

ANALISIS SENTIMEN KUALITAS APLIKASI DISCORD MENGGUNAKAN ALGORITMA NAÏVE BAYES DAN SUPPORT VECTOR MACHINE

Ichsan Dicki Fareza, Endah Tri Esti Handayani

Sistem Informasi, Universitas Nasional

Jl. Sawo Manila No.61, RT.14/RW.7, Pejaten Bar., Ps. Minggu, Jakarta, Indonesia

ichsandicki2021@student.unas.ac.id

ABSTRAK

Awalnya Discord merupakan aplikasi yang sering digunakan oleh para gamer sebagai platform komunikasi, namun seiring berjalannya waktu, Discord telah memperluas basis penggunaannya untuk mencakup beragam komunitas besar. Jumlah pengguna Discord yang kian bertambah, sangat dipengaruhi oleh persepsi dan ulasan positif dari para pengguna. Namun, tidak dapat dipungkiri bahwa masih ada ulasan negatif yang menunjukkan adanya aspek-aspek tertentu yang perlu diperbaiki. Sebagian besar ulasan pengguna Discord di *Google Play Store* memiliki rating di bawah 5 bintang dan banyak pengguna mengeluhkan adanya bug atau kendala dalam aplikasi. Penelitian ini bertujuan untuk mengetahui tingkat kepuasan pengguna terhadap aplikasi Discord, serta memberikan masukan atau support kepada pengembang Discord mengenai aspek aplikasi yang perlu ditingkatkan atau perlu diperbaiki. Dengan menggunakan algoritma *Naïve Bayes* dan *Support Vector Machine* untuk melakukan klasifikasi pada analisis sentimen ulasan pengguna aplikasi Discord, kita dapat memahami lebih jauh bagaimana persepsi tersebut memengaruhi kualitas Discord. Hasil penelitian menunjukkan bahwa model SVM memiliki kinerja yang lebih baik dibandingkan *Naive Bayes*, dengan nilai akurasi keseluruhan mencapai 88%. Sementara itu *Naive Bayes* memperoleh akurasi keseluruhan sebesar 78%.

Kata kunci : Analisis Sentimen, *Naïve Bayes*, *Support Vector Machine*, Discord, *Google Play Store*

1. PENDAHULUAN

Perkembangan teknologi informasi dan komunikasi telah mengubah cara manusia berinteraksi, bekerja, dan memperoleh informasi. Salah satu manifestasi paling signifikan dari perkembangan ini adalah penggunaan internet yang semakin meluas dan intensif. Internet tidak hanya digunakan untuk komunikasi dan hiburan, tetapi juga menjadi sumber utama informasi, pendidikan, dan bisnis. Dengan meningkatnya penggunaan internet, muncul kebutuhan untuk memahami pola perilaku pengguna dalam memanfaatkan layanan online [1].

Beragamnya platform komunikasi jarak jauh seperti Discord, Google Meet, Skype, dan Zoom telah memberikan pengguna keleluasaan dalam memilih layanan yang paling sesuai dengan kebutuhan mereka. Hal ini tentunya memberikan manfaat berupa ketersediaan fitur dan layanan yang bervariasi.

Awalnya Discord merupakan aplikasi yang sering digunakan oleh para gamer sebagai platform komunikasi, namun seiring berjalannya waktu, Discord telah memperluas basis penggunaannya untuk mencakup beragam komunitas besar. Discord dapat digunakan dalam sistem berbasis Windows, iOS, Android, Linux, bahkan Mac (Rakhmawan et al., 2020). Didirikan sejak tahun 2015 aplikasi ini sudah didownload sebanyak 100 juta lebih unduhan dengan rating 3,5 dan memiliki 5,68 juta ulasan baik ulasan positif, negatif, dan juga netral di *Google Play Store*. Jumlah pengguna Discord diperkirakan akan mencapai 614 juta pada tahun 2024. Ini merupakan peningkatan sebesar 9% atau 51 juta pengguna baru dari tahun 2023. Discord diperkirakan akan memiliki

pertumbuhan pengguna sebesar 12,21% pada tahun 2025 [2].

Namun, seiring dengan pengguna Discord yang kian meningkat perlu diketahui tingkat kepuasan pengguna dan kinerja teknis Discord yang menjadi faktor penting yang perlu diperhatikan untuk meningkatkan mutu layanan yang diberikan.

Discord memiliki 150 juta pengguna aktif bulanan dan 19 juta server aktif perminggu. Jumlah pengguna Discord yang kian bertambah, sangat dipengaruhi oleh persepsi dan ulasan positif dari para pengguna. Namun, tidak dapat dipungkiri bahwa masih ada ulasan negatif yang menunjukkan adanya aspek-aspek tertentu yang perlu diperbaiki. Sebagian besar ulasan pengguna Discord di *Google Play Store* memiliki rating di bawah 5 bintang dan banyak pengguna mengeluhkan adanya bug atau kendala dalam aplikasi. Untuk memahami lebih jauh bagaimana persepsi tersebut memengaruhi kualitas Discord, perlu dilakukan analisis yang mendalam, salah satu solusinya adalah dengan menerapkan analisis sentimen.

Penelitian ini menggunakan algoritma *Naïve Bayes* dan *Support Vector Machine* untuk melakukan klasifikasi pada analisis sentimen ulasan pengguna aplikasi Discord. Pemilihan algoritma *Naïve Bayes* dan SVM ini didasarkan pada penelitian tentang evaluasi kinerja algoritma *Naïve Bayes*, KNN, dan SVM pada analisis sentimen media sosial Twitter [3].

Hasil penelitian tersebut menjelaskan bahwa algoritma SVM memperoleh akurasi tertinggi sebesar 86.7%, diikuti oleh *Naïve Bayes* dengan 82.5%, dan KNN dengan akurasi 78.0%, ini menunjukkan kemampuan SVM dalam menemukan hyperplane yang

memisahkan data secara optimal dimana terbukti pada klasifikasi sentimen dalam teks, khususnya pada ulasan pengguna atau komentar *twitter*.

Mengacu pada penelitian sebelumnya, telah dilakukan analisis sentimen ulasan aplikasi Discord pada artikel yang berjudul “Analisis Sentimen Ulasan Pengguna Aplikasi Discord Menggunakan Metode *K-Nearest Neighbor*” dengan hasil model KNN memperoleh akurasi sebesar 82,76% [4]. Namun penelitian tersebut masih terdapat beberapa keterbatasan, yaitu hanya membandingkan klasifikasi sentimen ke dalam dua kategori, yaitu positif dan negatif, tanpa mempertimbangkan kategori netral. Selain itu, metode yang digunakan masih terbatas pada KNN saja.

