

ANALISIS SENTIMEN APLIKASI E-LEARNING QUIPPER SELAMA PANDEMI COVID-19 DENGAN MENGGUNAKAN METODE SUPPORT VECTOR MACHINE (SVM)

Ahsinil Amal Annur¹, Aang Alim Murtopo², Nurul Fadilah³

^{1,2,3}Program Studi Teknik Informatika, STMIK Tegal

E-mail : ahsinilam@gmail.com, aang.alim@gmail.com, nurulfadilah18@gmail.com

Jl. Pendidikan No.1, Pesurungan Lor, Kec. Margadana, Kota Tegal, Jawa Tengah 52122

Kata Kunci:

*Pandemi Covid-19,
Aplikasi E-learning,
Analisis Sentimen,
Support Vector
Machine, Quipper*

Abstrak

E-Learning atau Pembelajaran Online adalah pembelajaran formal dan informal yang dilakukan dengan menggunakan teknologi, sehingga siswa dan guru bisa melakukan proses belajar mengajar melalui media elektronik. Pembelajaran online memiliki kendala tersendiri dan berbeda-beda dari setiap pengguna. Ada yang merasa kesulitan untuk memahami materi atau penjelasan yang diberikan oleh dosen atau gurunya di saat pendidikan menjadi online, ada juga yang merasa mudah dalam memahami materi atau penjelasan saat melakukan pendidikan online. Pada penelitian ini metode SVM dipilih untuk mengklasifikasikan sentimen, karena metode SVM terbukti cukup efektif dalam melakukan klasifikasi sentimen. Alur awal yang dilakukan pada analisis sentimen di penelitian ini yaitu pengumpulan data ulasan dengan cara scrapping pada play store aplikasi Quipper. Selanjutnya adalah tahap preprocessing di mana proses ini memiliki lima tahapan yaitu case folding, text cleaning, tokenization, stemming, dan filtering (stopword removal). Setelah mendapatkan hasil preprocessing, akan dilakukan pelabelan dan transformation data. Hasil transformation data, yaitu data train akan digunakan dalam pembentukan model algoritma SVM. Selanjutnya data test dimasukkan ke dalam model algoritma SVM yang sudah dibuat. Proses terakhir adalah melakukan perhitungan akurasi metode SVM. Sentimen masyarakat atau pengguna terhadap aplikasi Quipper menghasilkan 410 ulasan negatif 590 dan ulasan positif. Hasil evaluasi kinerja algoritma Support Vector Machine kernel linier dalam klasifikasi atau analisis sentimen ulasan aplikasi e-learning Quipper menggunakan tiga skenario split data, diperoleh hasil terbaik di skenario ketiga dengan rasio perbandingan data training 60% dan data testing 40% dan menghasilkan accuracy sebesar 90,25%, precision 86,85%, recall 94,39%, dan f1-score 90,46%.

Abstract:

*Covid-19 Pandemic,
E-learning
Application,
Sentiment Analysis,
Support Vector
Machine, Quipper*

E-Learning or Online Learning is formal and informal learning carried out using technology, so that students and teachers can carry out the teaching and learning process through electronic media. Online learning has its own constraints and varies from user to user. There are those who find it difficult to understand the material or explanations given by the lecturer or teacher when education becomes online, there are also those who find it easy to understand the material or explanations when conducting online education. In this study, the SVM method was chosen to classify sentiment, because the SVM method proved to be quite effective in classifying sentiment. The initial flow of sentiment analysis in this study is collecting review data by scrapping on the Quipper application play store. Next is the preprocessing stage where this process has five stages, namely case folding, text cleaning, tokenization,

stemming, and filtering (stopword removal). After getting the preprocessing results, labeling and data transformation will be carried out. The results of data transformation, namely train data will be used in the formation of the SVM algorithm model. Furthermore, the test data is entered into the SVM algorithm model that has been created. The last process is to calculate the accuracy of the SVM method. The community or user sentiment towards the Quipper app resulted in 410 negative reviews, 590 positive reviews. The results of the evaluation of the performance of the Support Vector Machine linear kernel algorithm in the classification or sentiment analysis of the Quipper e-learning application reviews using three data split scenarios, the best results are obtained in the third scenario with a ratio of 60% training data and 40% testing data and produces an accuracy of 90, 25%, precision 86.85%, recall 94.39%, and f1-score 90.46%.

Pendahuluan

COVID-19 adalah penyakit menular yang disebabkan oleh SARS-CoV-2 (virus corona). Orang yang positif COVID-19 dapat mengalami demam, batuk kering, dan kesulitan bernapas. Penyakit ini menyebar ke orang lain melalui percikan (droplet) dari saluran pernapasan yang dihasilkan ketika seseorang bersin atau batuk. Jarak droplet di saat seseorang bersin atau batuk bisa menjangkau hingga 1 meter [1]. Untuk menekan risiko penularan virus corona dari satu orang ke orang lainnya, pemerintah menganjurkan social distancing sebagai salah satu protokol kesehatan selama pandemi COVID-19. Contoh penerapan menjaga jarak yang umum dilakukan adalah bekerja dari rumah (working from home) dan belajar online dari rumah untuk sekolah dan kuliah [2].

