

KOMPARASI ALGORITMA DATA MINING UNTUK ANALISIS SENTIMEN APLIKASI RUANG GURU MENGGUNAKAN ALGORITMA NAIVE BAYES, SVM, RANDOM FOREST

Fedrico Simatupang¹, Yusran Timur Samuel ^{*2}

^{1,2}Fakultas Teknologi Informasi, Universitas Advent Indonesia
e-mail: ¹1981012@unai.edu, ^{2*}y.tarihoran@unai.edu

Abstrak

Sebuah aplikasi pada umumnya dikatakan baik hanya berdasarkan jumlah rating dan jumlah unduhan yang tinggi, dan hal itu tentu menjadi tidak relevan, sehingga peneliti tertarik untuk menjadikan opini tersebut sebagai bahan analisis sentimen pada aplikasi Ruangguru sehingga bisa menjadi evaluasi bagi aplikasi Ruangguru, sekaligus menentukan algoritma apa yang memiliki akurasi terbaik. Pada penelitian ini, peneliti menguji tiga model klasifikasi *Naive Bayes*, dan *Support Vector Machine*, dan *Random Forest* untuk menganalisis sentimen aplikasi Ruangguru. Teknik pengumpulan data diperoleh dengan menggunakan metode *web scraping* Google Play Store, dengan jumlah data yang dikumpulkan yaitu sebanyak 1629 mulai dari tanggal 16 oktober 2022 hingga 11 februari 2023 dan juga berdasarkan *review created version* 4.5.3 sampai dengan versi 6.6.1. Hasil penelitian ini ditemukan bahwa *Random Forest* merupakan algoritma yang cukup stabil dalam peningkatan persentase mulai dari jumlah data terendah ke tinggi yaitu; 100:100, 200:200, 300:300, 499:499 bahkan dengan data yang tidak balance dengan jumlah data positive 1088 dan 499 untuk jumlah data negatif. Algoritma *Random Forest* memiliki akurasi yang paling baik yaitu 85,25% dengan pembobotan CV dan 84,56% dengan pembobotan TF-IDF, dibanding dengan algoritma SVM memiliki akurasi 81,74% untuk di kedua pembobotan baik CV & TF-IDF, dan *Naive Bayes* memiliki akurasi 83,87% dengan pembobotan CV & 82,91% untuk TF-IDF.

Kata Kunci: Analisis Sentimen, Algoritma, Naive Bayes, Support Vector Machine, Random Forest.

COMPARISON OF DATA MINING ALGORITHMS FOR SENTIMENT ANALYSIS OF RUANGGURU APPLICATIONS USING NAIVE BAYES, SVM, RANDOM FOREST ALGORITHMS

Abstract

An app is generally said to be good only based on the number of ratings and a high number of downloads, which is irrelevant, so researchers are interested in using these opinions to analyze the sentiment of the Ruangguru app so that it can be an evaluation for the Ruangguru app, as well as determine which algorithm has the best accuracy. In this study, we tested three classification models-Naive Bayes, Support Vector Machine, and Random Forest-to analyze the sentiment of Ruangguru app. The data collection technique was obtained using the Google Play Store web scraping method, with a total of 1629 data collected from October 16, 2022 to February 11, 2023 and also based on the review created version 4.5.3 to version 6.6.1. The results of this study found that Random Forest is a fairly stable algorithm in increasing the percentage ranging from the lowest amount of data to high, namely; 100:100, 200:200, 300:300, 499:499 even with unbalanced data with the number of positive data 1088 and 499 for the number of negative data. Random Forest algorithm has the best accuracy which is 85.25% with CV weighting and 84.56% with TF-IDF weighting, compared to SVM algorithm which has 81.74% accuracy for

both CV & TF-IDF weighting, and Naïve Bayes has 83.87% accuracy with CV weighting & 82.91% for TF-IDF.

Keywords: *Sentiment Analysis, Algorithm, Naïve Bayes, Support Vector Machine, Random Forest.*

1. Pendahuluan

Salah satu industri yang terkena dampak kemajuan teknologi adalah pendidikan. Kemajuan teknologi yang pesat saat ini telah menjadi komponen kunci dalam dunia pendidikan [1], di tambah dikarenakan efek dari pandemi COVID-19 beberapa waktu yang lalu, membuat perubahan secara khusus dalam bidang pendidikan, dimana mengharuskan mengubah metode pembelajaran dari Luring ke Daring, sehingga *mobile learning* atau pembelajaran elektronik berupa aplikasi pun menjadi meningkat [2].

Salah satunya aplikasi *mobile learning* itu adalah aplikasi Ruangguru yang telah diunduh lebih dari 10 juta kali pengunduh dan aplikasi Ruangguru ini menjadi salah satu layanan bimbingan belajar yang cukup populer, terutama di kalangan pelajar Indonesia, mulai dari SD, SMP, SMA/Sederajat, atau bahkan mahasiswa. Ruangguru merupakan program atau media pembelajaran yang menggunakan pembelajaran elektronik yang tidak hanya bisa diakses melalui aplikasi namun juga bisa di akses melalui web, dengan begitu pastinya akan sangat memudahkan siswa untuk mengakses materi pembelajaran yang berkualitas ditambah dengan banyaknya fitur-fitur menarik yang disediakan aplikasi Ruangguru, sehingga bahan-bahan materi pembelajaran yang disediakan tersebut, akan dengan mudah untuk diakses kapan saja dan dari lokasi mana saja tanpa adanya batasan ruang dan waktu, sehingga kegiatan belajar mengajar dapat dilakukan dengan lebih sukses antara siswa dan guru [3].

