ISSN: 2339-2541
JURNAL GAUSSIAN, Volume 10, Nomor 3, Tahun 2021, Halaman 346 - 358
Online di: https://ejournal3.undip.ac.id/index.php/gaussian/

ANALISIS SENTIMEN ULASAN APLIKASI *TIKTOK* DI *GOOGLE PLAY* MENGGUNAKAN METODE *SUPPORT VECTOR MACHINE* (SVM) DAN ASOSIASI

Sola Fide[1], Suparti[2], Sudarno[3]

[1,2,3] Departemen Statistika, Fakultas Sains dan Matematika, Universitas Diponegoro

solafide34@gmail.com

ABSTRACT

Corona virus pandemic requires people to do activities from home so the number of internet usage in Indonesia has increased because information is carried out through social media. One of the popular social media in Indonesia is TikTok. However, the Tiktok's popularity cannot be separated from the footsteps of TikTok in Indonesia which was blocked by government for committing many violations. Each application allows users to provide a review about the application. To find out the users TikTok's sentiment, sentiment analysis was carried out to classify reviews into positive and negative sentiments. Classification is carried out using the Support Vector Machine (SVM) with kernel Radial Basis Function (RBF) method which is more effective classification algorithm and kernel function, seen from previous studies. The parameters used in the SVM gamma default 0.0004255 and the Cost (C) parameter experiment used is 0,01; 0,1; 1; 10; 100; 1000. The results can provide information that can be retrieved using the association method. The steps are scrapping data, data preprocessing, sentiment scoring, TF-IDF weighting, classifying using the SVM RBF kernel method and text association. Evaluation of the model using a confusion matrix with the value of accuracy and kappa. The greater the value of accuracy and kappa, the better the performance of the classification model. The review classification resulted in the best accuracy rate of 90.62% and the best kappa of 81.24% which means that it includes an almost perfect classification result. Based on the data association, positive reviews are given because users like and are comfortable with the current version of TikTok which contains funny videos on fyp. Meanwhile, negative reviews were given because the user failed to register and his account was blocked, so the user asked TikTok to continue to make improvements.

Keywords: TikTok, sentiment analysis, Suport Vector Machine (SVM), TF-IDF, accuracy, kappa, association

1. PENDAHULUAN

Pandemi virus corona menyebabkan terjadinya perubahan di masyarakat. Perubahan didukung dengan berkembangnya teknologi komunikasi yang membuat angka penggunaan internet meningkat yang mendorong persebaran informasi dengan sangat cepat melalui media social. Salah satu media sosial yang populer di Indonesia adalah aplikasi *TikTok*. Kepopuleran aplikasi *TikTok* tersebut tidak bisa lepas dari jejak *TikTok* di Indonesia yang sempat diblokir oleh pemerintah karena dianggap banyak melakukan pelanggaran, seperti pornografi, pelecehan agama, dan lainnya (CNN, 2018). Jumlah pengguna *TikTok* di Indonesia makin meningkat terlebih di masa pandemi. Menurut laporan perusahaan riset pasar aplikasi *mobile Sensor Tower*, pada periode Juli 2020, jumlah pengguna TikTok di Indonesia sebanyak 8,5 persen, terbanyak kedua setelah Amerika sebanyak 9,7 persen (Liputan 6, 2020). Berdasarkan pernyataan di atas, penulis akan meneliti bagaimana sentimen ulasan pengguna *TikTok* di *Google Play* dengan analisis sentimen.

Pada penelitian ini dilakukan klasifikasi ulasan pengguna *TikTok* ke dalam dua sentimen, yaitu sentimen positif dan sentimen negatif. Pengklasifikasian dilakukan dengan metode *Support Vector Machine* (SVM). SVM adalah algoritma yang sudah mendapatkan pengakuan luas untuk klasifikasi dengan akurasi yang baik. Umumnya, masalah dalam dunia nyata jarang yang bersifat *linear separable*. Untuk menyelesaikan *problem nonlinear*, SVM dimodifikasi dengan memasukkan fungsi *kernel* RBF, yang bisa membuat klasifikasi menggunakan SVM dapat beroperasi dalam ruang dimensi tinggi (Wang, 2005). Selanjutnya dilakukan eksplorasi data ulasan menggunakan metode asosiasi teks untuk mencari kata-kata berhubungan yang dapat memberikan informasi hubungan antarkata dalam ulasan sebagai bahan rujukan dalam proses evaluasi aplikasi (Santosa dan Nugroho, 2019).

2. TINJAUAN PUSTAKA

2.1. Analisis Sentimen

Analisis sentimen adalah metode untuk mengekstrak opini dan sentimen dari teks bahasa alami menggunakan metode komputasi. Opini dan sentimen terkait dengan evaluasi, penilaian, sikap, pengaruh, emosi, dan suasana hati (Liu, 2015).

2.2. *Text Mining*

Text mining adalah proses mengekstrak informasi melalui identifikasi dan eksplorasi pola yang menarik dari sumber data berupa kumpulan dokumen data tekstual tidak terstruktur. *Preprocessing* data dilakukan untuk mengubah data tidak terstruktur menjadi lebih terstruktur sehingga data siap diproses. *Preprocessing* yang dilakukan adalah *case folding, cleaning,* dan normalisasi kata. *Case folding* yaitu proses penyeragaman bentuk huruf pada dokumen. *Cleaning* yaitu penghapusan karakter selain yang ditentukan, seperti huruf atau karakter di luar dari alfabet a-z, tanda baca, angka, dan *emoticon*. Normalisasi kata yaitu perbaikan kata yang tidak sesuai pedoman (Fatmawati dan Affandes, 2017).