E-Learning atau Pembelajaran Online adalah pembelajaran formal dan informal yang dilakukan dengan menggunakan teknologi, sehingga siswa dan guru bisa melakukan proses belajar mengajar melalui media elektronik [3]. Faktor utama berkembang pesatnya layanan e-learning di Indonesia adalah kebijakan menjaga jarak (social distancing), di mana penggunaan e-learning meningkat drastis ketika pemerintah mengumumkan kebijakan social distancing, sehingga membuat layanan pendidikan offline berubah menjadi online. Pembelajaran online memiliki kendala tersendiri dan berbeda-beda dari setiap pengguna. Ada yang merasa kesulitan untuk memahami materi atau penjelasan yang diberikan oleh dosen atau gurunya di saat pendidikan menjadi online, ada juga yang merasa mudah dalam memahami materi atau penjelasan saat melakukan pendidikan online.

Landasan Teori

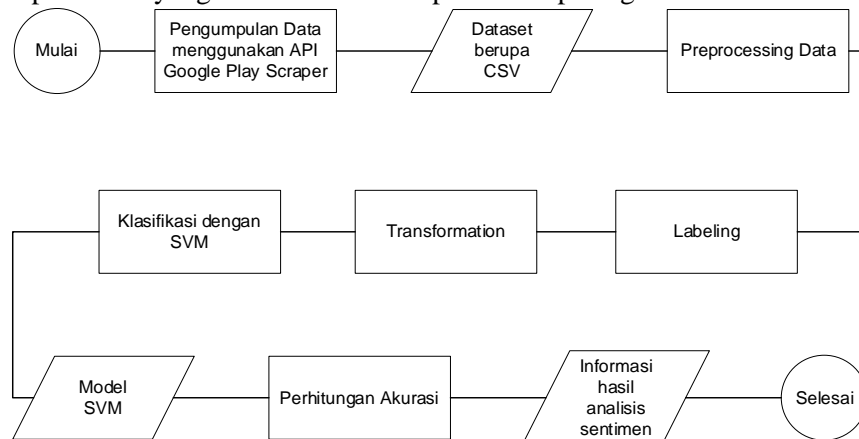
Pada penelitian ini aplikasi *e-learning* yang dipilih adalah Quipper, yang memiliki pengguna lebih dari 6.000.000 dan diunduh lebih dari 1.000.000 kali. Data ulasan untuk aplikasi *e-learning* Quipper diperoleh dari media *play store*. Penelitian ini bertujuan untuk menganalisis sentimen aplikasi *e-learning* di saat pandemi berlangsung. Komentar relevan yang telah dikumpulkan pada bulan Juli 2022 akan digunakan sebagai data untuk melakukan analisis sentimen. Data ulasan tersebut akan dianalisis untuk mengetahui sentimen masyarakat terhadap aplikasi *e-learning* tersebut, sekaligus dapat mengetahui apa yang bisa dikembangkan dan ditingkatkan dari aplikasi Quipper.

Penelitian yang berjudul Analisis Sentimen *Review* Pelanggan Restoran Menggunakan Algoritma *Support Vector Machine* Dan *K-Nearest Neighbor* [4] membuktikan hasil kinerja pada penelitian ini algoritma SVM lebih baik dibandingkan menggunakan algoritma k-NN di mana hasil perhitungan dari algoritma SVM memiliki nilai *accuracy* sebesar 81.92% sedangkan pada algoritma k-NN memiliki hasil *accuracy* sebesar 59.03%. Pada penelitian berjudul Analisis Sentimen pada *review* Aplikasi Grab di Google *Play Store* Menggunakan *Support Vector Machine* [5] terbukti bahwa metode SVM menghasilkan nilai akurasi sistem sebesar 85.54%.

Berdasarkan penelitian sebelumnya, pada penelitian ini metode SVM dipilih untuk mengklasifikasikan sentimen, karena metode SVM terbukti cukup efektif dalam melakukan klasifikasi sentimen. SVM juga dapat mengidentifikasi dengan cara memaksimalkan *margin* dan memperoleh *hyperplane* yang diinginkan dalam mengklasifikasikan data.

Metode Penelitian

Sebuah perencanaan analisis sentimen atau gambaran umum untuk menggambarkan rangkaian dari beberapa proses penelitian yang akan dilakukan dapat dilihat pada gambar 1 berikut.



Gambar 1. Alur Penelitian

Alur awal yang dilakukan pada analisis sentimen di penelitian ini yaitu pengumpulan data ulasan dengan cara *scraping* pada *play store* aplikasi Quipper. Selanjutnya adalah tahap *preprocessing* di mana proses ini memiliki lima tahapan yaitu *case folding*, *text cleaning*, *tokenization*, *stemming*, dan *filtering* (*stopword removal*). Setelah mendapatkan hasil *preprocessing*, akan dilakukan pelabelan dan *transformation data*. Hasil *transformation data*, yaitu *data train* akan digunakan dalam pembentukan model algoritma SVM. Selanjutnya *data test* dimasukkan ke dalam model algoritma SVM yang sudah dibuat. Proses terakhir adalah melakukan perhitungan akurasi metode SVM.