Tetapi disamping itu, setiap aplikasi pastinya memiliki kelebihan dan kekurangannya masing-masing, dan pengguna biasanya meluapkan respon kepuasan dan kekecewaan pada aplikasi tersebut. Dan sebuah aplikasi dikatakan baik hanya berdasarkan jumlah rating dan ulasan-ulasan pada aplikasi tersebut, hal itu tentu menjadi tidak relevan, sehingga peneliti tertarik untuk menjadikan opini tersebut sebagai bahan analisis sentimen pada aplikasi Ruangguru sehingga bisa menjadi evaluasi bagi aplikasi Ruangguru [4], dan sekaligus menentukan algoritma apa yang memiliki akurasi terbaik [5].

Pada penelitian ini dianalisis bahwa sentimen merupakan penggalian informasi yang diolah berdasarkan hasil analisa dari berbagai data berupa pandangan atau pendapat sehingga menghasilkan hasil analisis berupa sentimen positif, negatif ataupun netral [6]. Dan dengan bantuan analisis sentimen maka memungkinkan untuk mengekstrak informasi penting dari data yang tidak terstruktur, sehingga dapat mengungkapkan bagaimana ungkapan pengguna terhadap aplikasi Ruang Guru.

Lalu pada penelitian sebelumnya [7] yang berjudul Perbandingan Akurasi, Recall, dan Presisi Klasifikasi pada Algoritma C4.5, *Random Forest*, *SVM* dan *Naïve Bayes*. Dari ke empat algoritma tersebut diketahui bahwa algoritma SVM adalah metode yang terbaik dengan nilai akurasi 95% dan yang memiliki akurasi paling kecil adalah Random Forest dengan akurasi 83,33%. Pada penelitian selanjutnya [8] dengan judul *Sentiment Analysis on Twitter Data of World Cup Soccer Tournament Using Machine Learning*. Pada penelitian ini diperoleh bahwa *naïve bayes* memiliki akurasi paling tinggi dibanding dengan SVM, KNN, dan *Random Forest* dengan akurasi 88.17% meskipun *Random Forest* memiliki AUC paling tinggi yaitu 0.97. Dan pada penelitian terdahulu yang lain [9] dengan judul penelitian Penerapan Algoritma *Sentiment Analysis* Dan *Naïve Bayes* Terhadap Opini Pengunjung Di Tempat Wisata Pantai Pintu Kota, Kota Ambon. Yang menunjukkan hasil yang diperoleh dari penelitian tersebut menggunakan algoritma *Naïve Bayes* terdapat *recall negative* sebesar 97.62%, *recall positive* sebesar 83,33%. Nilai akurasi negatif sebesar 85,42%, sedangkan nilai akurasi positif sebesar 97,22%. Dan dari hasil tersebut diperoleh nilai akurasi sebesar 90,65%.

Berdasarkan uraian diatas, maka peneliti menggunakan algoritma *Naïve Bayes*, *SVM* dan *Random Forest* sebagai perbandingan ditambahkan dengan pembobotan kata untuk membantu dalam pengklasifikasian data dan untuk meningkatkan hasil akurasi analisis sentimen dari ketiga algoritma tersebut berdasarkan data berupa ulasan pengguna yang diperoleh dari *play store*. Dan diharapkan

penelitian ini dapat mengungkapkan analisis sentimen pada aplikasi Ruangguru dan mengidentifikasi algoritma yang terbaik pada aplikasi tersebut.

2. Tinjauan Pustaka

Algoritma Naïve Bayes

Metode *Naïve Bayes* merupakan *machine learning* yang menggunakan perhitungan probabilitas dari setiap kemungkinan klasifikasi termasuk dalam teknik yang cukup efektif dalam hal analisis sentimen [10].

$$P(X) = \frac{P(X|H) P(H)}{P(X)} \quad (1)$$

Dimana $P(H|X)$ adalah kondisi probabilitas hipotesis H terletak pada apa yang ada pada data sampel x , sedangkan $P(H|X)$ adalah kondisi probabilitas yang akan terjadi pada data sampel X akan diamati bila hipotesis H valid benar. Lalu H adalah label yang lebih unik untuk dugaan data x , X data sampel dengan kelas yang tidak diketahui. $P(H)$ merupakan probabilitas hipotesis H , sedangkan $P(X)$ Probabilitas sampel X (yang diamati) [11].

Support Vector Machine

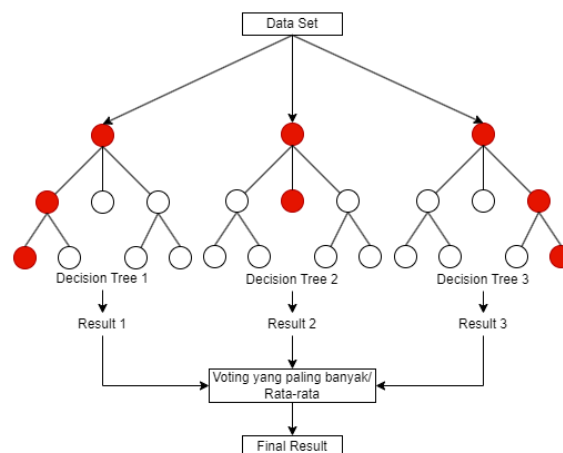
Support Vector Machine (SVM) adalah contoh *machine learning* yang lain yang sama baiknya dalam hal mengklasifikasikan data, dimana SVM ini memiliki *hyperplane* (bidang pemisah) yang berfungsi untuk menghitung jarak (*margin*) antara *support vector* (data) yang menjadi pemisah antara 2 kelompok/kategori, sehingga dapat menghasilkan prediksi yang akurat [12].

$$y(x) = wT\phi(x) + b \quad (2)$$

Di mana w adalah parameter bobot, $\phi(x)$ adalah fungsi basis, dan b adalah bias, x adalah vektorinput.

