Jurnal Elektronik Ilmu Komputer Udayana
Volume 13, No 1. August 2024

p-ISSN: 2301-5373
e-ISSN: 2654-5101

Analisis Sentimen Ulasan Aplikasi Solusi Kota Cerdas Menggunakan Algoritma *Naïve Bayes* dan *Support Vector Machine* (SVM) dengan Seleksi Fitur *Chi-Square*

Ni Luh Komang Indira Pramesti[a1], Made Agung Raharja[a2], Ngurah Agus Sanjaya ER[a3], I Gede Arta Wibawa[a4]

[a]Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Udayana
Badung, Bali, Indonesia
[1]indiprames064@student.unud.ac.id
[2]made.agung@unud.ac.id
[3]agus_sanjaya@unud.ac.id
[4]gede.arta@unud.ac.id

Abstrak

Masyarakat yang semakin bergantung dengan teknologi dalam kegiatan sehari-hari menyebabkan banyaknya aplikasi yang hadir dalam membantu kegiatan ini. Salah satunya adalah aplikasi SpeedID yang berfungsi sebagai solusi kota cerdas. Fitur yang dimiliki beragam, mulai dari verifikasi identitas *online*, antrean *online*, manajemen usaha kuliner, manajemen usaha UKM, dan masih banyak lagi. Popularitas aplikasi ini berujung pada banyaknya ulasan yang diberikan oleh pengguna, baik itu positif, negatif, maupun netral. Dengan demikian, perlu dilakukan suatu analisis sentimen ulasan guna mengetahui pandangan pengguna terhadap aplikasi. Metode klasifikasi sentimen yang digunakan adalah *Naïve Bayes* (NB) dan *Support Vector Machine* (SVM) dengan seleksi fitur *chi-square*. Hasil evaluasi model menunjukkan bahwa seleksi fitur *chi-square* memiliki pengaruh positif terhadap performa model NB yang ditandai dengan meningkatnya nilai akurasi hingga sebesar 3,12%. Namun, seleksi fitur *chi-square* ini tidak memiliki pengaruh terhadap performa model SVM yang tidak mengalami peningkatan atau penurunan nilai akurasi saat ditambahkan *chi-square*.

Keywords: *SpeedID, Naïve Bayes (NB), Support Vector Machine (SVM), Chi-Square, Sentiment Analysis*

1. Pendahuluan

Hidup di era digital menyebabkan masyarakat semakin bergantung dengan teknologi dalam kegiatan sehari-hari mereka. Hal ini menjadi suatu peluang bagi perusahaan yang bergerak di bidang teknologi untuk hadir dengan berbagai macam inovasi untuk membantu masyarakat melakukan aktivitas secara daring. Salah satu aplikasi tersebut adalah SpeedID yang dirilis pada tahun 2018 sebagai solusi untuk kota cerdas. Dikutip dari situs resminya, SpeedID memiliki visi untuk menjadi solusi bagi penerapan IT dan identitas digital baru kota cerdas di seluruh dunia. Berbagai fitur yang dimiliki, mulai dari verifikasi identitas *online*, antrian *online*, manajemen usaha kuliner, manajemen usaha UKM, dan masih banyak lagi, membuat semakin banyak orang menggunakan aplikasi ini. Per April 2023, aplikasi ini telah diunduh sebanyak lebih dari 100 ribu kali. Hal ini juga berdampak pada semakin banyaknya ulasan yang diberikan oleh pengguna, baik itu positif, negatif, maupun netral. Dengan demikian, perlu dilakukan suatu analisis sentimen ulasan pengguna terhadap aplikasi ini guna mengetahui seberapa baik performa aplikasi dan apa yang dapat ditingkatkan oleh perusahaan ke depannya. Analisis sentimen adalah metode yang mengelompokkan polaritas dari data tekstual ke dalam kategori positif, netral, dan negatif [1]. Analisis sentimen ini dapat membantu dalam memperoleh wawasan yang berguna dalam pengambilan keputusan terkait pengembangan aplikasi, pemasaran, dan layanan pelanggan. Dengan demikian, perusahaan diharapkan dapat mengidentifikasi kekuatan dan kelemahan aplikasi mereka, serta meningkatkan kepuasan pengguna.

Ada berbagai macam metode yang dapat dilakukan dalam analisis sentimen. Umumnya, metode tersebut dibagi menjadi dua, yaitu pendekatan *supervised learning* dan *lexicon-based*. *Lexicon-based* merupakan pendekatan yang memanfaatkan kamus kata yang telah berisi definisi kata-kata untuk mencari polaritasnya. *Supervised learning* adalah metode yang bergantung kepada data latih yang telah memiliki label atau kelas [2]. Beberapa algoritma yang banyak digunakan dalam analisis sentimen adalah *Naïve Bayes* dan *Support Vector Machine.*

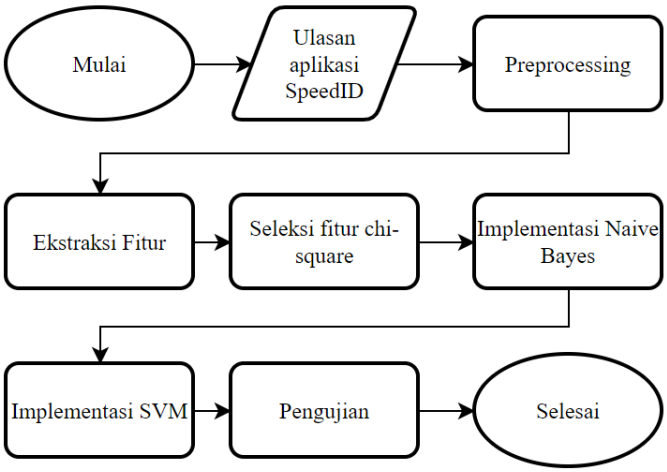
Naïve Bayes merupakan algoritma klasifikasi yang menggolongkan data ke dalam kategori yang tepat berdasarkan probabilitas dari data sebelumnya [3]. Salah satu kelebihan dari metode ini adalah dapat mencapai *accuracy* yang tinggi walaupun data latih yang digunakan sedikit [4]. Selain itu, metode *Naïve Bayes* juga mudah diimplementasikan dan dapat memberikan hasil yang baik pada berbagai kasus. Sedangkan, kekurangannya adalah fitur-fitur bersifat independen dan keterkaitan antar fitur tidak dapat dimodelkan oleh *Naïve Bayes* [5].