1.1. Pengumpulan Data

Pengambilan data yang dilakukan dengan metode *scraping* dengan menggunakan bahasa pemrograman *python* pada situs *play store* untuk mengambil ulasan dari pengguna aplikasi Quipper. Pada tahap pengumpulan data, peneliti melakukan pengumpulan data yang relevan dan dilakukan satu kali pada bulan Juni tahun 2022 untuk mengambil data yang terbaru. Data yang terkumpul diambil cukup 2 atribut saja yaitu *content*, dan *score*.

1.2. Preprocessing

Dataset yang didapat dari *scraping* belum bisa digunakan untuk klasifikasi menggunakan algoritma, karena isi di dalamnya masih memiliki kalimat atau kata-kata yang berantakan atau tidak terstruktur dan mempunyai *noise* [6]. Maka dari itu diperlukan proses untuk memperoleh inti informasi dari data ulasan agar bisa digunakan untuk kepentingan klasifikasi atau analisis. Penelitian ini menggunakan 5 tahapan *preprocessing* yang akan dilakukan, sebagai berikut:

- Case Folding* adalah proses mengganti semua huruf kapital (huruf besar) yang ada di *dataset* menjadi *lowercase* (huruf kecil) [7].
- Text cleaning* adalah proses menghilangkan atribut, tanda baca ataupun karakter yang tidak diperlukan [8].
- Stemming* adalah proses menghilangkan imbuhan dari sebuah kata atau mengubah kata turunan menjadi sebuah kata dasar [9].
- Filtering* adalah proses menghapus seluruh kata yang tidak terlalu berpengaruh dalam analisis sentiment [10].
- Tokenization* adalah proses memisahkan setiap kata berdasarkan karakter spasi yang terdapat di suatu kalimat atau suatu ulasan [11].

1.3. Labeling

Setelah data dibersihkan dari *noise*, kemudian dilakukan pelabelan data ulasan di mana setiap ulasan dengan *score* 4 dan 5 dianggap positif dan akan diberi label dengan angka 1, sedangkan jika ulasan dengan *score* 1 sampai 3 maka dianggap negatif dan diberi label dengan angka -1. Setelah semua kalimat ulasan telah diberi label, maka data sudah dapat dilakukan transformasi.

1.4. Transformation

Tahap *transformation*, di tahap ini data ditransformasikan menjadi format yang sesuai untuk *mining* [12]. Tahap ini akan melakukan pembobotan kata menggunakan TF-IDF, berguna untuk memproses data teks menjadi vektor berbobot. Perhitungan metode TFIDF dinyatakan dengan rumus sebagai berikut.

$$w_t = TF_{t,d} \times IDF_t = TF_{t,d} \times \log \frac{N}{DF_t} \quad (1)$$

Setelah itu dilakukan pemisahan data ke data latih (*data training*) dan data uji (*data testing*), dilakukan pemisahan data secara acak dan dijadikan dalam 3 skenario, yaitu skenario 1 (80% *data training* dan 20% *data testing*), skenario 2 (70% *data training* dan 30% *data testing*), skenario 3 (60% *data training* dan 40% *data testing*). Data uji digunakan untuk proses pelatihan mencari nilai *a* dan *b* di mana nilai tersebut digunakan untuk fungsi keputusan (*classifier*). Sementara *data testing* digunakan untuk proses pengujian dan menghasilkan klasifikasi data sebagai positif atau negatif.

1.5.Support Vector Machine (SVM)

Support Vector Machine adalah suatu algoritma *machine learning* dengan performa lebih baik dari algoritma *machine learning* lainnya untuk mengklasifikasi, regresi dan prediksi. Klasifikasi SVM bekerja dengan mencari *hyperplane* terbaik, didukung oleh parameter C yang memiliki fungsi untuk memperhatikan pencarian *hyperplane* [13]. Setelah melakukan *transformation*, *data train* dimasukkan ke dalam model SVM yang memiliki *split data* dengan skenario pertama 80:20, kedua 70:30 dan ketiga 60:40 dari total data yang berjumlah 1000 ulasan. Kemudian dimasukkan ke dalam *data test* untuk melakukan klasifikasi dengan memberikan hasil *hyperplane* untuk pemisahan ruang (*space*) dari kedua kelas (*class*), secara sederhana metode ini adalah pembuat klasifikasi terarah (*linear classifier*) kemudian berkembang sehingga dapat bekerja pada *problem non linear* [14].