2.3. *Sentiment Scoring*

Sentiment scoring dilakukan dengan menggunakan kamus sentimen dan *boosterwords*. Kamus sentimen berisi kumpulan kata yang telah diberi kekuatan sentimen, yaitu kekuatan negatif antara -1 hingga -5 dan kekuatan positif antara +1 hingga +5. *Boosterwords* adalah kata yang dapat meningkatkan atau mengurangi intensitas sentimen kata di sebelahnya dengan bobot 1-2 (Wahid dan Azhari, 2016).

2.4. *Feature Selection*

Feature selection merupakan tahapan untuk mengurangi dimensi data tekstual sehingga hasil *text mining* berkualitas lebih baik. *Feature selection* yang dilakukan adalah *stopword removal*, *stemming*, dan *tokenizing*. *Stopword removal* adalah tahap menghilangkan kata-kata yang dianggap tidak menggambarkan isi kalimat. *Stemming* adalah tahap menghilangkan imbuhan kata untuk mendapatkan kata dasar. *Tokenizing* adalah tahap pemisahan setiap kata dalam suatu kalimat (Indraloka dan Santosa, 2017).

2.5. Pembobotan *Term Frequency-Inverse Document Frequency* (TF-IDF)

Term frequency (tf) merupakan sistem pembobotan yang mengukur frekuensi kemunculan istilah dalam dokumen. Semakin tinggi tf berarti *term* sering muncul maka dapat dianggap *term* umum sehingga tidak penting nilainya. *Inverse Document Frequency* (idf) merupakan frekuensi berbanding terbalik. Persamaan TF-IDF (Christian *et al*., 2016).

$$TF = \frac{jumlah\ term\ \text{di}\ dok}{jumlah\ seluruh\ term\ \text{di}\ dok}$$

$$IDF = \log_2 \frac{jumlah\ seluruh\ dok}{jumlah\ dok\ pada\ term}$$

$$TF\ IDF = TF \times IDF \tag{2.1}$$

2.6. *Support Vector Machine* (SVM)

SVM adalah metode *supervised learning* yang menghasilkan fungsi pemetaan berupa fungsi klasifikasi melalui pembuatan *hyperplane* dengan margin maksimum (Wang, 2005). Margin adalah jarak antara *hyperplane* dengan *pattern* terdekat masing-masing kelas. *Pattern* yang paling dekat ini disebut sebagai *support vector* (Nugroho et al., 2003).

2.6.1 SVM *Linearly Separable Data*

Linearly separable data adalah data yang dapat dipisahkan secara linear. Misalkan $x_i \in R^d$ dan label kelas sebagai $y_i \in \{-1, +1\}$ untuk $i = 1, 2, ..., n$, n adalah banyaknya data. Model linier metode SVM untuk menghasilkan *hyperplane* (Wang, 2005).

$$y_i = \mathbf{w}^T \mathbf{x}_i + b, \quad i = 1, 2, .., n \tag{2.2}$$

dengan $\mathbf{w}$: vektor parameter bobot

$\mathbf{x}_i$: vektor variabel bebas

b : bias atau *error*

$y_i \in \{-1, +1\}$: nilai target dari himpunan data $\boldsymbol{x}$

Diasumsikan kedua kelas -1 (positif) dan +1 (negatif) terpisah secara sempurna oleh *hyperplane* $\mathbf{w}^T\mathbf{x} + b = 0$ (Wang, 2005). Jika $\mathbf{w}^T\mathbf{x_1} + b = +1$ adalah *hyperplane*-pendukung dari kelas +1 (positif) dan $\mathbf{w}^T\mathbf{x_2} + b = -1$ *hyperplane*-pendukung dari kelas −1 (negatif), margin dihitung dengan mencari jarak kedua *hyperplane*-pendukung kedua kelas.

$$\begin{aligned} &\mathbf{w}^T\mathbf{x_1} + b = +1 \\ &\mathbf{w}^T\mathbf{x_2} + b = -1 \quad - \\ &\mathbf{w}^T(\mathbf{x_1} - \mathbf{x_2}) = 2 \\ &\left(\frac{\mathbf{w}}{\|\mathbf{w}\|}(\mathbf{x_1} - \mathbf{x_2})\right) = \frac{2}{\|\mathbf{w}\|} \end{aligned} \tag{2.3}$$

Untuk mendapat *hyperplane* terbaik digunakan rumus *Quadratic Programming (QP) problem* yang merupakan salah satu bentuk persamaan optimasi yang banyak digunakan dengan mengoptimalkan invers persamaan (2.3), yaitu

$$\min \frac{1}{2}\|\mathbf{w}\|^2 \ , subject\ to\ y_i\left(\mathbf{w}^T\mathbf{x}_i + b\right) \geq 1, i = 1, 2, \ .., n \tag{2.4}$$

Optimalisasi dipecahkan dengan teknik *Lagrange Multiplier*, yang menghasilkan

$$L_p(\mathbf{w}, b, \boldsymbol{\alpha}) = \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{i=1}^{n} \alpha_i \left(y_i\left(\mathbf{w}^T\mathbf{x}_i + b\right) - 1\right) \tag{2.5}$$

dengan α_i merupakan nilai *Lagrange multiplier* dengan nilai $\alpha_i \geq 0$. Kemudian L_p diminimumkan dengan menurunkan parsial L_p terhadap $\mathbf{w}$ dan b lalu disamadengankan 0.

$$\frac{\partial}{\partial b} L_p(\mathbf{w}, b, \boldsymbol{\alpha}) = 0 \Rightarrow \mathbf{w} = \sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i \tag{2.6}$$

$$\frac{\partial}{\partial \mathbf{w}} L_p(\mathbf{w}, b, \boldsymbol{\alpha}) = 0 \Rightarrow \sum_{i=1}^{n} \alpha_i y_i = 0 \tag{2.7}$$