Random Forest

Random Forest (RF) yang sesuai dengan namanya, dimana merupakan sebuah metode klasifikasi dan regresi yang disusun berdasarkan pohon keputusan secara acak dan *node* sebagai pemisah [13].



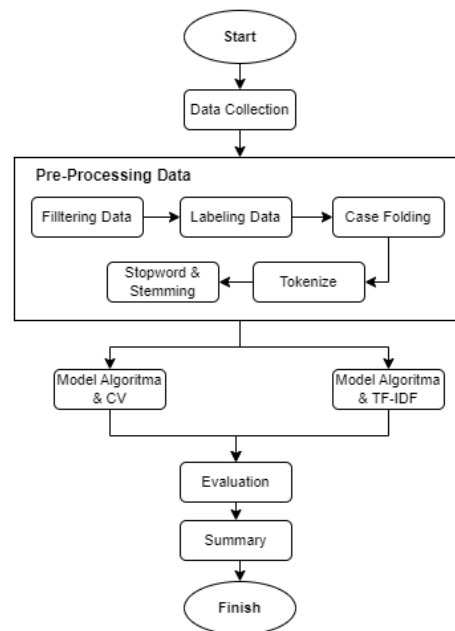
Gambar 1. Logika Algoritma Random Forest

Cara kerja dari pada random forest ini adalah dengan membuat *base classifier* yang dibangun dan digabungkan dari pohon keputusan (decision tree), selanjutnya dari masing2 pohon mengambil keputusan

prediksi secara acak, dan setelah itu barulah kita kombinasikan dari masing keputusan prediksi secara acak tersebut menggunakan suara terbanyak/rata sebagai hasil [14].

3. Metode Penelitian

Seperti terlihat pada Gambar 1 di bawah ini, metodologi penelitian yang akan digunakan dalam penelitian ini adalah sebagai berikut:



Gambar 2. Metode Penelitian

Pengumpulan Data

Kumpulan data untuk penelitian ini berasal dari review Google Play Store pada aplikasi Ruangguru. Dalam prosedur ini, data diekstrak menggunakan metode *web scraping* dari situs web Google Play store melalui program *Python* [15]. Sebanyak 1629 data dikumpulkan melalui data komentar teks berdasarkan *review created version* 4.5.3 sampai dengan versi 6.6.1 mulai dari tanggal 16 oktober 2022 hingga 11 februari 2023.

Tabel 1. Data Yang di Kumpulkan

<i>Time</i>	<i>Content</i>	<i>Score</i>	<i>Review Created Version</i>
2/11/2023 15:32	Roboguru harus beli paket... why?	3	6.39.0
2/11/2023 13:55	Sangat membantu jaya terus ruang guru	5	6.52.0
2/11/2023 6:55	Video nya ga bisa diputar, padahal udah diunduh :(:(4	6.52.0
10/16/2022 12:57	Error terus video nya, nyesel gue Makai ruang guru	1	
2/11/2023 6:39	Keren banget deh pokoknya	5	6.52.0

Berikut berdasarkan tabel diatas adalah data yang diperoleh berdasarkan *Web Scraping* dan telah di import ke file excel.

Preprocessing Data

Setelah dilakukan pengumpulan data maka tahap selanjutnya adalah melakukan *preprocessing data*, dimana pada tahap *preprocessing data* merupakan tahap awal sebelum dilakukannya proses klasifikasi data, untuk mendapatkan data yang baik, maka dari itu berikut adalah beberapa teknik *preprocessing data* yang digunakan oleh peneliti:

1. Filtering Data

Kegunaan *filtering data* sini adalah untuk mengurangi *attribute* yang kurang diperlukan dan memilih *attribute* yang penting saja, dalam hal ini data yang diperoleh seperti *time, score & review created version* dihilangkan, sebelum akhirnya diproses lebih lanjut [8].

Tabel 2. Data Filtering

Ulasan/Komentar
Roboguru harus beli paket... why?
Sangat membantu jaya terus ruang guru
Video nya ga bisa diputar, padahal udah diunduh :(
Error terus video nya, nyesel gue Makai ruang guru
Keren banget deh pokoknya

Pada tahap *filtering* ini, data yang diambil hanya sebuah ulasan, karena pada data sebelumnya, berupa *time, score dan review created version* kurang diperlukan dalam pengklasifikasian data, ditambah fokus pada penelitian ini adalah sentimen dari pengguna, jadi peneliti telah memfilter data dan data yang diambil hanya sebuah content/ulasan/komentar dari aplikasi Ruangguru.

2. Labeling Data

Selanjutnya setelah dilakukannya *filtering data*, yang perlu dilakukan adalah *Labeling data*, karena *labeling data* merupakan hal yang paling utama sebelum melakukan analisis sentimen, dan harus dipastikan bahwa *dataset* yang akan di proses memiliki label yang benar. Label sentimen pada penelitian ini dibagi menjadi kelas *sentiment positive* dan *sentiment negative*, yang sistem pelabelan dilakukan secara *crowdsourcing* dengan menggunakan beberapa orang pelabel untuk meningkatkan hasil pelabelan [16] sejalan dengan penilaian aplikasi Ruangguru dan ilustrasi hasil pada Tabel 3. Dan dalam hal ini terjadi ketidakseimbangan data setelah proses *labeling*, dimana ditemukan bahwa jumlah sentimen positif sebanyak 1088, dan sentimen negatif sebanyak 499.