Penelitian ini memiliki urgensi yang tinggi mengingat keterbatasan dalam penelitian sebelumnya yang hanya menggunakan metode *K-Nearest Neighbor (KNN)* dengan klasifikasi sentimen terbatas pada dua kategori sentimen. Penelitian ini bertujuan untuk memberikan hasil yang lebih komprehensif, serta menyediakan analisis perbandingan sentimen pada ulasan pengguna aplikasi Discord menggunakan algoritma *Naive Bayes* dan *Support Vector Machine* juga memberikan kontribusi akademik dalam pengembangan metode analisis sentimen berbasis *machine learning*.

2. TINJAUAN PUSTAKA

2.1. Literatur Review

Pada penelitian terdahulu yang berjudul “Analisis Sentimen Review Pengguna Aplikasi Fizzo Novel Di Google Play Menggunakan Algoritma *Naive Bayes*”, bertujuan untuk mengklasifikasikan sentimen pengguna aplikasi Fizzo dan meningkatkan pengalaman pengguna pada *platform* membaca digital. Hasil yang diperoleh yaitu model *Naive Bayes* mencapai akurasi 83%, *precision* 85%, *recall* 82% serta *F1 score* mencapai 84% [5].

Penelitian kedua yang berjudul “Analisis Sentimen Terhadap Layanan Aplikasi Grab Indonesia Menggunakan Metode *Naive Bayes*”, bertujuan untuk mengklasifikasikan umpan balik pengguna dan meningkatkan layanan Grab berdasarkan umpan balik pengguna. Hasil yang diperoleh yaitu model *Naive Bayes* yang digunakan mendapatkan hasil akurasi sebesar 84,36%, *presisi* 99,64%, dan *recall* sebesar 87,12% [6].

Penelitian ketiga yang berjudul “Analisis Sentimen Ulasan Aplikasi Shazam Di Google Play Store Menggunakan *Support Vector Machine*”, bertujuan untuk mengklasifikasikan ulasan tersebut menjadi dua kategori, positif serta negatif. Hasil yang diperoleh yaitu model *Support Vector Machine* memiliki akurasi 84%, serta terdapat lima kali lebih banyak ulasan positif (167 ulasan) dibandingkan ulasan negatif (33 ulasan) [7].

Penelitian keempat yang berjudul “Implementasi Algoritma *Naive Bayes* dan *Support Vector Machine* pada Analisis Sentimen Ulasan Aplikasi Canva”

bertujuan untuk memahami faktor-faktor yang memerlukan perbaikan serta peningkatan. Hasil yang diperoleh yaitu model *Support Vector Machine* yang digunakan mendapatkan hasil akurasi sebesar 83%, *presisi* 76,32%, dan *recall* sebesar 94,69%, dan *F1 score* mencapai 84,40%. Sedangkan *Naive Bayes* hanya memperoleh akurasi sebesar 78,20% [8].

Penelitian kelima yang berjudul “Analisis Sentimen Ulasan Aplikasi Jamsostek Mobile Menggunakan Metode *Support Vector Machine*” bertujuan untuk memberikan wawasan bagi calon pengguna dalam proses pengambilan keputusan. Hasil yang diperoleh yaitu model SVM mencapai tingkat akurasi yang tinggi yaitu 96%.

2.2. Analisis Sentimen

Analisis sentimen merupakan teknik yang digunakan untuk mengevaluasi persepsi masyarakat terhadap suatu entitas. Analisis sentimen digunakan dalam berbagai bidang seperti pemasaran, politik, dan perusahaan untuk mengevaluasi sikap dan emosi masyarakat terhadap produk, layanan, organisasi, individu, masalah, peristiwa, topik, dan atribut lainnya [9].

2.3. Algoritma *Naive Bayes*

Naive Bayes merupakan metode *machine learning* yang memanfaatkan perhitungan probabilitas dan statistik yang diperkenalkan oleh ilmuwan Inggris bernama Thomas Bayes. Metode ini bekerja dengan mengestimasi probabilitas di masa yang akan datang berdasarkan pengalaman di masa yang sudah terjadi [10]. Teorema Bayes memiliki bentuk umum sebagai berikut :

$$P(x|y) = \frac{P(y|x) \cdot P(x)}{P(y)}$$

Keterangan :

$P(x | y) \rightarrow$ Probabilitas kejadian x terjadi jika y sudah terjadi (probabilitas posterior).

$P(y | x) \rightarrow$ Probabilitas y terjadi jika x terjadi (probabilitas likelihood).

$P(x) \rightarrow$ Probabilitas awal terjadinya x (probabilitas prior).

$P(y) \rightarrow$ Probabilitas total terjadinya y (normalisasi).

2.4. Algoritma *Support Vector Machine*

Support Vector Machine (SVM) adalah salah satu teknik dalam *machine learning* yang berdasarkan teori struktural pembelajaran statistik. SVM mempunyai performa yang lebih baik apabila dibandingkan dengan metode *machine learning* lain [11].

SVM memiliki konsep sentral, yaitu menentukan *hyperplane* mana yang terbaik untuk menyediakan ruang atau pemisahan antara dua grup yang telah diprogram sebelumnya [12].

$$K(w, b) = w^T x + b$$

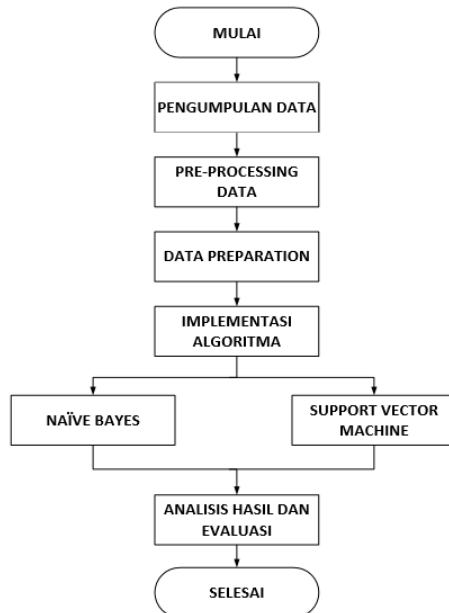
$K(w, b)$ merepresentasikan fungsi kernel dengan dua parameter, yaitu w dan b . Dimana w adalah vektor

bobot yang didapatkan selama proses pelatihan model sedangkan b adalah bias, yang berfungsi untuk menggeser hyperplane. $w^T x$ menunjukkan operasi dot product antara vektor bobot w dan vektor input x .