$\| \mathbf{w} \|^2$ dijabarkan berdasarkan persamaan (2.6)

$$\| \mathbf{w} \|^2 = \mathbf{w}^T \mathbf{w} = \sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i^T \sum_{j=1}^{n} \alpha_j y_j \mathbf{x}_j = \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j \left(\mathbf{x}_i^T \mathbf{x}_j \right) \tag{2.8}$$

L_p (*primal problem*) diubah ke dalam *dual problem* L_d dengan mensubsitusikan persamaan (2.6), (2.7), dan (2.8) ke persamaan L_p (2.5). Nilai optimal diperoleh dengan memaksimalkan L_d terhadap α untuk mendapat pemisahan bidang terbaik (Wang, 2005):

$$\max L_d = \max \left(\sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} y_i y_j \alpha_i \alpha_j \left(\mathbf{x}_i^T \mathbf{x}_j \right) \right), \text{ dengan syarat} \sum_{i=1}^{n} \alpha_i y_i = 0, \alpha_i \geq 0 (i = 1, 2, ..., n) \tag{2.9}$$

$\alpha_i > 0$ adalah *support vector*, sisanya $\alpha_i = 0$ tidak terletak di *hyperplane* (Nugroho *et al.*, 2003).

2.6.2 *SVM Nonlinearly Separable Data*

Umumnya dua buah *class* pada *input space* tidak dapat terpisah secara sempurna, maka SVM dirumuskan ulang dengan memperkenalkan teknik *softmargin*. Dalam *softmargin*, persamaan (2.4) dimodifikasi dengan memasukkan *slack variabel* $\xi_i \quad (\xi_i > 0)$.

$$\begin{aligned} & y_i \left(\mathbf{w}^T \mathbf{x}_i + b \right) \geq 1 - \xi_i, i = 1, 2, .., n, \xi_i \geq 0 \\ & \min_w \tau(\mathbf{w}, \boldsymbol{\xi}) = \min \left(\frac{1}{2} \| w \|^2 + C \sum_{i=1}^{n} \xi_i \right) \end{aligned} \tag{2.10}$$

Parameter C (*Cost*) dipilih untuk mengontrol *tradeoff* antara margin dan *error* klasifikasi $\boldsymbol{\xi}$. Nilai C yang besar berarti akan memberikan penalti yang lebih besar terhadap *error* klasifikasi tersebut (Nugroho *et al.*, 2003). Bentuk *primal problem* menjadi persamaan (2.11)

$$L_p(\mathbf{w}, b, \boldsymbol{\xi}, \boldsymbol{\alpha}, \boldsymbol{\beta}) = \frac{1}{2} \| \mathbf{w} \|^2 + C \sum_{i=1}^{n} \xi_i - \sum_{i=1}^{n} \alpha_i \left(y_i \left(\mathbf{w}^{\mathbf{T}} \mathbf{x}_{\mathbf{i}} + b \right) \right) - 1 + \xi_i - \sum_{i=1}^{n} \beta_i \xi_i$$

dimana α_i dan β_i merupakan *Lagrange multiplier*. Optimasi L_p dihitung dengan menurunkan L_p terhadap $\mathbf{w}, \boldsymbol{\xi}, b$ secara parsial lalu disamadengankan 0, sehingga diperoleh sebagai berikut.

$$\frac{\partial L}{\partial \mathbf{w}} = 0 \Rightarrow \mathbf{w} = \sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_{\mathbf{i}} \tag{2.12}$$

$$\frac{\partial L}{\partial b} = 0 \Rightarrow \sum_{i=1}^{n} \alpha_i y_i = 0 \tag{2.13}$$

$$\frac{\partial L}{\partial \xi} = 0 \Rightarrow C - \alpha_i - \beta_i = 0 \Rightarrow \alpha_i = C - \beta_i \tag{2.14}$$

dari persamaan (2.12), (2.13) dan (2.14) diperoleh persamaan L_d yang dimaksimalkan

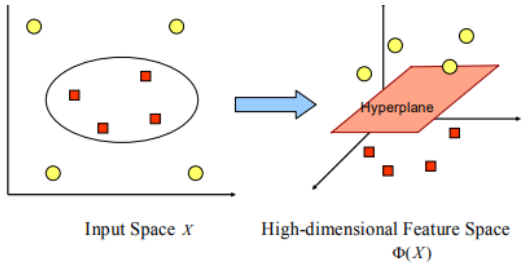
$$\max L_d = \max \left(\sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} y_i y_j \alpha_i \alpha_j \left(\mathbf{x}_i^T \mathbf{x}_j \right) \right) \tag{2.15}$$

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

Karena masalah dunia nyata jarang bersifat *linear*, tetapi *nonlinear*, maka SVM dimodifikasi dengan memasukkan fungsi *kernel*, yaitu fungsi yang sering digunakan untuk mengubah data masukan menjadi berdimensi tinggi sehingga dapat dipisahkan (Wang, 2005).

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

$$\Phi : R^d \rightarrow R^q \qquad d < q \tag{2.17}$$



Gambar 1. Fungsi Φ memetakan data ke ruang dimensi tinggi (Nugroho *et al*., 2003)

Selanjutnya proses pembelajaran pada SVM dalam menemukan titik-titik *support vector*, hanya bergantung pada *dot product* dari data yang sudah ditransformasikan pada ruang berdimensi lebih tinggi, yaitu $\Phi(x_i)\Phi(x_j)$. Umumnya transformasi Φ sulit untuk dipahami, maka perhitungan *dot product* digantikan fungsi kernel yang biasa disebut *Kernel Trick*.

$$K(x_i, x_j) = \Phi(x_i)^T \Phi(x_j) \tag{2.18}$$

Fungsi kernel yang umum digunakan salah satunya kernel *Radial Basis Function* (RBF).

$$K(x_i, x_j) = \exp\left(-\gamma \| x_i - x_j \|^2\right) \tag{2.19}$$

dengan x_i dan x_j adalah pasangan dua data. d adalah *degree*, makin tinggi maka batas keputusan yang lebih fleksibel. γ adalah *gamma*, yaitu tingkat pembelajaran yang mengatur lebar kurva berbentuk lonceng, jika makin besar maka kurva makin sempit.

