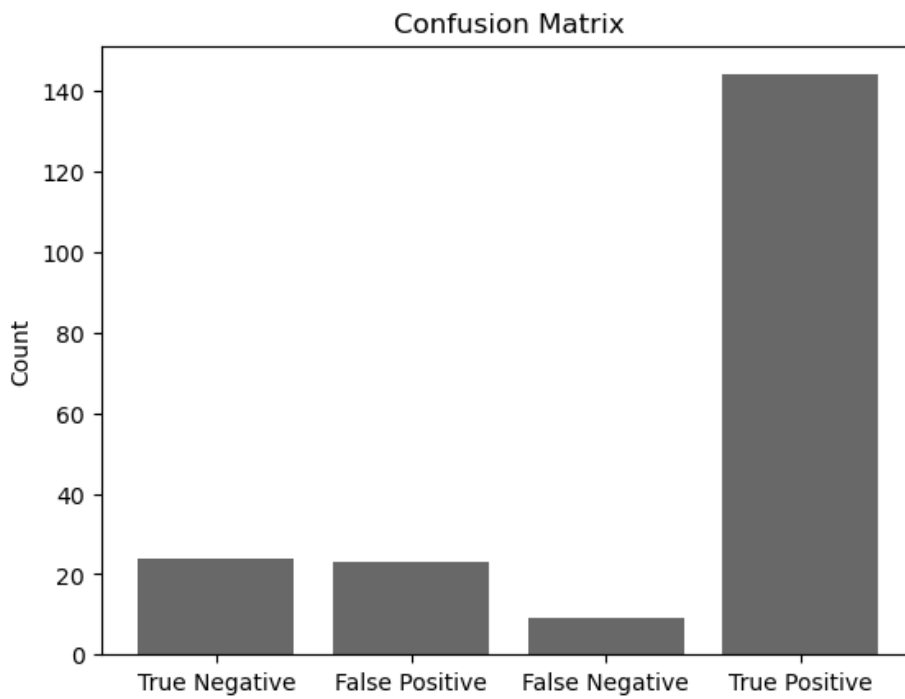
conf_mat = confusion_matrix(y_test, y_pred)
print(conf_mat)

Accuracy of SVM classifier on test set: 0.84
[[ 24  23]
 [  9 144]]
```

Gambar 8. Code Script Confusion Matrix



Gambar 9. Confusion Matrix



Gambar 10. Diagram Batang Confusion Matrix

Tabel 1. Confusion Matrix

Prediksi	Negatif	Positif
Negatif	24	23
Positif	9	144

Lebih spesifiknya adalah dengan menghitung secara manual untuk tingkat akurasi yang menunjukkan bahwa 84% (0.84) dengan menggunakan rumus pada persamaan (1) yang sudah dituliskan sebelumnya.

$$Accuracy = \frac{TP + TN}{(TP + TN + FP + FN)} \times 100\%$$

$$Accuracy = \frac{144 + 24}{(144 + 24 + 23 + 9)} \times 100\%$$

$$Accuracy = \frac{168}{200} \times 100\%$$

$$Accuracy = 0.84$$

$$Accuracy = 84\%$$

3.2.3 Evaluation Data Testing

Tahapan terakhir dari tahapan-tahapan selanjutnya adalah melihat hasil penilaian dari nilai *precision*, *recall*, *F1*, tertera pada Gambar 11.

```
print(classification_report(y_test, y_pred))
```

	precision	recall	f1-score	support
negatif	0.73	0.51	0.60	47
positif	0.86	0.94	0.90	153
accuracy			0.84	200
macro avg	0.79	0.73	0.75	200
weighted avg	0.83	0.84	0.83	200

Gambar 11. Laporan Klasifikasi

Pada gambar diatas menunjukkan *precision*, *recall*, dan *F1-score* untuk setiap kelas (negatif dan positif) serta *accuracy*, *macro average* dan *weighted average* dari model. Dalam hal ini, model SVM memiliki akurasi 0.84%, yang berarti model telah berhasil memprediksi 84% dari sampel di set data uji. *Precision* untuk kelas negatif adalah 0.73, yang berarti 73% dari sampel yang diprediksi sebagai negatif benar-benar negatif. *Recall* untuk kelas negatif adalah 0.51, yang berarti 51% dari sampel negatif sebenarnya berhasil diprediksi sebagai negatif. *F1-score* untuk kelas negatif adalah 0.60, yang diartikan sebagai rata-rata harmonik pada *precision* dan *recall*.