3. METODE PENELITIAN

3.1. Tahapan Penelitian

Pada gambar 1 menunjukkan penelitian ini dibagi menjadi beberapa tahapan yaitu :



Gambar 1. Tahapan penelitian

Pada gambar 1 menunjukkan penelitian ini dibagi menjadi beberapa tahapan yaitu :

- a. Pengumpulan Data
Mengumpulkan ulasan pengguna Discord di Google Play Store, menggunakan teknik *web scrapping* dengan *library google-play-scrapper*.
- b. *Pre-processing* Data
Melakukan prosedur pra-pemrosesan data pada kolom *content*, dimana data mentah diolah agar dapat diproses lebih lanjut oleh algoritma.
- c. *Data Preparation*
Tahap ini terbagi menjadi dua proses yaitu *translating* dan *labeling*. Proses *labeling* menggunakan *Textblob* berdasarkan skor polaritas. *TextBlob* menggunakan model leksikal yang dibangun berdasarkan dataset bahasa Inggris, seperti *SentiWordNet* untuk analisis sentimen. Oleh karena itu, menerjemahkan data ke bahasa Inggris diperlukan sebelum proses *labeling* sentimen dilakukan.
- d. Implementasi Algoritma
Sebelum algoritma diimplementasikan, data di *splitting* terlebih dahulu menjadi data *training* dan data *testing*, kemudian diubah menjadi fitur numerik menggunakan *library scikit-learn* yaitu *TfidfVectorizer* agar dapat di proses lebih lanjut.
- e. Analisis Hasil dan Evaluasi

Kinerja algoritma *Naïve Bayes* dan *SVM* dievaluasi menggunakan *confusion matrix*, dengan tolok ukur *accuracy*, presisi, *recall*, dan *F1-score* kemudian dibandingkan performanya.

3.2. Confusion Matrix

Tolok ukur yang digunakan untuk mengukur performa adalah *accuracy*, *precision*, *recall*, dan *f1-score*. Tolok ukur tersebut membutuhkan *confusion matrix* yang terdiri dari *True Positive (TP)*, *True Negative (TN)*, *False Positive (FP)*, dan *False Negative (FN)*. Berikut adalah persamaan rumusnya :

$$Accuracy = \frac{(TP + TN)}{Total\ Data\ Uji}$$

$$Precision = \frac{TP}{(TP + FP)}$$

$$Recall = \frac{TP}{(TP + FN)}$$

$$F1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

4. HASIL DAN PEMBAHASAN

4.1. Pengumpulan Data

[illegible]

Gambar 2. Hasil web scrapping

Gambar 2 diatas menunjukkan data yang berhasil diperoleh berjumlah 5000 ulasan pengguna selama periode bulan Mei – November 2024, data kemudian di sortir berdasarkan ulasan paling baru, dan difilter kolom yang relevan saja

4.2. Pre-processing Data

4.2.1. Data Cleaning

Tahapan dimulai dengan data cleaning, yaitu menghilangkan karakter yang tidak diperlukan seperti URL, hashtag (#), mention, emoji.

[illegible]

Gambar 3. Hasil data cleaning

Pada gambar 3 diatas menunjukkan ulasan asli yang mengandung emoji, seperti pada baris kedua, telah dihapus pada hasil cleaning, sehingga hanya menyisakan teks. Setelah proses data cleaning selesai dilakukan, jumlah keseluruhan data menjadi 4443.

4.2.2. Normalisasi

Normalisasi dilakukan untuk menghilangkan atau mengganti variasi kata seperti singkatan, ejaan tidak baku, atau penggunaan slang.

cleaning	normalization
aplikasi II mau daftar aja susah	aplikasi II mau daftar aja susah
asjaha amam maaf aaaaaku sjshsh setres	asjaha amam maaf aaaaaku sjshsh setres
aplikasi gak jelas mau nautin nomor tapi gak n...	aplikasi tidak jelas mau nautin nomor tapi tid...
saya seru main sambil ngobrol	saya seru main sambil ngobrol
ya bagus bagus aja sih	ya bagus bagus ajaah sih
...	...
banyak minta nya cuma masuk doang ke aplikasin...	banyak minta nya cuma masuk doang ke aplikasin...
loading terlalu lama aku jadi menghapus nya saja	loading terlalu lama aku jadi menghapus nya sajaa
heran ini app buat komunikasi tapi gak bisa ma...	heran ini aplikasi buat komunikasi tapi tidak ...
aplikasi ampas anj gw saranin mending jngn dow...	aplikasi ampas jelek gw saranin mending jngn d...
dc skrang makin berat di hp lowend diu aman aj...	dc skrang makin berat di hp lowend diu aman aj...

Gambar 4. Hasil normalisasi

Pada gambar 4 diatas menunjukkan dalam proses tersebut kata-kata tidak baku telah dinormalisasi menjadi kata-kata yang sesuai.

4.2.3. Stopword Removal

Stopword Removal dilakukan untuk menghapus kata-kata yang tidak memiliki kontribusi penting, seperti "dan", "di", dan lainnya.

cleaning	normalization	stopwords
aplikasi II mau daftar aja susah	aplikasi II mau daftar aja susah	II daftar aja susah
asjaha amam maaf aaaaaku sjshsh setres	asjaha amam maaf aaaaaku sjshsh setres	asjaha amam maaf aaaaaku sjshsh setres
aplikasi gak jelas mau nautin nomor tapi gak n...	aplikasi tidak jelas mau nautin nomor tapi tid...	tidak jelas nautin nomor tapi tidak nerima kod...
saya seru main sambil ngobrol	saya seru main sambil ngobrol	seru main sambil ngobrol
ya bagus bagus aja sih	ya bagus bagus ajaah sih	ya bagus bagus ajaah
...	...	...
banyak minta nya cuma masuk doang ke aplikasin...	banyak minta nya cuma masuk doang ke aplikasin...	minta cuma masuk doang aplikasinya ribet minta...
loading terlalu lama aku jadi menghapus nya saja	loading terlalu lama aku jadi menghapus nya sajaa	loading lama jadi hapus sajaa
heran ini app buat komunikasi tapi gak bisa ma...	heran ini aplikasi buat komunikasi tapi tidak ...	heran aplikasi komunikasi tapi tidak bisa masu...
aplikasi ampas anj gw saranin mending jngn dow...	aplikasi ampas jelek gw saranin mending jngn download n...	ampas jelek gw saranin mending jngn download n...
dc skrang makin berat di hp lowend diu aman aj...	dc skrang makin berat di hp lowend diu aman aj...	dc skrang makin berat hp lowend diu aman aja...