Selanjutnya hasil klasifikasi dari data x diperoleh dari persamaan berikut (Nugroho *et al.*, 2003)

$$f(\Phi(\mathbf{x})) = sign\left(\sum_{i=1}^{ns} \alpha_i y_i K(\mathbf{x}, \mathbf{x}_i) + b\right)$$

$$f(\Phi(\mathbf{x})) = \begin{cases} 1, jika \sum_{i=1}^{ns} \alpha_i y_i K(\mathbf{x}, \mathbf{x}_i) + b \geq 0 \\ -1, jika \sum_{i=1}^{ns} \alpha_i y_i K(\mathbf{x}, \mathbf{x}_i) + b < 0 \end{cases} \tag{2.20}$$

dengan $f(\Phi(\mathbf{x}))$: hasil klasifikasi dari data $\mathbf{x}$

y_i : kelas data

α_i : koefisien *lagrange*

$K(\mathbf{x}, \mathbf{x}_i)$: fungsi *kernel* data uji dan data latih

b : bias

ns : banyak *support vector*

2.7. Evaluasi Sistem Klasifikasi

Suatu sistem klasifikasi harus dievaluasi agar dapat diketahui tingkat akurasi dari prediksi yang dihasilkan oleh sistem klasifikasi tersebut. Untuk proses evaluasi akurasi klasifikasi dilakukan dengan memperhatikan *confusion matrix*. *Confusion matrix* digambarkan dengan tabel jumlah data uji yang benar dan salah diklasifikasikan (Rahman *et al*., 2017).

Tabel 1. *Confusion Matrix (Rahman et al., 2017)*

Corect Classification	***Classified as***	
	Predicted "+"	***Predicted "-"***
Actual "+"	*True Positives*	*False Negatives*
Actual "-"	*False Positives*	*True Negatives*

dengan *True Positives* (TP) : banyak *record* data positif yang diklasifikasikan positif
False Positives (FP) : banyak *record* data negatif yang diklasifikasikan positif
False Negatives (FN) : banyak *record* data positif yang diklasifikasikan negatif
True Negatives (TN) : banyak *record* data negatif yang diklasifikasikan negatif.

Tabel 2. Ukuran Evaluasi Model Klasifikasi

Ukuran	**Rumus**
Accuracy	$\frac{TP+TN}{TP+TN+FP+FN}$
Kappa statistic	$P_0 = Accuracy$ $P_c = \left[\left(\frac{TP+FP}{Total}\right)\left(\frac{TP+FN}{Total}\right)\right]+\left[\left(\frac{FN+TN}{Total}\right)\left(\frac{FP+TN}{Total}\right)\right]$ $K = \frac{P_0 - P_c}{(1-P_c)}$

2.8. Asosiasi

Asosiasi adalah hubungan antara dua variabel, yaitu antarkata pada ulasan sehingga mendapat informasi sebagai bahan rujukan dalam proses evaluasi aplikasi. Nilai asosiasi yang besar berarti korelasi semakin besar dan menunjukkan kata-kata tersebut sering muncul bersamaan dalam satu kalimat (Santosa dan Nugroho, 2019).

$$r_{xy} = \frac{n\left(\sum_{i=1}^{n} X_i Y_i\right) - \left(\sum_{i=1}^{n} X_i\right)\left(\sum_{i=1}^{n} Y_i\right)}{\sqrt{\left\{n\sum_{i=1}^{n} X_i^2 - \left(\sum_{i=1}^{n} X_i\right)^2\right\}\left\{n\sum_{i=1}^{n} Y_i^2 - \left(\sum_{i=1}^{n} Y_i\right)^2\right\}}} \quad (2.21)$$

dengan r_{xy} : Nilai korelasi antara variabel x dan variabel y
n : Banyaknya pasangan data x dan y
$\sum_{i=1}^{n} X_i$: Jumlah nilai pada variabel x atau variabel pertama
$\sum_{i=1}^{n} Y_i$: Jumlah nilai pada variabel y atau variabel kedua
$\sum_{i=1}^{n} X_i^2$: Kuadrat dari total nilai variabel x
$\sum_{i=1}^{n} Y_i^2$: Kuadrat dari total nilai variabel y
$\sum_{i=1}^{n} X_i Y_i$: Jumlah dari hasil perkalian antara nilai variabel x dan y

3. METODE PENELITIAN

3.1. Sumber Data dan Variabel Penelitian

Jenis data yang digunakan adalah data sekunder, yaitu data ulasan bahasa Indonesia aplikasi *TikTok* di *Google Play Store* pada bulan September 2020 hingga Februari 2021 sebanyak 3200 ulasan. Pengumpulan data menggunakan teknik *web scraping*, yaitu pengambilan sebuah dokumen semi-terstruktur dari halaman web (Ayani *et al.*, 2019). Variabel yang digunakan adalah tanggal memberi ulasan, nama akun *Google*, dan ulasan pengguna.