Support Vector Machine (SVM), merupakan metode pembelajaran mesin yang memanfaatkan hipotesis dalam bentuk fungsi linier dalam fitur dengan dimensi tinggi yang telah dilatih menggunakan algoritma pembelajaran berdasarkan teori optimisasi [6]. Kelebihan dari metode SVM adalah metode ini dapat menentukan *hyperplane* yang memaksimalkan margin untuk memisahkan kelas yang berbeda [7]. Selain itu, SVM juga mampu memberikan solusi untuk *overfitting* dan *optimal local*, serta memiliki rasio konvergensi yang rendah [8]. SVM juga memiliki kekurangan, yaitu data yang memiliki properti yang sama akan berpengaruh terhadap nilai akurasi secara signifikan [9].

Penelitian ini akan fokus pada analisis sentimen ulasan aplikasi SpeedID pada *Google Play Store* menggunakan metode *Naïve Bayes* dan *Support Vector Machine* (SVM) dengan menerapkan seleksi fitur *chi-square*. Seleksi fitur *chi-square*, merupakan metode yang digunakan untuk menguji ketergantungan dari dua kejadian [10]. Seleksi fitur ini diharapkan dapat mengurangi fitur yang kurang penting dan meningkatkan performa model algoritma dalam memprediksi kelas sentimen. Metode *Naïve Bayes* dan SVM akan dibandingkan untuk mengetahui algoritma mana yang memiliki performa lebih baik dalam klasifikasi sentimen.

2. Metode Penelitian

Tahapan penelitian dimulai dari pengumpulan data yang digunakan (ulasan aplikasi SpeedID), *preprocessing*, ekstraksi fitur, seleksi fitur, implementasi *Naïve Bayes*, implementasi SVM, dan pengujian atau evaluasi seperti yang ditunjukkan pada gambar 1.



Gambar 1. Alur Penelitian

2.1 Pengumpulan Data

Data yang digunakan merupakan ulasan aplikasi solusi kota cerdas, yaitu SpeedID. Data ini diperoleh dari 2 platform, yaitu *Google Play Store* dan *App Store* dengan cara *scraping*. Proses *scraping* dari *Play Store* menghasilkan data berbentuk file .csv yang terdiri dari 557 baris dan 11 kolom, yaitu thumbsUpCount, reviewCreatedVersion, at, replyContent, repliedAt, reviewId, userName, userImage, content, score, dan appVersion. Dari semua kolom tersebut, data yang digunakan adalah kolom content yang mengandung teks ulasan dan kolom score yang digunakan untuk menentukan label data.

Pengumpulan data dari platform *App Store* memperoleh data berupa file .csv yang terdiri dari 112 baris dan 7 kolom, yaitu date, review, rating, isEdited, username, title, dan developerResponse. Data yang digunakan adalah kolom review yang berisi teks ulasan dan rating untuk menentukan label sentimen.

2.2 *Preprocessing* Data

p-ISSN: 2301-5373
e-ISSN: 2654-5101

Sebelum masuk ke tahap implementasi algoritma, data mentah harus melalui tahap *preprocessing* terlebih dahulu. Tujuannya adalah untuk meningkatkan kualitas data dan memudahkan proses lebih lanjut. *Preprocessing* data terdiri dari beberapa bagian seperti berikut.

1. *Data cleansing*: Pembersihan atau penghilangan karakter-karakter tertentu, seperti tanda baca, *username*, atau url. Tahap ini juga meliputi *case folding*, yaitu mengubah semua karakter huruf dalam teks menjadi karakter huruf kecilnya.
2. Normalisasi kata: Mengubah kata-kata yang sebelumnya tidak baku menjadi bentuk kata bakunya.
3. *Stemming*: Mengubah kata-kata dalam teks menjadi bentuk dasar atau kata dasar, seperti mengubah kata "makanan", "makanan-makanan", dan "makan" menjadi kata dasar "makan".
4. *Stopwords removal*: Menghilangkan kata-kata yang umum dan tidak memiliki arti khusus, seperti "dan", "atau", dan "saja".

2.3 TF-IDF

Metode ekstraksi fitur yang digunakan adalah TF-IDF. TF-IDF merupakan metode yang menggabungkan antara *Term Frequency* (TF) dan *Inverse Document Frequency* (IDF). Metode ini akan menghasilkan nilai bobot yang tinggi untuk fitur yang banyak muncul pada suatu dokumen, tetapi jarang muncul pada kumpulan dokumen keseluruhan. Rumus untuk menghitung TF-IDF adalah berikut [11].

$$TF * IDF(d,t) = TF(d,t) * log\ \frac{N}{df(t)} \quad (1)$$

Keterangan:

$TF * IDF(d,t)$ = bobot TF-IDF

$TF(d,t)$ = Frekuensi kemunculan fitur t pada dokumen d

N = Jumlah semua kumpulan dokumen

$df(t)$ = Jumlah dokumen yang mengandung fitur t

2.4 *Chi-Square*

Dalam statistika, *chi-square* sering digunakan untuk menguji ketergantungan antara dua kejadian [12]. Metode *chi-square* juga merupakan salah satu metode yang banyak digunakan dalam melakukan seleksi fitur. Metode ini dikatakan mampu menghapus fitur-fitur yang tidak dibutuhkan tanpa mengurangi nilai *accuracy* yang diperoleh [13]. Rumus *chi-square* dijabarkan sebagai berikut [14].

$$X^2(t,c) = \frac{N(AD-CB)^2}{(A+C)(B+D)(A+B)(C+D)} \quad (2)$$

Keterangan:

X^2 = nilai *chi-square*

t = term atau fitur

c = kelas atau label

N = jumlah semua dokumen

A = jumlah kemunculan t dalam kelas c

B = jumlah kemunculan t dalam kelas selain c

C = jumlah kemunculan kata selain t dalam kelas c

D = jumlah kemunculan kata selain t dalam kelas selain c

2.5 *Naïve Bayes* (NB)

Algoritma *Naïve Bayes* merupakan algoritma klasifikasi yang bisa digunakan untuk memprediksi probabilitas kelas berdasarkan teorema Bayes [10]. Prinsip dasar algoritma *Naïve Bayes* adalah menghitung probabilitas posterior untuk setiap kelas target berdasarkan kemunculan fitur atau atribut dalam dokumen. Probabilitas posterior adalah probabilitas kelas target setelah melihat data. Berikut adalah rumus untuk mencarinya [15].

$$P(A|B) = \frac{P(B|A)\ P(A)}{P(B)} \quad (3)$$

Keterangan:

P(A|B) = probabilitas dari atribut B masuk ke kelas A

P(B|A) = probabilitas dari munculnya atribut B pada kelas A

P(A) = probabilitas dari data yang termasuk kelas A

P(B) = jumlah semua kata yang ada di dataset

P(B|A), disebut juga sebagai probabilitas kondisional, dapat dihitung dengan menggunakan rumus berikut.

$$P(B|A) = P(B_1|A) \times P(B_2|A) \times \ldots \times P(B_d|A) \quad (4)$$

$P(B_1|A)$ merupakan probabilitas dari munculnya fitur B_1 pada kelas A. Probabilitas munculnya fitur tertentu pada suatu kelas A dihitung menggunakan persamaan berikut.

$$P(B_i|A) = \frac{bobot\ TF{-}IDF\ fitur\ B_i\ di\ kelas\ A}{total\ bobot\ TF{-}IDF\ di\ kelas\ A} \quad (5)$$

P(A) atau probabilitas prior dapat diperoleh menggunakan persamaan berikut.

$$P(A) = \frac{jumlah\ dokumen\ pada\ kelas\ A}{jumlah\ semua\ dokumen\ pada\ training\ set} \quad (6)$$

2.6 *Support Vector Machine* (SVM)

SVM (*Support Vector Machine*) adalah algoritma *machine learning* yang digunakan untuk melakukan klasifikasi dan regresi pada data yang berdimensi tinggi. SVM bekerja dengan membuat sebuah *hyperplane* yang memisahkan antara dua kelas data yang berbeda secara maksimal [16]. Algoritma ini dapat digunakan untuk klasifikasi data yang memiliki lebih dari 2 kelas dengan menerapkan strategi *one-vs-all*. Strategi ini memberikan satu *classifier* per kelasnya. Untuk masing-masing *classifier*, satu kelas dipisahkan dengan kelas lainnya [17].

SVM dapat bekerja pada data yang tidak linier dengan memanfaatkan penggunaan kernel. Kernel adalah fungsi transformasi yang mengubah data ke dalam dimensi yang lebih tinggi, sehingga memungkinkan SVM untuk membuat *hyperplane* yang dapat memisahkan data yang tidak linier [18]. Beberapa kernel yang banyak digunakan adalah linear kernel, *polynomial* kernel, *sigmoid* kernel, dan *radial basis function* (RBF) kernel.

1. Linear kernel

Kernel ini adalah kernel SVM paling sederhana yang cocok digunakan ketika kelas data dapat dipisah secara linear. Linear kernel adalah kernel yang digunakan secara default pada pemodelan SVM [19].

$$f(x) = w^T x + b \quad (7)$$

2. *Polynomial* kernel

Polynomial kernel banyak digunakan saat semua data latih telah dinormalisasi.

$$Kx_i x = (x_i^T x + 1)^d \quad (8)$$

3. *Sigmoid* kernel

Sigmoid kernel merupakan pengembangan dari jaringan saraf tiruan yang dinyatakan dalam rumus berikut.

$$Kx_i x = \tanh(\gamma\, x_i^T x + \mathrm{r}) \quad (9)$$

4. *Radial basis function* (RBF)

RBF kernel menggunakan 2 parameter untuk optimalisasi SVM, yaitu *Gamma* dan *Cost*. *Gamma* merupakan nilai dari satu data, sedangkan *Cost* merupakan variabel yang bertujuan untuk optimalisasi informasi data [19].

$$Kx_i x = \exp[-\gamma||x_i - x||^2] \quad (10)$$

2.7 *Confusion Matrix*

Confusion matrix adalah alat yang digunakan untuk mengukur kinerja model klasifikasi. Di dalamnya, terdapat informasi mengenai nilai yang diprediksi oleh sistem dan nilai data yang sebenarnya [20].

Tabel 1. *Confusion Matrix*

	Kelas Prediksi	
	Positif	Negatif

p-ISSN: 2301-5373
e-ISSN: 2654-5101

Kelas Sebenarnya	Positif	TP	FN
	Negatif	FP	TN

Keterangan:

TP (*True Positive*) = data kelas positif yang diprediksi positif

FP (*False Positive*) = data kelas negatif yang diprediksi positif

FN (*False Negative*) = data kelas positif yang diprediksi negatif

TN (*True Negative*) = data kelas negatif yang diprediksi negatif

Dari *confusion matrix* yang dapat dilihat pada tabel 1, dapat diperoleh kinerja algoritma dengan menghitung nilai *accuracy*, *precision*, *recall,* dan *F1-score* dengan rumus berikut [21].

$$accuracy = \frac{TP+TN}{Total} \quad (11)$$

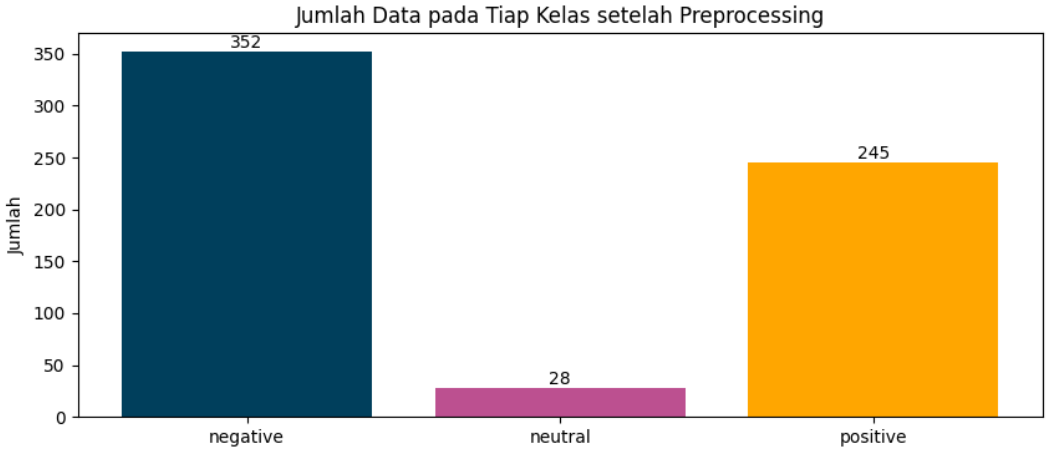
$$precision = \frac{TP}{TP+FP} \quad (12)$$

$$recall = \frac{TP}{TP+FN} \quad (13)$$

$$F1\ score = 2 * \frac{recall*precision}{recall+precision} \quad (14)$$

3. Hasil dan Pembahasan

Total data yang diperoleh dari tahap pengumpulan data adalah 669 data ulasan aplikasi SpeedID yang berasal dari *Play Store* dan *App Store*. Setelah melewati tahap preprocessing, terdapat sebanyak 44 data yang kosong sehingga data yang bisa digunakan adalah 625 data dengan distribusi kelas yang ditunjukkan pada gambar 2.