Gambar 5. Hasil stopwords removal

Gambar 5 diatas menunjukkan dalam proses tersebut kata-kata yang tidak memiliki kontribusi penting dihapus. Sebagai contoh, pada baris ketiga dan kesembilan, kata "aplikasi" di kolom stopwords telah dihapus. Selain itu, kata "saya" di baris keempat juga dihapus, dan kata "aku" dibaris ketujuh dihapus juga.

4.2.4. Tokenizing

Proses tokenizing ini bertujuan untuk memecah sebuah kalimat menjadi kata – kata untuk analisis teks lebih lanjut.

Gambar 6 diatas menunjukkan pada kolom *tokenized* di mana setiap ulasan yang telah dinormalisasi dipecah menjadi unit-unit yang lebih kecil, yaitu token-token kata. Sebagai contoh, pada ulasan keempat yang awalnya adalah "seru main sambil ngobrol", setelah melalui proses tokenisasi, teks tersebut dipecah menjadi token-token seperti ["seru", "main", "sambil", "ngobrol"].

cleaning	normalization	stopwords	tokenized
aplikasi II mau daftar aja susah	aplikasi II mau daftar aja susah	II daftar aja susah	[II, daftar, aja, susah]
asjaha amam maaf aaaaaku sjshsh setres	asjaha amam maaf aaaaaku sjshsh setres	asjaha amam maaf aaaaaku sjshsh setres	[asjaha, amam, maaf, aaaaaku, sjshsh, setres]
aplikasi gak jelas mau nautin nomor tapi gak n...	aplikasi tidak jelas mau nautin nomor tapi tid...	tidak jelas nautin nomor tapi tidak nerima kod...	[tidak, jelas, nautin, nomor, tapi, tidak, ner...
saya seru main sambil ngobrol	saya seru main sambil ngobrol	seru main sambil ngobrol	[seru, main, sambil, ngobrol]
ya bagus bagus aja sih	ya bagus bagus ajaah sih	ya bagus bagus ajaah	[ya, bagus, bagus, ajaah]
...	...	...	...
banyak minta nya cuma masuk doang ke aplikasin...	banyak minta nya cuma masuk doang ke aplikasin...	minta cuma masuk doang aplikasinya ribet minta...	[minta, cuma, masuk, doang, aplikasinya, ribet...
loading terlalu lama aku jadi menghapus nya saja	loading terlalu lama aku jadi menghapus nya sajaa	loading lama jadi hapus sajaa	[loading, lama, jad, menghapus, sajaa]
heran ini app buat komunikasi tapi gak bisa ma...	heran ini aplikasi buat komunikasi tapi tidak ...	heran aplikasi komunikasi tapi tidak bisa masu...	[heran, aplikasi, komunikasi, tapi, tidak, bis...
aplikasi ampas anj gw saranin mending jngn dow...	aplikasi ampas jelek gw saranin mending jngn download n...	ampas jelek gw saranin mending jngn download n...	[ampas, jelek, gw, saranin, mending, jngn, dow...
dc skrang makin berat di hp lowend diu aman aj...	dc skrang makin berat di hp lowend diu aman aj...	dc skrang makin berat hp lowend diu aman aja...	[dc, skrang, makin, berat, hp, lowend, diu, aman...

Gambar 6. Hasil tokenizing

4.2.5. Stemming

Pre-processing diakhiri dengan proses *stemming*, proses ini menghilangkan imbuhan atau akhiran kata sehingga hanya menyisakan kata dasar atau bentuk dasarnya.

normalization	stopwords	tokenized	stemming
aplikasi II mau daftar aja susah	II daftar aja susah	[II, 'daftar', 'aja', 'susah']	II daftar aja susah
asjaha amam maaf aaaaaku sjshsh setres	asjaha amam maaf aaaaaku sjshsh setres	[asjaha, 'amam', 'maaf', 'aaaaaku', 'sjshsh']	asjaha amam maaf aaaaaku sjshsh tres
aplikasi tidak jelas mau nautin nomor tapi tid...	tidak jelas nautin nomor tapi tidak nerima kod...	['tidak', 'jelas', 'nautin', 'nomor', 'tapi', ...]	tidak jelas nautin nomor tapi tidak nerima kod...
saya seru main sambil ngobrol	seru main sambil ngobrol	['seru', 'mai', 'sambil', 'ngobrol']	seru main sambil ngobrol
ya bagus bagus ajaah sih	ya bagus bagus ajaah	['ya', 'bagus', 'bagus', 'ajaah']	ya bagus bagus ajaah
...	...	...	...
banyak minta nya cuma masuk doang ke aplikasin...	minta cuma masuk doang aplikasinya ribet minta...	['minta', 'cuma', 'masuk', 'doang', 'aplikasin...']	minta cuma masuk doang aplikasi ribet minta ampun
loading terlalu lama aku jadi menghapus nya sajaa	loading lama jadi hapus sajaa	['loading', 'lama', 'jadi', 'menghapus', 'sajaa']	loading lama jadi hapus sajaa
heran ini aplikasi buat komunikasi tapi tidak ...	heran aplikasi komunikasi tapi tidak bisa masu...	['heran', 'aplikasi', 'komunikasi', 'tapi', 'L...']	heran aplikasi komunikasi tapi tidak bisa masu...
aplikasi ampas jelek gw saranin mending jngn dow...	ampas jelek gw saranin mending jngn download n...	['ampas', 'jelek', 'gw', 'saranin', 'mending...']	ampas jelek gw saranin mending jngn download n...
dc skrang makin berat di hp lowend diu aman aj...	dc skrang makin berat hp lowend diu aman aja...	['dc', 'skrang', 'makin', 'berat', 'hp', 'lowe...']	dc skrang makin berat hp lowend diu aman aja...

Gambar 7. Hasil stemming

Gambar 7 diatas menunjukkan pada kolom *stemming* di mana setiap ulasan dihilangkan imbuhan atau akhiran kata sehingga menyisakan kata dalam bentuk dasarnya. Sebagai contoh, pada ulasan keenam yang awalnya adalah kata "menghapus", setelah melalui proses stemming, teks tersebut menjadi kata "hapus".