3.2. Langkah-langkah Analisis

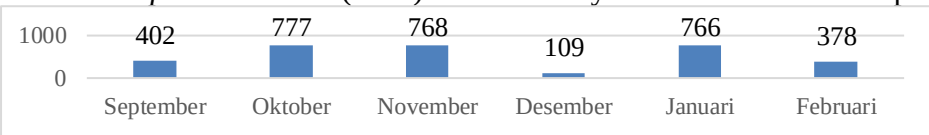
Penelitian dilakukan dengan bantuan *software Data Miner* untuk *web scrapping* dan *Rstudio* untuk analisis data. Adapun tahapan-tahapan analisis yang dilakukan:

1. *Scraping data* ulasan aplikasi *TikTok* di *Google Play*
2. Analisis deskriptif data ulasan aplikasi *TikTok* di *Google Play*.
3. *Pre-processing data (case folding, cleaning*, dan normalisasi kata).
4. Pelabelan data dengan *sentiment scoring*.
5. *Feature selection* (*stopwords*, *stemming*, dan *tokenizing*)
6. Pembobotan TF-IDF
7. Pembuatan data latih dan data uji dengan perbandingan 80%:20%
8. Membangun model klasifikasi *Suport Vector Machine* dengan fungsi *kernel* RBF
9. Menghitung akurasi dan *kappa* menggunakan *confusion matrix* dari data uji untuk mengevaluasi model klasifikasi
10. Membuat asosiasi teks
11. Interpretasi hasil.

4. HASIL DAN PEMBAHASAN

4.1. Pengumpulan Data

Pengumpulan data ulasan *TikTok* di *Google Play* menggunakan metode *scraping data* dengan aplikasi *Data Miner*. Data yang diambil berisi tanggal, nama, dan ulasan yang disimpan dalam bentuk *Comma Separated Value* (CSV). Berikut banyak data ulasan *TikTok* per bulan.



Gambar 2. Histogram banyak *ulasan TikTok* per bulan

4.2. *Pre-Processing Data*

Preprocessing data dilakukan untuk mengubah data tidak terstruktur menjadi terstruktur

Tabel 3. Contoh proses *pre-processing data*

Proses	Sebelum	Sesudah
Case folding	Apk ini bagus banget!!!!!!!!!!!!!!	apk ini bagus banget!!!!!!!!!!!!!!
Cleaning	apk ini bagus banget!!!!!!!!!!!!!!	apk ini bagus banget
Normalisasi	apk ini bagus banget	aplikasi ini bagus banget

4.3. Pelabelan Data dengan *Sentiment Scoring*

Pada proses pelabelan data dengan *sentiment scoring* menggunakan dua kamus, yaitu kamus *boosterwords* dan kamus sentimen dengan ketentuan menghitung skor sebagai berikut.

1. Kamus sentimen berisi kata berserta skor yang diberikan. Kata yang tidak terdapat pada

kamus sentimen diberi nilai 0. Contoh: kata "baik" bernilai 4, kata "buruk" bernilai -4, kata "ada" tidak terdapat pada kamus sentimen, maka bernilai 0.

2. Kamus *boosterwords* digunakan dengan melihat skor pada kamus sentimen. Jika kata pada kamus sentimen >0 dan diikuti kata *boosterwords*, maka skor ditambahkan. Sebaliknya, jika kata pada kamus sentimen <0 dan diikuti kata *boosterwords*, maka skor dikurangi.
3. Pada kalimat yang mengandung kata negasi, akan dilakukan *replace* dengan lawan katanya, seperti "tidak bagus" diganti dengan "jelek" dan sebaliknya "tidak jelek" diganti "bagus".
4. Ulasan dengan total skor akhir ≥0 maka akan diberi label positif, sedangkan ulasan dengan total skor akhir <0 maka akan diberi nilai negatif.

Contoh perhitungan *sentiment scoring* adalah sebagai berikut.

aplikasi ini bagus banget	bagus (kamus sentimen)	=1
	banget (*boosterwords*)	=4
	Total skor	**=5 (positif)**

Gambar 3. Contoh perhitungan *sentiment scoring*

Hasil pelabelan dari 3200 terdiri dari 1741 sentimen positif dan 1459 sentimen negatif.

4.4. *Feature Selection*

4.4.1 *Stopwords Removal*

Stopwords yang digunakan dalam penelitian ini sebanyak 875 kata. Dengan dilakukan proses *stopwords removing* jumlah *term* dalam dokumen menjadi berkurang.

4.4.2 *Stemming*

Stemming merupakan tahap mengubah kata-kata dalam dokumen teks menjadi kata dasar. *Stemming* pada *software* R dilakukan dengan menggunakan *package* katadasaR.

Tabel 4. Hasil *Stemming*

Ulasan ke-	Ulasan
1	suka main tiktok main tiktok seru iya main tiktok seru terimakasih aplikasi tiktok pandemi covid main tiktok biar hibur
2	aplikasi bagus banget

4.4.3 *Tokenizing*

Tokenizing merupakan tahap pemisahan setiap kata dalam suatu kalimat. Tujuan *tokenizing* adalah memotong kalimat berdasarkan tiap kata yang menyusun kalimat tersebut.

4.5. Pembobotan *Term Frequency-Inverse Document Frequency* (TF-IDF)

Term frequency (tf) untuk mengukur frekuensi kemunculan kata dalam setiap ulasan. Ketika tf tinggi yang berarti term sering muncul dalam ulasan. Berikut hasil perhitungan *tf*

Tabel 5. Hasil perhitungan *term frequency* (tf)

Ulasan ke	aba	…	aplikasi	…	bagus	…	zoom	Jumlah *term*
1	0	…	1	…	0	…	0	19
2	0	…	1	…	1	…	0	3
⋮	⋮	⋮	⋮	⋮	⋮	⋱	⋮	⋮
3200	0	…	0	…	0	…	0	10
Jumlah dok	1	…	1634	…	835	…	8	

Contoh perhitungan pembobotan TF-IDF.

$$w_{aplikasi,dok1} = \frac{jumlah\ term\ \text{di}\ dok}{jumlah\ seluruh\ term\ \text{di}\ dok} \times \log_2 \frac{jumlah\ seluruh\ dok}{jumlah\ dok\ pada\ term} = \frac{1}{19} \times \log_2 \frac{3200}{1634} = 0,051$$