1.5.1.Linearly Separable Data

Linearly separable data merupakan suatu data yang bisa dipisahkan secara linier. Misal $x_i \in R^d$ dan label kelas sebagai $y_i \in \{-1, +1\}$ untuk $i = 1, 2, \dots, n$, *n* adalah banyaknya data. Model linier metode SVM untuk menghasilkan *hyperplane* [15].

$$y_i = w^T x_i + b, i = 1, 2, \dots, n \quad (2)$$

dengan w : vektor parameter bobot
 x_i : vektor variabel beban
 b : bias atau *error*
 $y_i \in \{-1, +1\}$: nilai target dari himpunan data x

Diduga dua kelas -1 (positif) dan +1 (negatif) terpisah dengan sempurna oleh *hyperplane* $w^T x + b = 0$ (Wang, 2005). Jika $w^T x_1 + b = +1$ ialah *hyperplane* pendukung ke kelas +1 (positif) dan $w^T x_2 + b = -1$ *hyperplane* pendukung ke kelas -1 (negatif), margin dihitung dengan cara mencari *space* antara kedua *hyperplane* pendukung kedua kelas [16].

$$w^T x_1 + b = +1$$

$$w^T x_2 + b = -1 \quad (3)$$

$$w^T (x_1 - x_2) = 2$$

$$\left(\frac{w}{\|w\|} (x_1 - x_2) \right) = \frac{2}{\|w\|}$$

Untuk mendapat *hyperplane* terbaik digunakan rumus *Quadratic Programming* (QP) problem yang adalah suatu bentuk persamaan optimal yang banyak digunakan dengan mengoptimalkan *invers* persamaan (3), yaitu:

$$\min \frac{1}{2} \|w\|^2, \text{ subject to } y_i = w^T x_i + b \geq 1, i = 1, 2, \dots, n \quad (4)$$

Pengoptimalan dilakukan dengan teknik *Lagrange Multiplier*, yang menghasilkan:

$$L_p(w, b, a) = \frac{1}{2} \|w\|^2 \sum_{i=1}^n a_i (y_i (w^T x_i + b) - 1) \quad (5)$$

Dengan a_i merupakan nilai Lagrange multiplier dengan nilai $a_i \geq 0$. Kemudian L_p diminimumkan dengan menurunkan parsial L_p terhadap w dan b lalu disamadengankan 0.

$$\frac{\partial}{\partial b} L_p(w, b, a) = 0 \rightarrow w = \sum_{i=1}^n a_i y_i x_i \quad (6)$$

$$\frac{\partial}{\partial b} L_p(w, b, a) = 0 \rightarrow \sum_{i=1}^n a_i y_i = 0 \quad (7)$$

$\|w\|^2$ dijabarkan berdasarkan berdasarkan persamaan (6)

$$\|w\|^2 = w^T w = \sum_{i=1}^n a_i y_i x_i^T \sum_{j=1}^n a_j y_j x_j = \sum_{i=1}^n \sum_{j=1}^n a_i a_j y_i y_j (x_i^T x_j) \quad (8)$$

L_p (*primal problem*) diganti ke dalam *dual problem* L_d dengan penggantian persamaan (6), (7), dan (8) ke persamaan L_p (5). Nilai optimal diperoleh dengan memaksimalkan L_p terhadap a untuk mendapat pemisahan bidang terbaik [15]:

$$\max L_d = \max \left(\sum_{i=1}^n a_i - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n y_i y_j a_i a_j (x_i^T x_j) \right) \tag{9}$$

dengan syarat $\sum_{i=1}^n a_i y_i = a_i \geq 0 (i = 1, 2, \dots, n)$, $a_i > 0$ adalah *support vector*, sisanya $a_i = 0$ tidak terletak di *hyperplane* [17].

1.5.2. Nonlinearly Separable Data

Secara umum dua lapisan *class* di *input space* tidak dapat dipisahkan sepenuhnya, sehingga SVM diformat ulang dengan memperkenalkan teknik *softmargin*. Dalam *softmargin*, persamaan (4) dimodifikasi dengan memasukkan variabel *slack* ξ_i ($\xi_i > 0$) [18].

$$y_i (w^T x_i + b) \geq 1 - \xi_i, i = 1, 2, \dots, n, \xi_i \geq 0$$

$$\min \tau (w, \xi) = \min \left(\frac{1}{2} \|w\|^2 + C \sum_{i=1}^n \xi_i \right) \tag{10}$$

Parameter C (*Cost*) dipilih untuk mengawasi *tradeoff* antara *margin* dan *error* klasifikasi ξ . Nilai C yang besar berarti akan memberikan penalti yang lebih besar untuk klasifikasi *error* tersebut (Nugroho et al., 2003). Bentuk *primal problem* menjadi persamaan (11)

$$L_p(w, b, \xi, a, \beta) = \frac{1}{2} \|w\|^2 + C \sum_{i=1}^n \xi_i - \sum_{i=1}^n a_i (y_i (w^T x_i + b)) - 1 + \xi_i - \sum_{i=1}^n \beta_i \xi_i \tag{11}$$

di mana a_i dan β_i merupakan *Lagrange multiplier*. Optimasi L_p dihitung dengan menurunkan L_p terhadap w, b, ξ secara parsial lalu disamadengankan 0, sehingga diperoleh sebagai berikut.