4.3. Data Preparation

4.3.1. Translating

normalization	stopwords	tokenized	stemming	translate
login pake number dan lupa kata sandi tapi pas...	login pake number lupa kata sandi tapi pas dis...	['login', 'pake', 'number', 'lupa', 'kata', 'sandi', 'tapi', 'pas', 'sur...']	login pake number lupa kata sandi tapi pas sur...	login using number forgot password but when sur...
tidak bisa layar penuh	tidak bisa layar penuh	['tidak', 'bisa', 'layar', 'penuh']	tidak bisa layar penuh	can't full screen
sangat bagus	sangat bagus	['sangat', 'bagus']	sangat bagus	very good
siapa yang download discord hanya download pzy fusi...	siapa download discord hanya download pzy fusi...	['siapa', 'download', 'discord', 'hanya', 'dow...']	siapa download discord hanya download pzy fusi...	who download discord just download pzy fusion ...
yes free	yes free	['yes', 'free']	yes free	yes free
...	...	...	...	...
terimakasih atas nya sangat membantu dalam ber...	terimakasih atas nya sangat membantu dalam ber...	['terimakasih', 'atas', 'nya', 'sangat', 'memb...']	terimakasih atas nya sangat bantu dalam main g...	Thank you for it, it really helps in playing g...
please fix this problem i cant add my phone nu...	please fix this problem i cant add my phone nu...	['please', 'fix', 'this', 'problem', 'i', 'cant', 'add', 'my', 'phone', 'nu...']	please fix this problem i cant add my phone nu...	please fix this problem i cant add my phone nu...
orang gila mana sih chat pake ginian	orang gila mana chat pake ginian	['orang', 'gila', 'mana', 'chat', 'pake', 'gin...']	orang gila mana chat pake ginian	crazy people who chat using this
berat sekali sekarang dc	berat sekali dc	['berat', 'sekali', 'dc']	berat sekali dc	very heavy
kisamaas login aja susah pea	kisamaas login susah pea	['kisamaas', 'login', 'susah', 'pea']	kisamaas login susah pea	kisamaas login is hard pea

Gambar 8. Hasil translating

Pada gambar 8 diatas menunjukkan pada kolom *translate* di mana setiap ulasan pada kolom stemming sudah diterjemahkan dari bahasa Indonesia menjadi bahasa Inggris.

4.3.2. Labeling

Setelah proses *translating* selesai dilanjutkan dengan pelabelan sentimen dengan menggunakan *Textblob* yang terdiri dari tiga kelas sentimen yaitu positif, netral, dan negatif.

	stemming	translate	sentimen	polaritas
0	il daftar ajaa susah	il just register it's hard	Negatif	-0.291667
1	ajajha amam maaf aaaaak sjsah tres	oh my gosh sorry aaaaak sjsah tres	Negatif	-0.500000
2	tidak jelas nautin nomor tapi tidak nerima kod...	it's not clear, i'm navigating the number but ...	Positif	0.066667
3	seru main sambil ngobrol	fun to play while chatting	Positif	0.300000
4	ya bagus bagus ajaa	yeah, it's good, it's good	Positif	0.700000
...	...	...	...	...
4438	minta cuma masuk doang aplikasi ribet minta ampun	just ask to enter the application, it's so com...	Negatif	-0.350000
4439	loading lama jadi hapus saja	long loading so just delete it	Negatif	-0.050000
4440	heran aplikasi komunikasi tapi tidak bisa masu...	i'm surprised the communication application ca...	Positif	0.100000
4441	ampas jelek gw saranin mending jgn download n...	i suggest you don't download this, it's bad, i...	Negatif	-0.450000
4442	dc skrang makin berat hp lowend diu aman ajaa ...	it's getting heavier now that lowend phones ar...	Positif	0.054167

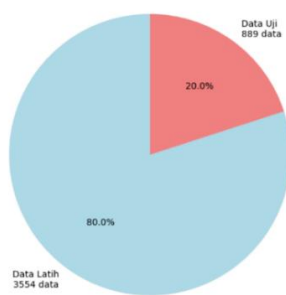
Gambar 9. Hasil labeling

Gambar 9 diatas menunjukkan hasil dari labeling dimana sentimen ditentukan berdasarkan polaritas. Dengan ketentuan jika nilai polaritas (sentimen) lebih besar dari 0.0, maka teks dikategorikan sebagai sentimen Positif apabila polaritas (sentimen) kurang dari 0.0, maka teks dikategorikan sebagai sentimen Negatif, sedangkan jika nilai polaritas (sentimen) sama dengan 0.0, maka teks dikategorikan sebagai sentimen Netral.

4.3.3. Splitting Data

Data yang berjumlah 4443 terlebih dahulu dibagi menjadi data *training* dan data *testing* dengan rasio 80% / 20%, kemudian diubah menjadi fitur numerik menggunakan *TfidfVectorizer*.

Pembagian Dataset menjadi Data Latih dan Data Uji



Gambar 10. Splitting data

Gambar 10 diatas menunjukkan 3554 (80%) data digunakan untuk data latih, dan 889 (20%) data yang tersisa digunakan sebagai data uji.

4.4. Implementasi Algoritma

4.4.1. Klasifikasi Naïve Bayes

Proses selanjutnya yaitu membuat model klasifikasi menggunakan algoritma *Naïve Bayes Multinomial*. Implementasi model algoritma *Naïve Bayes Multinomial* dapat dilihat pada gambar 11.

```
# Membangun Model NB
from sklearn.naive_bayes import MultinomialNB

# Membangun Model Naive Bayes Classifier
naive_bayes = MultinomialNB()
naive_bayes.fit(X_train_vec, y_train)
y_pred = naive_bayes.predict(X_test_vec)

# Laporan Klasifikasi
accuracy = accuracy_score(y_test, y_pred)
label_names = ['Positif', 'Netral', 'Negatif']
cmb = confusion_matrix(y_test, y_pred)
report = classification_report(y_test, y_pred, target_names=label_names)
```