Tabel 6. Hasil pembobotan dengan *Term Frequency-Inverse Document Frequency*

Ulasan ke	aba	…	aplikasi	…	bagus	…	zoom
1	0	…	0,051	…	0	…	0
2	0	…	0,323	…	0,646	…	0
⋮	⋮	⋮	⋮	⋮	⋮	⋱	⋮
3200	0	…	0	…	0	…	0

4.6. Klasifikasi

4.6.1 Pembuatan Data Latih dan Data Uji

Data latih dan data uji yang digunakan adalah data pembobotan TF-IDF beserta label kelas, dengan perbandingan data uji dan data latih adalah 20% : 80%. Pembuatan data latih dan data uji dilakukan dengan asumsi bahwa setiap data ulasan punya peluang yang sama untuk dapat digunakan sebagai data uji maupun data latih. Maka pemilihan data uji dan data latih berdasarkan urutan ulasan, yaitu dari total 3200 data ulasan, digunakan sebanyak 2560 data ulasan pertama sebagai data latih dan sisanya sebanyak 640 data sebagai data uji.

4.6.2 Klasifikasi *Support Vector Machine* (SVM)

Pada penelitian ini digunakan metode SVM dengan fungsi *kernel Radial Basis Function* (RBF). Pada *kernel* RBF terdapat dua parameter yaitu nilai *Cost* dan *gamma*. Pengoptimalan parameter C dapat dilakukan dengan cara *trial and error*. Nilai *Cost* (C) yang digunakan adalah 0,01; 0,1; 1 10, 100, 1000 dan *gamma* sesuai nilai *default,* yaitu $\gamma = \frac{1}{ncol} = \frac{1}{2350} = 0,0004255319$.

Proses SVM dalam menemukan titik-titik *support vector*, hanya bergantung pada *dot product* dari data yang sudah ditransformasikan pada ruang baru yang berdimensi lebih tinggi. Untuk menentukan *dot product* dilakukan dengan memasukkan *kernel* RBF pada setiap data. Seluruh data dihitung dengan cara sama sehingga menghasilkan nilai *kernel* yang membentuk matrix berukuran 2560×2560. Contoh perhitungan fungsi *kernel* RBF pada data latih.

$$K(x_i, x_j) = \exp\left(-\gamma \| x_i - x_j \|^2\right) = \exp\left(-\gamma \sum_{j=1}^{n} \left(x_i - x_j\right)^2\right)$$

$$K(x_1, x_1) = \exp\left(-0.0004255319 \begin{pmatrix} (0-0)^2 + ... + (0-0)^2 + (0-0)^2 + (0-0)^2 + (0-0)^2 \\ + (0-0)^2 + (0-0)^2 + (0-0)^2 + ... + (0-0)^2 \end{pmatrix}\right) = 1$$

Selanjutnya dicari nilai $\boldsymbol{\alpha}$ dan b paling optimal dengan *Quadratic Programming* (QP) pada *Cost* (C) yang dicoba, yaitu 0,01; 0,1; 1; 10; 100; 1000 dengan bantuan *software Rstudio* untuk memprediksi klasifikasi data uji. *Dot product* data latih dan data uji diperoleh dengan cara yang sama, yaitu memasukkan data uji ke data latih dalam fungsi *kernel* RBF sehingga menghasilkan matriks *kernel* ukuran 2560×640

Tabel 7. Nilai $\boldsymbol{\alpha}$ dan b pada tiap nilai C

Cost	0,01	0,1	1	10	100	1000
$\boldsymbol{\alpha}$	$\begin{bmatrix} 0,01 \\ 0 \\ \vdots \end{bmatrix}_{2560\times1}$	$\begin{bmatrix} 0,1 \\ 0 \\ \vdots \end{bmatrix}_{2560\times1}$	$\begin{bmatrix} 1 \\ 0 \\ \vdots \end{bmatrix}_{2560\times1}$	$\begin{bmatrix} 10 \\ 0 \\ \vdots \end{bmatrix}_{2560\times1}$	$\begin{bmatrix} 0 \\ 0 \\ \vdots \end{bmatrix}_{2560\times1}$	$\begin{bmatrix} 0 \\ 0 \\ \vdots \end{bmatrix}_{2560\times1}$

b	-0,9993469	-0,9934695	-0,927911	-0,3033061	2,071065	1,503933

Untuk C=0,01, persamaan untuk SVM kernel RBF adalah sebagai berikut.

$$y_i = sign\left(\sum_{i=1}^{ns}\alpha_i y_i K\left(\mathbf{x},\mathbf{x}_i\right)+b\right) = sign\left(\sum_{i=1}^{ns}\alpha_i y_i K\left(\mathbf{x}_{training},\mathbf{x}_{testing}\right)+b\right) = sign\left(\sum_{i=1}^{ns}\alpha_i y_i K\left(\mathbf{x}_{training},\mathbf{x}_{testing}\right)-0,9993469\right)$$

$$y_{ulasan2561} = sign\left(\sum_{i=1}^{ns}\alpha_i y_i K\left(\mathbf{x}_{training},\mathbf{x}_{2561}\right)-0,9993469\right)$$

$$y_{ulasan2561} = sign\left(\left[\left(0,01\times1\times1\right)-0,9993469\right]+\left[\left(0\times1\times1\right)-0,9993469\right]+...\right)=-1$$

Perhitungan yang sama dilakukan untuk semua data uji pada tiap C yang digunakan.

Untuk mengukur performa model klasifikasi, dapat dilihat *confusion matrix* data uji hasil *output software* R yang merupakan proses klasifikasi SVM *kernel* RBF di setiap C.

