

Sentimen Analisis Pada Ulasan Aplikasi Ajaib Di *Google Play Store* Dengan Algoritma *Support Vector Machine*

Alfi Syahri¹, Angraini², Fitriani Muttakin³

Universitas Islam Negeri Sultan Syarif Kasim Riau, Pekanbaru, Riau, Indonesia^{1,2,3},

Email: 11950311521@students.uin-suska.ac.id¹, angraini@uin-suska.ac.id², fitrianimuttakin@uin-suska.ac.id³

Abstract - The development of mobile applications in the financial sector has made it easier to invest. One application that can make investments is the Ajaib application. In using the application, you can see the ratings and reviews given by users on the Google Play Store platform. User reviews on the Ajaib App provide an important overview for potential users in understanding quality and user satisfaction. However, the large number of reviews makes manual analysis difficult and inefficient. Therefore, a review classification technique is needed that utilizes the Support Vector Machine (SVM) algorithm. The implementation is done through the Python programming language. The Support Vector Machine technique has shown remarkable accuracy in handling high-dimensional and imbalanced data. The purpose of this research is to facilitate the progress of the Magic App by utilizing the feedback provided, as well as reducing user complaints that exist in reviews by identifying and fixing problems that users often face. The data used is taken from application reviews as much as 5000 data with varying ratings from January to October 2023. Based on the results of sentiment analysis on Ajaib application reviews, it is found that user sentiment about the application tends to be positive which results in 68.6% positive sentiment polarity, negative sentiment polarity of 16.1%, and neutral sentiment polarity of 15.3%. From these results developers can gauge how well the app's features function in the face of user requests and provide a more seamless and satisfying experience.

Keywords – Sentiment Analysis, Ajaib Application, Support Vector Machine,

Intisari – Perkembangan aplikasi mobile di bidang keuangan telah memberikan kemudahan dalam berinvestasi. Salah satu aplikasi yang bisa melakukan investasi adalah aplikasi Ajaib. Dalam menggunakan aplikasi, Anda dapat melihat rating dan review yang diberikan oleh pengguna di platform *Google Play Store*. Ulasan pengguna pada Aplikasi Ajaib memberikan gambaran penting bagi calon pengguna dalam memahami kualitas dan kepuasan pengguna. Namun, banyaknya tinjauan membuat analisis manual menjadi sulit dan tidak efisien. Oleh karena itu diperlukan suatu teknik klasifikasi review yang memanfaatkan algoritma *Support Vector Machine* (SVM). Implementasinya dilakukan melalui bahasa pemrograman Python. Teknik *Support Vector Machine* menunjukkan akurasi luar biasa dalam menangani data berdimensi tinggi dan data tidak seimbang. Tujuan dari penelitian ini adalah untuk memfasilitasi kemajuan Aplikasi Ajaib dengan memanfaatkan umpan balik yang diberikan, serta mengurangi keluhan pengguna yang ada pada ulasan dengan mengidentifikasi dan memperbaiki masalah yang sering dihadapi pengguna. Data yang digunakan diambil dari *review* aplikasi sebanyak 5000 data dengan rating yang bervariasi pada bulan Januari hingga Oktober 2023. Berdasarkan hasil analisis sentimen pada ulasan aplikasi Ajaib diperoleh bahwa sentimen pengguna tentang aplikasi cenderung positif dimana menghasilkan 68,6 % polaritas sentimen positif, polaritas sentimen negatif sebesar 16,1 %, dan polaritas sentimen netral sebanyak 15,3%. Dari hasil ini pengembang dapat mengukur seberapa baik fitur-fitur aplikasi berfungsi dalam menghadapi permintaan pengguna dan memberikan pengalaman yang lebih mulus dan memuaskan.

Kata Kunci – Analisis Sentimen, Aplikasi Ajaib, *Support Vector Machine*,

I. PENDAHULUAN

Teknologi berkembang begitu pesat diberbagai bidang terutama pada bidang financial yang mana telah melakukan berbagai macam transaksi salah satunya kegiatan investasi. Sekarang investasi dapat dilakukan teknologi *mobile application* sehingga para investor dapat melakukan transaksi dengan nyaman[1]. banyaknya *mobile application* yang ada membuat investor memiliki pertimbangan khusus dalam memilih aplikasi untuk investasi. Faktor lain yang menjadi pertimbangan antara lain memiliki banyak fitur yang mudah digunakan, waktu pemrosesan yang cepat, memiliki kesederhanaan antarmuka, dan menawarkan banyak keuntungan untuk pengguna. Salah satu *mobile application* untuk investasi adalah aplikasi Ajaib di *Google Play Store*, yang memiliki berbagai fitur investasi serta menawarkan berbagai opsi pembayaran[2]. Aplikasi Ajaib sendiri telah didownload 5 juta lebih pengguna dan mendapatkan lebih 150 ribu ulasan di *Google Play Store*.

Banyaknya ulasan pada aplikasi Ajaib tentu memberikan kemudahan kepada calon pengguna untuk menilai aplikasi dan dapat menjadi acuan kepada calon pengguna untuk mendapatkan informasi terkait aplikasi. Ulasan dari pengguna biasanya berisi saran dan kritik sehingga dapat menjadi informasi penting untuk pihak aplikasi dalam mengembangkan aplikasinya[3]. Akan tetapi, banyaknya ulasan yang diberikan tersebut membuat pihak *developer* aplikasi maupun masyarakat umum memerlukan banyak waktu dalam melakukan analisa terhadap ulasan yang diberikan. Permasalahan lainnya adalah terdapat banyaknya *rating* yang diberikan oleh pengguna tidak sesuai dengan ulasan yang diberikan. Untuk mengelompokkan ulasan yang begitu banyak akan sulit jika diproses secara manual, maka diperlukan suatu klasifikasi dalam bentuk analisis sentimen[4]. Analisis sentimen adalah prosedur *teks mining* yang menggunakan teknik penambangan teks untuk mengekstrak informasi dari teks tidak terstruktur menggunakan algoritma *data mining*[5]. Dari informasi yang diolah dengan algoritma dihasilkan teks ulasan pengguna yang menyampaikan pendapat positif, negatif, dan netral[6].