Gambar 11 Model Naïve Bayes Multinomial

Pada Gambar 11 menunjukkan model ini diimpor dari pustaka *sklearn.naive_bayes* dan kemudian diinisialisasi menggunakan *MultinomialNB()*. Proses pelatihan dilakukan dengan memanfaatkan data latih (*X_train_vec*, *y_train*) menggunakan metode *.fit()*, sedangkan prediksi sentimen terhadap data uji (*X_test_vec*) dilakukan melalui metode *.predict()*, menghasilkan label prediksi *y_pred*. Hasil dari *classification report* dapat dilihat pada gambar 12.

```
=== Multinomial Naive Bayes ===
Classification Report :
```

	precision	recall	f1-score	support
Positif	0.75	0.83	0.79	323
Netral	0.86	0.73	0.79	313
Negatif	0.75	0.79	0.77	253
accuracy			0.78	889
macro avg	0.78	0.78	0.78	889
weighted avg	0.79	0.78	0.78	889

Gambar 12. Classification report NB

Gambar 12 diatas menunjukkan bahwa model memiliki akurasi sebesar 78. Pada kelas Positif, *precision* yang diperoleh yaitu 0.75, *recall* 0.83, dan *f1 score* 0.79. Sementara itu, pada kelas Netral, *precision* mencapai 0.86, *recall* 0.73, dan *F1-score* 0.79, yang mana model cukup ketat dalam memastikan suatu sampel termasuk dalam kategori netral, tetapi sering kali gagal mendeteksi beberapa sampel yang seharusnya dikategorikan sebagai netral. Pada kelas Negatif, *precision* tercatat 0.75, *recall* 0.79, dan *F1-score* 0.77, yang menunjukkan keseimbangan antara *precision* dan *recall* dalam mendeteksi sentimen negatif. Secara keseluruhan, model menunjukkan kinerja yang cukup baik, dengan nilai *macro average* dan *weighted average* sebesar 0.78.

4.4.2. Klasifikasi Support Vector Machine

Model dengan penerapan algoritma SVM digunakan sebagai komparasi atau pembandingan dengan model *Naïve Bayes Multinomial*. Implementasi model algoritma SVM dapat dilihat pada gambar 13.

```
# Implementasi Model SVM dengan Optimalisasi Hyperparameter Tuning

from sklearn.svm import SVC
from sklearn.model_selection import GridSearchCV

# Tentukan Parameter Grid untuk Hyperparameter Tuning
param_grid = {
    'C': [0.1, 1, 10, 100],
    'gamma': [1, 0.1, 0.01, 0.001],
    'kernel': ['rbf', 'linear'],
}

# Membuat Model SVM classifier
svm = SVC(probability=True)

# Melakukan GridSearchCV untuk menemukan hyperparameters terbaik
grid_search_svm = GridSearchCV(svm, param_grid, cv=10, scoring='accuracy')
grid_search_svm.fit(X_train_vec, y_train)

# Hyperparameters terbaik dan accuracy score
print("Best Hyperparameters :", grid_search_svm.best_params_)

# Mengevaluasi Model terbaik pada Test Set
best_svm = grid_search_svm.best_estimator_
y_pred = best_svm.predict(X_test_vec)
label_names = ["Positif", "Netral", "Negatif"]
accuracy = accuracy_score(y_test, y_pred)
cmsvm = confusion_matrix(y_test, y_pred)

# Laporan Klasifikasi
report = classification_report(y_test, y_pred, target_names=label_names)
```

Gambar 13 Model *Support Vector Machine*

Pada Gambar 13 menunjukkan model klasifikasi SVM dengan optimasi hyperparameter menggunakan metode GridSearchCV. Hyperparameter yang dioptimalkan meliputi nilai C, gamma, dan jenis kernel. Parameter C diuji pada nilai [0.1, 1, 10, 100], parameter gamma pada [1, 0.1, 0.01, 0.001], serta kernel dengan opsi ['rbf', 'linear']. Proses tuning dilakukan menggunakan cross-validation sebanyak 10 kali lipatan (10-fold cross-validation). Setelah menemukan kombinasi hyperparameter terbaik, model SVM terbaik dilatih kembali pada data latih (training set) dan diuji pada data uji (test set). Hasil dari classification report dapat dilihat pada gambar 14.

```
Best Hyperparameters : {'C': 10, 'gamma': 0.1, 'kernel': 'rbf'}

=== Support Vector Machine ===

Classification Report :


```

	precision	recall	f1-score	support
Positif	0.89	0.86	0.87	323
Netral	0.88	0.93	0.90	313
Negatif	0.87	0.83	0.85	253
accuracy			0.88	889
macro avg	0.88	0.88	0.88	889
weighted avg	0.88	0.88	0.88	889

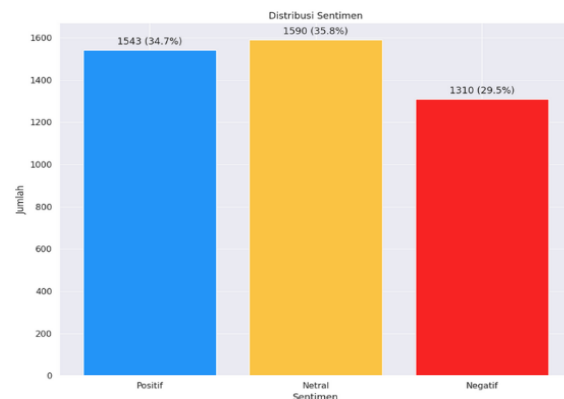
Gambar 14. Classification report SVM

Pada gambar 14 diatas menunjukkan, telah diperoleh *hyperparameter* terbaik yaitu dengan nilai (C: 10, gamma: 0.1, kernel: 'rbf'). Berdasarkan *classification report*, nilai akurasi keseluruhan mencapai 88%, Secara rinci, untuk sentimen positif, memperoleh nilai *precision* sebesar 89%, *recall* 86%, dan *f1-score* 87%. Sentimen netral menunjukkan performa terbaik dengan *precision* 88%, *recall* 93%, dan *f1-score* 90%. Sementara itu, sentimen negatif memiliki performa yang sedikit lebih rendah dengan *precision* 87%, *recall* 83%, dan *f1-score* 85%. Nilai rata-rata *precision*, *recall*, dan *f1-score* masing-masing adalah 0,88 untuk *macro average* maupun *weighted*

average, mengindikasikan konsistensi performa pada seluruh kelas.