Gambar 2. Distribusi Kelas setelah *Preprocessing*

Contoh hasil dari *preprocessing* data ditunjukkan pada tabel 2.

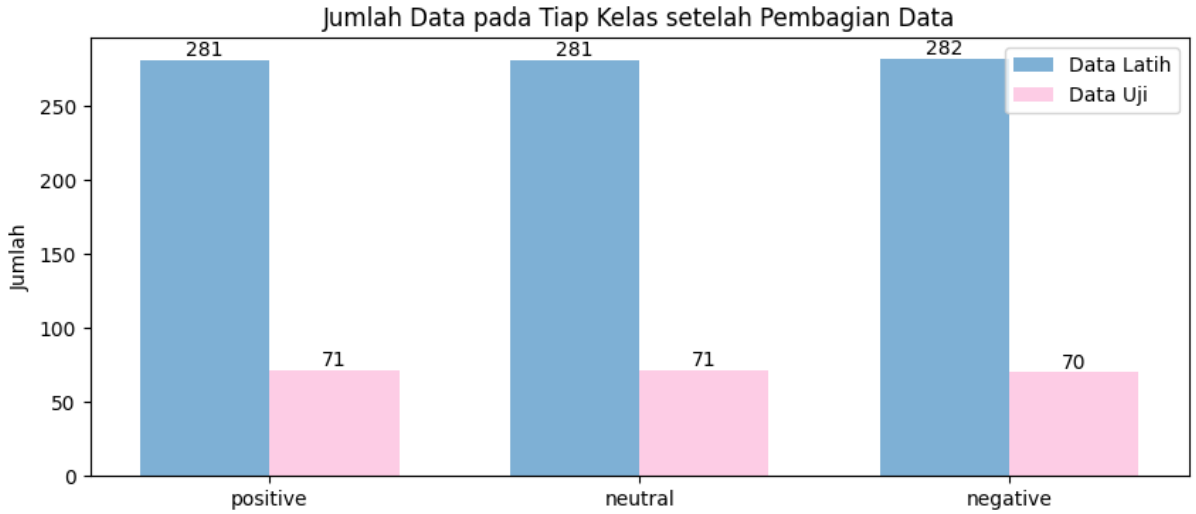
Tabel 2. Hasil *Preprocessing*

Sebelum Preprocessing	Setelah Preprocessing
Gak ada kelebihannya,cuma untuk daftar cari no antrean,,,,,,, no antrean juga tidak sesuai dengan yang tertera dilayar □□□□,,,,,, ngantre di bank bpd paling lama,,,,keteller nunggu lagi 9 orang perlu wakti 1 jam lebih□□□□□□□,,ada yang nyalib lewat samping juga□□□□ harus perlu trening ke bank swasta untuk perbandingan kinerja ,,,,,,, LOKASI BPD UBUNG	daftar cari antre antre sesuai tera layar antri bank bpd keteller tunggu orang wakti jam nyalib samping trening bank swasta banding kerja lokasi bpd ubung
Kenapa eror terus ya dari kemarin, tolong dong di perbaiki, padahal saya selalu menggunakan speedQ untuk daftar antrean rumah sakit,	eror kemarin speedq daftar antre rumah sakit
Aplikasi masih jauh dari efektif.. cara pendaftaran tidak praktis.. telalu bertele-tele. Boleh ribet tapi aplikasi harus bisa	aplikasi efektif daftar praktis talu tele tele ribet aplikasi simpan data

nyimpen data, supaya tidak mengetik berulang ulang.. Ini bukan speedid namanya.. tapi SLOWID..	etik ulang ulang speedid nama slowid

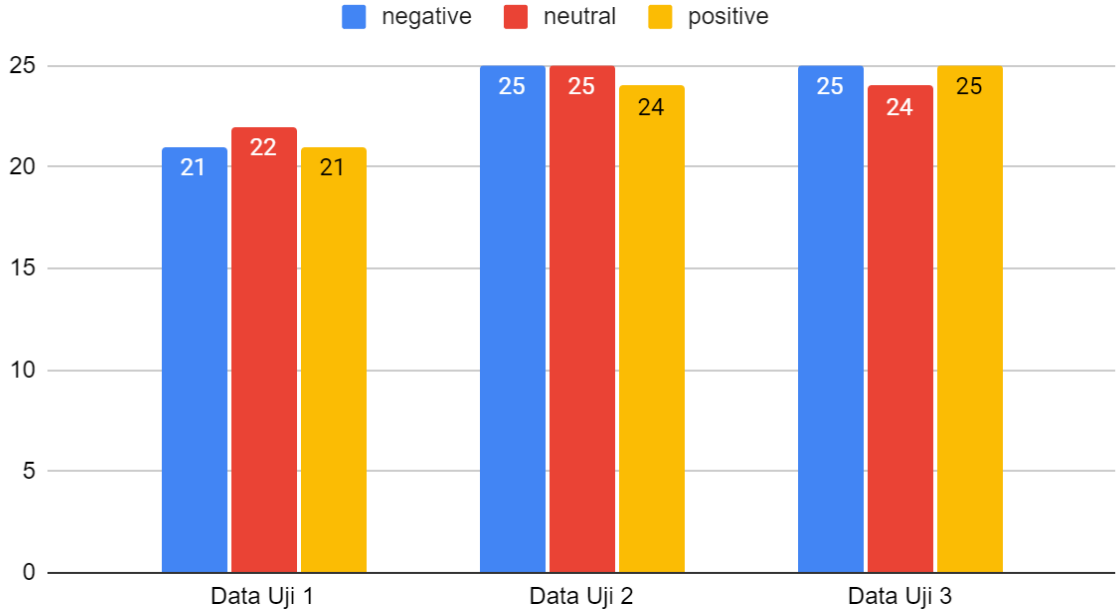
Karena jumlah kelas data yang tidak seimbang, dilakukan *over sampling* untuk menyeimbangkan data sebelum diproses dalam klasifikasi. *Over sampling* diterapkan secara acak pada data latih. *Over sampling* ini mengambil sampel pada kelas positif dan netral secara acak agar seimbang dengan kelas mayoritas, yaitu negatif. Data yang awalnya berjumlah 625, kini menjadi 1056 setelah dilakukan *over sampling* dengan jumlah data pada masing-masing kelas yaitu 352.