Untuk menganalisis dibutuhkan algoritma yang mampu menangani data yang kompleks. Algoritma *Support Vector Machine* (SVM) merupakan salah satu algoritma yang sesuai untuk menangani data yang kompleks. SVM adalah metode pembelajaran mesin yang sangat efektif yang telah menunjukkan keberhasilan besar dalam tugas-tugas seperti klasifikasi teks dan identifikasi pola. Metode ini bekerja dengan mencari pemisah *linier* optimal, dalam konteks ini ulasan positif, negatif dan netral. Dengan SVM, kita dapat mengembangkan model klasifikasi yang mampu mengidentifikasi sentimen ulasan pengguna dengan akurasi yang tinggi[7]. Teknik SVM digunakan untuk menganalisis sentimen pada aplikasi Ajaib berdasarkan uraian masalah yang diberikan. Algoritma *Support Vector Machine* digunakan karena kemampuannya menghasilkan akurasi yang signifikan dengan menggunakan daya komputasi yang lebih sedikit dan mudah untuk dilatih dibandingkan dengan algoritma *machine learning* lainnya[8]

Oleh karena itu, penelitian ini memiliki tujuan untuk membantu pengembangan aplikasi berdasarkan ulasan yang diberikan, serta memperbaiki keluhan pengguna dan meningkatkan kualitas aplikasi Ajaib. Selain itu diharapkan dapat meningkatkan sentimen pengguna tentang aplikasi dan mengetahui apakah sentimen pengguna lebih ke positif, negatif, atau netral. Signifikan dari penelitian ini terletak pada kontribusi terhadap pemahaman tentang bagaimana pengguna merespon aplikasi Ajaib. Pengembang dapat memanfaatkan umpan balik pengguna ini untuk meningkatkan kualitas aplikasi dan mengoptimalkan pengalaman pengguna.

Selain itu terdapat beberapa keterbatasan yang perlu diperhatikan, ukuran sampel yang terbatas dapat membatasi generalisabilitas temuan terhadap populasi pengguna aplikasi Ajaib secara keseluruhan. Selain itu menggunakan algoritma analisis sentimen yang kompleks, seperti *Support Vector Machine* (SVM), keterbatasan teknis dan metodologis dari algoritma

tersebut dapat mempengaruhi akurasi klasifikasi sentimen. Kemudian data ulasan pengguna yang di analisis mungkin memiliki keterbatasan tersendiri, seperti bias dalam jenis ulasan yang diunggah atau ketidakmampuan dalam menangani bahasa informal[9]. Penting untuk diingat bahwa sentimen dapat berbeda berdasarkan situasi dan pengalaman pribadi, yang mungkin tidak sepenuhnya dapat dipahami jika hanya mengandalkan analisis sentimen berbasis mesin.

II. SIGNIFIKANSI STUDI

A. Studi Literatur

Analisis sentimen, juga disebut sebagai *opinion mining*, adalah bidang studi dalam *teks mining* yang berupaya memastikan opini atau subjektivitas individu terhadap topik, peristiwa, atau isu tertentu [10]. *Text mining* terkait erat dengan pemrosesan bahasa alami (NLP). NLP adalah bidang studi yang berfokus pada pengembangan teknologi yang memungkinkan komputer memahami dan menafsirkan pentingnya bahasa manusia. NLP umumnya digunakan untuk tujuan penambangan teks, yang melibatkan ekstraksi informasi berharga dari teks. Hal ini meliputi identifikasi kata kunci, subjek, serta analisis sentimen dan klasifikasi teks.

1. Teks Processing

Tujuan dari pemrosesan awal teks (*Teks Processing*) adalah untuk mengubah dan mengekstrak kata-kata tidak terstruktur dari dokumen dan mengubahnya menjadi format yang dapat dipahami dan berharga. *Processing* adalah langkah penting dan penting dalam penambangan data, karena data yang diperoleh dari sumber data sering kali berisi beberapa istilah tidak beraturan dan sejumlah besar *noise*[11]. Ketika melakukan *Text processing* perlu menganalisis teks yang tidak hanya tema tetapi sudut pandang dan posisi yaitu teks orientasi[12]. Tahapan *preprocessing* teks yang digunakan dijelaskan di bawah ini;

1. *Case Folding*, Data tersebut berasal dari kata-kata yang menunjukkan ketidakberesan dalam bentuk tulisannya, seperti penggunaan huruf kapital, spasi, susunan kata, dan faktor sejenisnya. *Case Folding* mengacu pada tindakan mengubah konten dalam dokumen untuk mencapai struktur yang kohesif dan konsisten.
2. *Tokenizing*, Tokenisasi adalah tindakan mengelompokkan teks, seperti paragraf atau kalimat, menjadi unit-unit yang lebih kecil. Metode tokenisasi ini menggunakan pembatas atau pemisah seperti spasi, carriage return, atau *whitespace*.
3. Normalisasi, Normalisasi merupakan proses mengubah semua bentuk kata menjadi satu bentuk baku.
4. *Filtering (Stopword Removal)*, *Stopwords* adalah kata-kata yang sering muncul dan tidak dianggap penting dalam konteks dokumen. Item leksikal yang kurang memiliki makna semantik dihilangkan selama proses *filtering*.
5. *Stemming*, *Stemming* adalah prosedur memodifikasi kata yang mengandung imbuhan dan selanjutnya mengubahnya menjadi bentuk dasar dalam suatu dokumen.

2. TF-IDF

TF-IDF adalah metode umum untuk menimbang fitur dan menunjukkan tingkat presisi dan perolehan yang relatif tinggi. Pembobotan TF-IDF adalah metode pembobotan yang umum digunakan di bidang pencarian informasi. Bobot TF-IDF dinilai signifikan. Jika sebuah kata lebih sering muncul dalam satu dokumen, nilai kontribusinya lebih tinggi. Sebaliknya, jika istilah tersebut lebih sering muncul di beberapa dokumen, nilai kontribusinya akan lebih rendah. TF-IDF adalah teknik yang menggabungkan *Term Frekuensi* (TF) dan *Inverse Document Frekuensi* (IDF)[13]. Rumus TF dapat didefinisikan dengan [13]:

$$TF_{t,d} = (1 + \log(TF_{t,d})) \tag{1}$$

Keterangan:

$TF_{t,d}$ = *term frequent document*

Sedangkan rumus untuk menghitung nilai IDF yaitu :

$$IDF_t = \log \frac{N}{1 + DF_t} \tag{2}$$

Keterangan:

N = jumlah seluruh dokumen

DF_t = *document frequent term*

TF-IDF menggunakan rumus untuk menentukan bobot ($W_{t,d}$) setiap dokumen :

$$W_{t,d} = TF_{t,d} * IDF_t \tag{3}$$

Keterangan:

$TF_{t,d}$ = *term frequent document*

IDF_t = *inverse document frequent term*[14]