4.5. Analisis Hasil dan Evaluasi

4.5.1. Distribusi Sentimen



Gambar 15. Distribusi Sentimen

Gambar 15 diatas menunjukkan hasil klasifikasi sentimen setelah proses labeling dilakukan, sebagian besar data termasuk dalam kategori sentimen netral, yaitu sebanyak 1.590 data (35,8%) dari total keseluruhan. Sentimen positif berada pada posisi kedua dengan jumlah 1.543 data (34,7%). Sedangkan, sentimen negatif memiliki jumlah yang paling rendah, yakni sebanyak 1.310 data (29,5%). Hal ini mengindikasikan bahwa persepsi dan kepuasan pengguna terhadap aplikasi Discord cenderung didominasi oleh tanggapan netral, diikuti oleh tanggapan positif, dan diakhiri oleh tanggapan negatif.

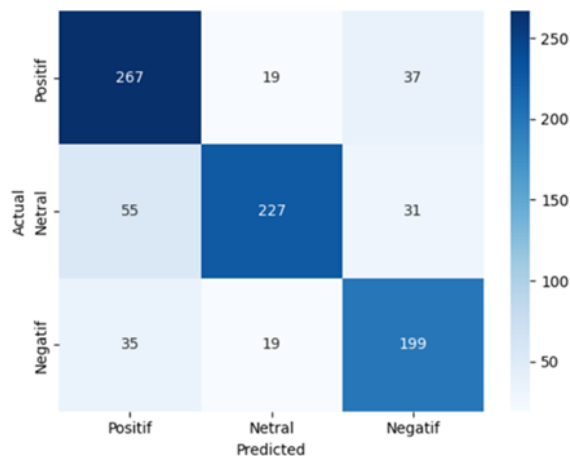
4.5.2. Wordcloud Modus Kata



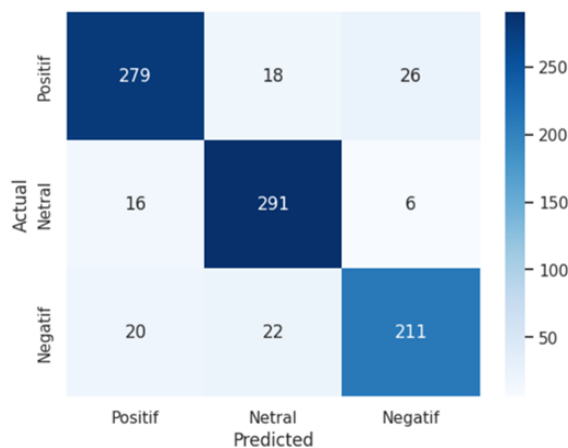
Gambar 16. Modus Kata

Pada gambar 16 diatas menunjukkan visualisasi wordcloud modus kata pada dataset, kata-kata yang sering muncul digambarkan dengan ukuran yang lebih besar, sedangkan kata-kata yang frekuensinya lebih rendah digambarkan dengan ukuran lebih kecil. Berdasarkan visualisasi, kata-kata seperti "tidak", "bisa", "discord", "login", dan "tolong" tampak dominan. Hal ini mengindikasikan bahwa permasalahan login pada aplikasi Discord menjadi topik yang sering dibahas dalam ulasan pengguna.

4.5.3. Evaluasi Kinerja Naïve Bayes



Gambar 17. Confussion Matrix NB



Gambar 18. Confussion Matrix SVM

Pada gambar 17 diatas menunjukkan *confussion matrix* memiliki 3 kelas sentimen yaitu positif, netral, dan negatif, dengan perhitungan sebagai berikut :

$$\begin{aligned}
 \text{Akurasi} &= \frac{267 + 227 + 199}{889} = 0,7796 \approx 0,78 \\
 \text{Presisi (Positif)} &= \frac{267}{267 + 35 + 55} = 0,7486 \approx 0,75 \\
 \text{Recall (Positif)} &= \frac{267}{267 + 19 + 37} = 0,8260 \approx 0,83 \\
 \text{F1 - Score (Positif)} &= 2 \times \frac{0,75 \times 0,83}{0,75 + 0,83} = 0,79 \\
 \text{Presisi (Netral)} &= \frac{227}{227 + 19 + 19} = 0,8566 \approx 0,86 \\
 \text{Recall (Netral)} &= \frac{227}{227 + 55 + 31} = 0,7259 \approx 0,73 \\
 \text{F1 - Score (Netral)} &= 2 \times \frac{0,86 \times 0,73}{0,86 + 0,73} = 0,79 \\
 \text{Presisi (Negatif)} &= \frac{199}{199 + 37 + 31} = 0,7453 \approx 0,75 \\
 \text{Recall (Negatif)} &= \frac{199}{199 + 19 + 35} = 0,7862 \approx 0,79 \\
 \text{F1 - Score (Negatif)} &= 2 \times \frac{0,75 \times 0,79}{0,75 + 0,79} = 0,77
 \end{aligned}$$

4.5.4. Evaluasi Kinerja Support Vector Machine

Pada gambar 18 diatas menunjukkan *confussion matrix* memiliki 3 kelas sentimen yaitu positif, netral, dan negatif, dengan perhitungan sebagai berikut :

$$\begin{aligned}
 \text{Akurasi} &= \frac{279 + 291 + 211}{889} = 0,8786 \approx 0,88 \\
 \text{Presisi (Positif)} &= \frac{279}{279 + 16 + 20} = 0,8857 \approx 0,89 \\
 \text{Recall (Positif)} &= \frac{279}{279 + 18 + 26} = 0,8637 \approx 0,86 \\
 \text{F1 - Score (Positif)} &= 2 \times \frac{0,89 \times 0,86}{0,89 + 0,86} = 0,87 \\
 \text{Presisi (Netral)} &= \frac{291}{291 + 18 + 22} = 0,8791 \approx 0,88 \\
 \text{Recall (Netral)} &= \frac{291}{291 + 16 + 6} = 0,9297 \approx 0,93 \\
 \text{F1 - Score (Netral)} &= 2 \times \frac{0,88 \times 0,93}{0,88 + 0,93} = 0,90 \\
 \text{Presisi (Negatif)} &= \frac{211}{211 + 26 + 6} = 0,8683 \approx 0,87 \\
 \text{Recall (Negatif)} &= \frac{211}{211 + 20 + 22} = 0,8339 \approx 0,83 \\
 \text{F1 - Score (Negatif)} &= 2 \times \frac{0,87 \times 0,83}{0,87 + 0,83} = 0,85
 \end{aligned}$$