$$\frac{\partial L}{\partial w} = 0 \rightarrow w = \sum_{i=1}^n a_i y_i x_i \tag{12}$$

$$\frac{\partial L}{\partial b} = 0 \rightarrow \sum_{i=1}^n a_i y_i = 0 \tag{13}$$

$$\frac{\partial L}{\partial \xi} = 0 \rightarrow C - a_i - \beta_i = 0 \rightarrow a_i = C - \beta_i \tag{14}$$

dari persamaan (12), (13) dan (14) diperoleh persamaan L_d yang dimaksimalkan

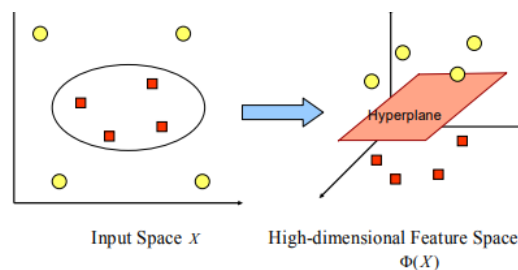
$$\max L_d = \max \left(\sum_{i=1}^n a_i - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n y_i y_j a_i a_j (x_i^T x_j) \right) \tag{15}$$

dengan batas $0 \leq a_i \leq C, i = 1, 2, 3, \dots, n$ dan $\sum_{i=1}^n a_i y_i = 0$

Karena masalah dunia nyata jarang bersifat linier, tetapi non linier, maka SVM dimodifikasi dengan menyisipkan fungsi kernel, yaitu fungsi yang sering digunakan untuk mengubah data masukan menjadi berdimensi tinggi sehingga dapat dipisahkan.

Dalam SVM *nonlinearly separable* data, data \vec{x} dipetakan oleh fungsi $\phi(\vec{x})$ ke dalam ruang vektor berdimensi lebih tinggi sehingga kedua kelas dapat dipisahkan secara linier oleh *hyperplane*. Notasi matematika dari pemetaan tersebut adalah sebagai berikut.

$$\phi: R^d \rightarrow R^q \quad d < q \tag{16}$$



Gambar 2. Fungsi Φ data dipetakan ke ruang dimensi tinggi

Selanjutnya, proses pembelajaran dalam SVM untuk mencari titik *support vector* hanya bergantung pada *dot product* dari data yang telah ditransformasikan ke dalam ruang berdimensi lebih tinggi, yaitu $\phi(x_i)\phi(x_j)$. Sering kali transformasi ϕ membuat bingung, sehingga *dot product* diganti dengan fungsi *kernel* yang biasa dikenal sebagai *kernel trick*.

$$K(x_i, x_j) = \phi(x_i)^T \phi(x_j) \tag{17}$$

2.6. Evaluation

Setelah proses pelatihan selesai dan nilai parameter *a* dan *b* sudah didapatkan, selanjutnya adalah proses pengujian untuk melihat hasil akurasi dan tingkat kesalahan sistem. Secara umum, untuk mengukur kinerja klasifikasi dilakukan dengan menggunakan *confussion matrix*. *Confussion matrix* adalah tabel catatan hasil dari proses klasifikasi [19]. Tabel 1 adalah contoh *confussion matrix* yang mengklasifikasikan masalah positif dan negatif menjadi dua kelas, yaitu kelas 1 dan kelas -1. Sel *True Positive* adalah jumlah data dari kelas 1 yang dipetakan dengan benar ke kelas 1 dan sel *False Positive* adalah data dari kelas 1 yang salah dipetakan ke kelas -1 dan begitu juga sebaliknya pada Sel *True Negative* adalah jumlah data di kelas -1 yang dipetakan dengan benar dan *False Negative* di kelas -1 yang salah dipetakan ke kelas 1.

Tabel 1. Tabel Matriks Konfusi

		Hasil prediksi	
		1	-1
Kelas Aktual	1	<i>True Positive</i> (TP)	<i>False Negative</i> (TN)
	-1	<i>False Positive</i> (FP)	<i>True Negative</i> (TN)

Berdasarkan isi tabel dari matriks konfusi maka dapat dihitung *accuracy*, *precision*, *recall* dan *f1-score* dari hasil klasifikasi yang telah dilakukan dengan menggunakan rumus sebagai berikut:

$$Akurasi = \frac{TP}{TP + FN} \times 100\%$$

$$Presisi = \frac{TP + TN}{TP + TN + FP + FN} \times 100\%$$

$$Recall = \frac{TP}{TP + FP} \times 100\%$$

$$F1score = 2 \times \frac{Presisi \times Recall}{Presisi + Recall}$$

HASIL DAN PEMBAHASAN

1.1. Pengumpulan Data

Data Quipper dalam penelitian ini dikumpulkan menggunakan metode *crawling* dengan bahasa pemrograman *python*. Pengambilan data ini prosesnya menggunakan API *google-play-scraper*. Hasil pengumpulan data dari aplikasi Quipper berjumlah 1000 data *ulasan*.