Tabel 3. Labeling

Sentiment	Content
Negative	Roboguru harus beli paket... why?
Positive	Sangat membantu jaya terus ruang guru
Negative	Video nya ga bisa diputar, padahal udah diunduh :(
Negative	Error terus video nya, nyesel gue Makai ruang guru
Positive	Keren banget deh pokoknya

Peneliti menggunakan metode *crowdsourcing*, dikarenakan jika menentukan sentimen hanya berdasarkan score seringkali ditemukan ketidakcocokan antara ulasan dengan score, dimana isi dari pada content itu merupakan ulasan yang bersifat positive tetapi diberikan score nilai 1 atau 2 oleh pengguna, begitu juga sebaliknya, sebagai contoh pada Gambar 3, berikut:

377	Negative	gila seh keren banget apk ini membantu aku dalam bidang olahraga	1	6.9.0
378	Positive	wow ini apps keren sekali, mamah bangga nilai ku naik dari 5 sekarang 100	1	6.9.0
379	Positive	keren nilaiku jadi bagus banget setelah pakai aplikasi ini, mamaku bangga kak	1	6.9.0
380	Negative	INII KOK RUANGGURU BUAT BUKA MATERI GABISA SIII?? LAYARNYA SELALU BERUBAH ITEM SETIAP PENCET SUBBAB. TRUS KELUAR SENDIRIII TO	1	6.9.0
381	Positive	Good	5	6.9.0

Gambar 3. Ketidakcocokan Antara Ulasan dengan *Score*

3. Case Folding

Selanjutnya adalah tahapan *case folding*, *case folding* adalah proses mengubah semua huruf menjadi huruf kecil, menghilangkan tanda baca, dan menghilangkan nominal (dalam hal ini hanya karakter dari a sampai z) guna mempercepat perbandingan selama proses pengindeksan [17] dan ilustrasi hasil pada Tabel 4.

Tabel 4. *Case Folding*

No	Label	Text Sebelum Case Folding	Text Sesudah Case Folding
1	Negative	Roboguru harus beli paket... why?	roboguru harus beli paket why
2	Positive	Sangat membantu jaya terus ruang guru	sangat membantu jaya terus ruang guru
3	Negative	Video nya ga bisa diputar, padahal udah diunduh...	video nya ga bisa diputar padahal udah diunduh
4	Negative	Error terus video nya, nyesel gue Makai ruang guru	error terus video nya nyesel gue makai ruang guru
5	Positive	Keren banget deh pokoknya	keren banget deh pokoknya

4. Tokenize

Tokenize adalah tahap dimana kita memecahkan suatu kalimat menjadi kata-kata individu untuk memudahkan pemrosesan data [18] dan ilustrasi hasil pada Tabel 5.

Tabel 5. *Tokenize*

No	Text Sebelum di Tokenize	Text Setelah di Tokenize
1	roboguru harus beli paket why	[roboguru, harus, beli, paket, why]
2	sangat membantu jaya terus ruang guru	[sangat, membantu, jaya, terus, ruang, guru]
3	video nya ga bisa diputar padahal udah diunduh	[video, nya, ga, bisa, diputar, padahal, udah, diunduh]
4	error terus video nya nyesel gue makai ruang guru	[error, terus, video, nya, nyesel, gue, makai, ruang, guru]
5	keren banget deh pokoknya	[keren, banget, deh, pokoknya]

5. Stopword & Stemming

Fungsi fitur *stopword* adalah menghilangkan kata-kata yang sering muncul dalam bahasa alami dan memiliki arti yang sedikit. Dan fungsi fitur *stemming* adalah untuk menghapus imbuhan dalam bentuk kata apapun menjadi bentuk kata yang paling dasar [19] dan ilustrasi hasil pada Tabel 6.

Tabel 6. Stopword & Stemming

No	Text sebelum di Stopword & Stemming	Stopword & Stemming
1	[roboguru, harus, beli, paket, why]	[roboguru, beli, paket]
2	[sangat, membantu, jaya, terus, ruang, guru]	[bantu, jaya, ruang, guru]
3	[video, nya, ga, bisa, diputar, padahal, udah, diunduh]	[video, putar, unduh]
4	[eror, terus, video, nya, nyesel, gue, makai, ruang, guru]	[eror, video, nyesel, gue, pakai, ruang, guru]
5	[keren, banget, deh, pokoknya]	[keren, banget, pokok]

Model Algoritma

Pada tahap ini setelah di lakukan teknik *Preprocessing data*, maka tahap selanjutnya adalah melakukan *build model development* pada ketiga algoritma yang digunakan, dimana pada penelitian ini algoritma yang digunakan ialah Algoritma *Naïve Bayes*, *Support Vector Machine* dan juga *Random Forest* di tambah dengan 2 Pembobotan yaitu *Count Vectorizer* & *TF-IDF Vectorizer*.