5. KESIMPULAN DAN SARAN

Hasil penelitian ini menunjukkan sebagian besar data termasuk dalam kategori sentimen netral, yaitu sebanyak 1.590 data (35,8%) dari total keseluruhan. Sentimen positif berada pada posisi kedua dengan jumlah 1.543 data (34,7%). Sedangkan, sentimen negatif memiliki jumlah yang paling rendah, yakni sebanyak 1.310 data (29,5%). Hal ini mengindikasikan bahwa persepsi dan kepuasan pengguna terhadap aplikasi Discord cenderung didominasi oleh tanggapan netral, diikuti oleh tanggapan positif, dan diakhiri oleh tanggapan negatif. Visualisasi wordcloud menunjukkan, kata-kata seperti "tidak", "bisa", "discord", "login", dan "tolong" tampak dominan. Hal ini menggambarkan bahwa permasalahan login pada aplikasi Discord menjadi topik yang sering dibahas dalam ulasan pengguna. Kata-kata lain seperti "verifikasi", "email", "password", "masuk", dan "sudah" juga menunjukkan adanya kendala teknis terkait proses autentikasi pengguna. Secara keseluruhan, model SVM memiliki kinerja yang lebih baik dibandingkan Naive Bayes, dengan nilai akurasi keseluruhan mencapai 88%.

Sementara itu Naive Bayes memperoleh akurasi keseluruhan sebesar 78%. SVM lebih unggul dalam menangani kelas netral dan lebih sedikit melakukan kesalahan prediksi terhadap kelas positif dan negatif dibandingkan Naive Bayes. Naive Bayes cenderung membuat lebih banyak kesalahan antar kelas, terutama pada kelas positif dan negatif, di mana prediksi yang salah sering terjadi antara kedua kelas tersebut. Adapun beberapa saran untuk penelitian selanjutnya antara lain yaitu jumlah data yang digunakan dapat ditingkatkan dengan memperpanjang periode pengambilan ulasan atau dengan mengombinasikan ulasan dari platform lain seperti App Store atau forum komunitas dan diharapkan untuk menggunakan

algoritma klasifikasi lainnya seperti *deep learning* yaitu *CNN*, *LSTM* atau *Transformer*.

DAFTAR PUSTAKA

- [1] A. S. Sitio and F. A. Sianturi, "Penerapan Algoritma Machine Learning dalam Analisis Pola Perilaku Penggunaan Internet," *DIKE J. Ilmu Multidisiplin*, vol. 2, no. 2, Art. no. 2, Aug. 2024, doi: 10.69688/dike.v2i2.102.
- [2] A. Turner, "Discord Users: How Many People Use Discord (2024)." Accessed: Oct. 06, 2024. [Online]. Available: <https://www.bankmycell.com/blog/number-of-discord-users/>
- [3] H. Purnomo, O. Arifin, A. Arpan, R. Permata, and D. Handoko, "EVALUASI KINERJA ALGORITMA NAÏVE BAYES, KNN, DAN SVM DALAM ANALISIS SENTIMEN MEDIA SOSIAL," vol. 9, no. 2, 2024.
- [4] R. Q. Rohmansa, N. Pratiwi, and M. J. Palepa, "ANALISIS SENTIMEN ULASAN PENGGUNA APLIKASI DISCORD MENGGUNAKAN METODE K-NEAREST NEIGHBOR," *JIPi J. Ilm. Penelit. Dan Pembelajaran Inform.*, vol. 9, no. 1, Art. no. 1, Feb. 2024, doi: 10.29100/jipi.v9i1.4943.
- [5] T. Arlovin, "ANALISIS SENTIMEN REVIEW PENGGUNA APLIKASI FIZZO NOVEL DI GOOGLE PLAY MENGGUNAKAN ALGORITMA NAIVE BAYES," vol. 6, no. 1, 2024.
- [6] A. Rifa'I, R. Ardhani, D. Pratama, and F. Fatihanursari, "ANALISIS SENTIMEN TERHADAP LAYANAN APLIKASI GRAB INDONESIA MENGGUNAKAN METODE NAÏVE BAYES," *JATI J. Mhs. Tek. Inform.*, vol. 8, no. 1, pp. 303–309, Feb. 2024, doi: 10.36040/jati.v8i1.8425.
- [7] A. Wijaya, Meilinda, and M. Maharani, "ANALISIS SENTIMEN ULASAN APLIKASI SHAZAM DI GOOGLE PLAY STORE MENGGUNAKAN SUPPORT VECTOR MACHINE," *ZONAsi J. Sist. Inf.*, vol. 6, no. 1, pp. 197–207, Feb. 2024, doi: 10.31849/zn.v6i1.17994.
- [8] H. Hariyadi, D. Firdo, and M. H. Al Rafi, "Implementasi Algoritma Naïve Bayes dan Support Vector Machine pada Analisis Sentimen Ulasan Aplikasi Canva," *J. Minfo Polgan*, vol. 13, no. 1, pp. 261–269, Mar. 2024, doi: 10.33395/jmp.v13i1.13568.
- [9] 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: Sentiment Analysis of Cash Direct Assistance Distribution for Fuel Oil Using Support Vector Machine," *MALCOM Indones. J. Mach. Learn. Comput. Sci.*, vol. 3, no. 1, pp. 27–35, May 2023, doi: 10.57152/malcom.v3i1.590.
- [10] M. I. Santoso and A. R. Dzikrillah, "Analisis Sentimen Pengguna Terhadap Kinerja Sistem Transportasi Umum Jakarta Menggunakan Algoritma Naive Bayes," *KLIK Kaji. Ilm. Inform. Dan Komput.*, vol. 4, no. 6, Art. no. 6, Jun. 2024, doi: 10.30865/klik.v4i6.1936.
- [11] D. Oktavia, Y. R. Ramadhan, and M. Minarto, "Analisis Sentimen Terhadap Penerapan Sistem E-Tilang Pada Media Sosial Twitter Menggunakan Algoritma Support Vector Machine (SVM)," *KLIK Kaji. Ilm. Inform. Dan Komput.*, vol. 4, no. 1, Art. no. 1, Aug. 2023, doi: 10.30865/klik.v4i1.1040.
- [12] S. Taliki, I. C. R. Drajana, and A. Bode, "SUPPORT VECTOR MACHINE BERBASIS CHI SQUARE UNTUK PREDIKSI HARGA BERAS ECER KABUPATEN POHUWATO," *J. Sci. Soc. Res.*, vol. 5, no. 2, Art. no. 2, Jul. 2022, doi: 10.54314/jsr.v5i2.899.