3. Support Vector Machine (SVM)

SVM adalah algoritma pembelajaran mesin yang menerapkan fungsi *hyperplane* pada data untuk membentuk wilayah tiap kelas. *Hyperplane* sendiri adalah fungsi yang digunakan untuk memisahkan antar kelas yang ada. Saat memprediksi kategori dari data, SVM memberi label pada data berdasarkan kategori yang termasuk di dalamnya. SVM sering digunakan pada kumpulan data besar yang diperoleh dari situs *online* dan menjadi populer karena penerapannya dalam klasifikasi teks[15]. Algoritma SVM dapat mengekstrak beberapa fitur dari kata dan gambar dan kemudian menentukan bobot fitur yang optimal [16]. Keberhasilan algoritma Support Vector Machine (SVM) untuk pengenalan pola mungkin terutama disebabkan oleh banyaknya hasil eksperimen luar biasa yang diperoleh di area aplikasi yang sangat berbeda [17]. SVM adalah metode yang sudah mapan dalam proses bahasa alami (NLP). Ini adalah teknik pembelajaran mesin yang sangat penting berdasarkan kernel dan salah satu algoritma yang paling banyak digunakan untuk pembelajaran khususnya dalam tugas klasifikasi [18].

4. Confusion Matrix

Confusion Matrix adalah model berharga untuk mengevaluasi keakuratan sistem klasifikasi dengan menentukan kualitas labelnya. *Confusion Matrix* digunakan dalam proses evaluasi model untuk memastikan metrik kinerja pendekatan yang berlaku, termasuk akurasi, *recall*, *precision*, dan *F1-Score*[19].

TABEL I
CONFUSION MATRIX

Kelas	Prediksi Negatif	Prediksi Netral	Prediksi Positif
Aktual Negatif	<i>True Negatif (TNg)</i>	<i>False Netral2 (FNt2)</i>	<i>False Positive1 (FP1)</i>
Aktual Netral	<i>False Negatif2 (FNg2)</i>	<i>True Netral2 (TNt)</i>	<i>False Positive2 (FP2)</i>
Aktual Positif	<i>False negatif1 (FNg1)</i>	<i>False Netral1 (FNt1)</i>	<i>True Positive (TP)</i>

Keterangan:

1. *True Negatif*/ Aktual Negatif = data aktual termasuk dalam kelas negatif dan prediksi modelnya negatif.
2. *True Positif*/ Aktual Positif = data aktual termasuk dalam kelas positif dan prediksi modelnya positif.
3. *True Netral*/ Aktual Netral = data aktual termasuk dalam di kelas netral dan prediksi modelnya netral.
4. *False Negatif*/ Prediksi Negatif = data sebenarnya termasuk dalam kelas positif atau netral, namun model memprediksi kelas negatif.
5. *False Positif*/ Prediksi Positif = data aktual masuk dalam kategori negatif atau netral, namun model memperkirakan positif.
6. *False Netral*/ Prediksi Netral = data sebenarnya termasuk dalam kelas negatif atau positif, tetapi model memprediksi netral.

Dari enam kemungkinan *confusion matrix*, dapat diperoleh nilai *accuracy*, *precision*, *recall*, dan *f1-score*. Berdasarkan pada persamaan dari:

1. *Accuracy* merupakan total keseluruhan nilai benar pada model dalam melakukan akurasi. Rumus akurasi dapat dilihat pada persamaan 4:

$$Accuracy = \frac{TP + TNg + TNt}{TP + FNt2 + \dots + FNt1 + TNg} \quad (4)$$

2. *Precision* merupakan seberapa sering prediksi benar saat model melakukan prediksi positif. Rumus *precision* dapat dilihat rumus 5,6, dan 7:

$$Positif = \frac{TP}{TP + FP1 + FP2} \quad (5)$$

$$Negatif = \frac{TNg}{TNg + FNg1 + FNg2} \quad (6)$$

$$Netral = \frac{TNt}{TNt + FNt1 + FNt2} \quad (7)$$

3. *Recall* mengacu pada frekuensi model memprediksi hasil positif dengan tepat ketika kelas sebenarnya positif. Rumus *recall* disajikan pada rumus 8, 9, dan 10:

$$Positif = \frac{TP}{FNg1 + FNt1 + TP} \quad (8)$$

$$Negatif = \frac{TNg}{FP1 + FNt2 + TNg} \quad (9)$$

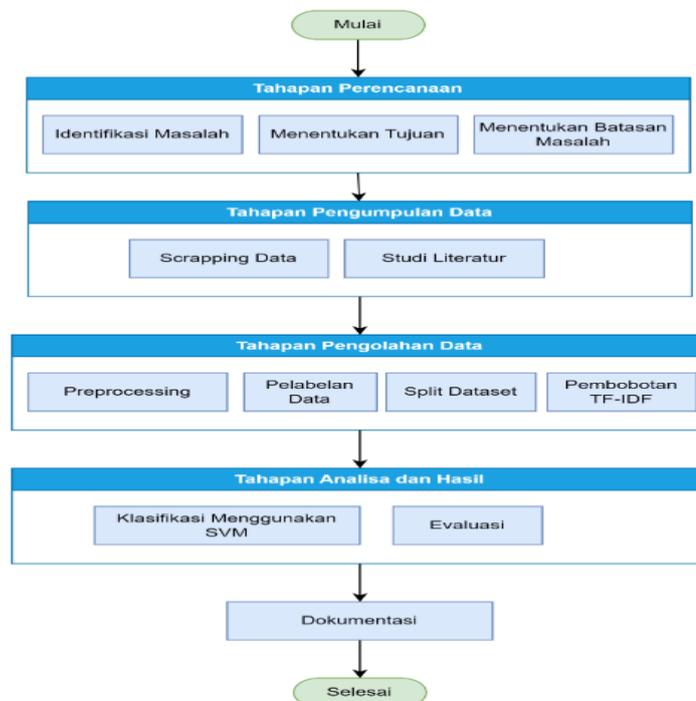
$$Netral = \frac{TNt}{FNg2 + FP2 + TNt} \quad (10)$$

4. *F1-score* merupakan rata-rata harmonik dari nilai *precision* dan *recall*. Rumus *F1-Score* dapat dilihat pada rumus 11:

$$2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (11)$$

B. Tahapan Penelitian

Penelitian ini melakukan analisis sentimen pada aplikasi Ajaib yang tersedia di Google Play Store. Tahapan awal *scrapping* data ulasan aplikasi Ajaib menggunakan menggunakan bahasa pemograman Python dan memanfaatkan *library google play scraper*. Selanjutnya melakukan *processing* data, pelabelan data, *Split dataset*, dan Pembobotan kata. Berikut adalah gambaran proses:



Gambar 1. Metodologi Penelitian

Pada gambar 1 dijelaskan bahwa alur penelitian ini dimulai dari tahapan perencanaan dengan mengidentifikasi masalah, menentukan tujuan, dan menentukan batasan masalah. Kemudian dilanjutkan dengan tahapan pengumpulan data dengan *scrapping* data untuk mengumpulkan ulasan yang ada pada aplikasi Ajaib. Tahapan selanjutnya pengolahan data, tahapan ini data yang telah dikumpulkan diolah dengan teks *processing* untuk membersihkan *noise* data. Selanjutnya, data yang diolah dikategorikan ke dalam label positif, negatif, dan netral berdasarkan ulasan. Selanjutnya, dataset dipartisi menjadi data *training* dan data *testing*, dengan 80% dialokasikan untuk pelatihan dan 20% untuk *testing*. Selanjutnya, setiap kata diberi bobot berdasarkan nilai TF-IDF-nya. Mengikuti prosedur pembobotan TF-IDF, evaluasi pengguna dikategorikan pada aplikasi menggunakan teknik *Support Vector Machine* dengan kernel *linear*. Proses klasifikasi dilakukan menggunakan bahasa pemograman Python.

C. Hipotesis Penelitian

Berdasarkan permasalahan yang ada maka apakah analisis sentimen pada ulasan aplikasi Ajaib dengan menggunakan algoritma *Support Vector Machine* dapat membedakan antara ulasan positif, netral, dan negatif dan menghasilkan prediksi yang akurat terkait sentimen ulasan pengguna aplikasi Ajaib.

III. HASIL DAN PEMBAHASAN

A. Scrapping Data

Pengumpulan data dilakukan dengan *scrapping data* dari ulasan pengguna aplikasi Ajaib di Google Play Store. Data yang digunakan yaitu data ulasan dalam rentang waktu Januari sampai Oktober 2023 dengan *rating* 1 sampai 5. *Scrapping data* dilakukan dengan menggunakan bahasa pemrograman Python yang menggunakan *library* google-play-scraper. Data yang digunakan dalam penelitian ini yaitu 5000 data. Gambar berikut merupakan hasil dari *scrapping data* yang telah dilakukan :

A	B	C	D	E	F
	reviewId	userName	userImage	content	score
0	5a2016b0	Zakia Balq	https://pl	Saya di suruh update,namun tidak bisa terpusat,bag	4
1	2be76dab	ady septiy	https://pl	Mantaap	5
2	a632c340	Budi Yono	https://pl	Apk yg sangat membantu	5
3	a35b68be	Mohamm	https://pl	Bagus	5
4	b6fb3ab3	Samudra I	https://pl	Aplikasi ajaib ini kenapa tidak real time, selalu ada je	3
5	ca656768	Ngurah an	https://pl	Peletane sesepe jak umi soleha	5
6	b7cf89b2	Pur Wind	https://pl	Mau daftar kode ajaib apaan ya	5
7	3cc007f4	Asep Husr	https://pl	bagus banget aplikasinya mudah di guna kan	5
8	8bd6af73	afrizal par	https://pl	Baik bagus keren	5
9	b8616669	Zavero Ne	https://pl	apk nya lumayan bagus untuk belajar saham dan lai	5
10	4a9fc470	Deki X	https://pl	nama ajaib.. tapi proses ini itu.. serba lelet... gak ajai	1
11	82c42e6c	Rama Prar	https://pl	Aku suka banget aplikasi ini. Udah gitu aja	5
12	6d272417	Masa Dep	https://pl	Kecewa ngadain event undang teman tapi gak mau b	1
13	9872f43e	Wira Lam	https://pl	Bagus banget	5
14	6f655091	Ismail	https://pl	Al hamdulillah aplikasinya bagus	1
15	70ceab59	Abde Faris	https://pl	Sangat baik dan mudah d fahami untuk pemula sepe	4
16	dc597784	Wira "Day	https://pl	Ok	5
17	dc9ad888	Arif Arif	https://pl	Ajarin dong puh sepuh	2
18	de6684a5	Oki Adine	https://pl	Woiiii.....!!!! Saldo RDN kemana ini?????!!!! Gak per	1
19	2b17740a	Suwardy V	https://pl	Min tolong buat bisa beli SBN Dong	5
20	f15e531d	Arifin XB	https://pl	Min kenapa hari ini portofolio saya hilang yaa?	3
21	91e33a88	m Yunus	https://pl	Sangat jelek, harga belum nyentuh stop Los saldo ud	1
22	76f80a96	KLIMPIJA	https://pl	Bagi pengguna baru ini ada kode untuk sama sama k	5

Gambar 2. Data Hasil *Scrapping*

B. Text Prossessing

1. Casefolding

Selama tahap *Casefolding*, ubah semua huruf besar menjadi huruf kecil dengan menggunakan *library* `re` dan fungsi `re.sub()`. Hasil dari *Casefolding* disajikan pada Tabel II:

TABEL II
CASEFOLDING

No	Text Input	Case Folding
1	Saya di suruh update,namun tidak bisa terpusat,bagaimana ini	saya di suruh updatenamun tidak bisa terpusatbagaimana ini
2	Mantaap	mantaap
3	Apk yg sangat membantu	apk yg sangat membantu
4	Bagus	bagus
5	Aplikasi ajaib ini kenapa tidak real time, selalu ada jeda di bagian beranda watchlist dan portofolio tidak sinkron, mohon dari team ajaib di tindak lanjuti, maaf jika bintang saya kurangi satu sampai ada perbaikan.	aplikasi ajaib ini kenapa tidak real time selalu ada jeda di bagian beranda watchlist dan portofolio tidak sinkron mohon dari team ajaib di tindak lanjuti maaf jika bintang saya kurangi satu sampai ada perbaikan

2. Tokenizing

Tokenizing merupakan proses dalam pemecahan kata menjadi beberapa bagian dan hasil kata yang dipecah disebut dengan token. Tahapan *tokenizing* dilakukan dengan memanfaatkan *library* `nltk.tokenize`. Hasil pada *tokenizing* dapat dilihat pada tabel III:

TABEL III
TOKENIZING

No	Text Input	Tokenizing
1	saya di suruh updatenamun tidak bisa terpusatbagaimana ini	['saya', 'di', 'suruh', 'updatenamun', 'tidak', 'bisa', 'terpusatbagaimana', 'ini']
2	mantaap	['mantaap']
3	apk yg sangat membantu	['apk', 'yg', 'sangat', 'membantu']
4	Bagus	['bagus']
5	aplikasi ajaib ini kenapa tidak real time selalu ada jeda di bagian beranda watchlist dan portofolio tidak sinkron mohon dari team ajaib di tindak lanjuti maaf jika bintang saya kurangi satu sampai ada perbaikan	['aplikasi', 'ajaib', 'ini', 'kenapa', 'tidak', 'real', 'time', 'selalu', 'ada', 'jeda', 'di', 'bagian', 'beranda', 'watchlist', 'dan', 'portofolio', 'tidak', 'sinkron', 'mohon', 'dari', 'team', 'ajaib', 'di', 'tindak', 'lanjuti', 'maaf', 'jika', 'bintang', 'saya', 'kurangi', 'satu', 'sampai', 'ada', 'perbaikan']

3. Normalisasi

Tahap normalisasi merupakan proses mengubah kata menjadi bentuk yang baku. Hasil normalisasi pada normalisasi dapat dilihat pada table IV:

TABEL IV
NORMALISASI

No	Text Input	Normalisasi
1	['saya', 'di', 'suruh', 'updatenamun', 'tidak', 'bisa', 'terpusatbagaimana', 'ini']	['saya', 'di', 'suruh', 'updatenamun', 'tidak', 'bisa', 'terpusatbagaimana', 'ini']
2	['mantaap']	['mantaap']
3	['apk', 'yg', 'sangat', 'membantu']	['apk', 'yang', 'sangat', 'membantu']
4	['bagus']	['bagus']
5	['aplikasi', 'ajaib', 'ini', 'kenapa', 'tidak', 'real', 'time', 'selalu', 'ada', 'jeda', 'di', 'bagian', 'beranda', 'watchlist', 'dan', 'portofolio', 'tidak', 'sinkron', 'mohon', 'dari', 'team', 'ajaib', 'di', 'tindak', 'lanjuti', 'maaf', 'jika', 'bintang', 'saya', 'kurangi', 'satu', 'sampai', 'ada', 'perbaikan']	['aplikasi', 'ajaib', 'ini', 'kenapa', 'tidak', 'real', 'time', 'selalu', 'ada', 'jeda', 'di', 'bagian', 'beranda', 'watchlist', 'dan', 'portofolio', 'tidak', 'sinkron', 'mohon', 'dari', 'team', 'ajaib', 'di', 'tindak', 'lanjuti', 'maaf', 'jika', 'bintang', 'saya', 'kurangi', 'satu', 'sampai', 'ada', 'perbaikan']

4. Filtering

Tahap *filtering* merupakan proses penghapusan kata yang tidak dibutuhkan berupa kata keterangan dan kata sambung dari hasil token. Hasil *filtering* dapat dilihat pada tabel V.

TABEL V
FILTERING

No	Text Input	Filtering
1	['saya', 'di', 'suruh', 'updatenamun', 'tidak', 'bisa', 'terpusatbagaimana', 'ini']	['saya', 'di', 'suruh', 'updatenamun', 'tidak', 'bisa', 'terpusatbagaimana', 'ini']
2	['mantaap']	['mantaap']
3	['apk', 'yang', 'sangat', 'membantu']	['apk', 'sangat', 'membantu']
4	['bagus']	['bagus']
5	['aplikasi', 'ajaib', 'ini', 'kenapa', 'tidak', 'real', 'time', 'selalu', 'ada', 'jeda', 'di', 'bagian', 'beranda', 'watchlist', 'dan', 'portofolio', 'tidak', 'sinkron', 'mohon', 'team', 'ajaib', 'tindak', 'lanjuti', 'maaf', 'bintang', 'saya', 'kurangi', 'satu', 'sampai', 'ada', 'perbaikan']	['aplikasi', 'ajaib', 'real', 'time', 'selalu', 'jeda', 'bagian', 'beranda', 'watchlist', 'portofolio', 'sinkron', 'mohon', 'team', 'ajaib', 'tindak', 'lanjuti', 'maaf', 'bintang', 'saya', 'kurangi', 'satu', 'sampai', 'ada', 'perbaikan']

No	Text Input	Filtering
	‘dari’, ‘team’, ‘ajaib’, ‘di’, ‘tindak’, ‘bintang’, ‘kurangi’, ‘satu’, ‘perbaiki’]	
	‘lanjuti’, ‘maaf’, ‘jika’, ‘bintang’, ‘saya’, ‘kurangi’, ‘satu’, ‘sampai’, ‘ada’, ‘perbaiki’]	

5. *Stemming*

Tahap *stemming* melibatkan transformasi kata menjadi bentuk akar atau dasarnya. Hasil dari proses *stemming* ditampilkan pada tabel VI:

TABEL VI
STEMMING

No	Text Input	Stemming
1	['saya', 'di', 'suruh', 'updatenamun', 'tidak', 'bisa', 'terpusatbagaimana', 'ini']	['suruh', 'updatenamun', 'terpusatbagaimana']
2	['mantaap']	['mantaap']
3	['apk', 'sangat', 'membantu']	['apk', 'sangat', 'bantu']
4	['bagus']	['bagus']
5	['aplikasi', 'ajaib', 'real', 'time', 'selalu', 'jeda', 'bagian', 'beranda', 'watchlist', 'portofolio', 'sinkron', 'mohon', 'team', 'ajaib', 'tindak', 'lanjut', 'maaf', 'bintang', 'kurangi', 'satu', 'perbaiki']	['aplikasi', 'ajaib', 'real', 'time', 'selalu', 'jeda', 'bagi', 'beranda', 'watchlist', 'portofolio', 'sinkron', 'mohon', 'team', 'ajaib', 'tindak', 'lanjut', 'maaf', 'bintang', 'kurang', 'satu', 'baik']

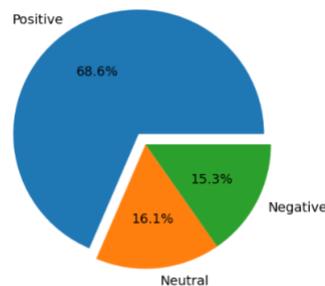
6. *Pelabelan*

Pelabelan data pada tahap ini melibatkan pengkategorian setiap ulasan pengguna ke dalam kategori positif, netral, atau negatif. Pelabelan dilakukan dengan menggunakan kamus *List of Opinion Words* berbahasa Indonesia yang dapat diakses pada url

TABEL VII
PELABELAN

No	Teks	Polaritas
1	suruh updatenamun terpusatbagaimana	negative
2	Mantaap	positive
3	apk sangat bantu	positive
4	Bagus	positive
5	aplikasi ajaib real time selalu jeda bagi beranda watchlist portofolio sinkron mohon team ajaib tindak lanjut maaf bintang kurang satu baik	positive