CountVectorizer

CountVectorizer ini merupakan salah satu pembobotan yang dipakai oleh peneliti, dimana pada dasarnya tujuan utama dari pada *CountVectorizer* ini adalah untuk menyesuaikan dan mempelajari setiap kata yang diberikan dalam kosakata, dan berdasarkan kosakata yang dibuat ini, terciptalah yang namanya matrix, yang dimana *CountVectorizer* ini merupakan model *boolean*, yang nantinya kata dalam kosakata ini akan di ubah ke dalam bentuk vektor/representasi matriks [21]. Dan Jumlah kemunculan untuk setiap kata yang unik dalam teks inilah yang digunakan sebagai masukan vektor. Adapun langkah untuk melakukan *CountVectorizer* yaitu: Fit dan Transformasi. Diawali dengan tahap fit dengan *vectorizer* mempelajari vokabuler dari setiap kata yang diberikan, dengan membaca dan menghitung frekuensi kata dalam dokumen tersebut. Setelah dilakukannya tahap fit, maka di lakukanlah tahap transformasi untuk mengubah dokumen menjadi representasi numerik berdasarkan kemunculan kata yang unik [22]. Jadi sebelum kita masuk ke metode *Term Frequency-Inverse Document Frequency* (TF-IDF), terlebih dahulu kita menghitung frekuensi *term* (tf) atau yang bisa kita sebut dengan *CountVectorizer* dari setiap kata (*term*)t dalam setiap dokumen (d). Frekuensi *term*, tf (t, d), menghitung berapa kali *term* t muncul dalam dokumen d dan dihitung sebagai rasio jumlah kemunculan kata t, n(t, d), dengan jumlah seluruh kata dalam dokumen (kosakata), V(d) [23].

Formula untuk menghitung frekuensi term (tf):

$$tf(t, d) = \frac{n(t, d)}{V(d)} \quad (6)$$

Dimana n (t, d) merupakan jumlah kemunculan kata t dalam dokumen d, sedangkan V(d) merupakan jumlah seluruh kata (kosakata) dalam dokumen d. selanjutnya, setelah kita mendapatkan frekuensi term (tf) untuk setiap kata dalam dokumen barulah kita bisa mengalikan dengan *Inverse Document Frequency* (IDF)

TF-IDF Vectorizer

Pembobotan TF-IDF adalah sebuah Metode yang memberi bobot pada setiap kata atau fitur dengan mengubah data tekstual menjadi data numerik. Dan TF-IDF ini digunakan juga untuk menilai seberapa penting kata dalam dokumen berdasarkan tingkat statistik bobot kata tersebut [24].

Formula untuk menghitung IDF setelah dilakukannya frekuensi term:

$$idf(t) = \log \left(\frac{N}{df(t)} \right) \quad (7)$$

Dengan keterangan dimana N adalah jumlah keseluruhan dokumen dalam korpus, lalu $df(t)$ merupakan nilai dokumen yang memuat kata t (yaitu, jumlah dokumen yang frekuensi nya tidak nol).

Teknik Evaluasi

Setelah selesai dilakukannya *build model development* ketiga algoritma, maka tahap selanjutnya adalah teknik evaluasi, dimana untuk melakukan evaluasi performa dari ketiga algoritma yang digunakan pada penelitian ini dikombinasikan dengan pembobotan *Count Vectorizer* dan *TF-IDF Vectorizer*.

Confusion Matrix adalah Salah satu teknik untuk mengevaluasi kinerja metode klasifikasi. Yang dimana matrix tersebut digambarkan pada Tabel 7.

Tabel 7. *Confusion Matrix*

True Negative (TN)	False Negative (FN)	Negative (0)	Predicted Values
False Positive (FP)	True Positive (TP)	Positive (1)	
Negative (0)	Positive (1)		
Actual Values			

Berdasarkan variabel diatas kita akan mendapatkan hasil performa kinerja, berupa nilai *accuracy*, *precision*, dan juga *recall* dengan menggunakan rumus yang ditunjukkan pada persamaan (3), (4), dan (5).

$$Accuracy = \frac{TP+TN}{TP+FP+FN+TN} \quad (3)$$

$$Precision = \frac{TP}{TP+FP} \quad (4)$$

$$Recall = \frac{TP}{TP+FN} \quad (5)$$

Dimana *accuracy* untuk melihat keakuratan jumlah total dari seberapa sering model mengklasifikasikan dengan benar. Lalu *precision* digunakan untuk mengukur seberapa baik klasifikasi model dengan benar pada kelas prediksi positif, dan *recall* sebaliknya dimana recall mengukur seberapa baik klasifikasi model dengan benar pada kelas aktual positif [20].

Word Cloud

(*Word Cloud*) adalah pengelompokan kata-kata yang sering muncul dalam data teks yang diteliti. Kata-kata tersebut terakumulasi menjadi sebuah awan kata. Ukuran huruf dalam sebuah kata berfungsi sebagai petunjuk tentang kuantitas kata yang digunakan. Sebuah kata jika semakin sering muncul maka akan semakin besar huruf dari kata tersebut muncul di dalam *word cloud* dan begitu pun sebaliknya semakin kecil huruf yang di visualisasikan dalam *word cloud*, maka frekuensi dari kata tersebut semakin sedikit [25].

3. Hasil dan Pembahasan

Pengujian Menggunakan Algoritma *Data Mining*

Pengujian dilakukan menggunakan bahasa pemrograman *Python* dengan *Jupyter Notebook* menggunakan *10-fold cross validation*, dengan algoritma *Data Mining* dengan rumus dan hasil sebagai berikut ini:

Tabel 8. Hasil Pengujian Algoritma Data Mining Secara Keseluruhan

Jumlah Data		Naïve Bayes & Count Vectorizer					Naïve Bayes & TF-IDF				
		Precision		Recall		Accuracy	Precision		Recall		Accuracy
Positive	Negative	Positive	Negative	Positive	Negative		Positive	Negative	Positive	Negative	
100	100	69,92%	91,30%	93,48%	63,00%	77,60%	71,07%	91,55%	93,48%	65,00%	78,65%
200	200	76,10%	86,29%	86,67%	75,50%	80,79%	76,00%	84,44%	84,44%	76,00%	80,00%
300	300	75,33%	87,21%	87,40%	75,00%	80,78%	76,19%	85,82%	85,50%	76,67%	80,78%
499	499	73,84%	86,43%	86,99%	72,89%	79,49%	74,75%	84,58%	84,47%	74,90%	79,38%
1088	499	84,80%	81,58%	91,92%	68,47%	83,87%	86,45%	75,83%	87,72%	73,69%	82,91%
Jumlah Data		SVM & Count Vectorizer					SVM & TF-IDF				
		Precision		Recall		Accuracy	Precision		Recall		Accuracy
Positive	Negative	Positive	Negative	Positive	Negative		Positive	Negative	Positive	Negative	
100	100	63,79%	76,32%	80,43%	58,00%	68,75%	74,26%	81,32%	81,52%	74,00%	77,60%
200	200	66,67%	81,58%	84,44%	62,00%	72,63%	66,67%	81,58%	84,44%	62,00%	72,63%
300	300	70,03%	83,67%	84,73%	68,33%	75,98%	70,03%	83,67%	84,73%	68,33%	75,98%
499	499	73,71%	84,33%	84,47%	73,49%	78,63%	73,71%	84,33%	84,47%	73,49%	78,63%
1088	499	83,40%	77,67%	90,14%	65,66%	81,74%	83,40%	77,67%	90,14%	65,66%	81,74%
Jumlah Data		Random Forest & Count Vectorizer					Random Forest & TF-IDF				
		Precision		Recall		Accuracy	Precision		Recall		Accuracy
Positive	Negative	Positive	Negative	Positive	Negative		Positive	Negative	Positive	Negative	
100	100	70,91%	82,93%	84,78%	68,00%	76,04%	69,23%	85,33%	88,04%	64,00%	75,52%
200	200	72,85%	88,05%	89,44%	70,00%	79,21%	71,75%	87,26%	88,89%	68,50%	78,16%
300	300	73,65%	87,85%	88,55%	72,33%	79,89%	72,47%	86,59%	87,40%	71,00%	78,65%
499	499	75,48%	89,37%	89,95%	74,30%	81,62%	75,73%	88,00%	88,36%	75,10%	81,30%
1088	499	85,63%	84,30%	93,18%	70,08%	85,25%	84,68%	84,25%	93,39%	67,67%	84,56%

Berdasarkan Tabel 8 diatas dengan jumlah sample data 1:1 antara positif dan negatif, untuk data 100:100, 200:200, 300:300, 499:499, dan pada pengujian yang ke-lima dengan data yang tidak *balance* dengan jumlah data positif itu 1088 dan 499 untuk jumlah data negatif, terjadi peningkatan dan penurunan akurasi di masing-masing algoritma dalam hal ini untuk algoritma *Naïve Bayes* & SVM. *Naïve Bayes* dengan *CountVectorizer* dari 77,60% naik ke 80,79% lalu turun menjadi 80,79% dan turun lagi menjadi 79,49%, setelah itu meningkat dikarenakan data yang tidak *balance* menjadi 83,87%, begitu juga dengan algoritma *Naïve Bayes* dengan pembobotan TF-IDF dari 78,65% naik menjadi 80,00%, naik lagi menjadi 80,78% lalu turun menjadi 79,38%, lalu pada SVM dengan pembobotan *CountVectorizer* terjadi peningkatan yang cukup stabil mulai dari 68,75% ke 72,63%, lalu 75,98% lalu naik lagi menjadi 78,64% dan 81,74% pada data yang tidak *i*, kemudian SVM dengan pembobotan TF-IDF, mulai dari 77,60% turun menjadi 72,63%, meningkat lagi menjadi 75,98% lalu 78,63% dan 81,74% pada data yang tidak *balance*. Namun hal itu tidak terjadi pada algoritma *Random Forest*, dimana tidak di temukan nya penurunan akurasi dengan pembobotan *CountVectorizer* & TF-IDF, terjadi peningkatan akurasi yang stabil bahkan dengan data yang tidak *balance* pada pengujian kelima yaitu untuk algoritma *Random Forest* dengan pembobotan *CountVectorizer* mulai dari 76,04% ke 79,04% lalu 79,89% naik lagi menjadi 81,62% dan pada data yang

tidak *balance* dengan akurasi 85,68%, begitu juga pada *Random Forest* dengan pembobotan TF-IDF akurasi mulai dari 75,52%, 78,16%, 78,65%, lalu 81,30% dan 84,56% pada data yang tidak *balance*.

3.2 Word Cloud

Visualisasi Word Cloud yang menggambarkan kata-kata yang sering muncul dengan label sentimen *negative* dan *positive*.



Gambar 4. Word Cloud Negative

Gambar 5. Word Cloud Positive

Dari Word Cloud diatas kita dapat menemukan kata yang sering muncul pada Sentimen Negative adalah "video", "ajar", "bagus", "langganan", "ruangguru", "jaring", "putar", "buka", "download", "banget", "fitur", "aplikasi" dimana pengguna sebenarnya merasa bahwa aplikasi ruang guru ini bisa dikatakan sudah cukup bagus karena terdapat banyak fitur di dalamnya yang membantu pengguna dalam belajar, namun masih sering ditemukan masalah dalam pemutaran video yang berputar2 atau loading yang lama, meskipun memiliki jaringan yang baik, dan sudah berlangganan yang membuat pengguna terpaksa harus mendownload video.