Data yang telah melalui tahap *over sampling* kemudian dibagi menjadi data latih sebanyak 80% dan data uji sebanyak 20% dari total data 1056. Data latih yang digunakan dalam pelatihan model klasifikasi terdiri dari 844 dan data uji yang digunakan dalam pengujian performa model terdiri dari 212 data. Adapun distribusi jumlah data pada masing-masing kelas di data latih dan data uji ditunjukkan melalui gambar 3 di bawah.



Gambar 3. Distribusi Kelas Data Latih dan Data Uji

Selain membagi data menjadi data latih dan data uji, data uji juga dibagi menjadi 3 bagian untuk melakukan pengujian sebanyak 3 kali. Distribusi kelas di tiap data uji ditampilkan pada gambar 4.



Gambar 4. Distribusi Kelas Data Uji

p-ISSN: 2301-5373
e-ISSN: 2654-5101

Setelah mendapatkan data latih yang seimbang di tiap kelasnya, tahap selanjutnya adalah melatih model klasifikasi NB dan SVM menggunakan data latih. Masing-masing pelatihan model dilakukan dalam 2 skenario, yaitu tanpa menggunakan seleksi fitur dan dengan menggunakan seleksi fitur *chi-square*. Model terbaik pada masing-masing skenario dinilai berdasarkan nilai akurasi tertinggi pada data latih. Model tersebut kemudian dievaluasi menggunakan data uji pada tahap evaluasi.

Pelatihan model tanpa seleksi fitur meliputi *hyperparameter tuning* untuk mengetahui kombinasi nilai parameter yang menghasilkan performa terbaik. Setelah itu, kombinasi *hyperparameter* terbaik tersebut digunakan untuk eksperimen lebih lanjut menggunakan seleksi fitur *chi-square*. Adapun nilai yang diuji dalam pelatihan model dengan *chi-square* adalah jumlah fitur yang dipertahankan seperti yang ditampilkan pada tabel 3.

Tabel 3. Eksperimen *Chi-Square*

Persentase fitur	Jumlah fitur
20%	174
40%	349
60%	524
80%	699

Dalam pelatihan model NB tanpa seleksi fitur, *hyperparameter* yang di-*tuning* adalah *alpha*, yaitu parameter untuk menangani agar probabilitas tidak bernilai 0. Hasil dari pelatihan model NB tanpa *chi-square* ditampilkan pada gambar 5.

```
Accuracy: 0.8626, alpha: 0.1
Accuracy: 0.8566, alpha: 0.5
Accuracy: 0.8531, alpha: 1
Accuracy: 0.8531, alpha: 1.5
Accuracy: 0.8507, alpha: 2
```

Gambar 5. Hasil Pelatihan NB

Terlihat bahwa akurasi data latih tertinggi, yaitu 86,26% diperoleh saat nilai *alpha*=0,1 sehingga nilai *hyperparameter* ini yang digunakan untuk melakukan eksperimen lebih lanjut dengan *chi-square*. Adapun hasil dari pelatihan model NB dengan *chi-square* dapat dilihat pada gambar 6 berikut.

```
Jumlah fitur = 174
Accuracy: 0.8400, alpha: 0.1
Jumlah fitur = 349
Accuracy: 0.8839, alpha: 0.1
Jumlah fitur = 524
Accuracy: 0.8791, alpha: 0.1
Jumlah fitur = 699
Accuracy: 0.8685, alpha: 0.1
```

Gambar 6. Hasil Pelatihan NB + *Chi Square*

Model NB terbaik diperoleh dengan *hyperparameter alpha*=0,1 dan akurasi data latih sebesar 86,26%. Saat dievaluasi menggunakan data uji sebanyak 3 kali, model NB tersebut menghasilkan performa yang dapat dilihat pada tabel 4.

Tabel 4. Hasil Pengujian Model Terbaik NB

Data	Akurasi (%)	*Precision* (%)	*Recall* (%)	*F1-Score* (%)
Data uji 1	92,19	93	92	91,67
Data uji 2	82,43	84,67	82,33	82,67
Data uji 3	83,78	84,33	84	84

Model NB + Chi Square terbaik memperoleh akurasi data latih sebesar 88,39% dengan mempertahankan jumlah fitur 40%. Adapun performa model terbaik NB dengan *chi-square* selengkapnya dapat dilihat pada tabel 5 berikut.

Tabel 5. Hasil Pengujian Model Terbaik NB + *Chi Square*

Data	Akurasi (%)	*Precision* (%)	*Recall* (%)	*F1-Score* (%)
Data uji 1	95,31	95,33	95	95
Data uji 2	82,43	84,67	82,33	82,67
Data uji 3	85,13	85,67	85,33	85,33

Secara keseluruhan, perbandingan performa kedua model NB dapat dilihat pada tabel 6 berikut.

Tabel 6. Perbandingan Akurasi (%) Model NB

Model	Data Uji 1	Data Uji 2	Data Uji 3
NB	92,19	82,43	83,78
NB + *chi square*	95,31	82,43	85,13

Berdasarkan tabel 6, seleksi fitur *chi-square* dapat meningkatkan nilai akurasi dari model NB sebanyak 2 kali. Nilai akurasi data uji 1 meningkat sebesar 3,12%, sedangkan akurasi data uji 3 meningkat sebesar 1.35%. Hal ini menunjukkan bahwa seleksi fitur *chi-square* memiliki pengaruh positif terhadap performa model NB.

Dalam pelatihan model SVM tanpa *chi-square*, *hyperparameter* yang diuji dapat dilihat pada tabel 7 berikut.

Tabel 7. Nilai *Hyperparameter* SVM yang Diuji

Hyperparameter	Nilai Parameter	Kernel
C	0,01; 0,1; 1; 10; 100	Linear, RBF, *sigmoid, polynomial*
Gamma	0,01; 0,1; 1; 10; 100	RBF, *sigmoid, polynomial*
Degree	2, 3, 4	*Polynomial*

Pelatihan model linear SVM tanpa seleksi fitur ini menghasilkan nilai akurasi pada data latih yang dapat dilihat pada gambar 7.

```
Accuracy: 0.4797, Parameters: {'C': 0.01, 'kernel': 'linear'}
Accuracy: 0.7559, Parameters: {'C': 0.1, 'kernel': 'linear'}
Accuracy: 0.8803, Parameters: {'C': 1, 'kernel': 'linear'}
Accuracy: 0.8886, Parameters: {'C': 10, 'kernel': 'linear'}
Accuracy: 0.8709, Parameters: {'C': 100, 'kernel': 'linear'}
Best Parameters: {'C': 10, 'kernel': 'linear'}
```