Tabel 8. *Confusion Matrix*

	C = 0,01		**C = 0,1**		**C = 1**		**C = 10**		**C = 100**		**C = 1000**	
Prediksi	**Aktual**		**Aktual**		**Aktual**		**Aktual**		**Aktual**		**Aktual**	
	N	**P**	**N**	**P**	**N**	**P**	**N**	**P**	**N**	**P**	**N**	**P**
N	277	47	284	61	286	66	286	67	282	36	272	39
P	29	287	22	273	20	268	20	267	24	298	34	295

Contoh perhitungan *accuracy* dan *kappa* pada *confusion matrix* C=0,01

$$Accuracy = \frac{prediksi\,data\,benar}{total\,data} = \frac{TP+TN}{TP+TN+FP+FN} = \frac{287+277}{287+277+29+47} = 0,8812$$

$$Kappa = \frac{P_0 - \mathrm{P}_c}{\left(1-\mathrm{P}_c\right)} = \frac{0,8812-0,4997}{\left(1-0,4997\right)} = 0,7626$$

Berdasarkan perhitungan *accuracy* dan *kappa* pada nilai C sebesar 0,01; 0,1; 1; 10; 100; 1000 yang telah dilakukan. Tabel hasil percobaan performa *kernel* RBF dengan nilai *gamma* tetap dan berbagai nilai *Cost* (C) adalah sebagai berikut.

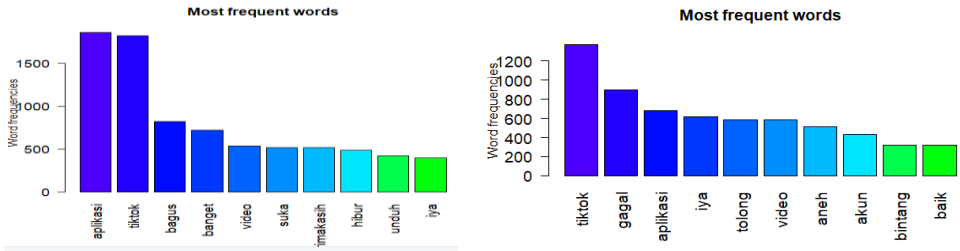
Tabel 9. Nilai *Accuracy* dan *Kappa* pada *Kernel* RBF

Evaluasi	**Cost (C)**					
Model	**0,01**	**0,1**	**1**	**10**	**100**	**1000**
Accuracy	0,8812	0,8703	0,8656	0,8641	0,9062	0,8859
Kappa	0,7626	0,7415	0,7324	0,7293	0,8124	0,7716

Maka nilai C yang paling optimal adalah 100, dengan nilai *accuracy* 0,9062 dan *kappa* 0,8124.

4.7. Visualisasi dan Asosiasi

Visualisasi data ulasan *TikTok* dilakukan pada masing-masing kelas untuk mengambil informasi tentang topik yang sering dibicarakan. Ulasan diidentifikasi berdasarkan banyaknya frekuensi kata tertinggi yang merupakan topik pembicaraan paling banyak ditulis pengguna. Berikut merupakan hasil visualisasi ulasan positif dan negatif.



Gambar 4. Histogram kata frekuensi terbanyak ulasan positif dan negatif

Asosiasi kata dari sepuluh kata paling sering muncul dengan batas bawah nilai asosiasi adalah 0,2 yang dianggap cukup menunjukkan besar hubungan antarkata berdasarkan kondisi data.

Tabel 10. Asosiasi kata positif

aplikasi		**tiktok**		**bagus**	**banget**		**video**	
semat	0,32	terimakasih	0,21	-	unduh	0,24	fyp	0,23
bikin	0,24				pokok	0,21	lucu	0,23
versi	0,23				suka	0,2		
suka		**terimakasih**		**hibur**	**unduh**		**iya**	
banget	0,2	semoga	0,23	-	buru	0,36	-	
		nyaman	0,22		sesal	0,32		
		tiktok	0,21		ayo	0,24		

Tabel 11. Asosiasi kata negatif

tiktok		**gagal**		**aplikasi**		**iya**	**tolong**	
tiktoker	0,23	syarat	0,31	bagus	0,25	-	baik	0,37
cipta	0,21	daftar	0,27	jelek	0,22		tiktoker	0,27
		lahir	0,25				cipta	0,23
		tanggal	0,25					
		masuk	0,23					
		coba	0,21					
video		**aneh**		**akun**		**bintang**	**baik**	
unggah	0,27	-		kembali	0,26	kasih 0,66	tolong	0,37
				blokir	0,24	*dislike* 0,27		
						maaf 0,24		

Contoh perhitungan asosiasi kata "video" dan "fyp" berdasarkan nilai *tf* pada kelas positif.

$$r_{XY} = \frac{n\left(\sum_{i=1}^{n} X_i Y_i\right) - \left(\sum_{i=1}^{n} X_i\right)\left(\sum_{i=1}^{n} Y_i\right)}{\sqrt{\left\{n\sum_{i=1}^{n} X_i^2 - \left(\sum_{i=1}^{n} X_i\right)^2\right\}\left\{n\sum_{i=1}^{n} Y_i^2 - \left(\sum_{i=1}^{n} Y_i\right)^2\right\}}} = \frac{(1741\times160)-(177)(537)}{\sqrt{\left\{(1741\times281)-(177)^2\right\}\left\{(1741\times995)-(537)^2\right\}}} = 0,23$$

Informasi yang bisa dibangun dari asosiasi kata di ulasan positif adalah pengguna yang memberikan ulasan positif karena suka sekali dengan aplikasi *TikTok* versi sekarang yang berisi video-video lucu di *fyp* atau beranda *TikTok*. Pengguna menyatakan tidak menyesal telah mengunduh *TikTok* dan mengajak pembaca untuk mengunduh juga. Pengguna berterima kasih karena merasa nyaman menggunakan aplikasi *TikTok*.