Pelabelan dilakukan dengan melakukan pengecekan kata yang terkandung dalam setiap kalimatnya. Apabila jumlah kata positif dalam suatu kalimat lebih banyak, maka dianggap polaritas positif. Apabila jumlah kata negatif dalam suatu kalimat lebih banyak, maka dianggap polaritas negatif. Sedangkan jika jumlah kata positif dan negatif sama maka dianggap polaritas netral. Hasil dari pelabelan ulasan pengguna pada aplikasi Ajaib yaitu 3428 data merupakan polaritas positif, 806 merupakan polaritas netral, dan 766 data merupakan polaritas negative. Berikut visualisasi hasil polaritas sentimen:



Gambar 3. Polaritas Sentimen

Dari visualisasi pada gambar menampilkan bahwa polaritas sentimen positif sebanyak 68,6%, polaritas netral sebanyak 16,1%, dan polaritas sentimen negatif sebanyak 15,3%.

7. *Split Dataset*

Tahap split dataset melibatkan pemartisian data menjadi dua subset: data *training* dan data *testing*. Data tersebut akan dipartisi menjadi 80% data *training* dan 20% data *testing*. Parameter random state akan disetel ke 42 untuk operasi ini.

8. TF-IDF

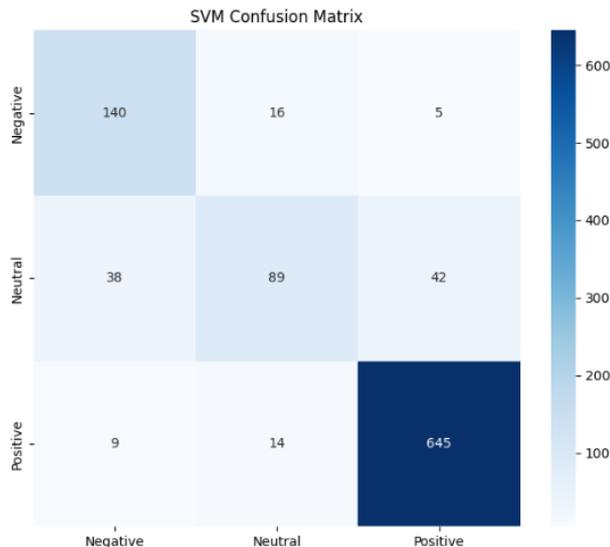
TF-IDF merupakan fitur pembobotan dengan presisi dan *recall* yang cukup tinggi. TF lebih memperhatikan istilah kata yang banyak dalam suatu dokumen. Pada saat yang sama, IDF memprioritaskan frasa dengan bobot lebih rendah yang disertakan dalam sejumlah besar dokumen. TF-IDF dilakukan dengan memanfaatkan *library* `TfidfVectorizer` yang diimport dari `sklearn.feature_extraction.text`.

9. Klasifikasi Algoritma *Support Vector Machine*

Saat ini, program Ajaib menggunakan teknik SVM dengan kernel linier untuk mengkategorikan evaluasi pengguna. Klasifikasi dilakukan setelah melalui tahap *preprocessing*, lalu tahap pelabelan, selanjutnya tahap pembagian dataset menjadi 80% data *training* dan 20% data *testing*. Selanjutnya dilakukan pembobotan dengan TF IDF menggunakan *library* `TfidfVectorizer` yang diimport dari `sklearn.feature_extraction.text`. Selanjutnya dilakukan klasifikasi menggunakan algoritma SVM.

10. Evaluasi

Setelah dilakukan klasifikasi maka selanjutnya dilakukan evaluasi terhadap performa algoritma *Support Vector Machine*. Evaluasi dilakukan dengan menggunakan *confusion matrix* untuk menghasilkan nilai *accuracy*, *precision*, *recall* dan *f1-score*. *Confusion Matrix* dapat dilihat pada Gambar



Gambar 4. SVM Confusion Matrix

Berdasarkan Gambar 4 dapat dijelaskan bahwa diperoleh pada label positif terklasifikasi dengan benar sebanyak 645 data, sedangkan kesalahan prediksi 9 data yang masuk ke dalam label negatif, dan 14 data masuk ke dalam label netral. Pada label netral dengan jumlah 89 data terklasifikasi dengan benar, sedangkan kesalahan prediksi dengan 38 data yang masuk ke dalam label negatif, dan 42 data masuk ke dalam label positif. Pada label negatif dengan jumlah sebanyak 140 data terklasifikasi benar, sedangkan kesalahan prediksi dengan 16 data yang masuk ke dalam label netral, dan 5 data masuk ke dalam label positif. Sehingga dapat disimpulkan bahwa sentimen pada ulasan aplikasi Ajaib cenderung positif.

```

Classification Report:
              precision    recall  f1-score   support

   Negative      0.75      0.87      0.80      161
   Neutral      0.75      0.53      0.62      169
   Positive      0.93      0.97      0.95      668

 accuracy              0.88      998
 macro avg              0.81      998
 weighted avg           0.87      998

Confusion Matrix:
[[140  16   5]
 [ 38  89  42]
 [   9  14 645]]
Accuracy Score:
0.875751503006012
    
```

Gambar 5. Clasification Report

Penerapan teknik SVM untuk menganalisis review pengguna aplikasi Ajaib menggunakan foto menghasilkan tingkat *accuracy* sebesar 87,57%. *Precision* pada kelas positif 93%, *recall* 97%, dan *f-1 score* 95%. Sedangkan pada kelas netral didapatkan *precision* sebesar 75%, *recall* 53%, dan *f-1 score* sebesar 62%. Serta pada kelas negatif mendapatkan *precision* sebesar 75%, *recall* 87%, dan *f-1 score* sebesar 80%.