Demikian pula, *precision*, *recall*, dan *F1-score* untuk kelas positif adalah 0.86, 0.94, dan 0.90, secara berurutan. Rata-rata makro dari *precision*, *recall*, dan *F1-score* adalah 0.79, 0.73, dan 0.75, secara berurutan. *Weight average* dari *precision*, *recall*, dan *F1-score* adalah 0.83, 0.84, dan 0.83, secara berurutan.

4. KESIMPULAN

Dari hasil-hasil yang sudah diperoleh, model klasifikasi SVM memiliki akurasi yang cukup tinggi dalam memprediksi sentimen ulasan aplikasi Shazam, yaitu sebesar 84%. Ini membuktikan bahwa model klasifikasi SVM dapat mengenali sebagian besar ulasan sebagai positif atau negatif dengan benar.

Selain itu, dapat dilihat bahwa analisis sentimen aplikasi Shazam menggunakan SVM mendapat respon positif yang tinggi sebanyak 167 ulasan, sedangkan hanya sebanyak 33 ulasan untuk respon negatif yang didapatkan dari 1000 ulasan di situs Google Play Store. Jika dihitung secara lebih terperinci, rasio yang didapatkan pada respon positif adalah $167/1000 = 0.167$ atau 16.7%. Sedangkan, rasio yang didapatkan pada respon negatif adalah $33/1000 = 0.033$ atau 3.3%. Ini dapat berarti bahwa aplikasi musik Shazam mendapatkan lima kali lebih besar dan lebih banyak untuk respon positif dibanding dengan respon negatif yang diperoleh.

SVM lebih baik dalam memprediksi ulasan positif dibanding dengan ulasan negatif. Hal ini terlihat dari nilai yang dihasilkan pada *precision*, *recall*, dan *F1-score* yang lebih tinggi terhadap kelas

positif. Ini sangat dimungkinkan bahwa ulasan positif lebih banyak dan lebih mudah dibedakan dibanding dengan ulasan negatif.

Daftar Pustaka

- [1] Liu B, *Sentiment Analysis and Opinion Mining*. Amerika Serikat: Morgan & Claypool, 2012. doi: 10.2200/s00416ed1v01y201204hlt016.
- [2] Alwasi'a A., "Analisis Sentimen pada Review Aplikasi Berita Online Menggunakan Metode Maximum Entropy (Studi Kasus: Review Detikcom pada Google Play 2019)," Universitas Islam Indonesia, Yogyakarta, 2020. [Online]. Available: <http://dspace.uui.ac.id/123456789/24006>
- [3] L. Wang, *Support Vector Machines: Theory and Applications*. Berlin, Jerman: Springer, 2005. doi: 10.1007/b95439.
- [4] N. Cristianini and John. Shawe-Taylor, *An Introduction to Support Vector Machines and other kernel-based learning methods*. Cambridge, Inggris: Cambridge University Press, 2000. doi: 10.1017/CBO9780511801389.
- [5] F. A. Indriyani, A. Fauzi, and S. Faisal, "Analisis sentimen aplikasi tiktok menggunakan algoritma naïve bayes dan support vector machine," *TEKNOSAINS : Jurnal Sains, Teknologi dan Informatika*, vol. 10, no. 2, pp. 176–184, Jul. 2023, doi: 10.37373/tekn.v10i2.419.
- [6] R. Nurlaely, D. S. Sartika, Kamdan, and I. L. Kharisma, "Analisis Sentimen Twitter Terhadap Cyberbullying Menggunakan Metode Support Vector Machine (SVM)," *Jurnal Computer Science and Information Technology (CoSciTech)*, vol. 4, no. 2, pp. 376–384, Aug. 2023, doi: 10.37859/coscitech.v4i2.5161.
- [7] L. A. Pramesti and N. Pratiwi, "Analisis Sentimen Twitter Terhadap Program MBKM Menggunakan Decision Tree dan Support Vector Machine," *Journal of Information System Research (JOSH)*, vol. 4, no. 4, pp. 1145–1154, Jul. 2023, doi: 10.47065/josh.v4i4.3807.
- [8] E. Fitri, Y. Yuliani, S. Rosyida, and W. Gata, "Analisis Sentimen Terhadap Aplikasi Ruangguru Menggunakan Algoritma Naïve Bayes, Random Forest Dan Support Vector Machine," *TRANSFORMATIKA*, vol. 18, no. 1, pp. 71–80, Jul. 2020, doi: 10.26623/transformatika.v18i1.2317.
- [9] I. S. K. Idris, Y. A. Mustofa, and I. A. Salihi, "Analisis Sentimen Terhadap Penggunaan Aplikasi Shopee Menggunakan Algoritma Support Vector Machine (SVM)," *Jambura Journal of Electrical and Electronics Engineering*, vol. 5, no. 1, pp. 323–332, Jan. 2023, doi: <https://doi.org/10.37905/jjee.v5i1.16830>.