Gambar 7. Hasil Pelatihan Linear SVM

Pelatihan linear SVM dengan *chi-square* selanjutnya dilakukan menggunakan nilai parameter C=10 dengan hasil yang ditampilkan pada gambar 8.

```
Accuracy: 0.8839, Parameters: {'C': 10, 'kernel': 'linear'}
Best Parameters: {'C': 10, 'kernel': 'linear'}

Accuracy: 0.8898, Parameters: {'C': 10, 'kernel': 'linear'}
Best Parameters: {'C': 10, 'kernel': 'linear'}

Accuracy: 0.8839, Parameters: {'C': 10, 'kernel': 'linear'}
Best Parameters: {'C': 10, 'kernel': 'linear'}

Accuracy: 0.8886, Parameters: {'C': 10, 'kernel': 'linear'}
Best Parameters: {'C': 10, 'kernel': 'linear'}
```

Gambar 8. Hasil Pelatihan Linear SVM + *Chi Square*

Untuk pelatihan model SVM dengan kernel *Radial Basis Function* (RBF), *hyperparameter* yang digunakan adalah C dan gamma dengan hasil pelatihan seperti pada tabel 8.

p-ISSN: 2301-5373
e-ISSN: 2654-5101

Tabel 8. Nilai Akurasi (%) Hasil Pelatihan RBF SVM

C	Gamma				
	0,01	0,1	1	10	100
0,01	48,09	48,68	47,38	44,30	44,30
0,1	48,09	50,10	71,33	64,33	64,33
1	49,51	80,57	89,81	84,36	84,36
10	81,16	89,45	**89,81**	84,48	84,36
100	88,62	88,63	89,81	84,48	84,36

Berdasarkan tabel 8, dapat dilihat bahwa nilai akurasi terbaik diperoleh ketika nilai C=10 dan gamma=1. Nilai *hyperparameter* tersebut kemudian digunakan dalam eksperimen dengan *chi-square* dengan hasil seperti pada gambar 9.

```
Accuracy: 0.8851, Parameters: {'C': 10, 'gamma': 1, 'kernel': 'rbf'}
Best Parameters: {'C': 10, 'gamma': 1, 'kernel': 'rbf'}

Accuracy: 0.8981, Parameters: {'C': 10, 'gamma': 1, 'kernel': 'rbf'}
Best Parameters: {'C': 10, 'gamma': 1, 'kernel': 'rbf'}

Accuracy: 0.8981, Parameters: {'C': 10, 'gamma': 1, 'kernel': 'rbf'}
Best Parameters: {'C': 10, 'gamma': 1, 'kernel': 'rbf'}

Accuracy: 0.8993, Parameters: {'C': 10, 'gamma': 1, 'kernel': 'rbf'}
Best Parameters: {'C': 10, 'gamma': 1, 'kernel': 'rbf'}
```

Gambar 9. Hasil Pelatihan RBF SVM + *Chi Square*

Pada pelatihan *sigmoid* SVM, *hyperparameter* yang digunakan adalah C dan gamma juga dengan hasil pelatihan yang ditunjukkan pada tabel 9.

Tabel 9. Nilai Akurasi (%) Hasil Pelatihan *Sigmoid* SVM

C	Gamma				
	0,01	0,1	1	10	100
0,01	47,97	47,97	47,85	45,85	50,12
0,1	47,97	47,97	75	65,16	56,40
1	47,97	75,59	85,54	59,24	53,79
10	75,59	88,03	83,77	56,99	54,15
100	88,03	**88,86**	84,12	58,30	55,68

Model dengan kombinasi nilai parameter terbaik pada *Sigmoid* SVM, yaitu C=100 dan gamma=0,1. Nilai *hyperparameter* tersebut kemudian digunakan dalam eksperimen dengan *chi-square* dengan hasil pelatihan seperti pada gambar 10.

```
Accuracy: 0.8815, Parameters: {'C': 100, 'gamma': 0.1, 'kernel': 'sigmoid'}
Best Parameters: {'C': 100, 'gamma': 0.1, 'kernel': 'sigmoid'}

Accuracy: 0.8898, Parameters: {'C': 100, 'gamma': 0.1, 'kernel': 'sigmoid'}
Best Parameters: {'C': 100, 'gamma': 0.1, 'kernel': 'sigmoid'}

Accuracy: 0.8839, Parameters: {'C': 100, 'gamma': 0.1, 'kernel': 'sigmoid'}
Best Parameters: {'C': 100, 'gamma': 0.1, 'kernel': 'sigmoid'}

Accuracy: 0.8886, Parameters: {'C': 100, 'gamma': 0.1, 'kernel': 'sigmoid'}
Best Parameters: {'C': 100, 'gamma': 0.1, 'kernel': 'sigmoid'}
```

Gambar 10. Hasil Pelatihan *Sigmoid* SVM + *Chi Square*

Pada pelatihan *polynomial* SVM, *hyperparameter* yang digunakan adalah C, gamma, dan *degree* dengan hasil pelatihan yang ditunjukkan di tabel 10, 11, dan 12.

Tabel 10. Nilai Akurasi (%) Hasil Pelatihan Polynomial SVM (Degree=2)

C	**Gamma**				
	0,01	0,1	1	10	100
0,01	46,31	46,31	46,31	88,86	**89,10**
0,1	46,31	46,31	71,57	89,10	89,10
1	46,31	46,31	88,86	89,10	89,10
10	46,31	71,57	89,10	89,10	89,10
100	46,31	88,86	89,10	89,10	89,10

Tabel 11. Nilai Akurasi (%) Hasil Pelatihan Polynomial SVM (Degree=3)

C	**Gamma**				
	0,01	0,1	1	10	100
0,01	45,36	45,36	45,36	87,08	87,08
0,1	45,36	45,36	69,67	87,08	87,08
1	45,36	45,36	86,97	87,08	87,08
10	45,36	45,36	87,08	87,08	87,08
100	45,36	69,67	87,08	87,08	87,08

Tabel 12. Nilai Akurasi (%) Hasil Pelatihan Polynomial SVM (Degree=4)

C	**Gamma**				
	0,01	0,1	1	10	100
0,01	45,01	45,01	45,01	86,61	86,61
0,1	45,01	45,01	68,60	86,61	86,61
1	45,01	45,01	86,61	86,61	86,61
10	45,01	45,01	86,61	86,61	86,61
100	45,01	45,01	86,61	86,61	86,61