Sedangkan, informasi yang bisa dibangun dari asosiasi kata di ulasan negatif adalah pengguna yang memberikan ulasan negatif karena gagal mencoba daftar atau masuk akun disebabkan tidak terpenuhinya syarat tanggal lahir. Pengguna tidak menyatakan *TikTok* jelek

secara keseluruhan, tapi juga mengakui ada bagian yang bagus. Pengguna meminta pihak *TikTok* mengembalikan akunnya yang diblokir. Pengguna yang memberikan ulasan negatif meminta maaf karena memberikan *dislike* atau sedikit bintang penilaian di *Google Play*. Pengguna *TikTok* meminta tolong pihak *TikTok* dan *TikToker* (sebutan untuk para pencipta atau pengunggah video di *TikTok*) bisa membuka *TikTok* menjadi semakin baik.

5. KESIMPULAN

Berdasarkan analisis dan pembahasan didapatkan beberapa kesimpulan sebagai berikut.

1. Persepsi pengguna *TikTok* berdasarkan pelabelan sentimen ulasan bulan September 2020 sampai Februari 2021 di *Google Play* sebanyak 3200 ulasan adalah jumlah ulasan positif lebih banyak, yaitu 1741 (54,41%) dibanding jumlah ulasan negatif, yaitu 1459 (45,59%).
2. Klasifikasi sentimen dari hasil *sentiment scoring* ulasan aplikasi *TikTok* di *Google Play* menggunakan metode *Support Vector Machine* (SVM) *kernel* RBF dengan perbandingan data latih dan data uji sebesar 80 : 20 menghasilkan tingkat *accuracy* dan *kappa* terbaik sebesar 90,62% dan 81,24% yang berarti termasuk hasil klasifikasi yang hampir sempurna.
3. Berdasarkan hasil klasifikasi dan asosiasi, pengguna memberikan ulasan positif karena suka sekali dengan *TikTok* versi sekarang yang berisi video-video lucu di *fyp* atau beranda *TikTok*. Pengguna menyatakan bahwa tidak menyesal telah mengunduh *TikTok* dan mengajak pembaca untuk mengunduh juga. Sedangkan pada kelas sentimen negatif, pengguna memberikan ulasan negatif karena gagal dalam mencoba daftar atau masuk ke akun *TikTok* karena tidak memenuhi syarat tanggal lahir. Pengguna meminta pihak *TikTok* mengembalikan akunnya yang diblokir. Pengguna yang memberi ulasan negatif biasanya juga memberikan *dislike* atau sedikit bintang penilaian di *Google Play* serta meminta tolong pihak *TikTok* dan *TikToker* (sebutan untuk para pencipta atau pengunggah video di *TikTok*) bisa membuat *TikTok* menjadi semakin baik.

DAFTAR PUSTAKA

Ayani, D. D., Pratiwi, H. S.,dan Muhardi, H. 2019. *Implementasi Web Scraping untuk Pengambilan Data pada Situs Marketplace*. Sistem dan Teknologi Informasi Vol 7, No 4, Hal. 257-262.

Christian, H., Agus, M. P., dan Suhartono, D. 2016. *Single Document Automatic Text Summarization Using Term Frequency-Inverse Document Frequency(TF-IDF)*. ComTech,Vol.7, Hal.285-294.

CNN, 2018. *Penuhi 9 dari 10 Syarat Kominfo, Blokir TikTok Dibuka*. https://www.cnnindonesia.com/teknologi/20180710162606-185-313025/penuhi-9-dari-10-syarat-kominfo-blokir-tik-tok-dibuka. Diakses: 2 September 2020.: s.n.

Fatmawati dan Affandes, M.. 2017. *Klasifikasi Keluhan Menggunakan Metode Support Vector Machine (SVM) (Studi Kasus : Akun Facebook Group iRaise Helpdesk)*. Jurnal CoreIT, Vol.3, No.1, Hal. 24-30.

Indraloka, D. S. dan Santosa, B. 2017. *Penerapan Text Mining untuk Melakukan Clustering Data Tweet Shopee Indonesia*. JURNAL SAINS DAN SENI ITS Vol. 6, No. 2.

Liputan 6. 2020. Orang Indonesia Kedua Paling Banyak Unduh TikTok per Juli 2020. https://www.liputan6.com/tekno/read/4324103/orang-indonesia-kedua-paling-banyak-unduh-tiktok-per-juli-2020. Diakses: 31 Agustus 2020.

Liu, B. 2015. *Sentiment Analysis Mining Opinions, Sentiments, and Emotions.* New York: Cambridge University Press.

Nugroho, A. S., Witarto, A. B., dan Handoko, D. 2003. *Support Vector Machine Teori dan Aplikasinya dalam Bioinformatika*. https://www.academia.edu/24381027/Support_Vector_Machine_Teori_dan_Aplikasinya_dalam_Bioinformatika_1. Diakses: 5 November 2020.

Rahman, M., Darmawidjadja, Alamsah, D. 2017. *Klasifikasi Untuk Dianosa Diabetes Menggunakan Metode Bayesian Regularization Neural Network(RBNN)*. Informatika, Vol 11, No 1.

Santosa, E. B. & Nugroho, A. 2019. *Analisis Sentimen Calon Presiden Indonesia 2019 Berdasarkan Komentar Publik di Facebook.* Eksplora Informatika, Hal 60-69.

Wahid dan Azhari. 2016. Peringkasan Sentimen Esktraktif di Twitter Menggunakan Hybrid TF-IDF dan Cosine Similarity. *Indonesian Journal of Computing & Cybernetics Systems,*Vol 10,No 2.

Wang, L. 2005. *Support Vector Machines: Theory and Applications.* Berlin: Springer.