- [10] D. Oktavia, Y. R. Ramadahan, and Minarto, "Analisis Sentimen Terhadap Penerapan Sistem E-Tilang Pada Media Sosial Twitter Menggunakan Algoritma Support Vector Machine (SVM)," *KLIK: Kajian Ilmiah Informatika dan Komputer*, vol. 4, no. 1, pp. 407–417, Aug. 2023, doi: 10.30865/klik.v4i1.1040.
- [11] Mohd. A. Saddam, E. K. D, and Indra, "Analisis Sentimen Fenomena PHK Massal Menggunakan Naïve Bayes dan Support Vector Machine," *Jurnal Informatika: Jurnal pengembang IT (JPIT)*, vol. 8, no. 3, pp. 226–233, Sep. 2023, doi: <https://doi.org/10.30591/jpit.v8i3.4884>.
- [12] M. R. Nadhif, Bratam Dwija Wisnu, and B. Rahayudi, "Analisis Sentimen Data Ulasan Pengguna Aplikasi TIX ID di Indonesia pada Google Play Store menggunakan Support Vector Machine," *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, vol. 6, no. 8, pp. 3932–3937, Aug. 2022, [Online]. Available: <http://j-ptiik.ub.ac.id>
- [13] Y. A. Singgalen, "Analisis Sentimen Top 10 Traveler Ranked Hotel di Kota Makassar Menggunakan Algoritma Decision Tree dan Support Vector Machine," *KLIK: Kajian Ilmiah Informatika dan Komputer*, vol. 4, no. 1, pp. 32–35, Aug. 2023, doi: 10.30865/klik.v4i1.1153.
- [14] S. H. Situmorang, *Analisis Data Untuk Riset Manajemen dan Bisnis*, 4th ed. Medan: USU Press, 2018.
- [15] M. Turland, *Phparchitect's Guide to Web Scraping with PHP*, 1st ed. Canada: musketeers.me, LLC, 2010.
- [16] R. Mitchell, *Web Scraping with Python: Collecting More Data from The Modern Web*, 2nd ed. O'Reilly Media, 2018. doi: 10.1017/CBO9780511801389.
- [17] M. Diki Hendriyanto, A. A. Ridha, and U. Enri, "Analisis Sentimen Ulasan Aplikasi Mola pada Google Play Store Menggunakan Algoritma Support Vector Machine," *Journal of Information Technology and Computer Science (INTECOMS)*, vol. 5, no. 1, pp. 1–7, Jun. 2022, doi: 10.31539/intecom.v5i1.3708.
- [18] D. Darwis, N. Siskawati, and Z. Abidin, "Penerapan Algoritma Naïve Bayes untuk Analisis Sentimen Review Data Twitter BMKG Nasional," *Jurnal Tekno Kompak*, vol. 15, no. 1, pp. 131–145, 2021, doi: 10.33365/JTK.V15I1.744.
- [19] R. Wahyudi and G. Kusumawardhana, "Analisis Sentimen pada Review Aplikasi Grab di Google Play Store Menggunakan Support Vector Machine," *Jurnal Informatika*, vol. 8, no. 2, pp. 200–207, Sep. 2021.
- [20] Q. Luo, "Advancing Knowledge Discovery and Data Mining," in *Proceedings - 1st International Workshop on Knowledge Discovery and Data Mining (WKDD)*, Adelaide, Australia, 2008, pp. 3–5. doi: 10.1109/WKDD.2008.153.

- [21] F. Septianingrum, J. H. Jaman, and U. Enri, “Analisis Sentimen pada Isu Vaksin Covid-19 di Indonesia dengan Metode Naive Bayes Classifier,” *Jurnal Media Informatika Budidarma*, vol. 5, no. 4, pp. 1431–1437, Oct. 2021, doi: 10.30865/mib.v5i4.3260.
- [22] B. P. Pratiwi, A. S. Handayani, and Sarjana, “Pengukuran Kinerja Sistem Kualitas Udara dengan Teknologi WSN Menggunakan Confusion Matrix,” *Jurnal Informatika*, vol. 6, no. 2, pp. 66–75, Dec. 2020, doi: 10.26877/jiu.v6i2.6552.
- [23] A. Rahim, Kusriani, and E. T. Luthfi, “Convolutional Neural Network untuk Klasifikasi Penggunaan Masker,” *Inspiration: Jurnal Teknologi Informasi dan Komunikasi*, vol. 10, no. 2, pp. 109–115, Dec. 2020.



ZONasi: Jurnal Sistem Informasi

is licensed under a [Creative Commons Attribution International \(CC BY-SA 4.0\)](https://creativecommons.org/licenses/by-sa/4.0/)