1.2. Preprocessing Data

Pada tahap ini proses modifikasi data dilakukan agar cocok dan dapat diproses pada langkah selanjutnya [20]. Berikut adalah contoh hasil dari 5 tahap *preprocessing* yang telah dilakukan:

a. *Case Folding*, proses mengubah huruf kapital ke huruf kecil.

Tabel 2. Tabel Contoh Hasil Case Folding

Dataset	Case Folding
Sangat membantu dan mudah untuk dipahami belajar kelas 9 di quipper	sangat membantu dan mudah untuk dipahami belajar kelas 9 di quipper

b. *Text cleaning*, proses menghilangkan karakter atau tanda baca yang tidak perlu.

Tabel 3. Tabel Contoh Hasil Text Cleaning

Dataset	Text Cleaning
sangat membantu dan mudah untuk dipahami belajar kelas 9 di quipper	sangat membantu dan mudah untuk dipahami belajar kelas di quipper

c. *Tokenization*, proses memisahkan teks atau kata yang ada di satu kalimat atau paragraf berdasarkan karakter spasi.

Tabel 4. Tabel Contoh Hasil Tokenization

Dataset	Tokenization
sangat membantu dan mudah untuk dipahami belajar kelas di quipper	['sangat', 'membantu', 'dan', 'mudah', 'untuk', 'dipahami', 'belajar', 'kelas', 'di', 'quipper']

d. *Filtering*, proses menghilangkan kata-kata yang dianggap tidak perlu.

Tabel 5. Tabel Contoh Hasil Filtering

Dataset	Filtering
sangat membantu dan mudah untuk dipahami belajar kelas di quipper	sangat membantu mudah dipahami belajar kelas quipper

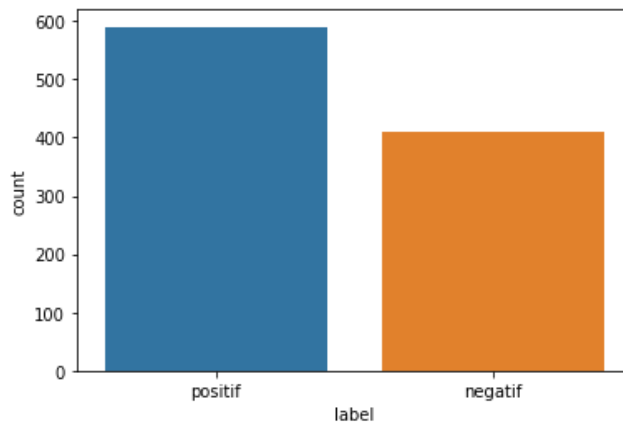
e. *Stemming*, proses melakukan perubahan kata menjadi kata dasar.

Tabel 6. Tabel Contoh Hasil Stemming

Dataset	Stemming
sangat membantu mudah dipahami belajar kelas quipper	sangat bantu mudah paham ajar kelas quipper

1.3. Labeling Data

Data dari *preprocessing dataset* Quipper yang berjumlah 1000, menghasilkan pelabelan data yang positif sebanyak 590, dan 410 data dengan label negatif yang bisa dilihat pada gambar 3 di bawah ini.



Gambar 3. Grafik Label Data

1.4. Transormation Data

Setelah data sudah diberi label, selanjutnya melakukan pembobotan kata dengan metode TF-IDF. Gambar 4 adalah contoh hasil bobot kata dengan TF-IDF pada *data training*.

Row No.	label	text	score	a	aakses	aamin	abis	abstrak	acu	ad	ada
1	positif	cukup bantu ajar moga upd...	4	0	0	0	0	0	0	0	0
2	negatif	aplikasi goblok hapus aplik...	1	0	0	0	0	0	0	0	0
3	positif	mudah paham sampai mat...	5	0	0	0	0	0	0	0	0
4	negatif	banyak bug klo matematika ...	1	0	0	0	0	0	0	0	0
5	positif	tampil aplikasi sangat bagu...	5	0	0	0	0	0	0	0	0
6	negatif	tampil video beri kurang tari...	1	0	0	0	0	0	0	0	0
7	negatif	woy apk apa sih ga jelas m...	1	0	0	0	0	0	0	0	0
8	positif	sebenemnya baguss banget ...	5	0	0	0	0	0	0	0	0
9	positif	aplikasi bagus cocok murid ...	5	0	0	0	0	0	0	0	0
10	negatif	tiap soal yg gambar nya ga ...	2	0	0	0	0	0	0	0	0
11	negatif	volume full suara terlalu kec...	3	0	0	0	0	0	0	0	0
12	negatif	gambar loading terus gak ...	1	0	0	0	0	0	0	0	0
13	negatif	quipper tugas ga buka udah...	2	0	0	0	0	0	0	0	0
14	negatif	aplikasi quipper nya buat ze...	3	0	0	0	0	0	0	0	0
15	negatif	kadang beri tugas guru tuga...	1	0	0	0	0	0	0	0	0
16	negatif	harus yang menggubakan ...	1	0	0	0	0	0	0	0	0
17	positif	kak notifikasi chatnya sedia ...	4	0	0	0	0	0	0	0	0
18	negatif	tiap masuk email sliaah teru...	2	0	0	0	0	0	0	0	0
19	negatif	guru ga asik klo kasih tau h...	3	0	0	0	0	0	0	0	0
20	negatif	punya kok tiba gak buka ma...	3	0	0	0	0	0	0	0	0
21	negatif	benar aplikasi bantu untuk t...	1	0	0	0	0	0	0	0	0
22	negatif	sangat jelek tugas guru mu...	1	0	0	0	0	0	0	0	0
23	negatif	kurang suka ngomongnya ti...	1	0	0	0	0	0	0	0	0
24	positif	terima kasih quipper saya a...	5	0	0	0	0	0	0	0	0
25	positif	video materi sering macet p...	4	0	0	0	0	0	0	0	0