Berdasarkan tabel 10, 11, dan 12 di atas, model dengan kombinasi nilai parameter terbaik pada *polynomial* SVM, yaitu C=0,01; gamma=100; dan *degree*=2. Nilai *hyperparameter* tersebut kemudian digunakan dalam eksperimen dengan *chi-square* dengan hasil seperti pada gambar 11.

```
Accuracy: 0.8720, Parameters: {'C': 0.01, 'degree': 2, 'gamma': 100, 'kernel': 'poly'}
Best Parameters: {'C': 0.01, 'degree': 2, 'gamma': 100, 'kernel': 'poly'}

Accuracy: 0.8637, Parameters: {'C': 0.01, 'degree': 2, 'gamma': 100, 'kernel': 'poly'}
Best Parameters: {'C': 0.01, 'degree': 2, 'gamma': 100, 'kernel': 'poly'}

Accuracy: 0.8756, Parameters: {'C': 0.01, 'degree': 2, 'gamma': 100, 'kernel': 'poly'}
Best Parameters: {'C': 0.01, 'degree': 2, 'gamma': 100, 'kernel': 'poly'}

Accuracy: 0.8862, Parameters: {'C': 0.01, 'degree': 2, 'gamma': 100, 'kernel': 'poly'}
Best Parameters: {'C': 0.01, 'degree': 2, 'gamma': 100, 'kernel': 'poly'}
```

Gambar 11. Hasil Pelatihan *Polynomial* SVM + *Chi Square*

Untuk menentukan 1 model SVM tanpa *chi-square* terbaik dan 1 model SVM dengan *chi square* terbaik, nilai akurasi data latih model SVM dirangkum pada tabel 13 di bawah.

Tabel 13. Rangkuman Hasil Pelatihan Model SVM

Model	**Akurasi Data Latih (%)**

p-ISSN: 2301-5373
e-ISSN: 2654-5101

Linear SVM	88,86
Linear SVM + *chi square*	88,98
Sigmoid SVM	88,86
Sigmoid SVM + *chi square*	88,98
RBF SVM	**89,81**
RBF SVM + *chi square*	**89,93**
Polynomial SVM	89,10
Polynomial SVM + *chi square*	88,62

Tabel 13 menunjukkan bahwa model terbaik dengan nilai akurasi data latih tertinggi adalah model SVM dengan kernel RBF yang memiliki *hyperparameter* C=10 dan gamma=1. Adapun performa model SVM terbaik selengkapnya dapat dilihat pada tabel 14 berikut.

Tabel 14. Performa Model Terbaik SVM

Data	Akurasi (%)	*Precision* (%)	*Recall* (%)	*F1-Score* (%)
Data uji 1	95,31	95,33	95	95,33
Data uji 2	83,78	85,67	84	84
Data uji 3	89,19	89,67	89,33	89,33

Model terbaik SVM dengan *chi-square* yang dihasilkan adalah model yang mempertahankan 80% jumlah fitur yang mencapai nilai akurasi data latih sebesar 89,93%. Performa model SVM dengan *chi-square* selengkapnya dapat dilihat pada tabel 15 berikut.

Tabel 15. Performa Model Terbaik SVM + *Chi Square*

Data	Akurasi (%)	*Precision* (%)	*Recall* (%)	*F1-Score* (%)
Data uji 1	95,31	95,33	95	95,33
Data uji 2	83,78	85,67	84	84
Data uji 3	89,19	89,67	89,33	89,33

Secara keseluruhan, perbandingan performa kedua model SVM dapat dilihat pada tabel 16.

Tabel 16. Perbandingan Akurasi (%) Model SVM

Model	Data Uji 1	Data Uji 2	Data Uji 3
SVM	95,31	83,78	89,19
SVM + *chi square*	95,31	83,78	89,19

Berdasarkan tabel 16 di atas, seleksi fitur *chi-square* sama sekali tidak dapat mengubah nilai akurasi dari model SVM. Nilai akurasi data uji 1, 2, dan 3 tidak mengalami peningkatan maupun penurunan saat ditambahkan *chi-square*. Hal ini menunjukkan bahwa seleksi fitur *chi-square* tidak memiliki pengaruh positif maupun negatif terhadap performa model SVM.

4. Kesimpulan

Penelitian ini bertujuan untuk mengetahui performa khususnya nilai akurasi dari model *Naive Bayes* (NB) dan *Support Vector Machine* (SVM) dalam analisis sentimen ulasan aplikasi SpeedID serta mengetahui pengaruh dari seleksi fitur *chi square* terhadap performa model NB dan SVM tersebut. Adapun beberapa kesimpulan yang dapat diambil dari hasil penelitian ini adalah sebagai berikut.

1. Model NB terbaik diperoleh dengan *hyperparameter alpha*=0,1 yang menghasilkan nilai akurasi sebesar 92,18%; 82,43%; dan 83,78%, sedangkan model SVM terbaik diperoleh dengan kernel *Radial Basis Function* (RBF), C=10, dan gamma=1 yang menghasilkan nilai akurasi sebesar 95,31%; 83,78%; dan 89,18 yang semuanya lebih tinggi dibandingkan dengan model NB.
2. Dari 3 kali pengujian yang dilakukan pada tiap model terbaik, seleksi fitur *chi-square* terbukti dapat meningkatkan nilai akurasi model NB hingga 3,12%. Namun, nilai akurasi model SVM

tidak mengalami peningkatan maupun penurunan saat ditambahkan dengan seleksi fitur c*hi-square*. Dengan demikian, seleksi fitur *chi-square* disimpulkan memiliki pengaruh positif terhadap performa NB, tetapi tidak memiliki pengaruh terhadap performa SVM.

References

[1] L. Ardiani, H. Sujaini, dan T. Tursina, "Implementasi Sentiment Analysis Tanggapan Masyarakat Terhadap Pembangunan di Kota Pontianak," *Jurnal Sistem dan Teknologi Informasi (Justin)*, vol. 8, no. 2, hlm. 183–190, Apr 2020, doi: 10.26418/justin.v8i2.36776.

[2] M. Cindo, D. P. Rini, dan Ermatita, "Literatur Review: Metode Klasifikasi Pada Sentimen Analisis," *Seminar Nasional Teknologi Komputer & Sains (SAINTEKS)*, hlm. 66–70, Jan 2019.

[3] F. S. Pattiiha dan Hendry, "Perbandingan Metode K-NN, Naïve Bayes, Decision Tree untuk Analisis Sentimen Tweet Twitter Terkait Opini Terhadap PT PAL Indonesia ," *JURIKOM (Jurnal Riset Komputer)*, vol. 9, no. 2, hlm. 506–514, Apr 2022.