Sedangkan pada Sentimen Positive, kata yang sering muncul adalah "ruangguru", "ajar", "bagus", "banget", "aplikasi", "seru", "bantu", "materi", "video" dimana pengguna merasa senang karena aplikasi ruang guru ini bagus banget, karena memiliki materi video pembelajaran yang seru sehingga sangat membantu para pengguna dalam hal belajar.

4. Kesimpulan

Berdasarkan hasil pengujian data dengan jumlah *sample* data 1:1, sebanyak 100:100, 200:200, 300:300, 499:499 dan pengujian yang kelima dengan jumlah data yang tidak *balance* dengan jumlah data positif sebanyak 1088 dan 499 untuk data negatif, untuk melihat algoritma apa yang lebih baik, diantara algoritma *Naïve Bayes*, *Support Vector Machine*, atau *Random Forest*, dan ditemukan bahwa *Random Forest* lah yang memiliki peningkatan presentasi yang lebih baik, dimana terlihat pada tabel 8, *Random Forest* cukup stabil dalam peningkatan persentase dari jumlah data 100:100, 200:200, 300:300 499:499 maupun data yang tidak *balance* 1088:499, bahkan algoritma *Random Forest* memiliki akurasi yang paling baik yaitu 85,25% dengan pembobotan *Count Vectorizer* dan 84,56% dengan pembobotan TF-IDF, dibanding dengan algoritma SVM, dan *Naïve Bayes*.

Selain itu, analisis sentimen dalam penelitian ini menunjukkan bahwa mayoritas sentimen yang diungkapkan adalah positif, yang mengindikasikan bahwa pengguna aplikasi pembelajaran elektronik Ruangguru secara umum puas dalam menggunakan aplikasi Ruangguru. Didapati juga hasil penelitian berdasarkan *word cloud* yang berisi informasi sentimen positif dan negatif terkait dengan kepuasan pengguna terhadap aplikasi Ruangguru, sehingga nantinya tim dari pada Ruangguru maupun *developer* bisa mengevaluasi apa yang kurang dari pada aplikasi Ruangguru untuk menjadi lebih baik ke depannya. Dalam penelitian berikutnya, perlu dicoba dalam mengimplementasikan opini publik dengan jumlah data

yang lebih banyak dan data yang *balance* dengan menggunakan algoritma yang berbeda ataupun dengan pembobotan yang lebih beragam seperti *Word 2 Vectorizer*, atau BERT.

5. Daftar Pustaka

- [1] A. Maritsa, U. Hanifah Salsabila, M. Wafiq, P. Rahma Anindya, and M. Azhar Ma'shum, "Pengaruh Teknologi Dalam Dunia Pendidikan," *Al-Mutharahah J. Penelit. dan Kaji. Sos. Keagamaan*, vol. 18, no. 2, pp. 91–100, 2021, doi: 10.46781/al-mutharahah.v18i2.303.
- [2] S. Aisyah and Muhammad Alif Kurniawan, "Penggunaan Media Pembelajaran Daring pada Masa Pandemi COVID-19," *J. Ris. Madrasah Ibtidaiyah*, vol. 1, no. 1, pp. 48–56, 2021, doi: 10.32665/jurmia.v1i1.195.
- [3] N. S. R. dan 2Mia Setiawati, "Aplikasi Pendidikan Online 'Ruang Guru' Sebagai Peningkatan Minat Belajar Generasi Milenial Dalam Menyikapi Perkembangan Revolusi Industri 4.0," *Seloka J. Pendidik. Bhs. dan Sastra Indones.*, vol. 3, no. 2, pp. 241–246, 2019.
- [4] A. Erfina, S. Basryah, A. Saepulrohman, and D. Lestari, "ANALISIS SENTIMEN APLIKASI PEMBELAJARAN ONLINE DI PLAY STORE PADA MASA PANDEMI COVID-19 MENGGUNAKAN ALGORITMA SUPPORT VECTOR MACHINE (SVM)," *Semin. Nas. Inform.*, 2020.
- [5] H. Yulianti, "Analisis Komparasi Algoritma Klasifikasi Data Mining Untuk Prediksi Penjurusan Siswa Sekolah Menengah Atas (Sma) Pramita Karawaci Tangerang," *Lensa*, vol. 2, no. 48, pp. 1–6, 2020, doi: 10.33050/lns.v2i48.1277.
- [6] A. Chamekh, M. Mahfoudh, and G. Forestier, "Sentiment Analysis Based on Deep Learning: A Comparative Study," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 13369 LNAI, pp. 498–507, 2022, doi: 10.1007/978-3-031-10986-7_40.
- [7] A. Purnamawati, W. Nugroho, D. Putri, and W. F. Hidayat, "Deteksi Penyakit Daun pada Tanaman Padi Menggunakan Algoritma Decision Tree, Random Forest, Naïve Bayes, SVM dan KNN," *InfoTekJar J. Nas. Inform. dan Teknol. Jar.*, vol. 5, no. 1, pp. 212–215, 2020, [Online]. Available: <https://doi.org/10.30743/infotekjar.v5i1.2934>
- [8] R. Patel and K. Passi, "Sentiment Analysis on Twitter Data of World Cup Soccer Tournament Using Machine Learning," *IoT*, vol. 1, no. 2, pp. 218–239, 2020, doi: 10.3390/iot1020014.
- [9] A. Harun and D. Putri Ananda, "Analisa Sentimen Opini Publik Tentang Vaksinasi Covid-19 di Indonesia Menggunakan Naïve bayes dan Decision Tree," *MALCOM Indones. J. Mach. Learn. Comput. Sci.*, vol. 1, no. 1, pp. 58–64, 2021, doi: 10.57152/malcom.v1i1.63.
- [10] R. W. Samsir¹, Ambiyar², Unung Verawardina³, Firman Edi⁴ and 1, "Analisis Sentimen Pembelajaran Daring Pada Twitter di Masa Pandemi COVID-19 Menggunakan Metode Naïve Bayes," *J. Media Inform. Budidarma*, vol. 5, no. 1, p. 149, 2021, doi: 10.30865/mib.v5i1.2604.
- [11] V. A. Permadi, "Analisis Sentimen Menggunakan Algoritma Naive Bayes Terhadap Review Restoran di Singapura," *J. Buana Inform.*, vol. 11, no. 2, pp. 141–151, 2020, doi: 10.24002/jbi.v11i2.3769.
- [12] H. Wisnu, M. Afif, and Y. Ruldevyani, "Sentiment analysis on customer satisfaction of digital payment in Indonesia: A comparative study using KNN and Naïve Bayes," *J. Phys. Conf. Ser.*, vol. 1444, no. 1, 2020, doi: 10.1088/1742-6596/1444/1/012034.
- [13] L. Breiman, *Random forests*, vol. 45, no. 1. California: Statistics Department University of California Berkeley, 2001. doi: 10.1023/A:1010933404324.
- [14] N. C. Alvita I. Kusumarini¹), Pandu A. Hogantara²), Muammar Fadhlurohman³), "Perbandingan Algoritma Random Forest, Naïve Bayes, Dan Decision Tree Dengan Oversampling Untuk Klasifikasi Bakteri E. Coli," *Senamika*, vol. 2, no. 1, pp. 792–799.
- [15] M. R. Rafsanjani, "ScrapPaper: A web scrapping method to extract journal information from PubMed