Gambar 4. Data Hasil dari TFIDF

Dalam langkah *transformation* data di bagi menjadi 2 data yaitu *data training* dan *data testing* dalam 3 skenario. Data ini dipisahkan menggunakan fungsi *train test split*. 3 skenario pemisahan data bisa dilihat pada Tabel 7.

Tabel 7. Tabel Skenario Pemisahan Data

Skenario Rasio Perbandingan	Data Training	Data Testing
80:20	800	200
70:30	700	300
60:40	600	400

1.5. Klasifikasi Support Vector Machine

Setelah melakukan tahap *preprocessing* dan perhitungan bobot dengan *tf-idf*, selanjutnya dilakukan proses *training* dengan penerapan algoritma *support vector machine*. Pada tahap ini digunakan metode *Support Vector Machine* (SVM) dengan fungsi *kernel* yang digunakan sebagai *kernel* linier. Pada *kernel* linier terdapat satu parameter yang akan diuji yaitu nilai *Cost* (C) dengan nilai parameter C : 0,01; 0,2; 0,4; 0,6; 1 untuk *data training*. Hasil dari pengujian pengaruh nilai *Cost* pada model linier ditunjukkan pada Tabel 8. Penelitian ini menggunakan 10 validasi silang (*cross-validation*) untuk menguji performa *machine* dalam melakukan klasifikasi data.

Tabel 8. Tabel Performa nilai C

Evaluasi Model	Cost (C)				
	0.01	0.25	0.5	0.75	1
Akurasi Keseluruhan	0.87	0.88	0.89	0.89	0.88

1.6. Evaluation

Langkah evaluasi ini dilakukan menggunakan *confusion matrix*. Perbandingan dari semua hasil evaluasi menggunakan *confusion matrix* dari seluruh skenario klasifikasi menggunakan algoritma *Support Vector Machine kernel* linier ditampilkan pada Tabel 9.

Tabel 9. Tabel Hasil Matriks Konfusi

Skenario	Accuracy	Precision	Recall	F1-score
80:20	87.0%	81.58%	94.90%	87.73%
70:30	89.33%	84.85%	95.24%	89.74%
60:40	90.25%	86.85%	94.39%	90.46%

Tabel 9 menunjukkan bahwa hasil evaluasi terbaik yang diperoleh pada skenario 3 menghasilkan *accuracy* sebesar 90,25%, *precision* 86,85%, *recall* 94,39%, dan *f1-score* 90,46%.

KESIMPULAN DAN SARAN

Kesimpulan

Setelah dilakukannya penelitian ini, dapat ditarik kesimpulan sebagai berikut, sentimen masyarakat atau pengguna terhadap aplikasi Quipper menghasilkan 410 ulasan negatif 590 dan ulasan positif. Hasil evaluasi kinerja algoritma *Support Vector Machine kernel* linier dalam klasifikasi atau analisis sentimen ulasan aplikasi *e-learning* Quipper menggunakan tiga skenario *split data*, diperoleh hasil terbaik di skenario ketiga dengan rasio perbandingan data *training* 60% dan data testing 40% dan menghasilkan *accuracy* sebesar 90,25%, *precision* 86,85%, *recall* 94,39%, dan *f1-score* 90,46%.

Referensi

- [1] Dinas Kesehatan Kabupaten Kendal, "Kenalan Dengan Covid-19," Pemerintah Kabupaten Kendal, 2021. <https://corona.kendalkab.go.id/berita/profil/kenalan-dengan-covid-19> (accessed May 21, 2022).
- [2] Kevin Adrian, "Pentingnya Menerapkan Social Distancing Demi Mencegah COVID-19," Alodokter.com, 2020. <https://www.alodokter.com/pentingnya-menerapkan-social-distancing-demi-mencegah-covid-19> (accessed May 21, 2022).