[4] R. Setiawan dan A. Triayudi, "Klasifikasi Status Gizi Balita Menggunakan Naïve Bayes dan K-Nearest Neighbor Berbasis Web," *JURNAL MEDIA INFORMATIKA BUDIDARMA*, vol. 6, no. 2, hlm. 777–785, Apr 2022, doi: 10.30865/mib.v6i2.3566.

[5] F. Solikhah, M. Febianah, A. L. Kamil, W. A. Arifin, dan S. J. S. Tyas, "Analisis Perbandingan Algoritma Naive Bayes Dan C.45 Dalam Klasifikasi Data Mining Untuk Memprediksi Kelulusan," *Tematik : Jurnal Teknologi Informasi Komunikasi (e-Journal)*, vol. 8, no. 1, hlm. 96–103, Jun 2021.

[6] I. M. Parapat, M. T. Furqon, dan Sutrisno, "Penerapan Metode Support Vector Machine (SVM) Pada Klasifikasi Penyimpangan Tumbuh Kembang Anak," *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, vol. 2, no. 10, hlm. 3163–3169, Okt 2018.

[7] F. Bei dan S. Saepudin, "ANALISIS SENTIMEN APLIKASI TIKET ONLINE DI PLAY STORE MENGGUNAKAN METODE SUPPORT VECTOR MACHINE (SVM)," *SISMATIK (Seminar Nasional Sistem Informasi dan Manajemen Informatika)*, vol. 1, no. 1, hlm. 91–97, Agu 2021.

[8] V. K. S. Que, A. Iriani, dan H. D. Purnomo, "Analisis Sentimen Transportasi Online Menggunakan Support Vector Machine Berbasis Particle Swarm Optimization," *Jurnal Nasional Teknik Elektro dan Teknologi Informasi*, vol. 9, no. 2, hlm. 162–170, Mei 2020.

[9] S. I. Nurhafida dan F. Sembiring, "Analisis Sentimen Aplikasi Novel Online Di Google Play Store Menggunakan Algoritma Support Vector Machine (SVM)," *Jurnal Sains Komputer & Informatika (J-SAKTI)*, vol. 6, no. 1, hlm. 317–327, Mar 2022.

[10] A. Falasari dan M. A. Muslim, "Optimize Naïve Bayes Classifier Using Chi Square and Term Frequency Inverse Document Frequency For Amazon Review Sentiment Analysis," *Journal of Soft Computing Exploration*, vol. 3, no. 1, hlm. 31–36, Mar 2022, doi: 10.52465/joscex.v3i1.68.

[11] C. J. E. Munthe, N. A. Hasibuan, dan H. Hutabarat, "Penerapan Algoritma Text Mining Dan TF-RF Dalam Menentukan Promo Produk Pada Marketplace," *Resolusi : Rekayasa Teknik Informatika dan Informasi*, vol. 2, no. 3, hlm. 110–115, Jan 2022, doi: 10.30865/resolusi.v2i3.309.

[12] U. I. Larasati, M. A. Muslim, R. Arifudin, dan A. Alamsyah, "Improve the Accuracy of Support Vector Machine Using Chi Square Statistic and Term Frequency Inverse Document Frequency on Movie Review Sentiment Analysis," *Scientific Journal of Informatics*, vol. 6, no. 1, hlm. 138–149, Mei 2019, doi: 10.15294/sji.v6i1.14244.

[13] L. Luthfiana, J. C. Young, dan A. Rusli, "Implementasi Algoritma Support Vector Machine dan Chi Square untuk Analisis Sentimen User Feedback Aplikasi," *Ultimatics : Jurnal Teknik Informatika*, vol. 12, no. 2, hlm. 125–126, Des 2020, doi: 10.31937/ti.v12i2.1828.

[14] W. Winata, A. Zaidiah, dan N. Chamidah, "ANALISIS SENTIMEN PADA ULASAN PRODUK MASKER DI MARKETPLACE SHOPEE MENGGUNAKAN SUPPORT VECTOR MACHINE DAN SELEKSI FITUR CHI SQUARE," *Seminar Nasional Mahasiswa Ilmu Komputer dan Aplikasinya (SENAMIKA)*, Agu 2022.

p-ISSN: 2301-5373
e-ISSN: 2654-5101

[15] M. Christianto, J. Andjarwirawan, dan A. Tjondrowiguno, “Aplikasi Analisa Sentimen Pada Komentar Berbahasa Indonesia Dalam Objek Video di Website YouTube Menggunakan Metode Naïve Bayes Classifier,” *JURNAL INFRA*, vol. 8, no. 1, 2020.

[16] R. W. Pratiwi, S. F. H, D. Dairoh, D. I. Af’idah, Q. R. A, dan A. G. F, “Analisis Sentimen Pada Review Skincare Female Daily Menggunakan Metode Support Vector Machine (SVM),” *Journal of Informatics, Information System, Software Engineering and Applications (INISTA)*, vol. 4, no. 1, hlm. 40–46, Des 2021, doi: 10.20895/inista.v4i1.387.

[17] S. Chatterjee, P. George Jose, dan D. Datta, “Text Classification Using SVM Enhanced by Multithreading and CUDA,” *International Journal of Modern Education and Computer Science*, vol. 11, no. 1, hlm. 11–23, Jan 2019, doi: 10.5815/ijmecs.2019.01.02.

[18] A. Z. Praghakusma dan N. Charibaldi, “Komparasi Fungsi Kernel Metode Support Vector Machine untuk Analisis Sentimen Instagram dan Twitter (Studi Kasus : Komisi Pemberantasan Korupsi) ,” *Jurnal Sarjana Teknik Informatika*, vol. 9, no. 2, 2021.

[19] A. Zeputra dan F. Utaminingrum, “Perbandingan Akurasi untuk Deteksi Pintu berbasis HOG dengan Klasifikasi SVM menggunakan Kernel Linear, Radial Basis Function dan Polinomial pada Raspberry Pi,” *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, vol. 5, no. 11, hlm. 4746–4757, Nov 2021.

[20] F. Novitasari dan M. D. Purbolaksono, “Sentiment Analysis Aspect Level on Beauty Product ReviewsUsing Chi-Square and Naïve Bayes,” *JOURNAL OF DATA SCIENCE AND ITS APPLICATIONS*, vol. 4, no. 1, hlm. 18–30, Jan 2021.

[21] D. Normawati dan S. A. Prayogi, “Implementasi Naïve Bayes Classifier Dan Confusion Matrix Pada Analisis Sentimen Berbasis Teks Pada Twitter,” *Jurnal Sains Komputer & Informatika (J-SAKTI)* , vol. 5, no. 2, hlm. 697–711, Sep 2021.

This page is intentionally left blank.