11. Diskusi

Hasil analisis sentimen ini memiliki implikasi yang signifikan bagi pengembang aplikasi Ajaib dan pengguna potensial. Pertama, pengembang dapat menilai kekuatan dan kekurangan

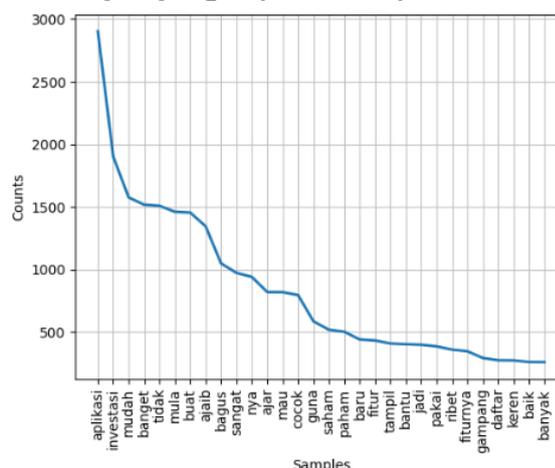
suatu aplikasi dengan lebih akurat ketika mereka memiliki pemahaman yang lebih baik tentang perasaan pengguna terhadap berbagai aplikasi. Untuk menjaga dan meningkatkan kualitas layanan, program ini memiliki kemampuan untuk mengidentifikasi wilayah di mana pelanggan cenderung memberikan ulasan yang baik. Hal ini dapat berfungsi sebagai titik fokus untuk aplikasi. Selain itu, data ini juga dapat membantu dalam proses penetapan taktik pemasaran yang lebih efektif. Secara umum, pemahaman yang lebih mendalam mengenai sentimen pengguna dapat menjadi instrumen yang sangat berguna bagi pengembang, memungkinkan mereka untuk terus meningkatkan kualitas aplikasi mereka, dan bagi calon pengguna, memungkinkan mereka membuat pilihan dan pilihan yang lebih cerdas ketika memilih aplikasi investasi.

Penggunaan TF-IDF dalam pemrosesan teks membantu dalam mengidentifikasi kata-kata yang paling relevan dan bermakna dalam mengekspresikan sentimen pengguna terhadap Aplikasi Ajaib. Dengan memberikan bobot yang lebih tinggi pada kata-kata yang jarang muncul namun memiliki dampak besar dalam mengidentifikasi sentimen, TF-IDF memungkinkan model untuk lebih sensitif terhadap fitur-fitur penting dalam data teks. Selanjutnya, penggunaan SVM sebagai algoritma klasifikasi memberikan kinerja yang stabil dan akurat dalam mengklasifikasikan sentimen dari data teks yang kompleks. SVM mampu mengatasi permasalahan dalam analisis sentimen dan menghasilkan temuan yang konsisten karena kemampuannya dalam menangani data yang berdimensi tinggi dan jumlah sampel yang relatif terbatas.

Oleh karena itu, penggunaan SVM dan TF-IDF bersama-sama dalam analisis sentimen Aplikasi Ajaib akan menghasilkan hasil yang dapat diandalkan dan kuat. Metodologi ini tidak hanya meningkatkan pemahaman kita tentang persepsi pengguna terhadap platform investasi ini, tetapi juga memberikan dasar yang kokoh untuk pengembangan strategi yang lebih efektif dalam meningkatkan layanan dan pengalaman pengguna di masa mendatang.

12. Visualisasi Kata

Berikut ini adalah visualisasi kata dengan memanfaatkan *library* matplotlib.pyplot pada python dengan menampilkan grafik terkait jumlah kata yang muncul. Grafik menampilkan 30 *sample* dengan jumlah kata terbanyak yang dapat dilihat pada Gambar dibawah ini:



Gambar 6. Jumlah Sampel Kata Terbanyak

Berdasarkan Gambar 6 dapat terlihat visualiasi kata terbanyak berdasarkan 30 *sample*. Adapun jumlah kata terbanyak terdapat pada kata “aplikasi” sebanyak 2903 kata, serta kata “investasi” sebanyak 1901. Jumlah kata yang paling banyak muncul menunjukkan bahwa data penelitian ini cocok dengan aplikasi investasi Ajaib. Sementara itu, hasil visualisasi dari kata-

kata dalam ulasan pengguna aplikasi Ajaib mencakup ulasan positif dan negatif. Untuk visualisasi lebih lanjut, silakan lihat Gambar 7.



Gambar 7. Visualisasi Kata Positif

Gambar 7 menampilkan hasil visualisasi kata-kata positif. Di sini, terlihat ulasan positif dari pengguna yang telah menggunakan aplikasi Ajaib, seperti "cocok banget", "mudah paham", "aplikasi Ajaib bagus banget", dan sebagainya.



Gambar 8. Visualisasi Kata Negatif

Pada Gambar 8 terlihat hasil visualisasi kata-kata negatif. Di sini, terdapat ulasan negatif dari pengguna yang telah menggunakan aplikasi Ajaib, seperti “aplikasi Ajaib tidak jelas”, “saham tidak masuk”, dan “verifikasi error”.

IV. KESIMPULAN

Berdasarkan analisis sentimen *Support Vector Machine* terhadap penilaian aplikasi Ajaib, diambil kesimpulan sebagai berikut:

1. Teknik *Support Vector Machine* beserta *tools Google Colaboratory* dan bahasa pemrograman Python telah dimanfaatkan secara efektif untuk melakukan analisis sentimen pada review aplikasi Ajaib. Pada penelitian ini peneliti menggunakan kamus *List of Opinion Word* berbahasa Indonesia dalam melakukan pelabelan, menggunakan proporsi pembagian dataset yaitu 80%:20%, menerapkan teknik TF-IDF untuk pembobotan pada setiap kata. Kemudian dilakukan klasifikasi sentimen dengan menggunakan teknik SVM, dan hasil klasifikasi akan dievaluasi dengan menggunakan *confusion matrix*.
2. Berdasarkan hasil analisis sentimen pada ulasan aplikasi Ajaib diperoleh bahwa sentimen pengguna tentang aplikasi cenderung positif dimana menghasilkan 68,6 % polaritas sentiment positif, polaritas sentiment negatif sebesar 16,1 %, dan polaritas sentimen netral sebanyak 15,3%.
3. Berdasarkan hasil evaluasi menggunakan *confusion matrix*, didapatkan hasil *accuracy* sebesar 87,57%. Pada sentiment positif didapatkan *precision* sebesar 93%, *recall* 97%, dan *f-1 score* sebesar 95%. Sedangkan pada sentimen netral didapatkan *precision* sebesar

75%, *recall* 53%, dan *f-1 score* sebesar 62%. Serta pada sentimen negatif mendapatkan *precision* sebesar 75%, *recall* 87%, dan *f-1 score* sebesar 80%.

4. Dari hasil sentimen yang positif maka pihak pengembang dapat membangun fitur-fitur baru yang disesuaikan dengan kebutuhan individu dan preferensi pengguna berdasarkan hasil analisis sentimen, yang biasanya menguntungkan. Dengan demikian fitur baru tersebut dapat meningkatkan daya tarik dan nilai tambah aplikasi.