- [3] N. Fitriani, "Pembelajaran E-Learning Sebagai Salah Satu Strategi Pembelajaran di Era Digitalisasi," *Dinkes ntbprov.co.id*, 2020. <https://dinkes.ntbprov.go.id/berita/pembelajaran-e-learning-sebagai-salah-satu-strategi-pembelajaran-di-era-digitalisasi/> (accessed May 21, 2022).
- [4] B. S. Amalia, Y. Umidah, and R. Mayasari, "Analisis Sentimen Review Pelanggan Restoran Menggunakan Algoritma Support Vector Machine Dan K-Nearest Neighbor," *SITEKIN J. Sains, Teknol. dan Ind.*, vol. 19, no. 1, pp. 28–34, 2021, doi: 10.24014/sitekin.v19i1.14861.
- [5] R. Wahyudi and G. Kusumawardana, "Analisis Sentimen pada Aplikasi Grab di Google Play Store Menggunakan Support Vector Machine," *J. Inform.*, vol. 8, no. 2, pp. 200–207, 2021, doi: 10.31294/ji.v8i2.9681.
- [6] L. Ardiani, H. Sujaini, and T. Tursina, "Implementasi Sentiment Analysis Tanggapan Masyarakat Terhadap Pembangunan di Kota Pontianak," *J. Sist. dan Teknol. Inf.*, vol. 8, no. 2, 2020, doi: 10.26418/justin.v8i2.36776.
- [7] R. L. Musyarofah, E. U. Utami, and S. R. Raharjo, "Analisis Komentar Potensial pada Social Commerce Instagram Menggunakan TF-IDF," *J. Eksplora Inform.*, vol. 9, no. 2, 2020, doi: 10.30864/eksplora.v9i2.360.
- [8] S. Khairunnisa, A. Adiwijaya, and S. Al Faraby, "Pengaruh Text Preprocessing terhadap Analisis Sentimen Komentar Masyarakat pada Media Sosial Twitter (Studi Kasus Pandemi COVID-19)," *J. MEDIA Inform. BUDIDARMA*, vol. 5, no. 2, 2021, doi: 10.30865/mib.v5i2.2835.
- [9] N. Saputra, "ANALISIS SENTIMEN DENGAN PREPROCESSING KATA (SENTIMENT ANALISYS WITH LEXICON PREPROCESSING)," *J. Din. Inform.*, vol. 7, no. 1, 2018.
- [10] R. L. Atimi and Enda Esyudha Pratama, "Implementasi Model Klasifikasi Sentimen Pada Review Produk Lazada Indonesia," *J. Sains dan Inform.*, vol. 8, no. 1, pp. 88–96, Jul. 2022, doi: 10.34128/jsi.v8i1.419.
- [11] F. Fathonah and A. Herliana, "Penerapan Text Mining Analisis Sentimen Mengenai Vaksin Covid - 19 Menggunakan Metode Naïve Bayes," *J. Sains dan Inform.*, vol. 7, no. 2, 2021, doi: 10.34128/jsi.v7i2.331.
- [12] G. Sani, G. Prawira, and H. Setiaji, *PENERAPAN DATA TRANSFORMATION PADA DATABASE SISTEM INFORMASI MANAJEMEN RUMAH SAKIT*. 2020.
- [13] E. Sugiarto and F. Budiman, "Optimasi Metode Support Vector Machine dengan Discrete Wavelet Transform Untuk Pengenalan Karakter Plat Nomor Kendaraan," *J. Transform.*, vol. 18, no. 2, 2021, doi: 10.26623/transformatika.v18i2.2694.
- [14] S. Budi, *Data Mining Teknik Pemanfaatan Data untuk Keperluan Bisnis*. 2007.
- [15] J. Han, M. Kamber, and J. Pei, *Data Mining: Concepts and Techniques*. 2012. doi: 10.1016/C2009-0-61819-5.
- [16] D. Lusiyanti and N. Nacong, "SISTEM SEDERHANA UNTUK MEMPREDIKSI RISIKO PEMBERIAN KREDIT," *J. Ilm. Mat. DAN Terap.*, vol. 15, no. 2, 2018, doi: 10.22487/2540766x.2018.v15.i2.11360.
- [17] A. S. Nugroho, A. B. Witarto, and D. Handoko, "Support Vector Machine – Teori dan Aplikasinya dalam Bioinformatika," *Kuliah Umum IlmuKomputer.Com*, 2003.
- [18] A. I. S. Azis, V. Suhartono, and H. Himawan, "Model Multi Class SVM Menggunakan Strategi 1V1 untuk Klasifikasi Wall-Following Robot Navigation Data," *J. Teknol. Inf.*, vol. 13, no. 2, 2017.
- [19] N. K. Fitriyani and A. D. Hartanto, "Analisis Sentimen Terhadap Tokoh Publik Menggunakan Support Vector Machine," *MEANS (Media Inf. Anal. dan Sist.*, 2020, doi: 10.54367/means.v5i1.615.
- [20] M. Diki Hendriyanto, A. A. Ridha, and U. Enri, "ANALISIS SENTIMEN ULASAN APLIKASI MOLA PADA GOOGLE PLAY STORE MENGGUNAKAN ALGORITMA SUPPORT VECTOR MACHINE SENTIMENT ANALYSIS OF MOLA APPLICATION REVIEWS ON GOOGLE PLAY STORE USING SUPPORT VECTOR MACHINE ALGORITHM," *J. Inf. Technol. Comput. Sci.*, vol. 5, no. 1, 2022, doi: 10.31539/intecom.v5i1.3708.