- and Google Scholar search result using Python.," *bioRxiv*, p. 2022.03.08.483427, 2022, doi: <https://doi.org/10.1101/2022.03.08.483427>.
- [16] V. O. Tama, Y. Sibaroni, and Adiwijaya, "Labeling Analysis in the Classification of Product Review Sentiments by using Multinomial Naive Bayes Algorithm," *J. Phys. Conf. Ser.*, vol. 1192, no. 1, 2019, doi: 10.1088/1742-6596/1192/1/012036.
 - [17] J. A. Josen Limbong, I. Sembiring, K. Dwi Hartomo, U. Kristen Satya Wacana, and P. Korespondensi, "Analisis Klasifikasi Sentimen Ulasan Pada E-Commerce Shopee Berbasis Word Cloud Dengan Metode Naive Bayes Dan K-Nearest Neighbor Analysis of Review Sentiment Classification on E-Commerce Shopee Word Cloud Based With Naïve Bayes and K-Nearest Neighbor Meth," *J. Teknol. Inf. dan Ilmu Komput.*, vol. 9, no. 2, pp. 347–356, 2019, doi: 10.25126/jtiik.202294960.
 - [18] S. D. Pramukti, A. Nugroho, and A. S. Sunge, "Analisis Sentimen Masyarakat Dengan Metode Naïve Bayes dan Particle Swarm Optimization," *Techno.Com*, vol. 21, no. 1, pp. 61–74, 2022, doi: 10.33633/tc.v21i1.5332.
 - [19] E. H. Muktafin and P. Kusriani, "Sentiments analysis of customer satisfaction in public services using K-nearest neighbors algorithm and natural language processing approach," *Telkomnika (Telecommunication Comput. Electron. Control.*, vol. 19, no. 1, pp. 146–154, 2021, doi: 10.12928/TELKOMNIKA.V19I1.17417.
 - [20] B. W. S. Irkham Widhi Saputro*1, "Uji Performa Algoritma Naïve Bayes untuk Prediksi Masa Studi Mahasiswa," *Creat. Inf. Technol. J. (CITEC JOURNAL)*, vol. 6, no. 1, pp. 1–11, 2019, doi: <https://doi.org/10.24076/citec.2019v6i1.178>.
 - [21] T. Turki and S. S. Roy, "Novel Hate Speech Detection Using Word Cloud Visualization and Ensemble Learning Coupled with Count Vectorizer," *Appl. Sci.*, vol. 12, no. 13, 2022, doi: 10.3390/app12136611.
 - [22] D. N. Sharma, D. P. Shankar, M. R. Raj, and M. C. Dalwadi, "Sentiment Analysis for Amazon Product Reviews Using Logistic Regression Model," *J. Dev. Econ. Manag. Res. Stud.*, vol. 09, no. 11, pp. 29–42, 2022, doi: 10.53422/jdms.2022.91104.
 - [23] S. Vijayaraghavan and D. Basu, "Sentiment Analysis in Drug Reviews using Supervised Machine Learning Algorithms," *arxiv (Cornell Univ.*, doi: <https://doi.org/10.48550/arXiv.2003.11643>.
 - [24] J. A. Septian, T. M. Fachrudin, and A. Nugroho, "Analisis Sentimen Pengguna Twitter Terhadap Polemik Persepakbolaan Indonesia Menggunakan Pembobotan TF-IDF dan K-Nearest Neighbor," *J. Intell. Syst. Comput.*, vol. 1, no. 1, pp. 43–49, 2019, doi: 10.52985/insyst.v1i1.36.
 - [25] I. T. Julianto, "Analisis Sentimen Terhadap Sistem Informasi Akademik Mahasiswa Institut Teknologi Garut," *J. Algoritma.*, vol. 19, pp. 458–465, doi: <https://doi.org/10.33364/algoritma/v.19-1.1112>.