Saran untuk pengembangan penelitian selanjutnya yaitu dapat dikembangkan dengan menggunakan ulasan pengguna aplikasi Ajaib dari media sosial lain serta dapat dikembangkan dengan menggunakan *feature selection* untuk meningkatkan akurasi. Kemudian menggunakan algoritma selain *Support Vector Machine* dalam melakukan analisis sentimen.

REFERENSI

- [1] S. Shellyna and W. Yuwono, "ktor-Faktor Potensial dalam Adopsi Fintech Lending di Kota Batam," *Organum: Jurnal Saintifik Manajemen dan Akuntansi*, vol. 5, no. 2, pp. 152–172, 2022, doi: 10.35138/organum.v5i2.286.
- [2] D. Safryda Putri and T. Ridwan, "Analisis Sentimen Ulasan Aplikasi Pospay Dengan Algoritma Support Vector Machine," *Jurnal Ilmiah Informatika*, vol. 11, no. 01, pp. 32–40, 2023, doi: 10.33884/jif.v11i01.6611.
- [3] R. D. Darmawan and H. Rohman, "Peningkatan Performa Pengalaman Pengguna Aplikasi Seluler Ajaib dengan Pendekatan Design Thinking dan Pengujian A/B: Studi Kasus UX Terhadap Ajaib - Platform Investasi Daring," *Jurnal Informatika dan Rekayasa Perangkat Lunak*, vol. 4, no. 1, p. 1, 2022, doi: 10.36499/jinrpl.v4i1.4762.
- [4] Y. A. Singgalen, "Analisis Performa Algoritma NBC, DT, SVM dalam Klasifikasi Data Ulasan Pengunjung Candi Borobudur Berbasis CRISP-DM," *Building of Informatics, Technology and Science (BITS)*, vol. 4, no. 3, pp. 1634–1646, 2022, doi: 10.47065/bits.v4i3.2766.
- [5] Y. Findawati, U. Indahyanti, Y. Rahmawati, and R. Puspitasari, "Sentiment Analysis of Potential Presidential Candidates 2024: A Twitter-Based Study," *Academia Open*, vol. 8, no. 1, pp. 1–17, 2023, doi: 10.21070/acopen.8.2023.7138.
- [6] N. C. Dang, M. N. Moreno-Garcia, and F. De la Prieta, "Sentiment Analysis Based on Deep Learning in E-Commerce," *Electronics (Basel)*, pp. 1–29, 2020, doi: 10.1007/978-3-031-10986-7_40.
- [7] A. Muhammadin and I. A. Sobari, "Analisis Sentimen Pada Ulasan Aplikasi Kredivo Dengan Algoritma Svm Dan Nbc," *Reputasi: Jurnal Rekayasa Perangkat Lunak*, vol. 2, no. 2, pp. 85–91, 2021, doi: 10.31294/reputasi.v2i2.785.
- [8] A. Tedyyana, O. Ghazali, and O. W. Purbo, "Machine learning for network defense: automated DDoS detection with telegram notification integration," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 34, no. 2, p. 1102, May 2024, doi: 10.11591/ijeecs.v34.i2.pp1102-1109.
- [9] A. Tedyyana, O. Ghazali, and O. Purbo, "Model Design of Intrusion Detection System on Web Server Using Machine Learning Based," in *Proceedings of the 11th International Applied Business and Engineering Conference, ABEC 2023, September 21st, 2023, Bengkulu, Riau, Indonesia*, EAI, 2024. doi: 10.4108/eai.21-9-2023.2342879.
- [10] F. F. Rachman and S. Pramana, "Analisis Sentimen Pro dan Kontra Masyarakat Indonesia tentang Vaksin COVID-19 pada Media Sosial Twitter," *Health Information Management Journal*, vol. 8, no. 2, pp. 100–109, 2020.

- [11] G. A. Lustiansyah *et al.*, “Analisis klasifikasi sentimen pengguna aplikasi pedulilindungi berdasarkan ulasan dengan menggunakan metode long short term memory,” *Seminar Nasional Mahasiswa Ilmu Komputer dan Aplikasinya (SENAMIKA)*, pp. 327–336, 2022.
- [12] K. X. Han, W. Chien, C. C. Chiu, and Y. T. Cheng, “Application of support vector machine (SVM) in the sentiment analysis of twitter dataset,” *Applied Sciences (Switzerland)*, vol. 10, no. 3, 2020, doi: 10.3390/app10031125.
- [13] C. H. Yutika, A. Adiwijaya, and S. Al Faraby, “Analisis Sentimen Berbasis Aspek pada Review Female Daily Menggunakan TF-IDF dan Naïve Bayes,” *Jurnal Media Informatika Budidarma*, vol. 5, no. 2, p. 422, 2021, doi: 10.30865/mib.v5i2.2845.
- [14] S. Wehnert, V. Sudhi, S. Dureja, L. Kutty, S. Shahania, and E. W. De Luca, “Legal norm retrieval with variations of the bert model combined with TF-IDF vectorization,” *Proceedings of the 18th International Conference on Artificial Intelligence and Law, ICAIL 2021*, pp. 285–294, 2021, doi: 10.1145/3462757.3466104.
- [15] M. I. Fikri, T. S. Sabrila, and Y. Azhar, “Perbandingan Metode Naïve Bayes dan Support Vector Machine pada Analisis Sentimen Twitter,” *Smatika Jurnal*, vol. 10, no. 02, pp. 71–76, 2020, doi: 10.32664/smatika.v10i02.455.
- [16] J. Cao, M. Wang, Y. Li, and Q. Zhang, “Improved support vector machine classification algorithm based on adaptive feature weight updating in the Hadoop cluster environment,” *PLoS One*, vol. 14, no. 4, pp. 1–18, 2019, doi: 10.1371/journal.pone.0215136.
- [17] G. Blanchard, O. Bousquet, and P. Massart, “Statistical performance of support vector machines,” *Ann Stat*, vol. 36, no. 2, pp. 489–531, 2008, doi: 10.1214/009053607000000839.
- [18] Z. Quan and L. Pu, “An improved accurate classification method for online education resources based on support vector machine (SVM): Algorithm and experiment,” *Educ Inf Technol (Dordr)*, vol. 28, no. 7, pp. 8097–8111, 2023, doi: 10.1007/s10639-022-11514-6.
- [19] B. T. Fadiyah, D. E. Ratnawati, and I. Arwani, “Analisis Sentimen Pengguna Twitter terhadap Pembayaran Cashless menggunakan ShopeePAY dengan Algoritma Random Forest,” *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, vol. 6, no. 3, pp. 1426–1433, 2022.